Technical University of Denmark

Integrated Rolling Stock Planning for Suburban Passenger Railways

Thorlacius, Per; Larsen, Jesper; Groth, Julie Jespersen; Laumanns, Marco

Publication date:
2017

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

Integrated Rolling Stock Planning for Suburban Passenger Railways

Per Thorlacius, M.Sc.

Ph.D. Thesis
May 2017

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

The carriage rolled noiselessly on the soft track, the shadows fell long on the dusty little plain interspersed with dark bushes, mounds of turned-up earth, low wooden buildings with iron roofs of the Railway Company; the sparse row of telegraph poles strode obliquely clear of the town, bearing a single, almost invisible wire far into the great campo – like a slender, vibrating feeler of that progress waiting outside for a moment of peace to enter and twine itself about the weary heart of the land.

Joseph Conrad, “Nostromo”, 1904

Summary

One of the core issues for operators of passenger railways is providing sufficient number of seats for passengers while keeping operating costs at a minimum. The process a railway operator undertakes in order to achieve this is called rolling stock planning. Rolling stock planning deals with deciding how to utilise the fleet of available train units in space and time.

In this thesis, rolling stock planning has been studied, using as case study DSB S-tog, the suburban passenger railway operator of the City of Copenhagen. At DSB S-tog, the rolling stock planning process is subdivided according to time horizon into two subprocesses. Firstly, there is the long-term circulation planning process, in which planning is conducted for anonymous, virtual train units months in advance. Secondly, there is the short-term train unit dispatching process, which covers the execution of the long term circulation plan. In the train unit dispatching process, the anonymous, virtual train units from the circulation planning process will have real, physical train units assigned to them. The train unit dispatching process has a short-term time horizon of days, hours and minutes and makes sure the actual, real-world train services are performed. Disruptions are also handled in this process.

In the long term circulation planning phase of rolling stock planning, a large number of railway-specific requirements must be taken into account: The physical railway infrastructure must be adhered to, e. g., platform and depot track capacities, the rules of the train control system and the order in which train units may be parked so as not to obstruct each other's movements; All trains services of the timetable must have a least one train unit assigned; Only the available rolling stock can be used in the plan; The plan should provide seating capacity according to the passenger demand and provide an even distribution of flexible space for bicycles etc.; Planned shunting operations in the depot should have sufficient personnel on duty; Train units must undergo interior and exterior cleaning, surface foil application and winter preparedness treatment at regular time intervals; At regular service distance intervals, train units must undergo scheduled maintenance etc., and consumables must be refilled; Certain train services must have train units with additional train control system equipment installed, special passenger counting equipment installed and/or perform predefined exposure of commercials.

In the short-term train unit dispatching phase of rolling stock planning, additional railway-specific requirements include: Exterior graffiti removal and unscheduled maintenance on demand and sometimes within a given time frame; Make available train units to meet surveillance video recording requests from the police within a given time frame.

Due to the large number of railway-specific requirements and their nature, rolling stock planning is traditionally conducted in a step-by-step manner, in which the individual planning processes are not integrated with each other. Needless to say, this yields rolling stock plans that are either suboptimal or infeasible with regard to the requirements.

In this thesis it is shown that it is possible to design and implement a rolling stock planning model integrating into one planning process all the railway-specific requirements of DSB S-tog, all at the same time. This integrated rolling stock planning model is implemented using a greedy heuristic and makes use of the novel (train) unit order conservation principle, implemented as

special side constraints to a resource constrained shortest path algorithm. The integrated rolling stock planning model is tested extensively on 15 real-world, manually constructed rolling stock plan data instances. When run on these instances, the greedy heuristic can achieve an average economic gain of approx. 2% with processing times in all cases less than 1 hour 20 minutes. In addition to this, the greedy heuristic can make typically infeasible rolling stock plans feasible within just a few minutes of processing time.

Moreover, in this thesis a number of different economic net value upper bound calculation models are designed, implemented and tested. The net value upper bound calculation models implement the railway-specific requirements to a varying degree and consequently expose different properties with regard to tightness of bounds and processing times. The net value upper bound model having the highest degree of requirements integration adheres to 47% of the requirements by count. Using this tightest net value upper bound calculation model, it is shown that the greedy heuristic mentioned before is able to gain approx. 1/3 of the relative gap between the net value of the original, manual plans and the net value upper bound. Moreover, it is shown that in most cases, the net value of the original, manual plans already lie close to the upper bound.

Furthermore, a branch-and-price based matheuristic integrated rolling stock planning model is designed, implemented and tested. It is shown that this type of matheuristic model is able to adhere fully to all railway-specific requirements, and that the vast majority of requirements can be integrated into the optimisation steps of the matheuristic algorithm. The branch-and-price matheuristic model can solve small instances (e. g., in the form of matheuristic iterations) to optimality. Used in conjunction with the greedy heuristic, the two methods combined can achieve an additional small gain in objective value not achievable using each method by itself.

With a yearly cost of the rolling stock operation in the hundreds of million DKK, the potential benefit of a real-world application of the models to DSB S-tog is in the order of several million DKK per year. In addition to this, a substantial benefit can be gained by the way the models can automate the current, manual planning procedures. This will enable planners to invest more creativity and meticulousness into the planning process as a result of being liberated from manual planning procedures. For these reasons, DSB S-tog is eager to proceed with the real-world application of the models developed in this thesis.

Preface

This thesis has been submitted to the Department of Management Engineering at the Technical University of Denmark for the partial fulfilment of the requirements for acquiring the degree of Doctor of Philosophy (Ph.D.) in Engineering Science.

The work has been conducted within the framework of the industrial Ph.D. programme of Innovation Fund Denmark in collaboration with Danish State Railways (DSB), Technical University of Denmark (DTU) and IBM Research, Zürich Laboratory.

The work has been supervised by Professor Jesper Larsen, DTU; Julie Jespersen Groth, DSB and Marco Laumanns, IBM. A substantial contribution has been provided by Professor David Ryan, University of Auckland.

The thesis deals with the planning of rolling stock for the suburban passenger railway operator of the City of Copenhagen, DSB S-tog. The thesis consists of three parts:

Part I is an introduction to rolling stock planning for a suburban railway operator. It covers rolling stock planning processes and requirements.

Part II deals with ways to model and to solve the rolling stock planning problem. It contains an overview of related work and presents five new rolling stock planning models with different characteristics. Experimental results are presented.

Part III puts the work carried out into perspective, the results achieved and lessons learned are discussed and an outlook to further research is presented.

Copenhagen, May 31, 2017

Per Thorlacius

Acknowledgements

For their contribution to the work presented in this thesis, the author would like to thank (in alphabetical order): Christian Blome, TU-Berlin; Ralf Borndörfer, ZIB; Berit Dangaard Brouer, DTU; Jakob Dirksen, DSB; Simon Bull, DTU; Sofie Burggraeve, KU Leuven; Associate Professor Andreas Bærentzen, DTU; Torben Christensen, DSB; Tom Christiansen; Emily Clearwater, University of Auckland; Bernd Dammann; Knud Forsberg; Rasmus Dyhr Frederiksen, Rapidis; Ole Aagaard Frandsen; Julie Jespersen Groth, DSB; Henrik Gudmundsson, Concito; Jørgen T. Haahr, DTU; Jesper Morelli Hansen; Jan Erik Karlskov Jensen; Tor Fog Justesen, DTU; Christian Vad Karsten, DTU; Jacob Cramer Kjær, DSB; Kim Kjærgaard, DSB; David Franz Koza, DTU; Esthi Kunz; Professor Jesper Larsen, DTU; Marco Laumanns, IBM; Associate Professor Richard Lusby, DTU; Associate Professor Andrew J. Mason, University of Auckland, Professor Otto Anker Nielsen, DTU; Natalia Rezanova, DSB; Senior Lecturer Andrea Raith, University of Auckland; Professor David Ryan, University of Auckland; Professor Stefan Røpke, DTU; Nikolaj Schlegel, DSB; Elsebet Seest, DSB; Holger Simon, DSB; Associate Professor Thomas Stidsen, DTU; Anders Tipsmark, DSB; Stefan Vidgren, DSB; Thomas Viinblad, DSB; René Zbinden.

Contents

Summary 2

Preface 4

Acknowledgements 5

List of Tables 9

List of Figures 11

I Rolling Stock Planning For A Suburban Passenger Railway 13

1 Introduction 14

1.1 Introduction to Passenger Railway Operations Planning . . . 15
1.2 Introduction to Rolling Stock Planning . . . 17
1.3 Industrial and Scientific Goals of This Thesis . . . 18
1.4 Scientific Contributions of This Thesis . . . 19
1.5 How This Thesis Is Structured . . . 20
1.6 Terminology . . . 20

2 The Rolling Stock Planning Process 25

2.1 Circulation Planning . . . 29
2.2 Train Unit Dispatching . . . 31

3 Railway-Specific Requirements for Rolling Stock Planning 33

3.1 Infrastructure . . . 33
3.2 Timetable . . . 35
3.3 Rolling Stock . . . 39
3.4 Passenger Demand . . . 41
3.5 Personnel on Duty . . . 45
3.6 Scheduled Maintenance . . . 45
3.7 Unscheduled Maintenance . . . 48
3.8 Friction Sand . . . 48
3.9 Exterior Cleaning . . . 48
3.10 Exterior Graffiti Removal . . . 48
3.11 Interior Cleaning . . . 48
3.12 Winter Preparedness . . . 49
3.13 Exposure of Commercials . . . 50
3.14 Surveillance Video Requests . . . 50

3.15 Surface Foil Application . . . 50
3.16 Passenger Counting Equipment . . . 50
3.17 Train Control System Equipment . . . 50

II The Integrated Rolling Stock Planning Problem 51

4 Rolling Stock Planning Models and Solution Methods 52
4.1 Overview of Models and Solution Methods from Literature . . . 52
4.2 Overview of Rolling Stock Planning Models Proposed In This Thesis . . . 59

5 A Greedy Heuristic Integrated Rolling Stock Planning Model 63
5.1 Introduction . . . 63
5.2 The Integrated Rolling Stock Planning Problem . . . 67
5.3 Timetable and Infrastructure Data Model . . . 73
5.4 Train Unit Data Model . . . 76
5.5 Path Finding Algorithm . . . 79
5.6 Heuristic Framework . . . 83
5.7 Experimental Results . . . 85
5.8 Discussion . . . 92

6 Net Value Upper Bound Calculation Models for Rolling Stock Planning 93
6.1 Introduction . . . 93
6.2 Mathematical Formulations . . . 94
6.3 Experimental Results . . . 102
6.4 Discussion . . . 104

7 A Branch-and-Price Matheuristic Integrated Rolling Stock Planning Model 110
7.1 Introduction . . . 110
7.2 Mixed Integer Linear Program . . . 113
7.3 Column Generation Framework . . . 120
7.4 Branch-And-Bound Framework . . . 121
7.5 Matheuristic Framework . . . 126
7.6 Numerical Experiments . . . 129
7.7 Discussion . . . 135
7.8 Conclusions and Further Research . . . 138

III Perspectives on Integrated Rolling Stock Planning 140

8 Discussion 141
8.1 Comparison of Model Characteristics . . . 141
8.2 Heuristic and Matheuristic Algorithm Convergence . . . 142
8.3 Main Conclusions . . . 144

9 Future Research 146
9.1 Alternative Solution Methodologies . . . 146
9.2 Circulation Planning and Train Unit Dispatching Process Integration . . . 149
9.3 Are We Solving the Right Problem? . . . 150

Appendix **155**

A Visualisation Tools for Rolling Stock Planning **155**
A.1 Space-Time Diagrams . . . 155
A.2 Circulation Diagrams . . . 164
A.3 Infrastructure and Timetable Data Model Interactive Diagrams . . . 164
A.4 Branch-and-Price Algorithm Progress Interactive Diagrams . . . 166
A.5 Causal Diagram for Model Design Decisions . . . 174

B Auxiliary Methodologies for Rolling Stock Planning **176**
B.1 Retrofitting Incomplete Train Unit Trajectories Data . . . 176
B.2 Rolling Stock Plan Net Value Calculation . . . 179
B.3 The Unit Order Flow Conservation Principle . . . 181

C Implementation Details and Metrics **184**
C.1 Software Application Architecture and Design . . . 184
C.2 Software Platforms Used . . . 189
C.3 Software Application Code Metrics . . . 191

D Repairing Infeasible Rolling Stock Plans **192**
D.1 Characteristics of Infeasible Rolling Stock Plans . . . 192
D.2 Instruments to Make Infeasible Rolling Stock Plans Feasible . . . 195

References **198**

List of Tables

2.1 Overview of characteristics for processes *circulation planning* and *train unit dispatching with disruptions* 26

3.1 The requirements for rolling stock planning at DSB S-tog 34
3.2 A page from the DSB S-tog timetable 37
3.3 Number of days per year the standard timetable was in effect on the entire network of DSB S-tog 37
3.4 Overview of characteristics for train unit types of DSB S-tog 40
3.5 Train composition types for revenue train services at DSB S-tog 41
3.6 Passenger demand development at DSB S-tog 41
3.7 Contractual obligations of the workshop (DSB Vedligehold) to the operator (DSB S-tog) with regard to number of train units to be provided 46

4.1 Overview of the requirements for rolling stock planning at DSB S-tog and to which degree the requirements are implemented 61
4.2 Overview of the characteristics of the rolling stock planning models developed for this thesis 62

5.1 Overview of characteristics in selected and reviewed, recent literature specific to rolling stock planning 65
5.2 Sets and their corresponding indices in the mathematical formulation of the heuristically based integrated rolling stock planning model 70
5.3 Parameters in the mathematical formulation of the heuristically based integrated rolling stock planning model 71
5.4 Overview of the requirements for rolling stock planning at DSB S-tog and how they are implemented in the model 72
5.5 Overview of data instances and experimental results for the improvement of rolling stock plans using the integrated heuristic model 88

6.1 Sets in the upper bound calculation models, their corresponding indices, domains and definitions, ordered alphabetically by symbol 95
6.2 Parameters with symbols b to n in the upper bound calculation models, their domains and definitions, ordered alphabetically by symbol 96
6.3 Parameters with symbols p to z in the upper bound calculation models, their domains and definitions, ordered alphabetically by symbol 97
6.4 Decision variables of the upper bound calculation models, their domains and definitions, ordered alphabetically by symbol 97
6.5 Comparison of experimental results from the greedy heuristic with experimental results from the upper bound calculation models 103
6.6 Overview of the characteristics of the proposed upper bound models 105

6.7 Requirements for rolling stock planning at DSB S-tog and the degree to which they are implemented in the upper bound calculation models 106

7.1 Overview of the requirements for rolling stock planning at DSB S-tog and to which degree they are implemented in the integrated models 111
7.2 Sets of the matheuristic model, their corresponding indices, domains and definitions, ordered alphabetically by symbol 114
7.3 Parameters of the matheuristic model, their domains and definitions, ordered alphabetically by symbol 115
7.4 Decision variables of the matheuristic model, their domains and definitions, ordered alphabetically by symbol 116
7.5 Overview of the characteristics of the branching schemes used 123
7.6 Short processing time numerical experiments for the greedy heuristic and the branch-and-price matheuristic using the **enhanced** objective 130
7.7 Short processing time numerical experiments for the greedy heuristic and the branch-and-price matheuristic using the **standard** objective 131
7.8 Long processing time numerical experiments for the greedy heuristic, the branch-and-price matheuristic and a hybrid thereof 132

8.1 Overview of combinatorial characteristics for the selection of k number of train units to modify from the set of U train units 144

B.1 Overview of rolling stock operating cost types for a suburban passenger railway operator like DSB S-tog 180
B.2 Overview of the unit costs for the rolling stock operation at DSB S-tog as used in the models in this thesis 181

C.1 Overview of code metrics, contents and example classes by package for the models implemented in this thesis 190

List of Figures

1.1 Overview of the overall planning processes for the operation of a passenger railway and the order in which they are traditionally carried out 16

2.1 The rolling stock planning process at DSB S-tog and its subprocesses 27
2.2 Example of a DSB S-tog rolling stock circulation diagram 28
2.3 Impressions of the rolling stock operation 29

3.1 Topological map of the railway infrastructure available to DSB S-tog 35
3.2 The train service line map of the Greater Copenhagen area 38
3.3 The DSB S-tog train unit types 39
3.4 An impression of a DSB S-tog train unit 40
3.5 Example of a space-time diagram of the passenger seat demand 42
3.6 Passenger demand by day type and time expressed as demand for number of specific train composition types 43
3.7 A space-time diagram showing the passenger demand expressed as demand for specific train composition types 44
3.8 The DSB S-tog train unit SA8120 undergoing a major overhaul in the workshop at Høje Tåstrup 46
3.9 The service distance distribution of the DSB S-tog train units 47
3.10 The DSB S-tog train unit SA9156 undergoing exterior cleaning in the cleaning facilities at the workshop at Høje Tåstrup 49

5.1 A schematic illustration of the space-time graph G 74
5.2 An example of train unit trajectories in the space-time graph 78
5.3 Examples of shunting situations at Farum station 78
5.4 Sankey diagram of the flows for benefits and costs (including penalties) in the proposed heuristic 85
5.5 Convergence diagram for different values of k 90
5.6 A convergence diagram showing the mean value and the median of the gain for 15 test runs 91

6.1 Relative net value gain over manual plan comparison for results from upper bound calculation models and greedy heuristic 107
6.2 Absolute net value comparison for results from upper bound calculation model B10, the greedy heuristic and the original, manual plan 108

7.1 Convergence diagram for different values of k for the branch-and-price matheuristic for runs with stopping criterion “stop at first integer” 136
7.2 A convergence diagram showing the mean value and the median of the gain for $n = 8$ test runs 137

9.1 An example of paper based passenger information 151
9.2 An example of a modern, electronic passenger information system 151

A.1 Example of a seat demand space-time diagram 157
A.2 Example of a seat supply space-time diagram 158
A.3 Example of a seat surplus space-time diagram for an original plan 159
A.4 Example of a seat surplus space-time diagram for a modified plan 160
A.5 Example of a train composition space-time diagram 161
A.6 Example of a train composition supply space-time diagram for an original plan 162
A.7 Example of a train composition supply space-time diagram for a modified plan 163
A.8 Example of a DSB S-tog rolling stock circulation diagram for Line F 164
A.9 A infrastructure and timetable data model interactive diagram example for real-world data 165
A.10 An example of a branch-and-bound tree diagram showing a small branch-and-bound tree 167
A.11 An example of a branch-and-bound tree diagram showing a branch-and-bound tree constructed using the normal, non-strong branching 168
A.12 An example of a branch-and-bound tree diagram showing an example branch-and-bound tree constructed using full strong branching 168
A.13 An example of a branch-and-bound tree diagram showing the branch-and-bound tree for the solution of Line H 169
A.14 An example of a fractional flow diagram with few fractional flows 171
A.15 An example of a fractional flow diagram with many fractional flows 172
A.16 An example of a fractional flow diagram with crossover trajectories 173
A.17 A causal diagram for the integrated matheuristic rolling stock planning model . 175

B.1 An example of a conflict graph for the retrofitting process for the depot of Høje Tåstrup 178

C.1 UML class diagram showing class hierarchies for Segment, Event and Train-Unit classes 186
C.2 UML class diagram showing the node and branching class hierarchies, the node queue and branching helper classes 188

Part I

Rolling Stock Planning For A Suburban Passenger Railway

Chapter 1

Introduction

Passenger railway transport systems form an important part of modern, urban societies. Every day, around the world, millions and millions of people are transported to and from their daily activities by means of passenger railway transport services.

As cities grow larger and more dense, a strong, global trend, passenger railway transport becomes an essential part of the urban societal infrastructure. Compared to other modes of transport, a passenger railway can supply an unsurpassed transport capacity with very modest land use requirements. By providing efficient transport services, passenger railways play a key role in preventing road traffic congestion, a major problem in most cities.

Moreover, passenger railway transport has, by far, the smallest environmental footprint of all motorised transport modes, with energy use per transported passenger kilometre for a typical journey lower than that of automobile transport by a factor in the order of 4 [107]. Moreover, greenhouse gas emission rates per passenger kilometre are lower by a factor in the order of 7 compared to automobile transport [107]. Most passenger railways are provided with an electric traction system, for which reason there is no local air pollution, a much wider range of sustainable energy sources at hand and a range of technologies to reduce environmental impact of energy production in electrical power plants. Passenger railway transport emits less noise than its road-based passenger transport counterparts, automobiles and busses, and the noise emitted is generally perceived far less aggravating [41]. As such, the noise-related external costs per passenger kilometre are in the order of 7 times lower for passenger railway transport than for automobile transport [41]. From a safety perspective, the fatality risk per passenger kilometre for railway is 3 times lower than for busses and 24 times lower than for automobiles [50].

All of these factors make passenger railways a major contributor to sustainable, societal, economic progress. This role is likely to be strengthened in the future with the emergence of driverless on-demand road vehicles. Driverless vehicles will likely make road transport easier, this in turn leading to an increase in demand, likely to lead to even more road congestion. However, the emergence of driverless road vehicles may in turn enable the provision of a road/railway integrated transport system in which the strengths of the different modes of transport can be utilised better than today.

This thesis deals with one of the essential aspects of providing an efficient passenger railway transport system, namely the planning of the utilisation of railway rolling stock. DSB S-tog, the suburban passenger railway operator of the City of Copenhagen is used as case study in the thesis. In 2015, DSB S-tog transported more than 114 million passengers [2].

1.1 Introduction to Passenger Railway Operations Planning

The operation of a passenger railway is an immensely complex undertaking. Numerous conditions must be fulfilled, all at the same time, for a passenger train service to be performed. Not only must the railway infrastructure of tracks, points (switches), energy transmission networks, bridges, tunnels, station buildings, platforms and communication networks exist and be in a functional state; for the individual train units to move, there must also be available capacity on the tracks. There must be a functional train control system to secure that operations may be conducted safely. The state of maintenance of the train units must be in accordance with the legislation. There needs to be space in the depot for the train units to be parked when they are not in use. The train units, platforms and station buildings must be cleaned regularly. Time tables must be constructed and distributed to passengers. The fare tariff must be determined and tickets and passes sold. Revenue must be collected and distributed to other operators within the same tariff. There needs to be personnel in the train units to drive them around and personnel to perform the inspection of tickets. There needs to be personnel to oversee and monitor the railway operation and to take action to prevent or recover from disruptions. All personnel needs to have the right education and certification, e. g., for the operation of specific types of train units. The union agreements for the personnel must be adhered to. These are just some examples of all the conditions that must be fulfilled for a passenger railway to operate.

Thus, in order for a railway to operate, a lot of people with very different backgrounds need to work closely together. As such, railway planning is a highly multi-disciplinary task.

Moreover, railway operations planning must be conducted within highly different time horizons and levels of detail, ranging from strategic decisions it may take years to realise (for instance the construction of new railway tracks or the acquisition of new rolling stock) to detailed, operational decisions on what is going to happen within the next minutes (for instance if a particular train service should be cancelled because of a technical fault on one of the front doors of the foremost train unit). For the reasons mentioned, railway operations planning is traditionally split into individual planning tasks by time horizon and area of expertise to be carried out individually in a step-by-step manner in the order shown in Figure 1.1.

Railway operations planning processes relate to three time horizons, the *strategic time horizon*, by which plans are created for events to happen years ahead, the *tactical time horizon*, by which plans are created for events to happen months or weeks ahead, and by the *operational time horizon*, by which plans and decisions are taken for events to happen days, hours or minutes ahead. Seen from an overall perspective, railway operations planning thus starts in the strategic time horizon with the process of *infrastructure planning*, i. e., the planning and provision of the railway infrastructure. Next, *fleet planning* deals with the planning and acquisition of the rolling stock to operate on the infrastructure. Next, *train service line planning* is conducted, involving strategic decisions as to which train service lines to operate, which stations to serve and with what frequency, etc.

The next overall process is in the tactical time horizon, namely the *timetabling* process in which the details of which train services to operate at what time are decided. Next, in the tactical process of *circulation planning* it is decided how the rolling stock should be operated to execute the timetable. In the following process *personnel planning* it is decided where and when the personnel should be on duty in order to execute the timetable and the circulation plan.

The next processes are in the operational time horizon dealing with the real-world execution of the hithereto planned tasks and services. The process *train service dispatching* deals with deciding which train services from the timetable to cancel or delay in the case of disruptions, the *train unit dispatching* process deals with making sure physical train units are assigned to all train services and *personnel dispatching* that there is personnel to perform all tasks. The

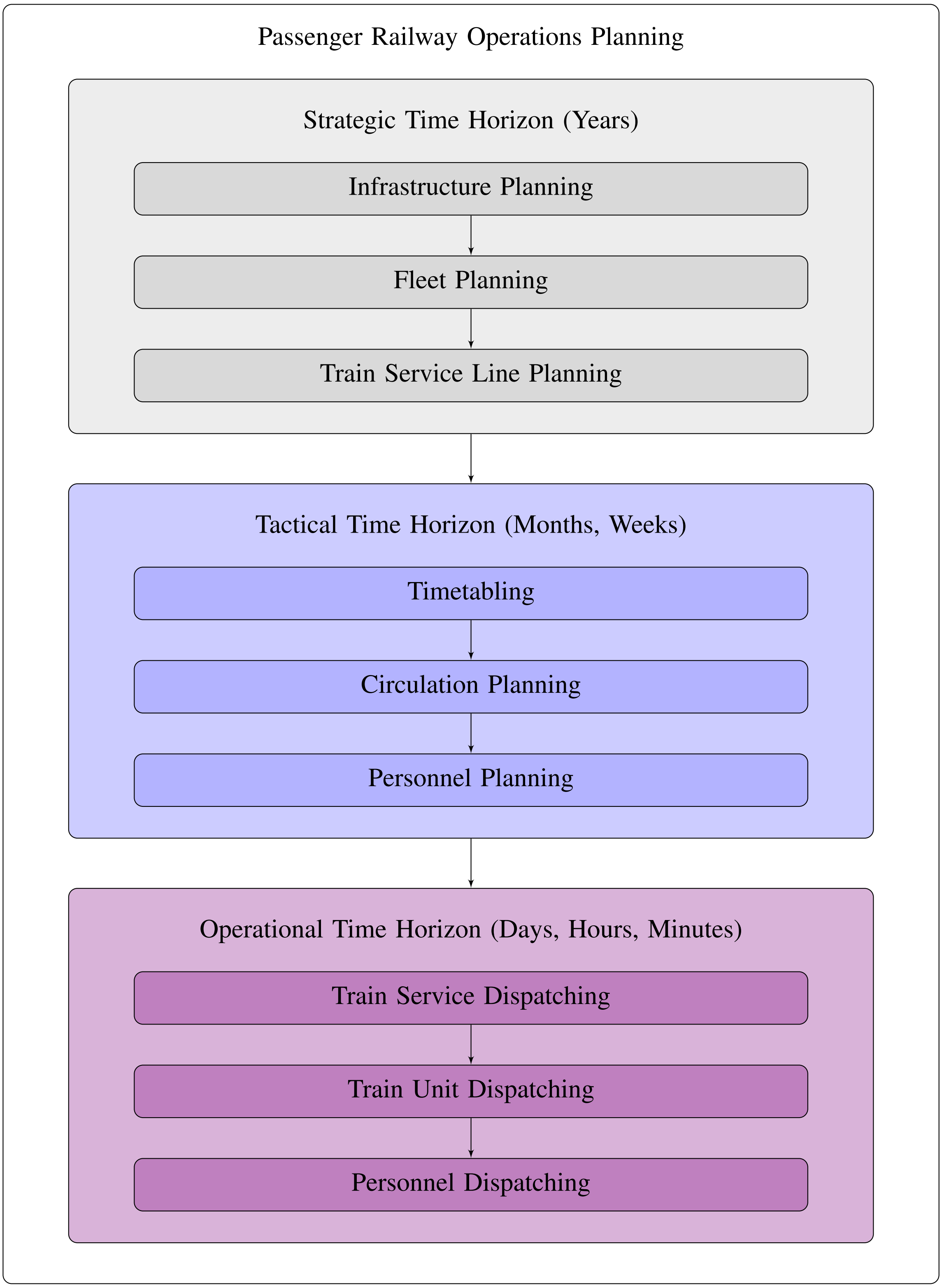

Figure 1.1: Overview of the overall planning processes for the operation of a passenger railway and the order in which they are traditionally carried out. The diagram shows overall planning processes carried out in the strategic time horizon (grey), in the tactical time horizon (blue) and in the operational time horizon (violet).

processes in the operational time horizon will be reiterated each time operational conditions change, e. g., in the event of disruptions. Focus is then shifted to recovery, i. e., to bring back the operation to be in accordance with the original plan. Processes in the operational time horizon will then be reiterated until all disruptions have been resolved.

As may be understood, the sequential manner by which railway operations planning is traditionally conducted may not be ideal. It may, for instance, be the case that a very good timetable may be constructed from a passenger perspective, but as it turns out, this timetable is difficult to execute with regard to the available rolling stock. The constructed timetable may also be prone to delays or just expensive. Moreover, it may turn out that the personnel plan corresponding to the constructed timetable may violate the union agreements with regard,e. g., to maximum duty duration.

The art of railway planning lies in conducting the planning in each individual step by taking into account the complex interactions with both preceding and succeeding planning steps. This may not be an easy task since very different areas of expertise, levels of detail, focus and planning objectives may prevail in the different planning steps.

If multiple planning subprocesses, different by nature, are handled all-together in one single planning process, we speak of *integrated planning* and say that the subprocesses are integrated into the single planning process. Ideally, in integrated planning, the subprocesses form an integral part of the integrated planning process. However, integrated planning can also, to some degree, be achieved by ensuring there exists a *feedback mechanism* between the individual planning steps.

Using a feedback mechanism is what railway operators have been practising for more than a century, namely that different departments suggest changes in the area of other departments to make planning easier in their own. In this case integration (to a limited degree) becomes an organic part of the planning procedures of the railway operator. Over time this may generate a lot of tacit knowledge about how to conduct the railway operation for the conditions prevailing for the individual railway.

Using the feedback mechanism between planning steps as means to achieve a higher level of process integration is of course greatly facilitated if the individual planning steps are fast to perform. In our age this means if they are dynamic and automated. In this way a plan covering one step of the planning process can quickly be constructed based on a plan constructed in another step. One can then iterate back and forth between the two steps, gradually producing corresponding plans that have a higher degree of integration and represent good solutions to both planning steps.

This thesis deals with integrating into one process, subprocesses which are usually executed in a step-by-step manner. The subprocesses treated are those contained in the processes *circulation planning* and *train unit dispatching* on Figure 1.1, collectively designated *rolling stock planning*, to which subject an introduction is given next.

1.2 Introduction to Rolling Stock Planning

Rolling stock planning is the collective term for the processes that a passenger railway operator undertakes in order to ensure the most efficient usage of its rolling stock. The main purpose of rolling stock planning is to accommodate the demand for seats in the individual train services running between stations as demanded by passengers. This demand must be accommodated while minimising operational costs.

The demand for seats is given as the number of expected passengers for a given train service, but whether this demand can be accommodated by the railway operator depends on a number of limiting factors.

The main limiting factor is of course the number of individual train units in the rolling stock fleet, as well as where the train units are located at any given time. As such, rolling stock planning is about determining the distribution of train units as resources in space and time.

Another important limiting factor lies in the fact that each train unit must be parked in a depot when not in use. The parking must be conducted in such a way that the depot capacity is utilised as good as possible, and so that all train units can be driven in and out of the depot when needed without being obstructed by each other. Most often depot tracks are only accessible from one end. This has the consequence that train units arriving to a given depot track first will also need to be the last to leave.

Since suburban railways are situated around larger cities, land is a scarce resource. For this reason the amount of space in the depots is often a highly limiting factor on the operation of suburban railways. This is to a substantial degree the case for DSB S-tog.

Furthermore, the topological layout of the tracks in the depots may also be a limiting factor. Some depot tracks may only be reached by from certain platform tracks, and sometimes even necessitating movements with multiple changes of direction and the use of the main line tracks.

Moreover, operational requirements as to the allowed types of train shuntings for coupling and decoupling train units also influence which rolling stock plans can be constructed.

Finally, each train unit must undergo scheduled maintenance at given time and service distance intervals. This is also a limiting factor in that it prescribes for which distance a train unit may be in service before it needs to be in a specific place at a specific time, namely at the maintenance workshop.

All of these requirements together (and more) constitute the highly complex problem of rolling stock planning, a problem that needs to be solved both within the tactical and operational time horizons.

A good rolling stock plan is a plan that adheres to all of the railway-specific requirements while at the same time minimising the economic costs of the operation. The total service distance completed is the cause for the largest portion of the economic costs for the rolling stock operation, however train shuntings in and out of depots may also be associated with an additional cost if more personnel is needed to perform the operations. A good rolling stock plan is also robust toward external influences such as delayed or cancelled train services.

The overall purpose of rolling stock planning is thus to supply seats in time and space to fulfil the passenger demand while minimising operating costs. This is conducted by assigning rolling stock train units to train services. As such, a train service is served by one or more train units. When passenger demand is high, train units providing a high seating capacity should be assigned to the individual train service. When demand is low, a minimum of train units may be assigned to the individual train service.

1.3 Industrial and Scientific Goals of This Thesis

Representing an industrial Ph.D. project, this thesis has both scientific and industrial goals. The industrial goals are related to the practical applicability of the research conducted.

As may be obvious by the explanations in the previous section, to construct a rolling stock planning model that takes all of the described requirements into account in an integrated manner is a difficult task. Nevertheless, this is the concrete and practical problem of the suburban railway operator, DSB S-tog, a practical problem for which a solution needs to be found.

To provide this solution, to provide an integrated rolling stock planning model, is the primary industrial goal of this Ph.D. thesis. This includes to demonstrate the real-world applicability of the integrated rolling stock model using real-world production data instances. Representing an industrial Ph.D. project, the primary goal of this thesis thus aims to stretch beyond an academic demonstration of the viability of a particular set of methods to solve isolated subproblems, subproblems that may even have been cherry-picked for academic reasons. The aim is to adhere to all railway-specific requirements (no exceptions) in an integrated manner and to do so using real, railway operation production data. The goal is thus to show that integrated rolling stock planning can actually be performed in a way applicable to an actual railway operator.

The secondary industrial goal is to compare the different solution approaches to integrated rolling stock planning which may be identified, so as to evaluate the implementation effort required to solve a given set of requirements, and to which degree this makes the model better with regard to requirements integration, the real-world application and the economic value of its solutions. This also includes a qualitative evaluation of the operational cost of having specific requirements in place.

Moreover, since the research conducted in the course of this industrial Ph.D. project is performed with the goal of being rolled out in a real railway-operation production organisation, an emphasis has been put in documenting the processes, solution methods and results. This includes the development of tools for the visualisation of the rolling stock plans themselves, and the data structures used to construct them.

The scientific goals of this thesis are related to the result-focused, industrial goals, however, the scientific goals are focused more in the direction of understanding “the whats, the hows and the whys” of the rolling stock planning problem and its solution.

The primary scientific goal is thus to identify suitable ways in which an integrated rolling stock planning model can be systematically constructed. This includes to identify suitable solution techniques, to design new ones and to compare them according to their properties, including processing time and solution quality. The latter touches upon the secondary scientific goal, namely to explore ways to quantify the improvement potential of existing rolling stock planning solutions. Naturally, both of the mentioned scientific goals require a systematic description of the rolling stock planning process itself. As such, this description can be viewed as the zeroth scientific goal.

1.4 Scientific Contributions of This Thesis

With the offset of the industrial and scientific goals described above, the main scientific contributions of the work described in this thesis are:

- A complete, structured and detailed description of the entire rolling stock planning process at DSB S-tog. This includes a description of the railway-specific requirements for rolling stock planning;

- The design, implementation and testing of a fully functional, integrated rolling stock planning model taking into account all the described railway-specific requirements in an integrated manner. In particular, the integration of train unit to train service assignment, maintenance planning (by distance) and depot planning is not known from literature. The model uses a greedy heuristic;

- The design and implementation of special side constraints to a resource constrained shortest path algorithm, side constraints that can handle the individual order of train units in train compositions so that no train unit will obstruct the movement of another. This novel concept is called *unit order flow conservation;*

- The design, implementation and testing of three different types economic value upper bound calculation models for rolling stock planning. With varying accuracy and processing time characteristics, theses models can be used to quickly calculate the upper bound of the economic value as an approximate measure of how good a rolling stock plan can be constructed based on a given set of input data;

- The design, implementation and testing of a fully functional branch-and-price based matheuristic integrated rolling stock planning model taking into account all railway-specific requirements, with the vast majority of requirements integrated into the optimisation part of the algorithm. This model is able to solve small instances (in the form of matheuristic iterations) to optimality. It is generally less time-effective than the greedy heuristic. However used in combination with the greedy heuristic model, it can achieve slightly better solutions than can be achieved with any of the models alone.

1.5 How This Thesis Is Structured

This thesis consists of nine chapters, divided into three parts.

Part I is an introduction to rolling stock planning for a suburban railway operator. The current introductory Chapter 1 leads forward to Section 1.6 (below) in which the terminology used throughout the thesis is defined. Next, Chapter 2 describes the rolling stock planning process of DSB S-tog and its subprocesses. A rolling stock plan must adhere to a number of practical, railway-specific requirements. These requirements are described in detail in Chapter 3.

Part II deals with ways to model and to solve the rolling stock planning problem. Its first Chapter 4 contains an overview of related work and presents a short overview of the five rolling stock planning models developed for this thesis. Next, in Chapter 5, the greedy heuristic based integrated rolling stock planning model is introduced. In Chapter 6, three different economic value upper bound calculation models for rolling stock planning are presented. Then in Chapter 7, the branch-and-price matheuristic integrated rolling stock planning model is presented.

Part III puts the work carried out into perspective, the results achieved and lessons learned. Chapter 8 discusses the implications of the developed models, whereas Chapter 9 presents an outlook to further research.

The appendix contains additional information related to the work, including a presentation of the developed visualisation tools for rolling stock planning, auxiliary rolling stock planning methodologies, implementation details and metrics and a description of the types of infeasibilities that may occur in rolling stock plans as a consequence of the step-by-step manner by which they may be constructed.

1.6 Terminology

This section defines the most important terms used throughout this thesis. References to more detailed descriptions of the terms and their implications are given where applicable. Terms in *italics* refer to other definitions in the list. The terms appear in a logical order for their explanation. The list can be read as a dense and detailed introduction to rolling stock planning.

- A **station** is a point in space where a *train service* may stop to allow passengers to get on or off;

- A **train service** is the concept of transport using *train units* on the main line railway tracks, provided as a service to passengers and/or to perform *positioning* of *train units*.

A *train service* runs between an *origin station* and a *terminal station* stopping at or *skipping* zero or more *intermediate stations* on the way at points in time as scheduled in the *timetable*;

- A **revenue train service** is a *train service* provided for the transport of passengers for revenue;
- A **non-revenue train service** is a *train service* that runs without passengers in order to *position* the *train units*;
- **Skipping** is when a *train service* passes a *station* without stopping. *Skipping* occurs as scheduled in the *timetable* for express *train services*, in disruption management for delayed *train services* to catch up, and for *non-revenue train services* that carry no passengers;
- **Positioning** is the process of moving *train units* in one or more *train services* in order to meet later demand for seats or technical maintenance etc. at other points in space;
- **Revenue positioning** is the *positioning* of *(virtual) train units* by providing more *(virtual) train units* in a *revenue train service* than is in demand by passengers, thus offering excess seating capacity;
- **Non-revenue positioning** is *positioning* in *non-revenue train services* with no passengers. This is also known as dead-heading;
- A **timetable** is a complete list of *revenue* and *non-revenue train services* for a given period of time. Only the *revenue train services* are published to the general public. For details, see Section 3.2 on page 35;
- An **origin station** is a *station* from which a *train service* starts;
- A **terminal station** is a *station* at which a *train service* ends;
- An **intermediate station** is a *station* during the run of a *train service* at which the *train service* may either be stopping or *skipping*;
- The **train drivers** constitute the staff group performing *train services*. *Train drivers* also perform those *train shuntings* being to and from *side tracks*;
- A **depot** is the entire infrastructure at a *station* used for *train shunting* and parking of *(virtual) train units* at *depot tracks*. All *depots* have facilities for cleaning and some have facilities for maintenance;
- A **maintenance depot** is a *depot* with maintenance facilities;
- A **depot station** is a *station* which has a *depot*;
- A **depot track** is a track at a *depot* where *(virtual) train units* may be parked when not running as a *train service*;
- A **split depot** is a *depot* at a *station* at which some of the *depot tracks* are only reachable from some of the *platform tracks* and vice verse. Hillerød station has as split depot, as seen on Figure 3.1 on page 35;
- A **terminal depot** is a *depot* located at a *terminal station*;

- A **same direction depot** is a *terminal depot* located so that *train services* arriving to its *depot station* may reach the *same direction depot* by continuing through the *depot station* in the same direction of movement. *Train shuntings* arriving from a *same direction depot* may also continue through its *depot station* in the same direction of movement to become *train services* departing from that *depot station*. Høje Tåstrup station has a *same direction depot*, as seen on Figure 3.1 on page 35. Note that the tracks below the station on Figure 3.1 belong to the maintenance workshop;

- An **opposite direction depot** is a *terminal depot* located so that *train services* arriving to its *depot station* may only reach the *opposite direction depot* by changing direction of movement at the *depot station*. *Train shuntings* arriving from an *opposite direction depot* must also change their direction of movement at the *depot station* to become *train services* departing from that *depot station*. Køge station has a *opposite direction depot*, as seen on Figure 3.1 on page 35;

- A **intermediate depot** is a *depot* located at an *intermediate station*. Train units entering an *intermediate depot* may continue in the same direction or must change direction depending on the direction in which they are arriving at the *intermediate depot*. København H station has an *intermediate depot*, as seen on Figure 3.1 on page 35;

- A **platform track** is a track at a *station* where a *train service* may stop and allow passengers to get on or off. A *platform track* may also be temporary used for the parking of *(virtual) train units*;

- A **side track** is a track that can only be used for parking in the day time. There is no internal *train shunting* between side tracks;

- A **side track station** is a station which has *side tracks*. As opposed to a *depot station*, there are no facilities for cleaning or maintenance;

- A **train shunting** is the operation of coupling and decoupling *(virtual) train units* at *depot stations* or *side track stations* as well as moving the *(virtual) train units* to and from *platform tracks, depot tracks* and *side tracks*;

- The **depot drivers** constitute the staff group generally performing those *train shuntings* that are in and out of a *depot*;

- A **train service line** is an aggregation of similar *train services* according to the time of day they are running, the stations they are visiting etc.;

- A **train service sequence** is a consecutive sequence of *train services*, on the same (or related) *train service line*, for which it is a natural choice that the *(virtual) train units* be reused from one *train service* to the next, without visiting a *depot* in between. The train service sequences of a *timetable* may be determined from the layout of tracks, the minimum and maximum turnaround times between two consecutive *train services* at the *origin* and *terminal stations* and by the *braiding* policy. The concept of *train service sequences* is used to ease the manual rolling stock planning process. The first and last *train service* in a *train service sequence* may not require a *depot driver* to perform *train shunting* of the corresponding *(virtual) train units* out from or into the depot, since the *train driver* for the *train service* has time to perform this operation as his first or last task in his duty. See Figure 2.2 on page 28 for an example of a circulation diagram showing *train service sequences*;

- **Braiding** is when there are *train services* from different *train service lines* in the same *train service sequence*. *Braiding* may yield better utilisation of the *(virtual) train units* at the cost of a lower robustness since disruptions may then propagate between *train service lines*. Under certain conditions *braiding* may produce *train service sequences* that represent one direction of one *train service line* and the opposite of another, a highly undesirable feature from a robustness point of view. Forced *braiding* may also be used to raise the robustness by forcing higher turnaround times;

- **Depot internal shunting** is the process of *train shunting* between depot tracks at the same *depot station*. A *platform track* may be involved in the process, but *depot internal shunting* starts at one *depot track* and finishes at another;

- A **train service segment** is the individual part a *train service* performs between *depot stations* or *side track stations* for that particular *train service*. Since there are no *depot stations* or *side track stations* en route on a *train service segment*, the *train composition* of a *train service segment* will remain constant. If there are more than one *train service segment* to a particular *train service* the *train composition* can change in the course of the *train service*;

- A **train shunting segment** is the individual part of a *train shunting* between *platform tracks*, *depot tracks* or *side tracks* for a particular *train shunting*. Analogous to *train service segments*, the *composition* remains constant in a *train shunting segment*, but since (by DSB S-tog business rule) there can only be one *train shunting segment* per *train shunting*, the *train composition* remains constant also for each *train shunting*;

- A **parking segment** captures the possibility of parking one or more *(virtual) train units* at a specific *depot track, side track* or *platform track* for a given period of time. *Parking segments* have a capacity corresponding to the entity they represent;

- A **train unit** is the actual, physical, individual, inseparable railway vehicle. For details, see Section 3.3 on page 39;

- A **virtual train unit** is an anonymous *train unit*. As such it has all the characteristics of a *train unit*, except the fact that it is not known which *train unit* it actually is. *Virtual train units* are used in planning in the tactical time horizon when one does not know which of the physical *train units* will be available when the plan is to be executed in the operational time horizon;

- The **train unit type** is the technical type of a *train unit*. For details see Section 3.3 on page 39;

- A **train unit trajectory** is the path a *train unit* moves through consecutive *train service segments, train shunting segments* and/or *parking segments* to fulfil its tasks over a period of time;

- A **train composition** is the ordered sequence of coupled, individual *(virtual) train units* assigned to an individual *train service segment, train shunting segment* or *parking segment*;

- A **train composition type** is the anonymous, non-ordered *train composition* specifying only *train unit types*. For details, see Section 3.3 on page 39;

- A **total composition exchange** is when all *(virtual) train units* in two consecutive *train service segments* in a *train service sequence* are exchanged, see Figure 2.2 on page 28;

- A **partial composition exchange** is when only some of the *(virtual) train units* in two consecutive *train service segments* in a *train service sequence* are exchanged, see Figure 2.2 on page 28;

- A **rolling stock plan** is the assignment of all available *(virtual) train units* to *train unit trajectories* that combined satisfy the operational requirements. The set of *train unit trajectories* implicitly determines the *train composition* of each *train service segment, train shunting segment* and *parking segment*. This implies how much seating capacity is offered in the individual *train service segments* and when and where *(virtual) train units* are parked in the depots and in what order.

Chapter 2

The Rolling Stock Planning Process

Rolling stock planning is the process a railway operator conducts in order to plan how to utilise its rolling stock for the conveyance of passengers.

At DSB S-tog, the rolling stock planning process is currently subdivided into two main subprocesses. The first main subprocess is the long-term, tactical time horizon *circulation planning* in which planning is conducted for anonymous, virtual train units months or weeks in advance. The second main subprocess is the short-term, operational time horizon *train unit dispatching*, which is the execution of the long-term circulation plan, conducted days, hours or even just minutes before the actual operation takes place. In the train unit dispatching process, the anonymous, virtual train units from the circulation plan will have real, physical train units assigned to perform the actual, real-world train services. This process also deals with recovery when disruptions occur.

Figure 2.1 shows an overview of the current rolling stock planning process at DSB S-tog and its subprocesses. In Table 2.1 different characteristics for the two specific subprocesses *circulation planning* and *train unit dispatching with disruptions* are compared. For a concrete impression of the contents of a rolling stock plan, see Figure 2.2, and for a real-world impression of the rolling stock operation itself, see Figure 2.3.

Currently, the rolling stock planning process at DSB S-tog is handled in two different information systems:

1. **The manual rolling stock planning system**, an online transaction processing system (OLTP) facilitating the manual construction and editing of circulation plans by the planners. Upon completion, the long-term circulation plans are executed as short-term train unit dispatching plans and the entire train unit dispatching process is managed using this system. The system is a legacy system providing no decision support, however, all railway-specific requirements of the planning process can be handled manually in it;

2. **The automated circulation planning system**, an offline decision support system (DSS) in which planners can construct circulation plans using automated optimisation tools. The circulation plans can then be exported to the rolling stock planning and dispatching system mentioned above. As will be clear from Appendix D, the automated system does not handle all of the railway-specific requirements. For this reason, manual changes to the circulation plan created with the automated circulation planning system will have to be performed upon import into the manual rolling stock planning system. The automated circulation planning system is still a non-legacy system.

At DSB S-tog, the routing of train services on the main line and through stations is not considered a part of the rolling stock planning process, but rather a part of the timetabling process.

Table 2.1: Overview of characteristics for processes *circulation planning* and *train unit dispatching with disruptions*.

Subject	**Characteristics of circulation planning**	**Characteristics of train unit dispatching w/disruptions**
Planning scope	The everyday operation	For occasionally occurring disruptions
Time horizon	Tactical	Operational
Time for processing	Long	Very short
Goal	Stable operation	To recover to stable operation
Instruments used	A limited number (to ensure simplicity and robustness)	Every possible one (to ensure operation is resumed quickly)
Requirements	Strict (planned violation not tolerated)	Not so strict (one-time violations tolerated)
Cost of plan	Important (cost is incurred every day)	Not so important (cost is incurred only once)

In the rolling stock planning process one can thus assume that there is sufficient capacity on the main line tracks and in the stations to operate the given timetable.

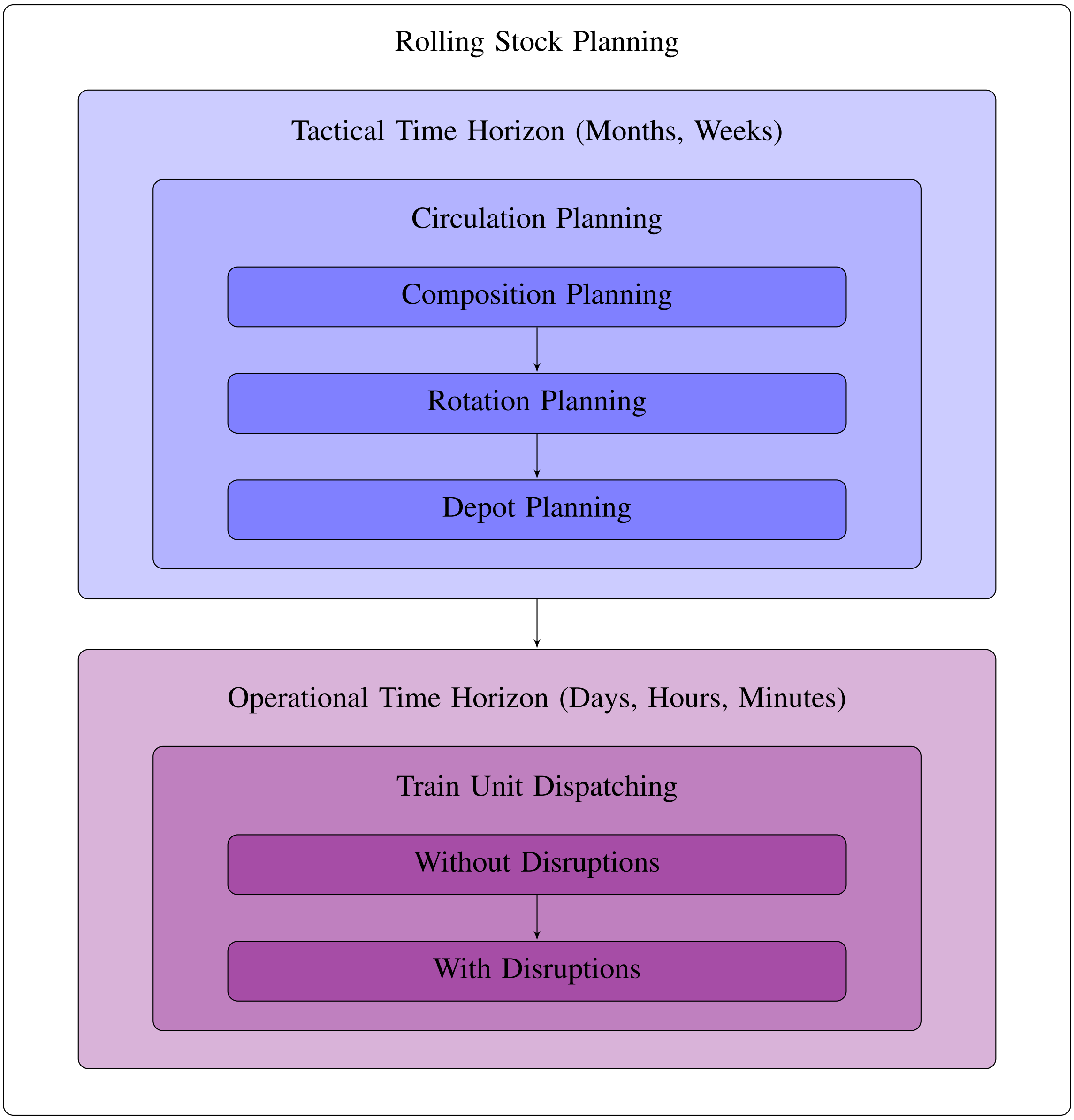

Figure 2.1: The rolling stock planning process at DSB S-tog and its subprocesses. As in Figure 1.1 on page 16, processes in the tactical time horizon are coloured blue and processes in the operational time horizon violet. The current DSB S-tog *manual rolling stock planning system* may handle the entire rolling stock planning process, the current DSB S-tog *automated circulation planning system* the circulation planning process only.

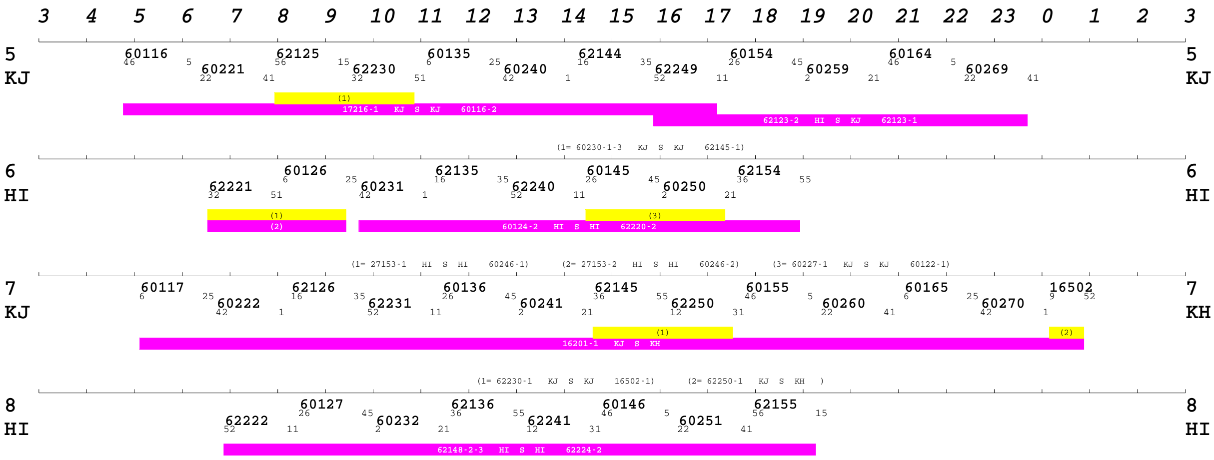

Figure 2.2: Example of a DSB S-tog rolling stock circulation diagram as printed by the current manual rolling stock planning system. The diagram shows four train service sequences on line E for weekdays in the week starting with Monday 2014-03-31. Train service sequences are groups of train services for which it is a natural choice to execute in sequence by the same train unit. Train service sequences are determined by the current braiding policy and minimum and maximum turnaround times. Train service sequences numbers 5 to 8 are written to the left and right of the diagram, along with their origin and terminal station abbreviations, HI for Hillerød, KJ for Køge and KH for København H. Each bar in the diagram represents a virtual train unit. Magenta bars are virtual train units of the longer type 1, yellow bars of the shorter type $\frac{1}{2}$. The 5-digit train service numbers of the train services belonging to the train service sequences are written above the bars. Hours are written on the top of the diagram, departure and arrival times in minutes past the hour (two digits) just below the train service numbers. For instance, train service sequence 6 starts with the departure from Hillerød (HI) at 06:32 with train service 62221 arriving in Køge (station not explicitly shown on diagram) at 07:51 and continuing from Køge again at 08:06 with train service 20126 back to Hillerød with arrival 09:25. For this part of the train service sequence a train composition type $1\frac{1}{2}$ is assigned, with the yellow short train unit type $\frac{1}{2}$ virtual train unit being in the north end of the train composition (i. e., at the top of the bar), and the magenta long train unit type 1 virtual train unit at the South end of the train composition (i. e., at the bottom of the bar). A total train composition exchange is then performed with the former train composition being shunted into the depot at Hillerød at 09:25, and a new train composition formed by one train unit type 1 virtual train unit being shunted to the platform to perform train service 60231 at 09:42. At 14:26 this train composition is being changed to a type $1\frac{1}{2}$ again in Køge by the addition of a train unit type $\frac{1}{2}$ virtual train unit. This virtual train unit is decoupled again after train services 60145 and 60250 have been performed. The last train service of train service sequence 6 is thus performed only with the train unit type 1 virtual train unit ending in Hillerød at 18:55.

Figure 2.3: Impressions of the rolling stock operation just south of København H station, Thursday 2012-12-13 at 12:20. The red DSB S-tog train unit to the left is heading south on the main line tracks towards Dybbølsbro station. The middle, red train unit is in the process of being shunted from the depot to København H station to enter revenue train service. Note the special livery to advertise the then new flexible space sections in the middle of the train unit. The right, red train unit is heading north on the main tracks towards platforms 9/10 at København H [73].

2.1 Circulation Planning

The tactical time horizon, long-term part of the current rolling stock planning process at DSB S-tog is called circulation planning. According to the current protocol, the circulation planning process must be started at least three months before and completed at least three weeks before the plan is to be executed.

For this reason, at the time the circulation planning is conducted, it is not known which physical train units are available when the plan is to be executed. Some train units may be in unscheduled maintenance (see Section 3.7 on page 48). Furthermore, it is not known where the physical train units are located at the time the plan is to be executed, since changes to the previous plan may have occurred. For this reason the circulation planning is performed for *virtual train units*, that is, anonymous train units which have no individual characteristics apart from those involved in the plan.

In the following, the processes of circulation planning at DSB S-tog are described as they occur in the automated circulation planning system currently in operation. As will be clear in Appendix D, the current system has its deficiencies, leading to some circulation plans being constructed completely manually, that is, with no interaction with the automated system. The manual construction process is somewhat in reverse order of the automated process, where the first step of the manual process represents the last step of the automated process and vice versa.

No attempt shall be made here to describe the specifics of the manual planning process.

In the automated circulation planning system currently in operation at DSB S-tog, the circulation planning is performed as three separate subprocesses that are executed one after the other as *composition planning, rotation planning* and *depot planning*. These three subprocesses are described in the following sections, and may also be seen in Figure 2.1.

2.1.1 Composition Planning

The subprocess of deciding how many virtual train units to assign to a train service in order to meet passenger demand is designated *composition planning*. The main requirements to this process are given by the infrastructure, the rolling stock, the timetable and the passenger demand (see Sections 3.1 to 3.4).

The output of the process is a composition plan defining the amount of virtual train units by type to assign to each train service segment, making sure that the overall depot track length of each of the depots is not exceeded by parked train units at any time, that each virtual train unit can only be part of one train service at a time, and that train unit balance is kept at all depot stations.

2.1.2 Rotation Planning

The next subprocess is designated *rotation planning* and deals with deciding which virtual train unit is to move from a position in a train composition serving one train service segment to a position in a next train composition serving the next train service segment (possibly with a stop-over at a depot). This subprocess also takes into account when and where the virtual train units should undergo scheduled maintenance (see Section 3.6). The composition plan and scheduled maintenance form the main requirements for this subprocess.

Typically, a train unit in service on a particular train service line stays on that train service line when changing direction at the terminal station. Thus a train unit used in a train service on a train service line in one direction typically "rotates" to become part of a train service in the opposite direction on the same train service line when it reaches the terminal station.

As mentioned in Section 1.6, a series of train services for which it is a natural choice that the rolling stock train units will be rotating from one train service to the next is called a *train service sequence*. Normally, train service sequences cover only one train service line. Under certain conditions however, it may be more desirable to *braid* (to intertwine) two train service lines to use the same rolling stock train units so that the train units change the train service line at one common terminal station.

2.1.3 Depot Planning

The subprocess of deciding how a virtual train unit is to be parked in a depot (when not needed to perform train services) and how it is to be retrieved again (when it is needed once more) is designated *depot planning*. Apart from the rotation plan produced in the previous step, the main requirements for this subprocess are given by the infrastructure and the personnel on duty (see Sections 3.1 and 3.5).

Some train operators do not consider depot planning a subprocess of circulation planning, but rather a separate process in itself [10]. This is presumed to be for the reason that those operators have sufficient capacity in the depots to perform the depot planning process completely independent of the rest of the rolling stock planning processes. As mentioned, this is not the case for DSB S-tog.

In literature, the term *shunting* is used for the process of coupling and decoupling of train units and parking them in the depot [55]. At DSB S-tog, the more general term *depot planning* is used including other processes like cleaning as well, not only shunting.

2.2 Train Unit Dispatching

At DSB S-tog, the operational time horizon process of executing a circulation plan (that is, putting the plan into motion) is called *train unit dispatching*. The time horizon for this process ranges from days to minutes. The overall purpose of the process is to assign actual, physical train units to the virtual train units in the circulation plan.

The train unit dispatching process has two subprocesses each of which are performed under different conditions. The process of dispatching train units is as such different depending on whether disruptions are occurring or not, and different requirements must be taken into account in the two subprocesses. The subprocesses themselves are shown in Figure 2.1 on page 27 and explained in the following two sections.

2.2.1 Train Unit Dispatching in a Situation without Disruptions

When no disruptions occur, the objective of the train unit dispatching process is to assign one physical train unit to each virtual train unit in the circulation plan. While doing so, it must be taken into account that the virtual train unit will need to go into maintenance before the maintenance service distance limit of the physical train unit has been reached. Furthermore, all of the requirements which were not taken into consideration in the circulation planning process need to be considered. This is shown in Table 3.1 on page 34.

The general idea (and purpose) is that the requirements already handled in the circulation planning process need not be considered in the train unit dispatching process when no disruptions are occurring.

2.2.2 Train Unit Dispatching in a Situation with Disruptions

In reality disruptions may occur at any time, influencing how the plan (or parts of it) may be realised. At DSB S-tog, when a disruption occurs, the *train controller* of the infrastructure manager (Banedanmark) is in charge of the overall recovery process, with the *train unit dispatchers* of DSB S-tog as co-operating partners.

In the case of disruptions, the train controller and train unit dispatchers may decide to disregard some of the requirements that needed consideration in the situation without disruptions. This is as to be able to handle a disruption before it gets out of hand. In the event of a disruption, it is of the utmost importance that sufficient action is taken sufficiently quickly so as to contain the disruption and to prevent it from propagating onto the entire network. Furthermore, the action taken must also facilitate the return to normal service as quickly and with as few changes to the original plan as possible. The instruments the train controller may use to conduct the recovery process may include:

- Cancelling individual train services, partially or completely;
- Cancelling train service lines, partially or completely;
- Making individual train services skip planned stops at stations to make up for lost time;
- Delaying train services.

The characteristics of the disruption and the instruments applied to remedy it provide the conditions the train unit dispatchers will have to compensate for.

In the event of a disruption directly caused by a sudden train unit breakdown, requiring that the train unit must undergo unscheduled maintenance, the train controller and train unit dispatchers may choose solution strategies depending on the speed and acceleration characteristics of the train unit, and whether it is allowed to carry passengers. Solution strategies may include off-loading passengers and getting the train unit temporarily out of the way or directly into the workshop. Another train unit can then be picked from the reserves to replace the one that has been taken out of service.

In the recovery process after a disruption, one goal of the train unit dispatchers is to reach the depot balance of the original circulation plan. By doing so, the dispatchers may ensure that the actual operations can actually return to being according to the original circulation plan.

When reinstating train services on cancelled train service lines, certain rules must be adhered to. For example, if reinstating a cancelled train service, all subsequent train services on the given train service line must also be reinstated.

In the event of a disruption, the train unit dispatchers have at their disposal instruments that are otherwise not available. These instruments include the cancelling of train services, disregarding business rules regarding train unit order in the train composition, disregarding rules about number of train units in a train composition, disregarding rules as to how trains units may be coupled, and others. These "dirty tricks" are allowed in the recovery process to prevent worse things from happening.

After a disruption, the physical train units may not be parked according to the circulation plan at all. Which physical train units may run as which virtual train units (as defined in the circulation plan) is then highly dependent on how the physical train units are actually parked in the depot.

As may be concluded from the previous paragraphs, in the event of disruptions, the work of the train unit dispatchers always deals with compensating for the unexpected events that may occur. If a train unit breaks down, a compensation train unit must be provided from the reserves. If train services are cancelled in the event of a disruption and the depot balance of a depot is not according to plan, train units must be positioned to compensate for this.

Chapter 3

Railway-Specific Requirements for Rolling Stock Planning

This chapter describes the practical, railway-specific requirements for rolling stock planning for the case of DSB S-tog. For an overview of the requirements and in which subprocess they are handled, refer to Table 3.1. For a full description of the requirements, refer to [103].

3.1 Infrastructure

The perhaps most important requirements for rolling stock planning are given by the railway infrastructure. The railway infrastructure consists of tracks, points (switches), stations, platforms, depots and maintenance workshops.

The infrastructure available to DSB S-tog is shown in Figure 3.1. At present, DSB S-tog is the sole operator on this infrastructure. As may be seen, the infrastructure consists of 6 *fingers*, with a shared double track *central segment* between stations Svanemøllen and København H and the so-called *circular line* around this central segment.

The track part of the infrastructure requirements relates to track topology and physical track capacities. Track topology is how the individual tracks are interconnected. The physical track capacities limit how many train units may use a particular part of the infrastructure at a time.

For the case of DSB S-tog, the track infrastructure at the central segment is a highly limiting factor as to which train operations may be performed. Since most train services pass through it, the track capacity in this segment is highly utilised.

In relation to rolling stock planning, the depot infrastructure is also is a highly limiting factor on the possible train operations. The track capacity of the individual depots is in itself very limited. Moreover, the track topology of some depots is strongly limiting the movements that may be conducted with the rolling stock.

At DSB S-tog, the routing of train services on the main line and through stations is a part of the timetabling process. As such, the infrastructure requirements (headway, routing, etc.) for the train services to be performed on the main line tracks have already been met with a given timetable and need not be considered in the rolling stock planning process.

Some stations have depot tracks while other stations have side tracks, the difference being that cleaning and overnight parking can be undertaken at depot tracks, whereas side tracks can only be used for day parking. At some stations overnight parking at the platform tracks is allowed in special cases.

Other infrastructure related requirements are the *platform track usage rules*, business rules stating which track to use for a specific line and direction.

Table 3.1: The requirements for rolling stock planning at DSB S-tog, their independence to the individual train unit and the subprocess in which they are handled.

			Subprocess		
Section	Requirement	Independent of individual train unit	Circulation planning	Train unit dispatching without disruptions	Train unit dispatching with disruptions
3.1	Infrastructure	•	•		•
3.2	Timetable	•	•		•
3.3	Rolling Stock	•	•		•
3.4	Passenger Demand	•	•		•
3.5	Personnel on Duty	•	•		•
3.6	Scheduled Maintenance		•	•	•
3.7	Unscheduled Maintenance			•	•
3.8	Friction Sand			•	•
3.9	Exterior Cleaning			•	
3.10	Exterior Graffiti Removal			•	
3.11	Interior Cleaning			•	
3.12	Winter Preparedness			•	
3.13	Exposure of Commercials			•	
3.14	Surveillance Video Requests			•	
3.15	Surface Foil Application			•	
3.16	Passenger Counting Equipment			•	
3.17	Train Control System Equipment		•	•	•

Train composition movements, that is, the motion, coupling and decoupling of train units are governed by the infrastructure in four ways:

1. The train control system enforces rules for the train composition movements depending on whether the train service is a revenue or non-revenue train service and whether the last train composition in the coupling process is arriving from the depot or from the main line. Simplified, it is allowed to have any train composition parked at a platform track and then coupling this train composition with a train composition arriving from the depot or a non-revenue train service arriving from the main line. It is disallowed to couple a train composition running as a revenue train service arriving from the main line with any train composition parked at the platform track;

2. In the process of decoupling a train composition from another, a business rule disallows to decouple and let the train composition that is going to continue as a revenue train service depart before the other train composition is driven into the depot. This is because passengers may get confused when parts of the train composition are in service, and other parts not. Furthermore, a train shunting conducted after the train composition serving the revenue train service has departed may not leave time for a depot driver to perform

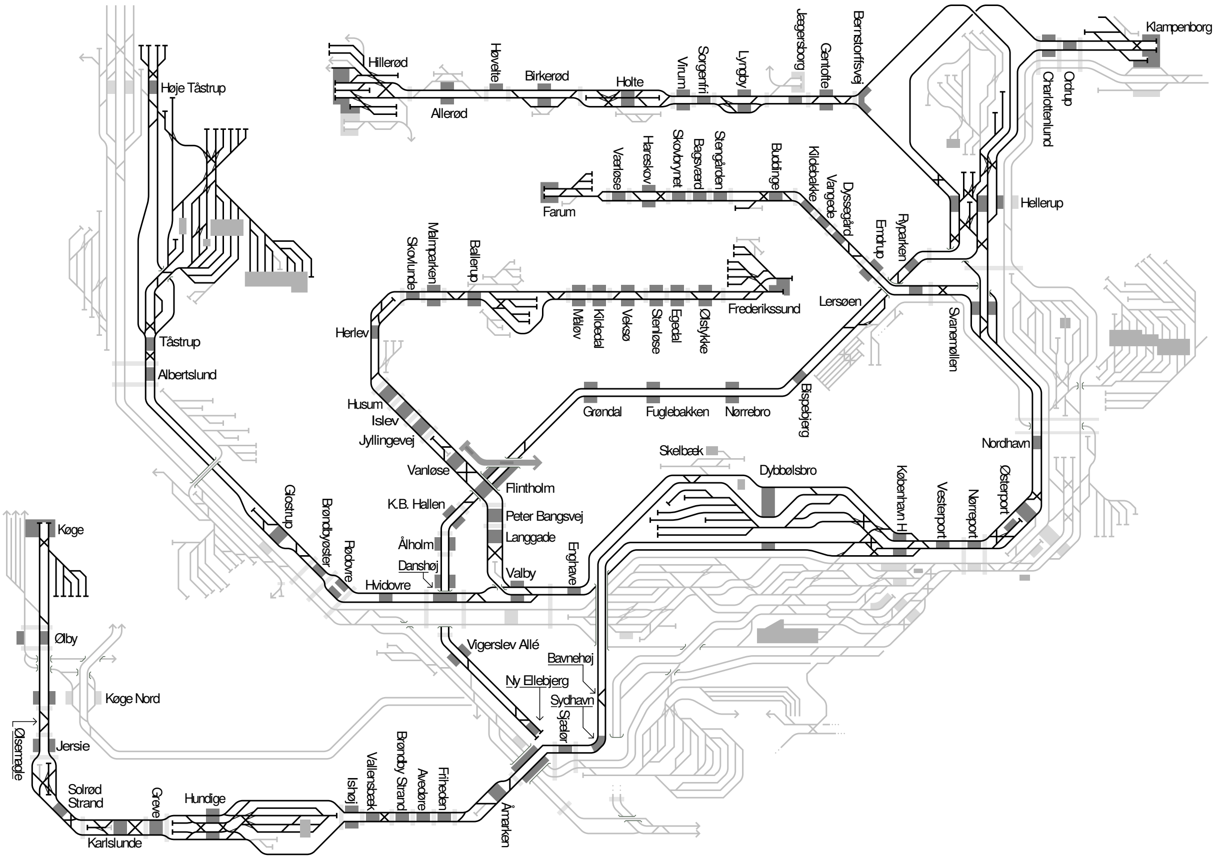

Figure 3.1: Topological map of the railway infrastructure available to DSB S-tog. Tracks which may be used by DSB S-tog are printed in black. Other tracks used by the long distance train services and the metro (only connecting tracks are printed) are printed in grey [78].

another train shunting for the next train service, if this movement is to occur before the next revenue train service departs;

3. Another business rule states that no coupling or decoupling may take place on a train composition en route from its origin station to its terminal station. This would run the risk of being too time consuming and thus delay other train services. When changes in the number of train units in a train composition must be performed en route, this must be conducted by performing a total composition exchange;

4. A business rule states that there can be only one train shunting to each coupling or decoupling. As such, it is disallowed to split a train composition into two before driving each of the two new train compositions into the depot. The opposite movement is also disallowed. This business rule is in place to enforce simplicity and robustness, if more movements were allowed, platform tracks would be occupied for longer periods of time, running the of risk of delaying otherwise uninvolved train services.

3.2 Timetable

Another main requirement for rolling stock planning is given by the timetable. The timetable states which train services must be run, and each train service must have rolling stock assigned to it in order to run. Table 3.2 shows an example of a DSB S-tog timetable. The timetable implicitly states which train service lines will be running between stations and where and when train services will be stopping at stations, see Figure 3.2.

For the case of DSB S-tog, the timetable is highly dependent on the existing contract with the Ministry of Transport [106]. This contract states a minimum number of train services to be run on given fingers at given time intervals and at which minimum frequency the different stations must be served. As such, DSB S-tog may only to a very limited degree vary the supplied seats in time and space by varying the frequency of operation. Thus, the seat supply must be varied using different train composition types. Also, DSB S-tog is allowed to redistribute a certain number of train service kilometres within the contract.

Timetables come in various sorts: The *standard timetable* covers the base, standard case when no extraordinary events are planned anywhere on the network. Subordinate *extraordinary timetables* cover special cases, for example when infrastructure works on a particular part of the network are performed. Typically an extraordinary timetable is very much like the standard timetable, since a minimal difference is desired. When the differences are small, planning is easier, and most importantly, the job of informing the public and employees of the changes is much less complicated. When larger infrastructure works are conducted, however, it may not be possible to achieve this similarity. In this case a *standard extraordinary timetable* for the general case of the infrastructure works is constructed with subordinate *extra-extraordinary timetables* relating to the standard extraordinary timetable.

In some years, the standard timetable may only be in effect on a very limited number of days, see Table 3.3. All the other days, extraordinary timetables are in effect.

Table 3.2: A page from the DSB S-tog timetable valid from December 2013 to December 2014 for weekdays. The table shows the departure times in minutes past the hour for both directions of the train service lines E and A, respectively. Hours of operation are noted at the bottom. Train services with white background are only operated in the day time. There are different timetables in effect on weekdays and at night, with fewer train service lines operating.

Mandag-fredag Monday-Friday

Dagtimer Daytime

E Hillerød

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

E Køge

Hillerød - København H
ma-fr 5.02 - 0.02* 5.52 - 18.32

København H - Køge
ma-fr 5.22 - 0.22** 6.12 - 18.52

*Linje B fra Hillerød afg. 0.26
fortsætter til København H ank. 1.09.
Standser ved alle stationer.

**Sidste linje A fra København H 0.54
fortsætter til Køge ank 1.37

Køge - København H
ma-fr 4.46 - 23.46* 5.56 - 18.16

København H - Hillerød
ma-fr 5.05 - 0.25** 6.15 - 18.55

*Sidste tog fra Køge er linje A afg. 0.09 og 0.29
ank. København H 0.53 og 1.13.
Standser ved alle stationer.

**Sidste linje B fra København H 0.37
fortsætter til Hillerød ank 1.19

Mandag-fredag Monday-Friday

Dagtimer Daytime

A Farum

↓	↓	↓	↓	↓	↓		↑	↑	↑	↑	↑	↑
16	26	36	46	56	06	Farum	10	20	30	40	50	00
20	30	40	50	00	10	Værløse	06	16	26	36	46	56
23	33	43	53	03	13	Hareskov	03	13	23	33	43	53
25	35	45	55	05	15	Skovbrynet	00	10	20	30	40	50
27	37	47	57	07	17	Bagsværd	58	08	18	28	38	48
29	39	49	59	09	19	Stengården	57	07	17	27	37	47
31	41	51	01	11	21	Buddinge	55	05	15	25	35	45
32	42	52	02	12	22	Kildebakke	53	03	13	23	33	43
34	44	54	04	14	24	Vangede	51	01	11	21	31	41
36	46	56	06	16	26	Dyssegård	49	59	09	19	29	39
38	48	58	08	18	28	Emdrup	47	57	07	17	27	37
40	50	00	10	20	30	Ryparken	45	55	05	15	25	35
42	52	02	12	22	32	Svanemøllen	43	53	03	13	23	33
44	54	04	14	24	34	Nordhavn	41	51	01	11	21	31
47	57	07	17	27	37	Østerport	39	49	59	09	19	29
49	59	09	19	29	39	Nørreport	36	46	56	06	16	26
51	01	11	21	31	41	Vesterport	34	44	54	04	14	24
54	**04**	**14**	**24**	**34**	**44**	**København H**	**33**	**43**	**53**	**03**	**13**	**23**
55	05	15	25	35	45	Dybbølsbro	29	39	49	59	09	19
58	08	18	28	38	48	Sydhavn	27	37	47	57	07	17
00	10	20	30	40	50	Sjælør	25	35	45	55	05	15
01	11	21	31	41	51	Ny Ellebjerg	23	33	43	53	03	13
04	14	24	34	44	54	Åmarken	21	31	41	51	01	11
06	16	26	36	46	56	Friheden	19	29	39	49	59	09
08	18	28	38	48	58	Avedøre	17	27	37	47	57	07
10	20	30	40	50	00	Brøndby Strand	15	25	35	45	55	05
13	23	33	43	53	03	Vallensbæk	12	22	32	42	52	02
15	25	35	45	55	05	Ishøj	10	20	30	40	50	00
18	28	38	48	58	08	Hundige	07	17	27	37	47	57
20		40		00		Greve	04		24		44	
23		43		03		Karlslunde	02		22		42	
27		47		07		Solrød Strand	58		18		38	

A Hundige / Solrød Strand

Farum - København H
ma-fr 4.56 - 0.16 6.06 - 18.26

København H - Hundige
ma-fr 5.14 - 0.54* 6.24 - 18.44

København H - Solrød Strand
ma-fr 5.54 - 17.54

*Sidste tog fra København H afg. 0.54
fortsætter til Køge ank. 1.37.
Standser ved alle stationer.

Solrød Strand - København H
ma-fr 6.38 - 18.38

Hundige - København H
ma-fr 4.47 - 0.47 5.57 - 18.17

København H - Farum
ma-fr 5.13 - 0.33* 6.23 - 18.23

* Fredag kører sidste tog 0.53

Table 3.3: Number of days per year the standard timetable was in effect on the entire network of DSB S-tog.

Year	2008	2009	2010	2011	2012	2013	2014	2015
Number of days in effect	72	177	27	53	80	177	124	61

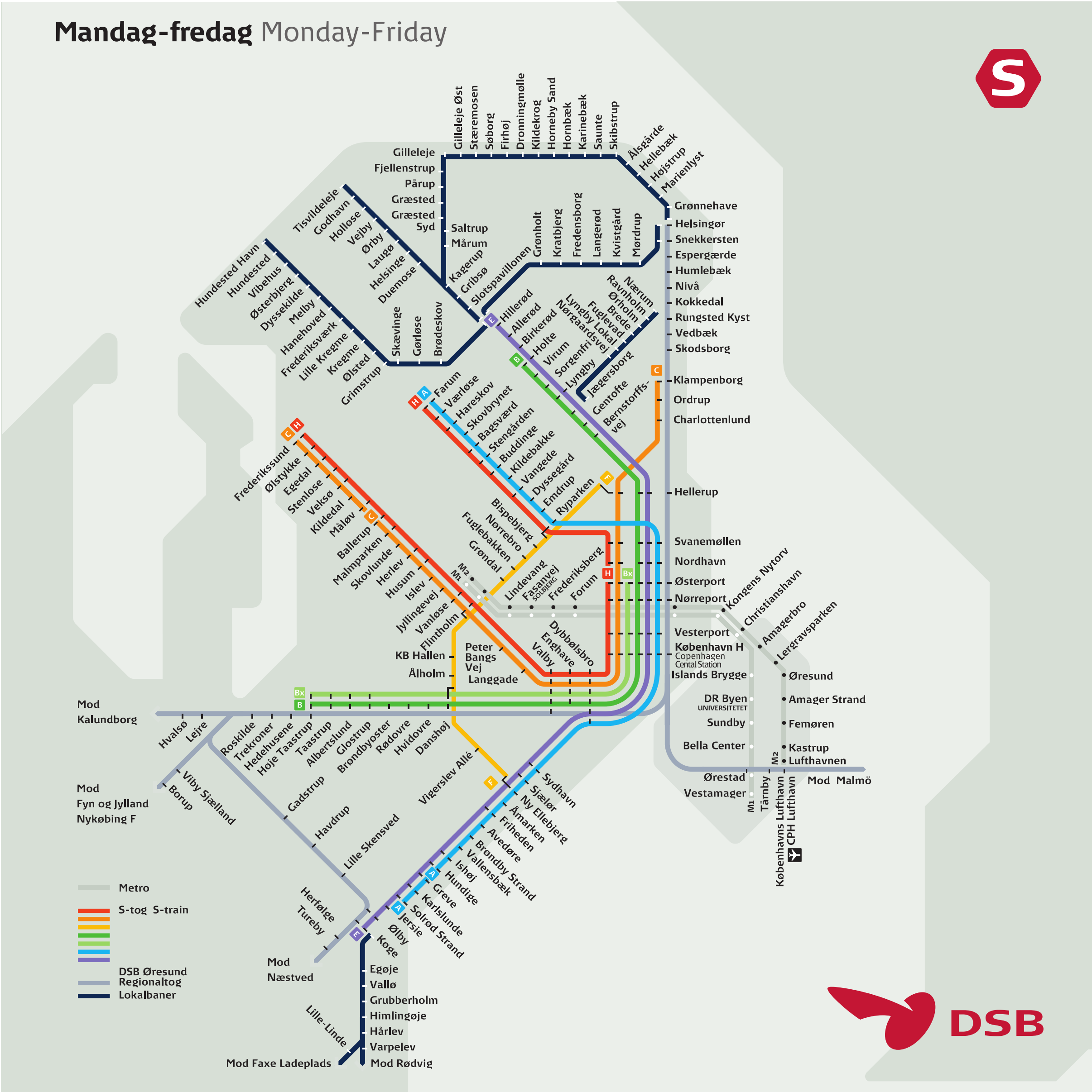

Figure 3.2: The train service line map of the Greater Copenhagen area for the timetable valid from December 2013 to December 2014 for weekdays. DSB S-tog train service lines are in rainbow colours. The train service lines of the metro, the local/regional and the long distance train services are shown in light grey, black and dark grey respectively. The DSB S-tog timetable is a cyclic timetable with the following characteristics on a normal weekday in the day time: The red H line runs every 20 minutes. So does the light green Bx line, but in the morning rush hour only. The yellow, circular F line runs every 5 minutes. All other lines run every 10 minutes. For weekends there is an all-together different line concept in use, with train services operating in frequencies of 10 and 20 minutes. For all day types, the frequency of operation is reduced in the evening. Nights after Friday and Saturday have yet another line concept and train services are then only operated with a frequency of 30 minutes.

3.3 Rolling Stock

DSB S-tog has two types of rolling stock train units with the technical designation *Litra SA* and *Litra SE*. For brevity and clearness these are designated 1 and $\frac{1}{2}$ respectively, indicating that the Litra SE are approx. half as long as the Litra SA. Other characteristics of the train unit types are given in Table 3.4 and the visual appearance is shown in Figures 3.3 and 3.4.

An important distinction between the two types of rolling stock is the amount and distribution of flexible space for bicycles, baby carriages and wheelchairs. Type 1 train units have a symmetric distribution of flexible space at both ends of the train unit and in the middle, in carriages number 1, 4, 5 and 8. Type $\frac{1}{2}$ train units have an asymmetric distribution, with flexible space only in the most northern carriage, which is the leftmost carriage on Figure 3.3.

Due to three different constraints (described below), train units in revenue train service may only be coupled together to form five different *train composition types* as shown in Table 3.5.

1. **Platform length and train control system block length:** For revenue train services, it must be possible to stop a train composition so that all doors are at the platform. The train composition must also fit in the existing train control system blocks at stations, so as to enable other train compositions to use adjacent points (switches) to neighbouring tracks. Stations fall in two categories, those with a platform and block length fitting at most a train composition type $\frac{2}{2}$ (on the circular line, Line F), and those with a length fitting at most a train composition type 2 (on the rest of the network);

2. **Number of train units in the train composition:** Generally, train compositions may not consist of more than two train units at a time. In the event of a disruption, more train units may be coupled together, however, for various reasons this runs the risk of further delays;

3. **Amount of flexible space and its distribution:** Type $\frac{1}{2}$ train units only have flexible space in the most northern of its carriages. In order to achieve that there is always flexible space in the first and last carriage of all train compositions, a business rule states that a type $\frac{1}{2}$ train unit may only be coupled to the north of a type 1 train unit. If it were to be coupled to the south, the train composition would have no flexible space in the most southern carriage. The business rule is in place to prevent delays arising when passengers are not able to find a carriage with flexible space to enter.

In Tables 3.4 and 3.5 *nominal number of seats* denotes the actual number of physical seats in the train unit, while *perceived number of seats* denotes the number of passengers that may be transported in the train unit when perceived by the passengers as being full with regard to seats. If the perceived number of seats is exceeded by the number of passengers, DSB S-tog experiences a sharp rise in the amount of customer complaints for seat shortage. Passengers are not using all available, physical seats because of the physical layout of the interior of the train units combined with their desire to sit in a particular carriage.

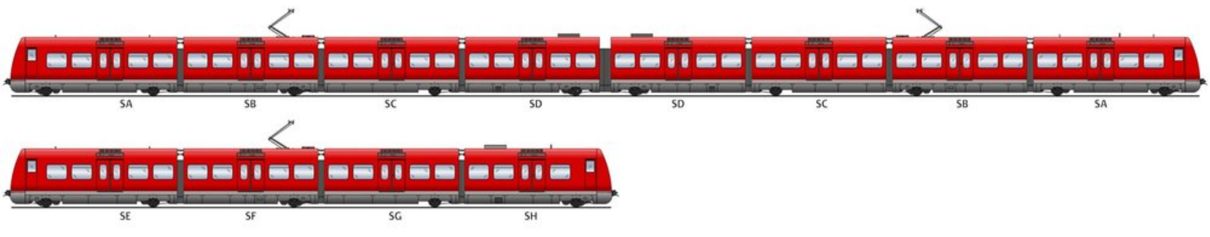

Figure 3.3: The DSB S-tog train unit types 1 (top) and $\frac{1}{2}$ (bottom).

Figure 3.4: An impression of a DSB S-tog train unit type 1 shortly before arriving at Sorgenfri Station on the B line [102].

Table 3.4: Overview of characteristics for train unit types of DSB S-tog. All train units are *Electrical Multiple Units (EMUs)*. *Nominal number of seats* is the actual number of physical seats in the train unit. *Perceived number of seats* is the number of passengers that may be transported in the train unit perceived by the passengers as being full with regard to seats.

Train unit type	Technical designation	Nominal # of seats	Perceived # of seats	Carriages	Length [m]	# of train units
$\frac{1}{2}$	Litra SE	150	125	4	42.58	31
1	Litra SA	336	300	8	83.78	104

Table 3.5: Train composition types for revenue train services at DSB S-tog. *Perceived number of seats* is the number of passengers that may be transported in the composition perceived by the passengers as being full with regard to seats. Additionally, compositions of type $\frac{3}{2}$, 3 and others exist but these are only allowed under special conditions.

Train composition type	Train unit types in composition	Perceived # of seats	Total Length [m]	Allowed on lines
$\frac{1}{2}$	$\frac{1}{2}$	125	42.58	F only
$\frac{2}{2}$	$\frac{1}{2}+\frac{1}{2}$	250	85.16	F only
1	1	300	83.78	All but F
$1\frac{1}{2}$	$1+\frac{1}{2}$	425	126.36	All but F
2	1+ 1	600	167.56	All but F

3.4 Passenger Demand

Another main requirement for rolling stock planning is of course the expected passenger volume, how much demand for seats is expected in the train services given by the timetable. The recent trend of the overall passenger demand for DSB S-tog is shown in Table 3.6.

DSB S-tog is in the very fortunate position of having very good data on passenger demand, since all train units measure the weight of passengers at the time of departure from every station. The passenger count can be determined with an accuracy of three to five persons in a type 1 train unit. On an average weekday approx. 28,000 measurements are recorded, one for each train service leaving a station.

DSB S-tog has developed an advanced statistical model to handle the passenger demand data. This passenger demand model uses exponential smoothing to handle the daily fluctuations of the passenger demand data while preserving information about possible trends. The passenger demand model delivers figures to be used as the dimensioning passenger demand by day type, using the concept of *comfort level*. Comfort level is the fraction of passengers that will have a seat in each individual train service over a number of days. If the comfort level is set to 0.95, the model calculates the dimensioning passenger demand so that 95% of all passengers in this train service (over a number of days) will have a seat.

The dimensioning passenger demand for each train service is the demand between those two consecutively visited stations where the passenger demand is the greatest. Stations pairs on the central segment are disregarded in the process, since travelling times here are short and frequencies high.

Figure 3.5 shows an example of seat demand data in space and time created with visualisation tools developed for this thesis. For other examples see Appendix A.

Figure 3.6 shows a graphical representation of current passenger demand data, converted to train composition type according to seat capacity, see Table 3.5. As may be seen from the

Table 3.6: Passenger demand development at DSB S-tog [2].

Year	2009	2010	2011	2012	2013	2014	2015
Transported passengers [Million]	92	93	103	106	109	112	114

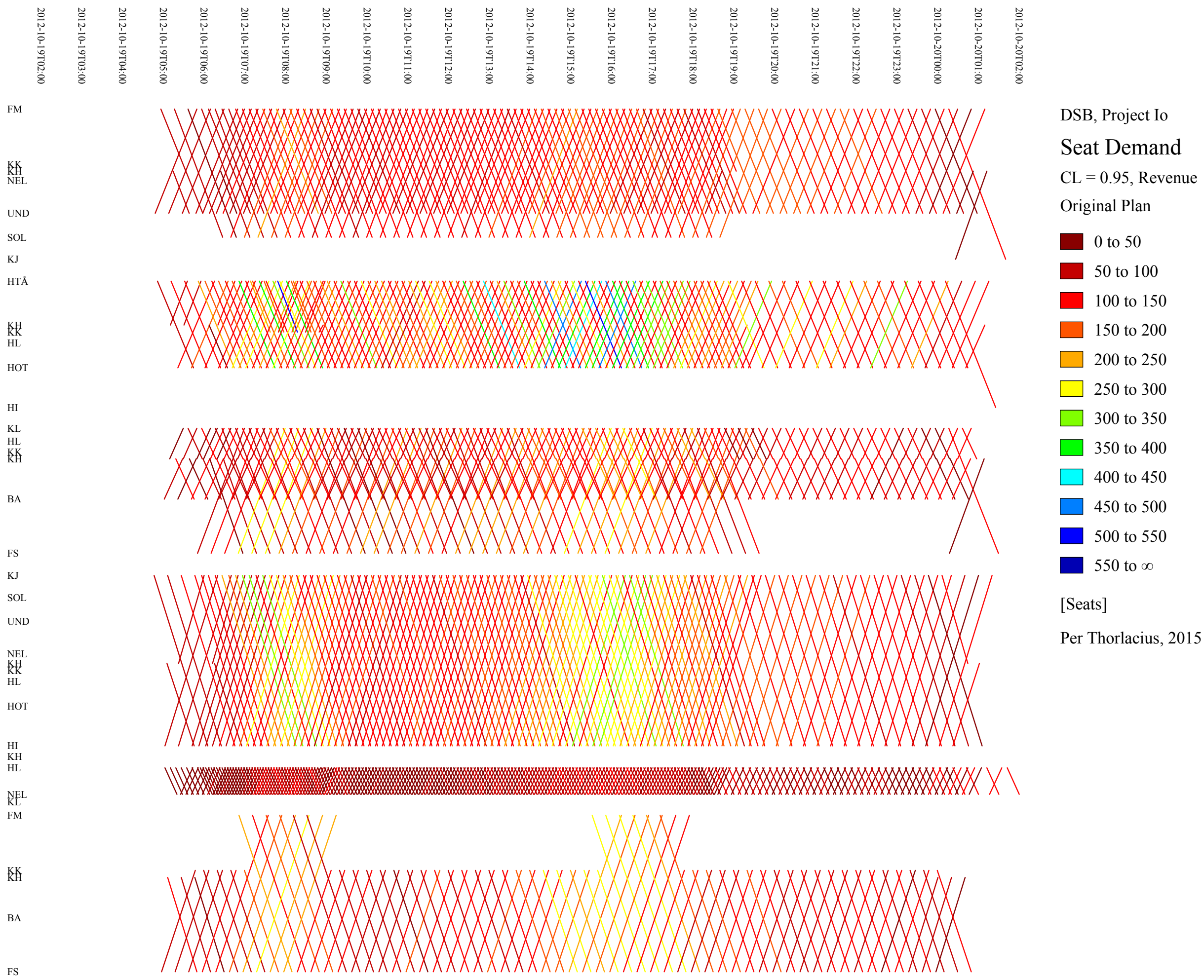

Figure 3.5: Example of a space-time diagram of the passenger seat demand with comfort level CL = 0.95. The time axis is on the top, the space axis with station abbreviations to the left. The 6 blocks of train services refer from top to bottom to the lines A, B (including Bx), C, E, F and H, respectively. The diagram refers to passenger data measurements from 2012 processed with the DSB S-tog statistical prognosis tool and rolled out on the date of 2012-10-19.

figure, the train composition type 2 with the highest capacity is only demanded in the rush hours on weekdays and Fridays. The same applies to the train composition type $1\frac{1}{2}$, the train composition type with the second highest capacity, the only difference being that a few train services at Sunday afternoon also demand this train composition type. Figure 3.7 shows how the train composition type demand is distributed in space and time.

When comparing the demand for the train composition $\frac{1}{2}$ in Figure 3.6, to the number of train units of type $\frac{1}{2}$ available to DSB S-tog (Table 3.4 on page 40), the demand for train composition types $\frac{1}{2}$ is often much higher than the number of train units of type $\frac{1}{2}$ available. In most of these cases, DSB S-tog is forced to assign train composition type 1 to the train services thus providing excess seat capacity and having to bear the extra cost.

When a rolling stock plan does not provide enough seats to meet passenger demand, the consequences for the train operator include increased risk of delays due to overcrowding, customer dissatisfaction and customer complaints. All of these consequences may eventually have negative economic implications. If, on the other hand, a rolling stock plan provides too many seats, this also has negative economic implications in that the surplus seats add an extra cost to the operator at virtually no extra benefit to passengers. Finally, there is also a political demand

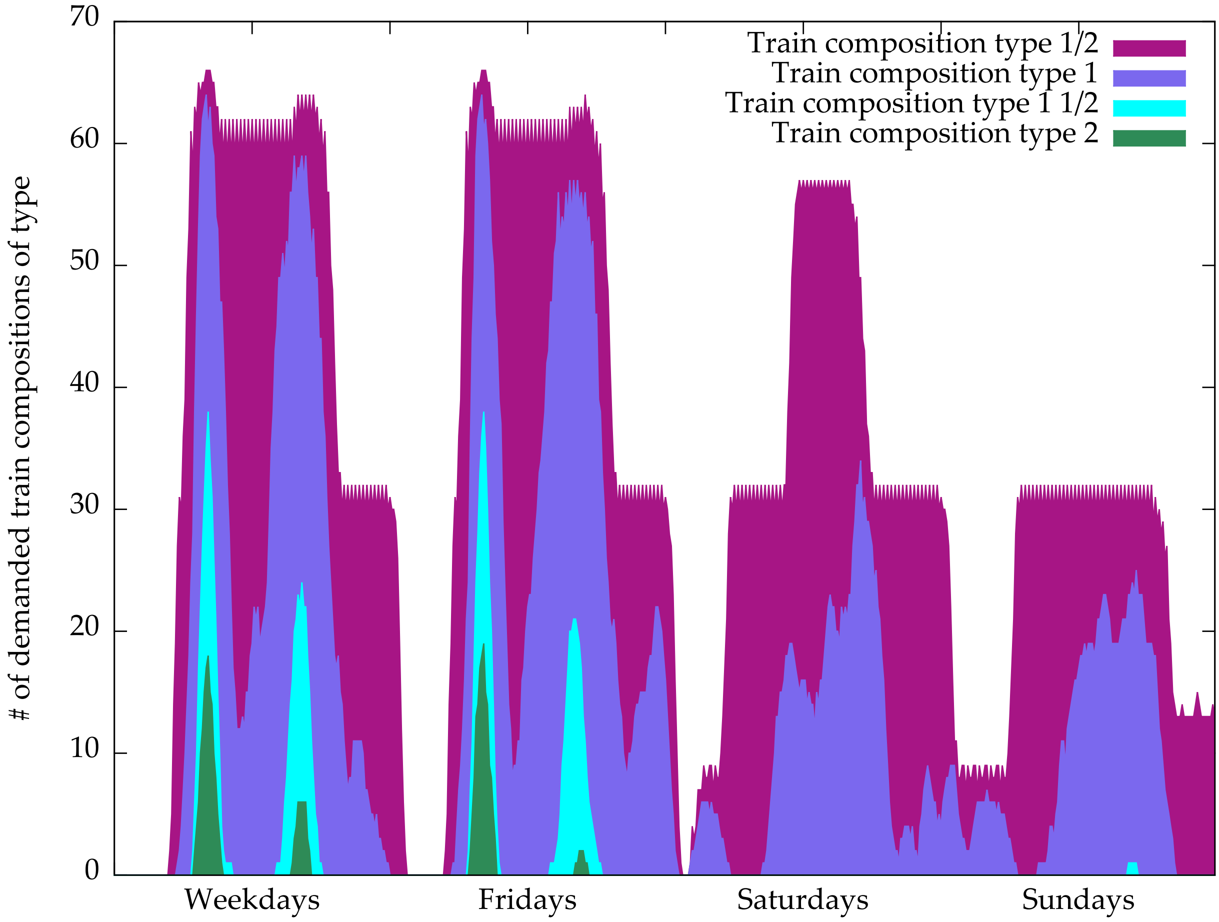

Figure 3.6: Passenger demand by day type and time expressed as demand for number of specific train composition types, as defined in Table 3.5 on page 41. The composition types are stacked in the figure, their common total indicating the total number of train services in service by day type and time. Train composition type $\frac{2}{2}$, which is suboptimal for cost reasons, is not shown. The data refer to the date of 2012-10-19.

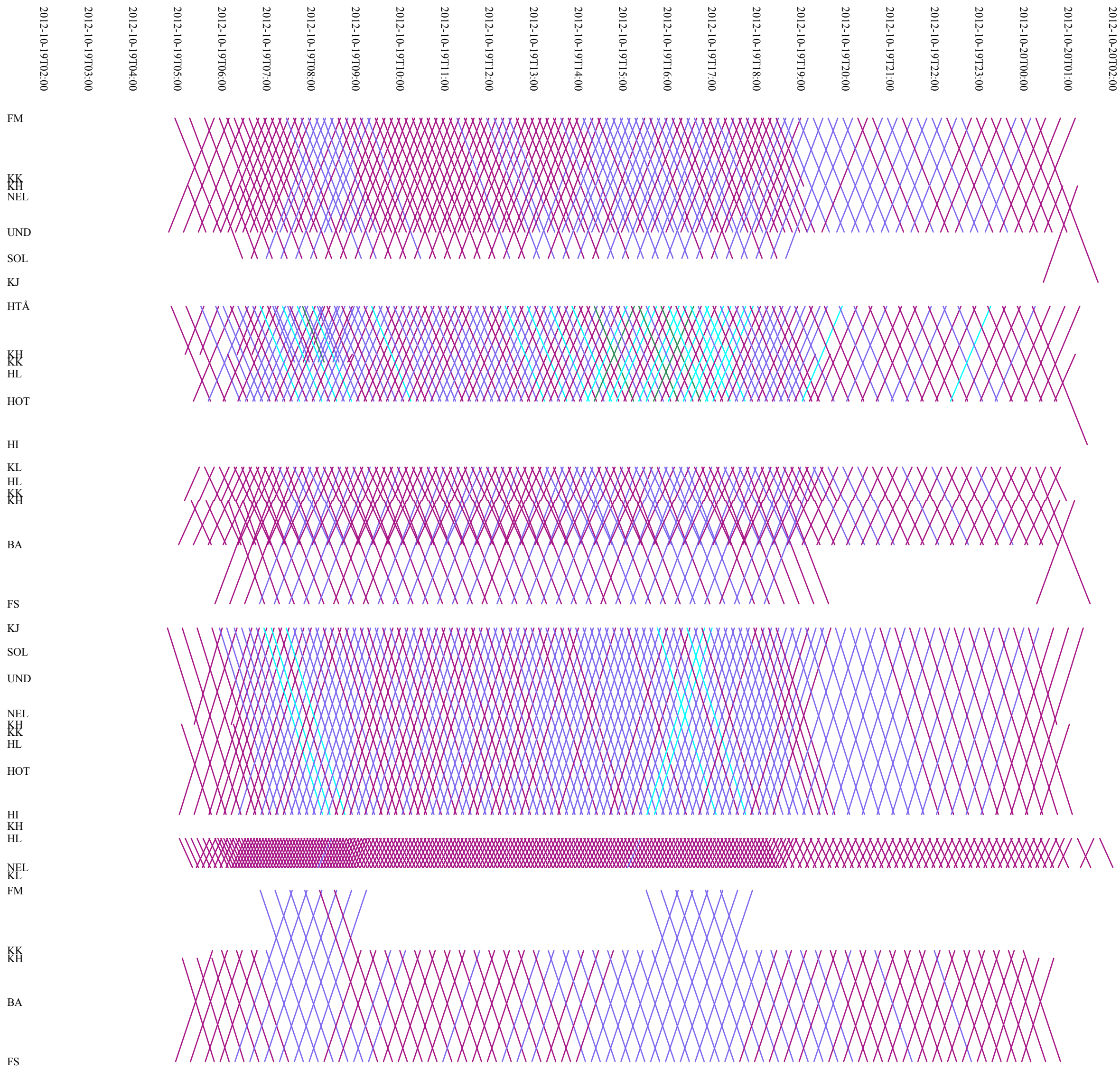

Figure 3.7: A space-time diagram showing the passenger demand expressed as demand for specific train composition types, as defined in Table 3.5 on page 41. The same legend is used as in Figure 3.6. The time axis is on the top, the space axis with station abbreviations axis on the left. The 6 blocks of train services refer from top to bottom to the lines A, B (including Bx), C, E, F and H, respectively. The diagram refers to passenger data measurements from 2012 processed with the DSB S-tog statistical prognosis tool and rolled out on the date of 2012-10-19.

from society that DSB S-tog maintains an efficient operation. Only by conducting an efficient operation may DSB S-tog gain future transport contracts. This underlines the importance of having a rolling stock plan that meets passenger demand as closely as possible.

3.5 Personnel on Duty

In order to couple and decouple train units and to drive train units into the depot for parking and back to the platform for the next train service, a designated personnel group of *depot drivers* operate at DSB S-tog. Each depot has a number of depot drivers on duty at different hours of the week, and any rolling stock plan must of course adhere to this number, and not demand more depot operations than it is possible to conduct with the personnel on duty.

How many depot drivers are hired and when they should be on duty is decided every time a *standard timetable* is planned (see definition in Section 3.2), based on the demand in the corresponding rolling stock plan. *Extraordinary timetables* must then adhere to the number of depot drivers hired and their duties for the standard timetable.

Recently, the *train drivers*, the personnel group driving the revenue and non-revenue train services, do perform some of the operations previously conducted by the depot drivers the morning and evening on weekends.

3.6 Scheduled Maintenance

Each train unit belonging to DSB S-tog must undergo scheduled maintenance at given intervals. A standard maintenance check of the individual train unit must be scheduled every 50,000 km of service distance. In addition to this, minor overhauls of the individual train unit must be scheduled every 100,000 km (for train unit type $\frac{1}{2}$) and 150,000 km (for train unit type 1). Finally, major overhauls (see Figure 3.8) of the train units must be scheduled every 600,000 km.

Figure 3.9 shows a snapshot of the service distance distribution of the DSB S-tog train units. On the average, a DSB S-tog a type 1 train unit travels a service distance in the order of 500 km a day, a type $\frac{1}{2}$ train unit only in the order of 250 km. However, the service distance travelled may vary a lot from day to day. Some days a train unit may be in maintenance preventing the train unit from running at all. Other days a train unit may be the part of a train composition that is running the whole day, travelling a service distance in the order of 1,000 km. Having a different scheme of utilisation, type 1 train units may enter the 50,000 km scheduled maintenance in the order of every 100 days, type $\frac{1}{2}$ train units in the order of every 200 days.

For DSB S-tog, the maintenance workshop itself is responsible for requesting the individual train units in for scheduled maintenance. When a train unit is requested in for scheduled maintenance, it is the responsibility of the train unit dispatchers (see Section 2.2.2) to get the train unit into the workshop. This is also the case if the train unit has encountered a break down and needs to undergo unscheduled maintenance (see Section 3.7). Getting the train unit to the workshop can be done directly by performing revenue or non-revenue positioning. It can also be done indirectly by making the train unit run on a line passing the workshop and let the workshop itself pick the individual train unit out for maintenance when it passes by. Table 3.7 shows how many train units of the different types must be provided to DSB S-tog by the workshop.

Whenever the workshop gets a train unit in for maintenance (scheduled or unscheduled) it is contractually responsible for delivering another working train unit back to the operator, DSB S-tog. As such, the workshop carries the risk of train units breaking down.

Methods to even out the use of the rolling stock are currently being investigated so that

Figure 3.8: The DSB S-tog train unit SA8120 undergoing a major overhaul in the workshop at Høje Tåstrup, September 2012.

Table 3.7: Contractual obligations of the workshop (DSB Vedligehold) to the operator (DSB S-tog) with regard to number of train units to be provided. Figures from 2015.

	For operation			On stand by		
Day type	# Type 1	# Type $\frac{1}{2}$	Total	# Type 1	# Type $\frac{1}{2}$	Total
Weekdays	94	28	122	3	0	3
Weekend	54	6	60	3	1	4

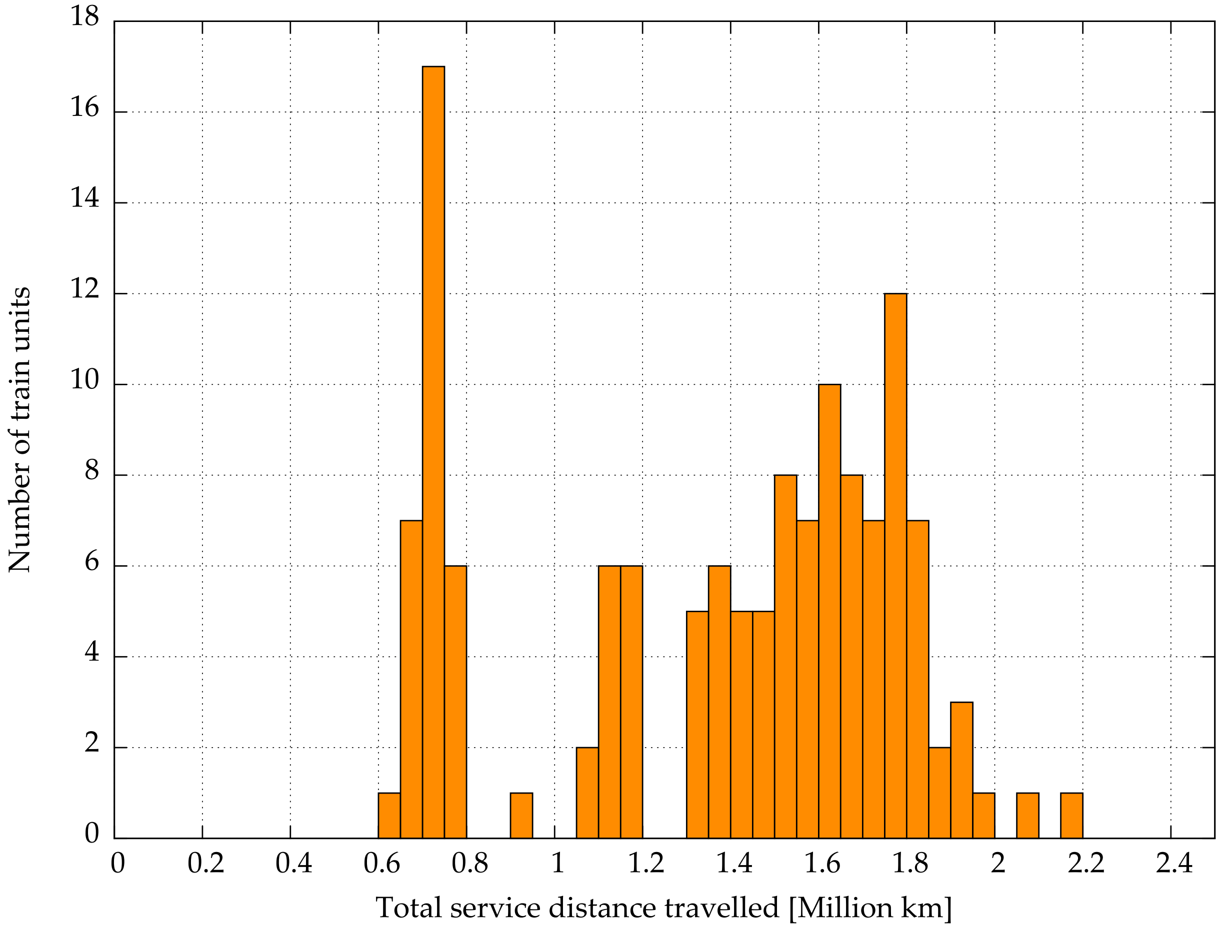

Figure 3.9: The service distance distribution of the DSB S-tog train units as of January 2013. As may be seen from the figure, the train units form four groups. Starting with the group of train units having the longest service distance (from 2.2 down to 1.3 million km), this group contains the first series of type 1 train units delivered from the year 1996 and onward. The second group (from 1.2 down to 1 million km) represents the second series of type 1 train units. The third group (from 1 down to 0.8 million km) consists of only one type 1 train unit which has travelled less than the others due to repairs after a collision accident with a truck in 2006. The fourth group (from 0.8 down to 0.6 million km) are the type $\frac{1}{2}$ train units, which are utilised differently from the type 1 train units and travel shorter distances each day.

the 50,000 km maintenance may be carried out evenly distributed in time in order not to cause bottlenecks in the workshop.

The maintenance workshop is located in Høje Tåstrup with a subsidiary workshop for minor repairs located in Hundige.

3.7 Unscheduled Maintenance

Apart from the scheduled maintenance mentioned above, a train unit may break down at any time and require immediate (and thus unscheduled) maintenance. Breakdowns are grouped in six categories depending on the severity of the breakdown and the desired time frame in which it must be remedied. Some types of breakdowns may prevent the train unit of performing revenue train services.

In addition to the workshops in Høje Tåstrup and Hundige, DSB Vedligehold also has a mobile repair crew that may head out to remedy broken down train units at any given location at any time.

3.8 Friction Sand

In order to enhance friction on tracks made slippery by fallen leaves etc., the train units have equipment installed to disperse sand on the tracks in front of the some of the wheels. The sand tanks need to be filled to a certain level, otherwise the train unit may not perform revenue train services and a speed restriction is enforced. The filling level of the sand tanks must be checked every 10,000 km of service distance.

Friction sand can be refilled at the workshops at Høje Tåstrup and Hundige and new facilities are under selection.

3.9 Exterior Cleaning

The exterior cleaning of the train units is conducted automatically in facilities which the train units pass through at low speed, see Figure 3.10. Facilities are in Høje Tåstrup and Hundige. It is the goal of DSB S-tog to clean the exterior of all train units every 15 days. Since the exterior graffiti removal facility in Høje Tåstrup is on the same track as the facilities for exterior cleaning, this goal can not always be achieved, and the removal of exterior graffiti has precedence.

3.10 Exterior Graffiti Removal

Removing graffiti from the outside of a train unit may take anything from half an hour to an entire day depending on the area of the train unit covered and in particular how much time has elapsed since the graffiti was painted onto the train unit. Graffiti removal is performed by DSB S-tog staff in facilities in Hundige and Høje Tåstrup.

3.11 Interior Cleaning

The interior cleaning of the train units is performed on a daily basis at the depots. Train units for day train services are cleaned at night and train units for night train services in the morning.

Figure 3.10: The DSB S-tog train unit SA9156 undergoing exterior cleaning in the cleaning facilities at the workshop at Høje Tåstrup, September 2012.

In order to facilitate day to day interior cleaning, a business rule states that train units to enter night time service must be put into service by performing a total train composition exchange (see definition in Section 3.1) with newly cleaned train units. As such, a train composition for a night train service will always consist of newly cleaned train units. A similar business rule states that, in the morning, a train composition from a night train service must be driven into the depot without being split up.

The two business rules are in place to make sure that the cleaning standard is as high as possible at all times and to even out the workload of the personnel cleaning. It is well known that train units with a low cleaning standard attract much more dirt and garbage than train units with a high level of cleanliness.

Interior graffiti removal is performed as part of the interior cleaning process, taking place at any depot.

3.12 Winter Preparedness

In order to prevent ice from accumulating underneath the train units in winter time, the undercarriage of all train units must be treated with anti ice fluid every 6 days when the weather is cold. Facilities for doing so are at Høje Tåstrup and København H. Six train units may be treated per hour in each of the two facilities.

3.13 Exposure of Commercials

Some train units of DSB S-tog have commercials mounted internally and/or externally. In order to expose commercials in certain geographic regions, it may be required that a certain train unit be running on a particular line on a particular set of days.

3.14 Surveillance Video Requests

All train units of DSB S-tog have 24 hour video surveillance. The video recordings of all cameras in a train unit are stored on hard disks in the train unit and may be retrieved for investigative purposes upon order by the Police. When the Police requests a video recording from a particular train unit, this train unit must be driven into the main workshop in Høje Tåstrup for video retrieval within a week from the time of the event the Police wants to investigate. Otherwise the recording is overwritten. DSB S-tog is working on a solution to make it possible to remotely download the video recordings in the future, thus eliminating this requirement from consideration.

3.15 Surface Foil Application

The train units of DSB S-tog are all covered by a protective surface foil applied on demand. This foil has better resistance to graffiti and may, in the event of damage, be replaced much faster and cheaper than the alternative - a conventional repaint of the train unit. In addition to this foil, some train units have a commercial foil applied on top of the protective foil. The commercial foil is replaced at irregular intervals. Facilities for foiling are in Hundige.

3.16 Passenger Counting Equipment

Some of the train units, 12 of the train unit type 1 and 4 of the train unit type $\frac{1}{2}$ have infrared passenger counting equipment installed. The infrared counting system is used to count the passengers getting on and off. In order to achieve good data sample coverage, the train units with infrared counting equipment may need to run as specific train services on specific days.

3.17 Train Control System Equipment

A new communication based train control system (CBTC) without external visual signals is in the process of being rolled out. Since 2014, this system has been in operation on parts of the network, requiring special train control system equipment installed on train units to run on these parts of the network.

Part II

The Integrated Rolling Stock Planning Problem

Chapter 4

Rolling Stock Planning Models and Solution Methods

In this chapter, Section 4.1 presents an overview of models and solution methods from literature that either specifically relate to rolling stock planning or have any kind of methodological similarity. To relate, Section 4.2 then presents an overview of the models and solution methods proposed in this thesis. Later chapters of this thesis feature further references to literature where relevant.

4.1 Overview of Models and Solution Methods from Literature

Historically, the adaptation of optimisation and other operations research (OR) techniques in the railway industry has been slower compared to the in many ways similar airline industry [61, 14]. Major airlines started forming OR groups from the late 1960s and onward and the widespread and early use of information technology in the airline sector has also strongly facilitated the use of OR [14]. The difference in adaptation speed between the two industries may be explained also by the fact that the airline industry experienced growth parallel to advances in OR (from the 1950s and onward), whereas the railway industry experienced a general decline in the same period due to increased competition from road and air transport. Furthermore, as OR techniques emerged, the railway industry was a much more established industry with presumed higher organisational inertia.

Driven by, among other things, congestion problems and rising environmental concerns regarding road and air transport, the railway industry has experienced a renaissance from the 1990s and onward. This renaissance has diminished the implementation gap between the railway and airline industries with OR techniques now being applied to a wide range of railway-specific problems, see surveys [39, 7, 64, 34, 71]. At present, the challenges in the adaptation of OR techniques in the railway industry seem not only to lie in finding solutions to each specific problem, but even more so in integrating the individual solution methods to the highly interconnected specific problems into holistic, integrated models. By integrating the specific models with each other, suboptimal solutions can be avoided. These tendencies for the integration of models are currently also seen in the airline industry [98].

The following is an overview of selected and reviewed literature related to railway rolling stock planning. The literature review is structured with a widening scope, firstly focusing specifically on railway rolling stock planning applications (Section 4.1.1), then on related railway planning applications in the field of train service line planning and timetabling (Sections 4.1.2

and 4.1.3), then widening the scope to look at ways to tackle the integration of a large number for different railway-specific requirements (Section 4.1.4). Next, the scope is widened once more to look at personnel planning applications in the railway and airline industries, and in the health sector (Section 4.1.5). Finally, the scope is widened to include planning problems from the maritime industry (Section 4.1.6). Throughout the review, descriptions are given as to how different planning problems are currently solved within DSB S-tog.

In the selected and reviewed literature, different overall modelling and solution methodologies from operations research are applied, including:

- Linear programming (LP);
- Mixed integer linear programming (MIP);
- Constraint programming (CP);
- Dynamic programming;
- Heuristics;
- Metaheuristics;
- Matheuristics.

Problems to be solved are formulated in different ways, including as:

- Assignment problems;
- Arc based multi-commodity flow problems;
- Path based multi-commodity flow problems;
- Independent set problems;
- Generalised set partitioning problems.

Most models from literature feature graphs that are either:

- Event-activity graphs (or similar);
- Conjugated edge-to-vertex dual graphs of the above;
- Conflict graphs.

Moreover, specific solution methodologies and algorithms from literature include:

- Column generation;
- Branch-and-bound;
- Branch-and-price (i. e., the combination of the two methods above);
- Shortest path algorithms;
- Resource constrained shortest path algorithms.

4.1.1 Railway Rolling Stock Planning Applications

For a tabular overview of most of the reviewed railway rolling stock planning literature, see Table 5.1 on page 65. Table 5.1 shows an overview of the following characteristics: Overall topic, planning processes, model type, model graph properties, railway-specific requirements integrated, model objective and solution methodologies applied.

As justified in Chapters 1 to 3, rolling stock planning is an immensely complicated undertaking. For this reason, rolling stock planning and rolling stock optimisation models have historically been domain specific (as opposed to integrated), in which each model has only adhered to a small number of the many railway-specific requirements at a time.

In the reviewed literature, almost all rolling stock planning models adhere to the common requirements, including timetable requirements, (overall) infrastructure requirements, rolling stock requirements and passenger (or freight) demand. Other requirements handled are train unit order in train compositions [55, 10, 53, 72, 86, 46, 17], maintenance etc. by time [38, 80, 109, 57], maintenance etc. by distance [38, 80, 46, 20, 21, 57], depot capacity [24, 55, 53, 72, 46, 17, 28, 21, 60] and depot topology [55, 53, 72, 46, 17]. It is interesting to note that none of the reviewed models integrate requirements regarding personnel on duty. As described in Section 3.5, this is an important requirement for DSB S-tog.

The models in the reviewed literature all integrate the railway-specific requirements to a varying degree, with a slight tendency that recent models integrate requirements to a higher degree than earlier ones do. Subsidiary requirements like cyclicity, robustness and disruption recovery with minimal changes have not been analysed in this review since this is out of scope for this thesis.

Most of the selected and reviewed rolling stock planning literature feature *arc based multi-commodity flow models* or similar [37, 38, 24, 89, 9, 10, 51, 80, 86, 109, 28]. In such types of models the flow of train units (in some cases locomotives) is modelled in a flow graph, with flow conservation constraints on each vertex of the graph to ensure the in-flow to the vertex equals the out-flow from it. Arc based flow models are typically low in complexity, a probable reason for their widespread use. However, in an arc based flow model, it is difficult to express sub-path constraints, such as recurring maintenance at regular distance intervals.

On the other hand, in *path based multi-commodity flow models* [67, 60], the potential sequence of movements of the individual train unit is modelled (e. g., by enumeration), facilitating also to take recurring distance related constraints such as maintenance into account.

Other types of models include assignment models or similar [5, 53, 72, 46, 27], set partitioning models [55, 17], hyper arc multi-commodity flow models, [20, 21] and even a model using Hamiltonian cycles [57].

In almost all of the reviewed rolling stock planning literature a graph is featured. The most prevalent graph type is a *space-time graph* (also known as a time-expanded graph), where each vertex in the graph is an event in space and time. This can, e. g., be the arrival of a train at a given time at a given station (in space). Correspondingly, the arcs in a space-time graph may, e. g., represent train services. This type of graph is also known as an *event-activity graph*, with vertices as events and arcs as activities, and it is also well known in the airline industry [96, 15, 14, 88, 98].

In some literature, the *edge-to-vertex dual* or *line graph* [62] graph type is used [10, 51, 86, 20, 21, 27, 57]. This type is conjugated from the space-time graph mentioned above. In the line graph type, vertices represent train services whereas arcs may, e. g., represent the option for a train unit to perform train services in sequence. Similar "line graph" style graph types are used to model train composition changes at stations [55, 10, 51, 53, 72, 86, 20, 21, 27], the approach direction in depot planning [55, 51, 53, 72] and rolling stock maintenance constraints [38, 20, 21, 57]. Models may use multiple modelling schemes from the ones mentioned.

In the majority of reviewed models from literature, it is the objective to minimise operational costs and/or penalties. Some models are used to minimise seat shortages [5, 51], number of train units required [89, 51, 27, 57] or number of shuntings [51]. Depot planning is mainly a matter of fixed costs, and for this reason, some depot planning models only strive to find a feasible solution [53, 46, 17], not to optimise a specific objective. Some of these models are constraint programming (CP) models.

All the reviewed solution methods from literature involve commercial solvers and/ or heuristics. Moreover, decomposition techniques in the form of column generation, branch-and-price and Benders decomposition are used in some cases.

With regard to the overall topic and focus of the reviewed literature: [37, 38] assign train units to train services using Benders decomposition. [24] assign different types of train units to train services, also taking depot capacity into account. [5] circulate train units for passenger train services using an assignment type model to minimise seat shortage, while [89] circulates train units for freight train services. In [9], train services have train units assigned to them by different types. [55] distribute train compositions of train units to depot tracks using a set partitioning model. In [10], train compositions have train units assigned to them. [51] assign train units to train services using multiple objectives. [80] perform maintenance planning of train units in a train unit dispatching context. [67] assigns collections of train units to freight train services using a variety of methods, among these branch-and-price. [71] assigns train units from train services to depot tracks, thus taking both depot capacity as well as track topology into account. [86] circulate rolling stock using a branch-and-price approach and a transition graph. In [109], freight train services have train unit types assigned to them. [20, 21] circulate the entire German ICE high speed fleet on a weekly basis in a hypergraph model. Train units are assigned to train services using an arc based multicommodity flow type model in [28] and a assignment type model in [27]. [57] plans maintenance for a timetable train services using a Hamilton cycle. Using DSB S-tog data, [53, 46] perform depot planning, [17] circulates rolling stock using set partition and [60] recover from a disrupted rolling stock plan in a train unit dispatching context using branch-and-price (but disregard the individual depot tracks).

The terminology used in rolling stock planning literature shows great variation. Some articles use a terminology that has weak connotations to the actual railway operation. This does not facilitate the understanding nor the comparison of methods across literature.

4.1.2 Train Service Line Planning Applications

As shown on Figure 1.1 on page 16, one of the overall strategic processes of railway operations planning is that of train service line planning. Based on the available railway assets, the expected passenger demand and desired service level, the objective of train service line planning is to determine which lines should operate on the railway network and with what frequency. Available railway assets include available infrastructure, track capacities, minimum head-ways and available rolling stock. The service level may e. g., be stated as the minimum frequency of operation and the passenger demand as an origin-destination passenger travel demand matrix. [99] presents a comprehensive and structured review on line planning models (also for other modes of transport), dividing models into the following categories:

- **Cost oriented models** that minimise costs for personnel, energy, etc.;
- **Passenger oriented models** that maximise number of direct connections for passengers or minimise passenger overall travelling time;
- **Game theoretic models** in which lines are modelled corresponding to players deciding their frequencies with regard to a benefit function (sometimes including robustness to delays);
- **Location based models** in which each new line is constructed so that the passenger access times to the new line is minimised. Location based models are thus oriented towards determining where high speed lines should be provided in a network.

Budget constraints play an important role in some models. Models are formulated as multicommodity flow problems, and integer, linear and/or non-linear programs and solved using a wide variety of methods including column generation, branch-and-bound, branch-and-cut and last but not least heuristics.

In conjunction with the preparation of the 2016 timetable, DSB S-tog used the line planning model described in [91].

4.1.3 Timetabling Applications

As shown on Figure 1.1 on page 16, timetabling is one of the processes in the tactical time horizon of railway operations planning. Based on the previously constructed train service lines, the objective of timetabling is to determine when train services belonging to the defined train service lines should run, while always adhering to track capacities, minimum head-ways and other operational constraints. [30] provide an overview of assumptions and properties of well-established timetabling models and present applications in practice. Some further representative references are given in the following.

[31] present a theoretic description of the timetabling problem and solve it for a single track instance using a two-phase heuristic working with dual information associated with Lagrangian multipliers. In [52], different methodologies to improve a given timetable with regard to robustness are given. Both cyclic and non-cyclic cases are treated. Methods include the use of linear and stochastic programming, each with different solution characteristics.

In [71] the timetabling problem is formulated as a *periodic event scheduling problem* (PESP), and a timetable is constructed so as to facilitate passenger interchange between train service lines at stations. A two-phase solution approach is used: Firstly, constraint programming is used to solve the problem to feasibility. Secondly, a heuristic is applied to improve the timetable created. [71] also describes the process of creating a new timetable in relation to all the other railway planning processes as shown in Figure 1.1.

DSB S-tog is a passenger railway operator providing high-frequency train services in a network in which few passengers need to change from one line to another. Moreover, DSB S-tog is the public transport provider with “highest priority”, in that other public transportation operators (local trains, busses, etc.) adapt their timetables to that of DSB S-tog for passenger interchange, not the other way around. For this reason, PESP models are currently less relevant for DSB S-tog. In [81], using DSB S-tog as case study, a mixed integer linear programming model is used to construct a timetable minimising the number of train service sequences, this in itself yielding lower cost. The braiding policy is given as input.

Currently, the DSB S-tog timetable is constructed using a semi-automated planning tool in which lines and braiding policy are defined by the user. In this tool, the user must then decide the order in which the train services of the defined train service lines should use the central segment. The tool then detects capacity conflicts, any of which the user must remedy in order for the tool to finally construct the timetable. As such, the tool does not apply any optimisation methods.

4.1.4 Integrated Railway Operations Planning Applications

With regard to *integrated* railway operations planning, the challenge is of course that models become very complex and difficult to solve when having to integrate a large number of requirements. In literature this challenge is addressed e. g., by defining layers in the models or by providing feedback between submodels.

[19] propose a *coarse-to-fine* approach for integrated rolling stock planning, in which a multi-layer model is built with different requirement types categorised as being either in the coarse layer or in the fine layer. In the model, the major decisions are taken in the coarse layer, while minor details are handled in the fine layer. The fine layer is restricted to a subset of

variables and is iteratively extended using information from the coarse layer. The coarse layer is, as such, used to identify which parts of the fine layer are relevant in the optimisation process.

[29] use a Benders decomposition based heuristic to solve a two-layer model for integrated rolling stock planning, solving, in the primary layer, the train unit to train service assignment problem, and, in the secondary layer, the routing of train services through stations. Results are compared to real-world rolling stock plans.

[112] propose a way to integrate railway timetabling with passenger assignment using a two-layer model. The timetabling is formulated as a periodic event scheduling problem (PESP). The passenger assignment problem uses a given passenger flow in space and time to predict transportation services used, i. e., the passenger flow in the individual train service.

[26] describe a method to integrate passenger railway line planning with timetabling using DSB S-tog data. Their integration approach is *feedback oriented* (see Section 1.1) in that two exact models, one for line planning and one for timetabling, communicate with each other in a heuristic feedback loop to generate plans that suit both models well. Main focus is robustness.

For a review of other integrated approaches involving public transport service line planning, see [99, Chapter 4].

4.1.5 Personnel Planning Applications in Railway, Airline and Health

There is some resemblance between rolling stock planning and personnel planning in the railway and airline industries and in the health sector. The mentioned problems involve tasks (duties, flights, train services) that need to be fulfilled by entities that may fulfil them (nurses/physicians/conductors, pilots/flight attendants, train units).

An important part of personnel planning models is played by the often very complex and large set of constraints reflecting workers union agreements, including rules for breaks and days off. These constrains bare some resemblance to the service distance or service time related constraints in rolling stock planning.

For reasons of complexity, personnel planning is traditionally conducted in a two-phase approach in which tasks are first grouped together or *paired* into *rosters*, and afterwards individual personnel is then assigned to the roster. Recently, research has been conducted into integrating the pairing and rostering steps and assigning tasks directly to the individual personnel [98].

With regard to personnel planning in the railway industry, [32] describe different ways to model personnel planning using the before mentioned two-phase approach. Formulations include a set covering model, to which an advanced, heuristically based method of solution for large-scale instances is provided in [33]. DSB S-tog personnel is planned using the TURNI software package based on methods from the above mentioned literature.

[70] also uses a set covering formulation to plan train drivers and conductors. The problem is solved using a combination of column generation, Lagrangian relaxation and heuristics. A key issue is robustness to disruptions, and variance in duties for the sake of the personnel. Following a train driver and conductor strike in 2001 caused by personnel dissatisfaction with duty structures, [4] provide a model to "share the sweet and the sour" between employees while also addressing economic efficiency.

As a specific example of a personnel related railway application of the branch-and-price methodology, see [90], in which train driver duties are recovered during disruptions. In this application the restricted master problem linear program is formulated so as to be relatively simple and to expose very good properties with regard to natural integrality. Moreover, the objective is to find a feasible solution fast, so branch-and-bound trees need not be fully explored. This is in line with the fact that processing time is of high importance, since a disruption must be handled fast in order not to propagate further.

[104] deal with the planning of the inspection personnel for ticket spot checking at DSB S-tog in relation to avoiding fare evasion. Based on historical data, a mixed integer linear programming model is used to determine an optimal duty schedule for the ticket inspectors in order to maximise the revenue from claimed penalty fares.

Personnel planning in the airline industry is similar to personnel planning in the railway industry. [15] perform airline personnel assignment using a two-phase approach in which column generation is used to find the optimal solution to an LP-relaxed version of the set partitioning problem. In the second phase this solution is made integer by branching, but no further columns are generated in the branching process.

[11] perform airline crew pairing, i. e., personnel planning for anonymous crew members. Their overall approach is an enhancement to [15], however there are more steps in the algorithm and a strong focus on performing all steps fast. This entails using approximate methods rather than exact ones. Moreover, column generation is stopped prior to optimality, i. e., when the improvement of the objective function in each iteration is negligible. Column generation may be focused on selecting flights with expensive pairings. When certain criteria are met, the column generation process is stopped and the problem is solved as a mixed integer linear program (MIP) using branching and employing the special Sprint LP solution algorithm with a fallback to the simplex algorithm when the former fails.

Both [15] and [11] employ *constraint branching*, however they refer to it using the more specific term *branch on follow-ons*.

[77] deal with airline ground crew roster creation by applying a cutting stock based integer programming model solved with a column generation based heuristic and special variable fixing. Robustness is a main goal.

Other non-personnel related problems from the airline industry with some resemblance to railway rolling stock planning exist, including tail assignment (i. e., aircraft to flight assignment) using primarily constraint programming (CP) [59]. However, since these problems occur in a specific and non-integrated way, they will not be treated further here.

With regard to the health sector, [12] deal with the scheduling of nurses in a hospital using a heuristic to create new columns that are fed to a mixed integer linear program (MIP) set covering model which is then solved. If the solution does not meet predefined criteria more columns are created and the MIP is solved again, repeating until the stopping criteria are met. In [12], the term *column generation* is used, however, this does not refer to the method of finding an optimal solution to a linear program by using dual information in a subproblem, it merely refers to the creation of new columns for a mixed integer linear program.

In a similar setup also from the health sector, [25] deal with the scheduling of physicians in hospitals. The scheduling of physicians is performed taking into account requirements including their level of experience, staffing policies and union agreements. The model generates flexible shifts as part of the solution process. The problem is formulated as a mixed integer linear program and a column generation heuristic is used to feed columns to this program.

Still in the health sector, [16] propose a model to solve the scheduling of trainees in hospitals. Two different decomposition approaches are examined, firstly the decomposition as a multicommodity flow problem decomposed on the activities to be performed, secondly as a set partitioning problem decomposed on staff members. Both approaches were solved using branch-and-price methodology. In the experiments the decomposition on activities outperform the one on staff members, however, the latter may be more suited for the implementation of further, future requirements.

4.1.6 Maritime Transportation Applications

With regard to maritime transportation applications, [35] provide an overview of the modelling aspects from both a strategic, tactical and operational point of view, pointing out that maritime planning has a very large variation in operating environments. Hence, seen from a modelling point of view, there is also a very large variation in problem structures. It is claimed that there is much more uncertainty in maritime transportation than in other modes of transportation. Moreover, most maritime transportation today is freight rather than passengers. For these reasons, from a general perspective, there may be only limited resemblance between maritime modelling applications and railway rolling stock planning.

Nevertheless, some maritime transportation problems still bare some resemblance to railway rolling stock planning. In liner shipping network design, cyclic routes between ports and schedules for a periodic maritime transportation shipping service are designed to maximise the gained revenue for the transportation of freight minus the costs of providing the transportation service. This resembles the choice of which train services to serve with which train units and which benefit can be gained from doing so. Moreover, the cost structures with fixed and variable costs are also similar.

[23] deal with the liner shipping network design problem by constructing a multi-commodity flow model and using heuristics that are column generation or tabu search based.

[68] provide an improvement heuristic based on an integer program which is solved iteratively to perform moves in a large neighbourhood search heuristic. The heuristic is incorporated in a simulated annealing metaheuristic framework.

Seen from a more general perspective, [36] deal with ship routing and scheduling and provide four general, archetype models for solutions to a variety of problems, including liner shipping network design and fleet deployment, the latter of which deals with the assignment of ships to liner shipping routes. Maritime fleet deployment bears some resemblance to assigning train units to train services, at least in a more general, assignment sense. [36] also provide a multi-commodity flow model for tramp shipping cargo routing and scheduling with similarity to rolling stock planning models.

4.2 Overview of Rolling Stock Planning Models Proposed In This Thesis

A total of five different rolling stock planning models have been developed for this thesis. To which degree the different models implement the railway-specific requirements described in Chapter 3 is shown in Table 4.1. Other characteristics of the models are shown in Table 4.2. The developed models are:

1. **A greedy sequential resource constrained shortest path based heuristic model**, described in Chapter 5, employing an event-activity graph, special side constraints for the resource constrained shortest path algorithm and a hill-climbing heuristic applied on a greedy and sequential train unit trajectory modification scheme. This model adheres to all railway-specific requirements;

2. **A simple, train composition type assignment model**, designated A2 and described in Section 6.2.1, used as an objective value upper bound calculation model, formulated as a mixed integer linear program;

3. **An enhanced train composition type arc multi-commodity flow model**, designated A4 and described in Section 6.2.2, used as an objective value upper bound calculation model, formulated as a mixed integer linear program;
4. **A advanced train unit type arc multi-commodity flow model**, designated B10 and described in Section 6.2.3, used as an objective value upper bound calculation model, formulated as a mixed integer linear program;
5. **A branch-and-price based matheuristic model**, described in Chapter 7 using components from model 1 above in addition to a mixed integer linear program solved with column generation and branch-and-bound. This model integrates all railway-specific requirements with the vast majority of requirements also integrated in the optimisation part of the algorithm.

Table 4.1: Overview of the requirements for rolling stock planning at DSB S-tog and to which degree the requirements are implemented in the rolling stock planning models developed for this thesis. In the table, ▼ symbolises full requirement implementation in the heuristic part of algorithm, ○ partial implementation in the optimisation part of algorithm, and ● full implementation in the optimisation part of algorithm.

		Developed model				
Requirement Category	Requirement Detail	1. Greedy heuristic	2. Upper bound calculation A2	3. Upper bound calculation A4	4. Upper bound calculation B10	5. Branch-and-price matheuristic
Infrastructure	Adhere to track length capacities for parking	▼		●	●	●
	Handle order of train units in train composit.	▼				▼
	Use platform tracks for temporary parking	▼		○	○	●
	Use side tracks for temporary parking	▼		●	●	●
	Adhere to train control system rules	▼				▼
	Adhere to coupling and decoupling rules	▼				▼
	Keep train unit balance in depot over time	▼				●
	Only one shunting per arrival/departure	▼			●	▼
	Handle split depots and track usage rules	▼		●	●	●
Timetable	Assign train units to all revenue train services	▼	●	●	●	●
	Enable non-revenue services for positioning	▼		●	●	●
	Adhere to braiding & train service seq. rules	▼		●	●	●
Rolling Stock	Adhere to platform lengths by train line	▼	●	●	●	●
	Adhere to rules on # of train units per train	▼	●	●	●	●
	Handle train composition flexible space distr.	▼				▼
Passenger Demand	Provide seats according to demand	▼	●	●	●	●
Personnel on Duty	Perform shuntings only when personnel avail.	▼			●	●
Scheduled Maintenance	Get train unit to workshop within dist. limit	▼				●
	Even out the flow of train units to workshop	▼				●
Unscheduled Maintenance	Get train unit to workshop within time limit	▼				●
Friction Sand	Get train unit to facility within distance limit	▼				●
Exterior Cleaning	Get train unit to workshop within time limit	▼				●
Graffiti Removal	Get train unit to workshop within time limit	▼				●
Interior Cleaning	Allow time to clean train units	▼				●
	Put newly cleaned train units into service	▼				●
Winter Preparedness	Get train unit to facility within time limit	▼				●
Exposure of Commercials	Expose commercials in certain regions	▼				●
Surveillance Video Requests	Get train unit to workshop within time limit	▼				●
Surface Foil Application	Get train unit to facility within time limit	▼				●
Passenger Counting Equip.	Assign specific train unit to spec. train lines	▼				●
Train Control System Equip.	Assign specific train unit to spec. train lines	▼				●
Operating Costs	Minimise energy costs	▼	●	●	●	●
	Minimise maintenance costs	▼	●	●	●	●
	Minimise infrastructure usage costs	▼	●	●	●	●
	Minimise train driver costs	▼	○	●	●	●
	Minimise depot driver costs	▼		●	●	●

Table 4.2: Overview of the characteristics of the rolling stock planning models developed for this thesis. *Requirements implemented* indicate to which degree (by count) the individual models implement the railway-specific requirements from Chapter 3. *Greedyness* indicates the degree of greedy aspects in the model. *k-optimality* indicates to which degree the individual models are solved to optimality for k selected train unit trajectories for modification (in each iteration for the heuristic and matheuristic models; in each model run, for the non-iterative upper bound calculation models). *Hot start* indicates to which degree and how models may be hot started for better model performance. *Symmetry* indicates to which degree there is unwanted symmetry in the model formulation. *Model variables cardinality* indicates number of variables in the models. *Iteration effectiveness* indicates how good with regard to objective value gain the model performs in each iteration (or model run, if applicable), and *time effectiveness* how well by time.

	Model				
Characteristic	1. Greedy heuristic	2. Upper bound calculation A2	3. Upper bound calculation A4	4. Upper bound calculation B10	5. Branch-and-price matheuristic
Requirements implemented					
Greedyness					
k-optimality					
Hotstart with uncov. revenue train services					
Hotstart with incomplete trajectories					
Symmetry in model formulation					
Model variables cardinality					
Iteration effectiveness					
Time effectiveness					

Chapter 5

A Greedy Heuristic Integrated Rolling Stock Planning Model

This chapter is published as a scientific article in *Journal of Rail Transport Planning and Management*, 5(4): 240–262, 2015, with the title *An Integrated Rolling Stock Planning Model for the Copenhagen Suburban Passenger Railway*, [105].

5.1 Introduction

5.1.1 Background and Terminology

Rolling stock planning is the process a passenger railway operator performs in order to plan how to use the rolling stock for the conveyance of passengers. The goal of the rolling stock planning process is to provide sufficient seats for passengers while at the same time keeping operating costs as low as possible. This goal is of course a highly important matter for operators of passenger railways since it is the core question of their very existence: *Can the passenger railway convey its passengers at an acceptable price?*

A passenger railway operates a *timetable* of *train services* for the conveyance of passengers for *revenue*. Rolling stock planning is performed by assigning individual *train units* to the train services from the timetable.

When producing rolling stock plans for a passenger railway, a large number of practical, railway-specific requirements need to be taken into account. These requirements relate to the railway infrastructure, the timetable, the rolling stock itself, the passenger demand, maintenance scheduling and a large number of other aspects of the railway operation.

Due to the large number of practical, railway oriented requirements and their complexity, rolling stock planning is often performed in a step-by-step manner, taking only some of the many requirements into consideration in each step. This is also the case in the rolling stock planning system currently used at DSB S-tog, the suburban passenger train operator of the City of Copenhagen. DSB S-tog is considered as case study for this paper.

In the rolling stock planning system of DSB S-tog, as it is typical for the industry, the first step is to decide how much seating capacity should be allocated to each train service. This step is called *composition planning*. Based on this, in the next step, individual train units are assigned to train services in a process called *rotation planning*. Finally, in the last step it is decided where the train units are to be parked in the *depots* when not in use. This step is called *depot planning*. Needless to say, the step wise approach may produce plans that are neither optimal nor feasible.

For DSB S-tog this is especially the case due to the very limited space in the depots where train units are parked when not in use. For this reason, the most constraining requirement for the rolling stock planning at DSB S-tog is that of being able to move the train units in and out of the depots. Planning this as the **last** step may prove highly problematic, since decisions taken in the earlier steps may limit the degrees of freedom for the depot planning steps to an extent that no feasible solution can be found. Such infeasible plans will have to be corrected manually, most often incurring extra cost.

Other suburban passenger train operators may have similar, challenging conditions that make sequential planning equally problematic. For this reason, an integration of **all** the different rolling stock planning processes is essential if an automated model is to produce plans that are usable in practice. This is achieved by the integrated rolling stock planning model proposed in this paper.

The combined process of composition planning, rotation planning and depot planning is called *circulation planning*. The circulation planning phase of rolling stock planning has a tactical scope and is conducted months before the plan is set into motion. The process of setting a circulation plan into motion is called *train unit dispatching*. This is the operational, short-term or real-time phase of rolling stock planning where last minute changes are made based on which physical train units are available, whether delays or disruptions have occurred, etc.

5.1.2 Literature Review

Until recently, operations research (OR) techniques have been applied to a wide range of specific problems in the railway industry, which are summarised in various surveys [39, 7, 64, 34, 71]. At present, the challenges in the adaptation of OR techniques in the railway industry seem not only to lie in finding solutions to each specific problem, but even more so in integrating the individual solution methods to the highly interconnected specific problems into holistic, integrated models. By integrating the specific models with each other, sub-optimal solutions can be avoided. The tendencies for the integration of models are currently also seen in the airline industry [98].

Table 5.1 shows an overview of characteristics of selected and reviewed, recent literature for rolling stock planning. The characteristics are grouped as follows: The overall topic of article; The railway planning processes it addresses; The type of the model proposed; The properties of the model graph (all reviewed models feature a graph); The railway-specific requirements the model integrates; The objective of the model; And finally, the solution method applied.

As may be seen from Table 5.1, a large portion of the reviewed methods use an *arc based multi-commodity flow* or similar modelling scheme. In such a scheme the flow of train units or locomotives is modelled in a flow graph, with flow conservation constraints on each vertex of the graph making sure the flow into the vertex equals the flow out of it. Arc based flow models are typically relatively low in complexity, a presumed reason for their widespread use. In arc based flow models, however, it adds to complexity to model sub-path constraints such as recurring maintenance at regular distance intervals.

In *path based multi-commodity flow models* on the other hand, each potential sequence of movements of the individual train unit or locomotive is modelled (e. g., by enumeration), making it easier to also take recurring distance related constraints such as maintenance into account.

In Table 5.1, the literature reviewed is also categorised according to the properties of the graphs involved in the models. As may be seen, most models use a *space-time graph* type (also called time-expanded graph type), where each vertex in the graph is an event in space and time, e. g. the arrival of a train at a given time at a given station (in space). Correspondingly, arcs

Table 5.1: Overview of characteristics in selected and reviewed, recent literature specific to rolling stock planning. Characteristics of the integrated rolling stock planning model proposed in this paper are listed at the bottom for comparison. *) Common requirements include: Timetable, overall infrastructure, rolling stock, passenger or freight demand requirements. Note that subsidiary requirements like cyclicity, robustness and disruption recovery with minimal changes have been omitted.

		Topic		Process				Model Type					
Authors	Year	Passenger railway	Freight railway	Composition planning	Rotation planning	Depot planning	Train unit dispatching	Assignment etc.	Set partitioning	Arc multi-commodity flow etc.	Hyper arc multi-commodity flow	Hamiltonian cycle	Path multi-commodity flow
Cordeau [37]	2000	●		●	●					●			
Cordeau et al. [38]	2001	●		●	●					●			
Brucker et al. [24]	2003	●		●	●					●			
Abbink et al. [5]	2004	●		●				●					
Razen [89]	2004		●	●	●					●			
Ahuja et al. [9]	2005		●	●	●					●			
Freling et al. [55]	2005	●				●			●				
Alfieri et al. [10]	2006	●		●	●					●			
Fioole et al. [51]	2006	●		●	●					●			
Føns [53]	2006	●				●		●					
Maróti and Kroon [80]	2007	●					●			●			
Jha et al. [67]	2008		●	●	●								●
Kroon et al. [72]	2008	●				●		●					
Peeters and Kroon [86]	2008	●		●	●					●			
Vaidyanathan et al. [109]	2008		●	●	●					●			
Dirksen [46]	2010	●				●		●					
Beygo [17]	2011	●				●			●				
Borndörfer et al. [20]	2011	●		●	●						●		
Cadarso and Marín [28]	2011	●		●	●					●			
Borndörfer et al. [21]	2012	●		●	●						●		
Cacchiani et al. [27]	2012	●		●	●			●					
Giacco et al. [57]	2014	●		●	●							●	
Haahr et al. [60]	2014	●					●						●
(Model proposed in this paper)	2015	●		●	●	●	●						●

		Graph Properties					Requirements Integration						
Authors	Year	Vertex is space-time event	Vertex is train service	Vertex is composition	Vertex is approach type	Vertex is maintenance	Common requirements *)	Train composition order	Maintenance etc. (by time)	Maintenance etc. (by distance)	Personnel on duty	Depot capacity	Depot topology
Cordeau [37]	2000	●					●						
Cordeau et al. [38]	2001	●				●	●		●	●			
Brucker et al. [24]	2003	●					●					●	
Abbink et al. [5]	2004						●						
Razen [89]	2004	●					●						
Ahuja et al. [9]	2005	●					●						
Freling et al. [55]	2005			●	●			●				●	●
Alfieri et al. [10]	2006	●	●	●			●	●					
Fioole et al. [51]	2006	●	●	●	●		●						
Føns [53]	2006			●	●			●				●	●
Maróti and Kroon [80]	2007	●					●		●	●			
Jha et al. [67]	2008	●					●						
Kroon et al. [72]	2008			●	●			●				●	●
Peeters and Kroon [86]	2008		●	●			●	●					
Vaidyanathan et al. [109]	2008	●					●		●				
Dirksen [46]	2010							●		●		●	●
Beygo [17]	2011							●				●	●
Borndörfer et al. [20]	2011	●	●	●		●	●			●			
Cadarso and Marín [28]	2011	●					●					●	
Borndörfer et al. [21]	2012	●	●	●		●	●			●		●	
Cacchiani et al. [27]	2012		●	●			●						
Giacco et al. [57]	2014		●			●	●		●	●			
Haahr et al. [60]	2014	●					●					●	
(Model proposed in this paper)	2015	●					●	●	●	●	●	●	●

		Objective							Solution Method				
Authors	Year	Minimise seat shortage	Minimise cost	Minimise penalties	Minimise # of train units etc.	Maximise benefits	Minimise # of train shuntings	Feasibility	Commercial solver etc.	Heuristics etc.	Column generation etc.	Branch-and-price etc.	Benders decomposition
Cordeau [37]	2000		●						●				●
Cordeau et al. [38]	2001		●						●				●
Brucker et al. [24]	2003		●	●						●			
Abbink et al. [5]	2004	●							●				
Razen [89]	2004				●				●	●			
Ahuja et al. [9]	2005		●	●					●	●			
Freling et al. [55]	2005			●					●		●		
Alfieri et al. [10]	2006		●						●	●			
Fioole et al. [51]	2006	●			●		●		●	●			
Føns [53]	2006							●	●			●	
Maróti and Kroon [80]	2007			●					●	●			
Jha et al. [67]	2008		●	●					●	●			
Kroon et al. [72]	2008			●					●				
Peeters and Kroon [86]	2008		●						●			●	
Vaidyanathan et al. [109]	2008		●	●					●				
Dirksen [46]	2010							●	●				
Beygo [17]	2011							●	●				
Borndörfer et al. [20]	2011		●	●					●	●			
Cadarso and Marín [28]	2011		●	●					●				
Borndörfer et al. [21]	2012		●	●					●	●			
Cacchiani et al. [27]	2012				●					●			
Giacco et al. [57]	2014				●				●				
Haahr et al. [60]	2014		●	●					●		●		
(Model proposed in this paper)	2015		●	●		●				●			

in a space-time graph may e. g. represent train services. Such a graph is also an *event-activity graph*, referring to the vertices as events and the arcs as activities. This type of graph is also well known in the airline industry [96, 14, 88, 98].

Some authors use the *edge-to-vertex dual* or *line graph* [62] graph type, conjugated from the space-time graph mentioned before. In the line graph type, vertices represent train services whereas arcs may e. g. represent the possibilities of a train unit to perform train services in sequence.

Similar "line graph" style graph types are used to model train composition changes at stations, the approach direction in depot planning and maintenance constraints. In two papers a hypergraph is used [20, 21].

As may be seen from Table 5.1, the models in the reviewed literature integrate a different number of railway-specific requirements, with a slight tendency that recent models integrate more requirements than earlier ones.

In most of the reviewed models the objective is to minimise operational costs and/or penalties. Some models are used to minimise seat shortages or the number of train units needed etc. Since depot planning is mainly a matter of fixed costs, some depot planning models have only feasibility as their objective.

All solution methods applied in the reviewed literature involve commercial solvers and/or heuristics. In addition, decomposition techniques such as column generation, branch-and-price and Benders decomposition are used in some cases.

The following is a brief overview of the size of the data instances used in the experiments reported in the reviewed literature. [37, 38] assign train units to 300 train services for the period of one week. [24] assign 6 different types of train units to 200 train services. [5] circulate train units for 188 passenger train services, while [89] circulates train units for 86 freight train services. In [9] 3,324 train services have up to 1,600 train units assigned to them by 5 types. [55] distribute 600 train compositions with 1,100 train units on 19 depot tracks. In [10] 12 train compositions have train units assigned to them on an intercity line with 30 min. frequency. [51] assign 85 train units in 2 categories to 67 train services. [80] perform maintenance planning of 47 train units serving 800 train services per day for a period of up to 5 consecutive days. [67] assigns 1,200 collections of train units to 350 freight train services. [71] assigns up to 600 train services with 1,100 train units to 19 depot tracks. In [109] 388 freight train services have 6 train unit types in 8 possible compositions assigned to them. [20, 21] circulate the entire German ICE high speed fleet for a week, resulting in a model graph with more than 60 million hyperarcs in it. [28] assign rolling stock to up to 400 train services, and [27] up to 76 train units in up to 10 categories to up to 600 train services. [57] plans maintenance for a timetable with up to 104 train services.

In comparison, DSB S-tog has approximately 1,350 train services per day with 122 train units in operation, routed to 53 depot tracks. In the integrated rolling stock planning model proposed here, this yields a model graph with up to approx. 28,000 arcs. The following references use DSB S-tog data: [53, 46] perform depot planning by each depot, [17] circulates rolling stock and [60] recover from a disrupted rolling stock plan (but disregard the individual depot tracks).

To the best of our knowledge, no integrated planning model for the requirements we are considering has been worked out or used, neither in the literature nor in industry practice.

On a general note, the terminology used in rolling stock planning literature shows great variation. In some articles a terminology is used that has weak connotations to the actual railway operation. Needless to say, this neither facilitates the understanding nor the comparison of methods across literature.

5.1.3 Scientific Contribution

The scientific contribution of the proposed integrated rolling stock planning model is that of integrating into one process, processes which are normally solved separately. In particular, the integration of train unit to train service assignment, maintenance planning (by distance) and depot planning is not known from literature. The processes are integrated using a heuristic framework.

A further scientific contribution is the development of special side constraints to a (resource constrained) shortest path algorithm that can handle the individual order of train units in train compositions so that no train unit will obstruct the movement of another. We call this new concept *unit order flow conservation*.

In addition to this, experiments are conducted using real data instances with all the peculiarities of actual production data in order to prove the scientific viability of the model in realistic conditions.

5.1.4 How This Paper Is Structured

Section 5.2 formulates the problem to be solved and presents an overview of the solution concepts. Moreover, an overview of the proposed model is presented, along with a mathematical formulation. Next, the different parts of the solution approach are presented in detail: Section 5.3 describes the data model for the timetable and the infrastructure (the space-time graph), Section 5.4 presents the data model for train units, Section 5.5 the path finding algorithm and Section 5.6 the surrounding heuristic framework. Section 5.7 describes the real-world data instances used in the evaluation of the integrated model and the results obtained. Lastly, Section 5.8 discusses the implications of the proposed methods and outlines further research.

5.2 The Integrated Rolling Stock Planning Problem

5.2.1 Problem Formulation

Seen from an overall business perspective, the goal of the rolling stock planning problem is to provide seats for passengers while at the same time keeping operational costs to a minimum. Seen from a more detailed operational perspective, rolling stock planning is about deciding which individual train unit should be assigned to which train service. By doing so, one has implicitly assigned seating capacity to the train services. At the same time, it must also be decided when and where train units should be parked at the depots when not in use. All these decisions must be taken in such a way that operational costs are minimised and all of the practical, railway-specific requirements are adhered to.

5.2.2 Solution Concepts

The underlying solution idea to the problem presented here is to look at the above mentioned assignment of train units to train services in an aggregated way: A rolling stock plan may be entirely described by the movement of its train units in space and time. The movement of a particular train unit in space and time for a particular period of time is called a *train unit trajectory*. A typical train unit trajectory starts off with the train unit being parked at a depot track before being *shunted* to the platform. From the platform, the train unit may then be assigned to a revenue train service starting at this *origin station*. At the *terminal station* of that particular train service, the train unit may *turn around* to be assigned to another train service in the opposite direction. This train service may be a *non-revenue train service* running without passengers with the purpose of positioning the train unit for later use. Typically, a train unit trajectory ends by having the train unit being shunted back into a depot track for parking for the remaining time of the given period.

As such, a train unit trajectory describes which train services the train unit in question is assigned to for the given period, including information of used turnaround times between train services, and at which depot tracks the individual train unit is parked when not in use. Formulated in the context of this solution idea, one can say that the rolling stock planning problem is to decide the individual train unit trajectories of the train units, thereby offering enough seating capacity for the passengers and at the same time keeping cost at a minimum and adhering to all railway-specific requirements.

To be able to find new candidate train unit trajectories that are attractive (that is, new ways that the individual train units should move in space and time), we need a measure of the attractiveness of each train unit trajectory. This measure is called the *additional net value*, defined as the additional benefit that may be achieved by assigning a train unit of a given type to perform the operations represented by the train unit trajectory in question, minus the incurred penalties and factual costs for doing so. Penalties are awarded for undesirable aspects of the rolling stock plan.

A positive additional net value for a given train unit trajectory indicates that there is good "value for money" in letting the train unit perform the given train unit trajectory, since the benefits of doing so outweigh the costs and penalties. A negative value would indicate that the costs and penalties outweigh the benefits, in most cases an unattractive option.

5.2.3 Requirements Overview

The following is a brief overview of the practical, railway-specific requirements for rolling stock planning at DSB S-tog. For a list of all requirements, see the left part of Table 5.4 on page 72. For a full description of all the requirements, see [103].

In the long term circulation planning phase of rolling stock planning, the following requirements must be taken into account: The physical railway **infrastructure** must be adhered to, e. g., depot track capacities, the rules of the train control system and the order in which train units may be parked so as not to obstruct each other's movements; All trains services of the **timetable** must have a least one train unit assigned; Only the available **rolling stock** can be used in the plan; The plan should provide seating capacity according to the **passenger demand** and provide an even distribution of flexible space for bicycles etc.; Planned shunting operations in the depot should have sufficient **personnel on duty**; Train units must undergo **interior** and **exterior cleaning**, **surface foil application** and **winter preparedness** treatment at regular time intervals; At regular service distance intervals train units must undergo **scheduled maintenance** etc., and consumables such as **friction sand** must be refilled; Certain train services must have train units with additional **train control system equipment** installed, special **passenger counting equipment** installed and/or perform predefined **exposure of commercials**.

In the short-term or real-time train unit dispatching phase of rolling stock planning, additional requirements include: **Exterior graffiti removal** and **unscheduled maintenance** on demand and sometimes within a given time frame; Make available train units to meet **surveillance video requests from the Police** within a given time frame.

5.2.4 Model Overview

The integrated rolling stock planning model proposed here integrates all the mentioned requirements using four main components. The first two components constitute a data model for the rolling stock plan. The last two components are algorithms applied to modify a given rolling stock plan in order to improve it. The four components are:

1. **The combined timetable and infrastructure data model:** A space-time graph with extended arc and vertex attributes, describing the timetable, the infrastructure, passenger demand, personnel on duty, which train service has which train unit assigned to it and in which individual, relative order, etc. This component is described in Section 5.3;

2. **The data model for train units**, interconnected with the space-time graph, describing the activities of the train units, e. g. which train unit is assigned to which train service. This component is described in Section 5.4;

3. **A special-purpose resource constrained shortest path algorithm with side constraints** operating on the space-time graph. This algorithm is used to find new candidate train unit trajectories taking into account the maximum service distance a train unit may perform as a resource constraint. As a side constraint, the individual, relative position of the train unit in relation to the other train units in the space-time graph is handled, determining which movements the train unit can perform based on its relative position. Also the flexible space distribution is handled as a side constraint. This whole component is described in Section 5.5;

4. **A heuristic framework** to accept or reject the candidate train unit trajectories found using the previously described components. The overall concept of the heuristic framework is to remove a number of train unit trajectories from an existing rolling stock plan and then, one by one, to create a new trajectory and insert it into the plan. The newly inserted trajectories are accepted if they produce an increase in the objective function value; if not, they are rejected, and the previous ones are re-inserted. The heuristic component is described in Section 5.6.

For an overview of the first three components, their aspects and which requirements they implement, see Table 5.4 on page 72.

5.2.5 Mathematical Formulation

The mathematical formulation of the proposed integrated rolling stock planning model presented in this paper is based on sets (with corresponding indices) and parameters as defined in Tables 5.2 and 5.3.

The heuristic component of the integrated rolling stock planning model (Component 4 in the overview in Section 5.2.4) is governing the program flow. It works by (in each iteration) selecting k number of train units U^* and removing each of their original train unit trajectories $j_u^- \in J$ (Component 2) from the graph G (Component 1). Next, for each train unit $u \in U^*$ a new train unit trajectory $j_u^+ \in J$ (Component 2) is then found using the shortest path algorithm (Component 3). This train unit trajectory is then inserted into the graph. (As will be seen in Section 5.5, the path finding algorithm takes into account all practical, railway oriented requirements, including train unit order, so no additional feasibility check is needed prior to the successive insertion of each newly found train unit trajectory.) This process is repeated for all train units in U^*.

Table 5.2: Sets and their corresponding indices in the mathematical formulation of the heuristically based integrated rolling stock planning model, their domains and definitions, ordered alphabetically by symbol. Sets have symbols in upper case, and their corresponding indices have the same symbol in lower case without subscripts or superscripts.

Symbol	Description	Index, domain, definition
A	Arcs in the space-time graph G, each arc going from one vertex to another. As such each arc also has a corresponding time interval $p \in P$	$a \in A$; $a = (v_1, v_2)$ $v_1, v_2 \in V$
A_j	The arcs of train unit trajectory j	$a \in A_j \subset A$
G	The directed and acyclic space-time graph with vertices V and arcs A. The graph has extended attributes as described in Section 5.3 on page 73	$G = (V, A)$
I	Train unit types	$i \in I = \{\frac{1}{2}, 1\}$
J	All possible train unit trajectories. A train unit trajectory is a path through the space-time graph G representing the movement in space and time of a train unit	$j \in J$ $j = a_1, a_2, ..., a_{\lvert j \rvert}$ $a \in A$
P	All possible time intervals, a time interval being a sorted 2-tuple of point in time. Each arc $a \in A$ represents a time interval	$p \in P$; $p = (t_1, t_2)$ $t_1, t_2 \in T$ $t_1 < t_2$
Q	Points in space, being the union of each depot track at every station, each side track at every station, all platform tracks [as a whole] at every station	$q \in Q$
T	Points in time	$t \in T$
U	Individual train units currently available	$u \in U$
U^*	The set of train units selected for train unit trajectory subtraction, creation and addition in the heuristic	$u \in U^* \subset U$ $\lvert U^* \rvert = k$
V	Vertices in the space-time graph G, each vertex being a *point in space, point in time* tuple	$v \in V$; $v = (q, t)$ $q \in Q$; $t \in T$

Table 5.3: Parameters in the mathematical formulation of the heuristically based integrated rolling stock planning model, their domains and definitions, ordered alphabetically by symbol. All parameters have symbols in lower case.

Symbol	Description	Domain, definition
$b^-(a,u)$	Benefits lost by subtracting train unit u from the arc a	$\in \mathbb{R}_0^+$
$b^+(a,u)$	Benefits gained by adding train unit u to the arc a	$\in \mathbb{R}_0^+$
$c^-(a,u)$	Costs saved by subtracting train unit u from the arc a	$\in \mathbb{R}_0^+$
$c^+(a,u)$	Costs added by adding train unit u from the arc a	$\in \mathbb{R}_0^+$
j_u^-	The original train unit trajectory belonging to train unit $u \in U^*$ scheduled for subtraction from the graph G	$j_u^- \in J$
j_u^+	The new train unit trajectory belonging to train unit $u \in U^*$ scheduled for addition into the space-time graph G	$j_u^+ \in J$
k	The number of train units to select in order to subtract their original train unit trajectories and add their newly found train unit trajectories	$k \in \mathbb{N}_1$ $k = \lvert U^* \rvert$
n	Number of test runs performed on each data instance for algorithm performance testing, see Table 5.5 on page 88	$n \in \mathbb{N}_1$
$p^-(a,u)$	Penalties saved by adding **or** subtracting train unit u to or from the arc a	$p^-(a,u) \in \mathbb{R}_0^+$
$p^+(a,u)$	Penalties awarded by adding **or** subtracting train unit u to or from the arc a	$p^+(a,u) \in \mathbb{R}_0^+$
z_Δ	The iteration net value increase, calculated according to Equation (5.1) on page 83 for the subset of trajectories that have been changed in the iteration	$z_\Delta \in \mathbb{R}$
$z^-(j_u^-)$	The subtractional net value, i. e., the net value of subtracting train unit trajectory j_u^- from the space-time graph G, calculated according to Equation (5.2) on page 83 before the train unit trajectory is subtracted	$z^-(j_u^-) \in \mathbb{R}$
$z^+(j_u^+)$	The additional net value, i. e., the net value of adding train unit trajectory j_u^+ into the space-time graph G, calculated according to Equation (5.3) on page 84 before the train unit trajectory is added	$z^+(j_u^+) \in \mathbb{R}$

Table 5.4: Overview of the requirements for rolling stock planning at DSB S-tog and how they are implemented in the model.

		Timetable and infrastructure data model (Section 5.3)															Train unit data model (5.4)			Path finding algorithm (sect. 5.5)	
		Arc attributes										Vertex attributes									
Requirement Category	Requirement Detail	Length of track repr. by arc	Service Distance	Assigned train units limit	Assigned train composition	Transition positions	Depot drivers required	Train service line	Additional/subtract. benefit	Additional/subtract. cost	Additional/subtract. penalty	Limit on # of shuntings	Assigned # of shuntings	Date and time	Depot drivers on duty (aggr.)	Graph Topology	Allowed on tr. srv. lines	Start/finish time and space	Service distance limit	Resource constraints	Side constraints
Infrastructure	Adhere to track length capacities	●			●																
	Handle order of train units in train composition				●	●															●
	Use platform tracks for temporary parking										●					●					
	Use side tracks for temporary parking															●					
	Adhere to train control system rules				●	●															
	Adhere to coupling and decoupling rules				●	●															●
	Keep train unit balance in depot over time																	●			
	Only one shunting per arrival/departure											●	●								
	Handle split depots and track usage rules										●					●					
Timetable	Assign train units to all revenue train services				●						●										
	Enable non-revenue services for positioning															●					
	Adhere to braiding and train service seq. rules															●					
Rolling Stock	Adhere to platform lengths by train service line	●						●													
	Adhere to rules on # of train units per train service			●	●																
	Handle train composition flexible space distribution				●	●															●
Passenger Demand	Provide seats according to demand								●												
Personnel on Duty	Perform shuntings only when personnel avail.						●						●	●	●						
Scheduled Maintenance	Get train unit to workshop within distance limit		●																●	●	
	Even out the flow of train units to workshop		●															●	●	●	
Unscheduled Maintenance	Get train unit to workshop within time limit																	●			
Friction Sand	Get train unit to facility within distance limit		●																●	●	
Exterior Cleaning	Get train unit to workshop within time limit																	●			
Graffiti Removal	Get train unit to workshop within time limit																	●			
Interior Cleaning	Allow time to clean train units																	●			
	Put newly cleaned train units into service																	●			
Winter Preparedness	Get train unit to facility within time limit																	●			
Exposure of Commercials	Expose commercials in certain regions							●									●				
Surveillance Video Requests	Get train unit to workshop within time limit																	●			
Surface Foil Application	Get train unit to facility within time limit																	●			
Passenger Counting Equip.	Assign specific train unit to specific train service lines							●									●				
Train Control System Equip.	Assign specific train unit to specific train service lines							●									●				
Operating Costs	Minimise energy costs		●							●											
	Minimise maintenance costs		●							●											
	Minimise infrastructure usage costs		●							●											
	Minimise train driver costs									●											
	Minimise depot driver costs									●											

Next, the heuristic component checks the change in objective value for the iteration in question, the *iteration net value increase* z_Δ. If z_Δ is positive, each newly created train unit trajectory j_u^+ for the selected train units U^* is kept in the graph G. If z_Δ is not positive, for each train unit $u \in U^*$, the newly found train unit trajectory j_u^+ is removed from the graph, and the original j_u^- reinserted, equivalent to the changes in the current iteration being rolled back, i. e., the train units U^* having their original train unit trajectories reinstated.

5.3 Timetable and Infrastructure Data Model

The first component of the integrated rolling stock planning model is the combined data model for the timetable and the railway infrastructure in the form of a directed, acyclic, space-time graph G. Space-time graphs are well known in the railway industry, the first use for timetabling is attributed to French engineer Ibry prior to 1885 [79]. Space-time graphs are also known from other industries, for an example of a recent application in the airline industry, see [14].

In the proposed space-time graph, the arcs A represent the possibility of a train unit to move in space and time or in time only. The vertices V represent space and time events, that is, points in space and time where a train unit may perform different movements later on. For instance, a train unit arriving as a revenue train service to one of the platforms of a station is an event, after which the train unit may either stay at the platform and turn around to the next departure of a train service (one outgoing arc from that vertex) or be shunted to a depot track (another outgoing arc from the same vertex).

A schematic illustration of the principles in the space-time graph is shown in Figure 5.1.

In the graph, all platform tracks of a given station are treated as a whole. However, to be able to model how one train unit may turn around from one train service to the next at the platform tracks, vertices representing arrivals to the platform tracks are separate from those representing departures, even though their point in time may be the same.

In the model, non-revenue train services for the positioning of train units are static in the sense that they are not decided when solving the model, but read from the timetable data as given input.

The space-time graph has three different aspects which are used in the integrated rolling stock planning model. These aspects are:

1. **The topology of the graph**, that is, which arcs are connected to which other arcs by their common vertices. This is described in Section 5.3.1.

2. **The arc attributes**, describing features related to train services, train shuntings, depot tracks, etc. This is described in Section 5.3.2;

3. **The vertex attributes**, describing features related to depot drivers on duty, number of shuntings assigned to each arrival and departure, etc. This is described in Section 5.3.3;

As mentioned earlier, the space-time graph is used as the data model for a resource constrained shortest path algorithm with side constraints. In order for this to work, the space-time graph exhibits two features to the path finding algorithm for each arc. These features are:

1. **Arc feasibility:** *Is it feasible for a given train unit to traverse the arc?* This is determined by factors like how many train units already traverse the arc compared to the capacity of the arc, etc.;

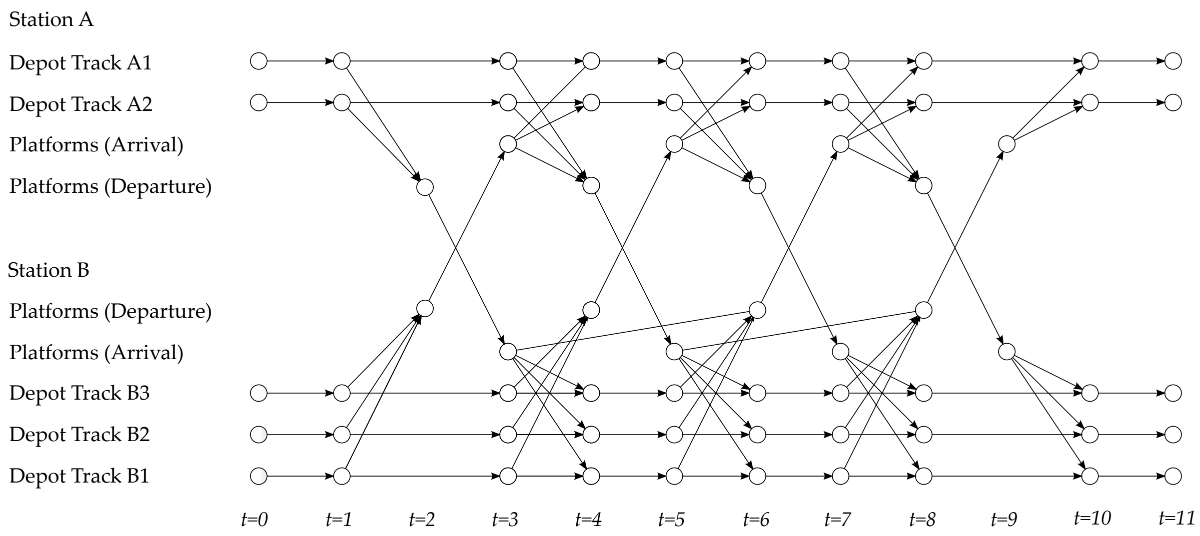

Figure 5.1: A schematic illustration of the space-time graph G. The time increases from left to right. Each vertex $v \in V$ represents an event, i. e., a space-time tuple $v = (q, t)$. Each arc $a \in A$ in the graph represents the possibility for a train unit to move in time and space (as a train service or train shunting) or in time only (as being parked at a depot track, or in the process of being turned around at a platform track). For example, the arc departing from station A at $t = 2$ arriving at station B at $t = 3$ is a train service. This arc has two corresponding vertices, the first one being the departure from the platforms at station A at $t = 2$, the second one the arrival at station B at $t = 3$. Note that the flow arcs going from the station source and to the station sink are omitted in this diagram. Also omitted are arcs representing overnight parking at the platform tracks. For a diagram including these types of arcs, see Appendix A.3 on page 164.

2. **Resources consumption:** *What is the resources consumption of traversing the arc with a given train unit?* In the model, two resources are defined: The first one is the net value of traversing the arc, measured as the benefit of doing so minus the cost and penalties incurred. The second resource is the service distance travelled, this resource is constrained since a train unit may e. g., only travel a certain distance after which it has to undergo maintenance.

The path finding algorithm (see Section 5.5) is relaying feasibility and resource consumption requests to the space-time graph through these two features. The features are exhibited based on the values of the arc and vertex attributes and the graph topology as described in the next sections. The attributes are described in the same order as they appear in Table 5.4.

5.3.1 Topology

The topology of the space-time graph reflects the allowed movements of train units as stated by the current business rules at DSB S-tog. For example, there are no arcs connecting a depot track with another depot track, since a current business rule prohibits depot internal shunting. The graph also incorporates turnaround times between train services.

5.3.2 Arc Attributes

This section gives an overview of the different arc attributes and how they are utilised to implement the different requirements.

The maximum length that train units assigned to an arc may utilise is governed by the *track length* attribute, set according to the minimum platform length on the stations visited by train service, or the actual track length for platform, side and depot tracks.

The *service distance* attribute represents the real-world service distance travelled in physical space for that arc. This attribute is used to calculate the distance related cost for energy, maintenance and infrastructure usage, etc.

The *assigned train units limit* attribute is set to two for arcs representing train services, since at DSB S-tog, no more than two train units may be coupled together when running on the main line tracks. For other arc types the train unit capacity is infinite.

The main arc attribute in the space-time graph is the *assigned train composition* attribute, indicating which train units are assigned to the activity represented by the arc, and in what individual order. This attribute is used to check for train unit order feasibility (see Section 5.4.2) and overall resources consumption in comparison with other arc attributes.

The attribute for *transition positions* keeps track of at which positions couplings and decouplings may take place for the train composition assigned to the arc in question (see Section 5.4.2).

The attribute for *depot drivers required* keeps track of whether a depot driver is required for the activity represented by the arc. Only train shunting arcs to and from depot tracks feature this attribute, shunting to and from side tracks is performed by the train drivers. Also on the first and last train services in the weekend, the train drivers themselves perform the actual shunting to and from the depot.

The *train service line* attribute keeps track of which train service line (as defined in the timetable) a given train service belongs to in order to make sure that platform lengths are adhered to, commercials are exposed and that the technical equipment of the train is in accordance with the requirements for that train service line.

The *additional benefit* attribute quantifies the benefit of assigning (adding) another train unit of a given type to the arc in the space-time graph. The *subtractional benefit* is equivalently the benefit of removing (subtracting) a train unit from the assigned train composition of that arc. As mentioned in Section 5.2.2, the additional net value is calculated as the additional benefit minus the additional penalties awarded and additional costs incurred. As such, the additional benefit attribute is the only positive driver of the integrated rolling stock planning model. The attribute represents the economic value of providing seats to passengers that demand them. The value of the additional benefit is calculated according to stated preference time penalties for having no seat, the specific value of time for commuters and the duration of an average travel [82]. For the case of DSB S-tog, this yields a value of 0.44 DKK/min. This is roughly equivalent to the actual specific revenue gained by DSB S-tog by means of ticket sales for conveying an average passenger. Note that only arcs representing revenue train services can provide a benefit since they are the only type of arc that represent the conveyance of passengers.

The *additional cost* attribute is a resources consumption attribute quantifying the factual total cost for the activity represented by the arc, as incurred by the assigning (adding) another train unit of a given type, calculated as the sum of the cost for train or depot drivers, the cost for technical maintenance, the energy cost and the cost for using the railway infrastructure. The *subtractional cost* attribute is analogously quantifying the cost saved when removing (subtracting) a given train unit from the assigned train composition of that arc.

For further details on how the costs and benefits are calculated, see Appendix B.2.

The *additional penalties* attribute quantifies the penalties awarded and saved by assigning (adding) a given train unit to the arc. The *subtractional penalties* attribute quantifies the penalties awarded and saved by removing (subtracting) a given train unit from the train composition assigned to the arc. Penalties are awarded for aspects of the plan that are not desired, e. g., uncovered train services or train shuntings from depot tracks to platform tracks where the train unit in question has to pass the main line tracks under way.

5.3.3 Vertex Attributes

All of the vertex attributes relate to the process of train shuntings to and from depot and side tracks. The limit on number of train shuntings quantifies the DSB S-tog business rule of an upper limit of one train shunting per train service departure or arrival. Vertices not being part of a train service have no limit on the number of train shuntings being performed.

The *assigned number of train shuntings* attribute keeps track of how many train shuntings are actually performed by assigned train units passing through the vertex.

The *date and time* attribute is used to map the time interval $p \in P$ of the arc being used for train shunting to and from depot tracks with the time intervals of the depot drivers on duty. If a train shunting arc is used, the date and time attributes on its vertices are matched with the supply in time of depot drivers for that particular depot station. If the demand from the current arc and the demand from other train shunting arcs being used at intersecting times may be supplied by the depot drivers on duty, the arc is feasible. If not, the arc is infeasible.

The *depot drivers on duty* attribute keeps track of the duties of the depot drivers by time and station. This is an aggregated attribute in the sense that the depot drivers on duty are shared between all the vertices on a given depot station for the time frame of that duty.

5.4 Train Unit Data Model

The second component of the proposed integrated rolling stock planning model is the data model for the train units. Its different attributes are described in the following.

The *allowed on train service lines* attribute keeps track of which train service lines (as defined in the timetable) the train unit in question may be assigned to. This is in order to adhere to restrictions on installed equipment and the exposure of commercials.

The *start/finish time-and-space* attribute is used to set the start and finish points in space and time for each train unit $u \in U$. This way the model can handle where train units are at the beginning and end of the plan period in order to e. g., send train units to maintenance at a given time (and by way of the service distance attribute also within the service distance limit). Furthermore, the balance of train units at each depot can be controlled using this attribute.

The *service distance limit* attribute keeps track of the service distance that the train unit in question can perform before it has to go into maintenance or have consumables refilled.

Two important features of the train unit data model are the train unit trajectories and the train unit order. These features are described in Sections 5.4.1 and 5.4.2.

5.4.1 Train Unit Trajectories

A train unit trajectory $j \in J$ keeps track of which activities the train unit in question is assigned to. As such, train unit trajectories are paths (i. e., ordered collections of arcs, $j = a_1, a_2, ..., a_{|j|}$) from the space-time graph representing the movement of their respective train units in space and time or time only. See Figure 5.2 on page 78 for an illustration of train unit trajectories in the space-time graph. Figure 5.2 shows an example of how four train units move through time and space along their train unit trajectories. If two train unit trajectories use the same arc, this means that the two train units are coupled together as a *train composition*. Depending on the type of arc, they may either be coupled together and running as a train service, or be coupled and parked on at depot track. The order by which the train units are coupled is maintained by the space-time graph. Note that in the example the red and blue train units exchange places in the execution of the plan, however the balance of train units on the individual depot tracks remains constant.

The space-time graph is used to find new candidate trajectories for train units to improve the plan. Since the graph already contains information on the existing trajectories assigned, only candidate trajectories feasible in conjunction with the existing ones can be found.

New candidate trajectories are found using a specially constructed resource constrained shortest path algorithm with side constraints. This algorithm is described in Section 5.5.

5.4.2 Train Unit Order

The train unit data model also keeps track of the order of the individual train units relative to each other. In the following, the logic for coupling and decoupling train compositions in relation to the order of the train units will be explained. This logic is specific to the train control system rules and business rules currently in effect at DSB S-tog.

The simple explanation is this: *At the platform, train units are being coupled and decoupled in the direction that is facing the depot. At the depot, train units are being coupled and decoupled in the direction facing the platform.* This is illustrated in Figure 5.3.

A *train composition* is the ordered sequence of one or more train units coupled together. At DSB S-tog, train compositions consisting of one or two train units may be assigned to revenue train services. Train compositions of more than two train units may be formed when parking train units at a depot (and thereby coupling them).

Current business and train system control rules at DSB S-tog state that when a decoupling takes place at the platform, the train composition moving away must move to the depot. It may not be assigned to a train service. Furthermore, when a coupling is to take place at a platform, the train composition moving in to couple must come from the depot. It may not come from the main line, i. e., from a train service.

The term *platform train composition* is used to denote the train composition in the operation that is facing the platform. Similarly, the *depot train composition* is the train composition that is facing the depot.

The term *relative position* denotes how an object (platform track, depot track, train composition, train unit) is oriented relative to another. For DSB S-tog, the relative position can be either *North* or *South*. For example, the relative position of a depot track to a platform track at its corresponding station may be South, meaning that to reach the depot track from the platform track, train units must move towards the South. Equivalently, a train unit may have the relative position South of another train unit. At the same time, this also means that the other train unit has the opposite relative position, i. e. North, of the first one.

The individual train units in a train composition have a relative position to each other. With the proposed definition, the relative position of train units in compositions is conserved in all feasible coupling and decoupling operations.

Picture the **coupling** of two train compositions at a **platform track**, one being a platform train composition, the other being a depot train composition. The situation before the depot train composition is shunted in from the depot is depicted in Figure 5.3d, the situation after coupling in Figure 5.3b. After coupling, the original depot train composition will have the relative position to the original platform train composition (in the new train composition) equal to the relative position of the depot to the platform (in other words, equal to the relative position of where the train shunting **started**). If, like in the example in Figure 5.3, the relative position of a depot track is to the South of a platform track, then the original depot train composition being

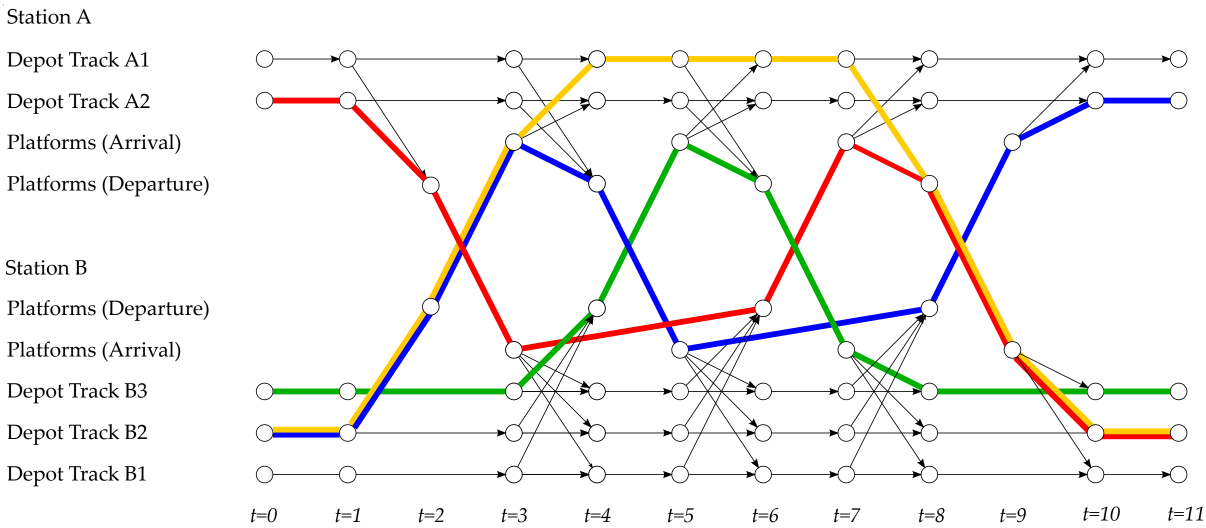

Figure 5.2: An example of train unit trajectories in the space-time graph, represented in colours red, green, blue and yellow. In this example, the red train unit starts at station A on depot track A2 at time $t = 0$. At $t = 1$ the red train unit is shunted to the platform, from which it departs as a train service at $t = 2$, arriving at station B at $t = 3$. At station B no arc exists to connect the arrival with the departure at $t = 4$. This is because the time difference between arrival and subsequent departure is less than the minimum turnaround time. For this reason, the red train unit waits at the platform from which it departs at $t = 6$, arriving at station A at $t = 7$. Station A has a shorter turnaround time and the red train unit may thus depart again at $t = 8$. Prior to departure it is coupled with the yellow train unit that is being shunted in from depot track A1 at $t = 7$. See Figure 5.3 for examples of how train units are parked at different times.

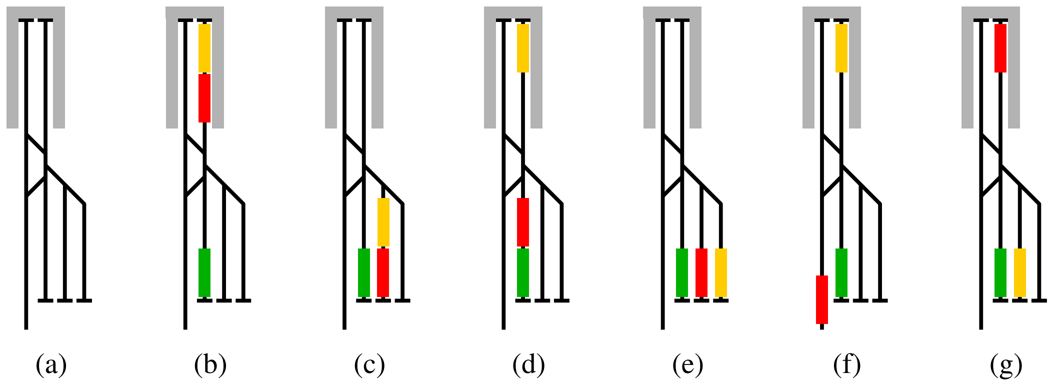

Figure 5.3: Examples of shunting situations at Farum station, equivalent to Station B in Figure 5.2. (a) shows the station layout: To the North there are two platform tracks, to the South there is the main line track (left) and three depot tracks (right). (a) also shows the situation at $t = 5$ for Station B in Figure 5.2. (b) shows the situation at $t = 9$ with two train units parked at the platform, the yellow one to the North of the red one. (c) shows the situation at $t = 10$, in which both train units have been shunted to the middle depot track. Business and train control system rules state that feasible transitions are (b) to (c), (b) to (d) and vice versa. The following transitions (and vice versa) are infeasible: (b) to (e) would require two train shuntings ; (b) to (f) since the moving red train unit is on the main line track; (b) to (g) is infeasible because the red train unit obstructs the movement of the yellow one.

shunted in to the platform on this station will have the relative position South to the original platform train composition in the new train composition. This is the result of the transition between the situations from Figure 5.3d to Figure 5.3b.

In the case of a **coupling** taking place at a **depot track**, the relative position of the platform train composition (which is the one undergoing movement in the operation) will also be the same as the relative position of the place from which the train shunting **started**, in this case the platform track. If, like in the example above, the relative position of a depot track is to the South of a platform track, then the relative position of the platform track is to the North of the depot track. The original platform train composition after coupling will then have the same relative position to the original depot train composition as the platform track has to the depot track, which in this case is North. This is the result of the transition between the situations from Figure 5.3b to Figure 5.3d.

In the case of a **decoupling** taking place at a **platform track**, only the train units at the same relative position to the others in the original train composition as the relative position of the depot track to the platform track may be decoupled to form the new depot train composition to be shunted into the depot. This is equivalent to the relative position of where the train shunting **ended**.

The ordering of train units in a train composition can be found by sorting the individual train units according to their relative positions.

5.5 Path Finding Algorithm

The third component of the integrated rolling stock planning model is the path finding algorithm, used to find new candidate train unit trajectories. The path finding algorithm is operating on the space-time graph described in Section 5.3 to find a path between a start vertex and a finish vertex. In the context of the integrated rolling stock planning model, the goal of the algorithm is to find the path through the space-time graph between these two vertices, having the largest additional net value (i. e. for which it is most advantageous to add a new train unit trajectory).

For convenience, the weights on the arcs in the space-time graph are set as the **negated** additional net value. This makes the algorithm work as a shortest path algorithm. The path finding algorithm concept is thus that of a *single-source shortest path for a directed, acyclic graph* [40], with resource and side constraints. It is implemented as a label setting algorithm and is using dominance to keep the set of potential paths to a minimum [66]. The algorithm traverses the space-time graph and creates potential paths through the graph by setting and processing labels on the vertices it traverses. Upon termination, the shortest path may be found by backtracking the processed labels.

The path finding algorithm consist of two parts: The outer part of the algorithm which is traversing the space-time graph (described in Section 5.5.1, shown as Algorithm 1) and the inner part which is processing the labels (described in Section 5.5.2 shown as Algorithm 2). The inner part is called in each iteration of the outer part.

When finding a new path, that is, a new train unit trajectory, the algorithm must ensure that the found train unit trajectory is feasible. This is ensured in three ways:

1. **Resource constraints:** The total resource consumption of the potential paths is checked in each iteration so that no resources are exhausted (meaning that the service distance limit of the individual train unit is never exceeded). This check is handled in the inner, label processing part, Algorithm 2;

2. **Side constraints:** The check for train unit order feasibility in decoupling operations is built into the path finding algorithm itself. So is the check for flexible space distribution for bicycles etc. These two checks constitute the side constraints in the algorithm. Both these checks are handled in the inner, label processing part, Algorithm 2;

3. **Space-time graph constraints:** The handling of all other practical, railway-specific requirements is relayed to the space-time graph as previously described in Section 5.3. The relaying is performed when traversing the graph in the outer part, Algorithm 1 (see Line 3).

In the path finding algorithm, each vertex from the space-time graph has associated to it a number of labels. The labels are used to mark potential paths through the space-time graph as the algorithm progresses. Each label refers to one arc on the potential path. Furthermore, each label belongs to a vertex (the to-vertex on the arc to which the label refers), but there may be many labels to the same vertex, since different, potential paths may pass through the vertex and since more arcs may be connected to it. Each label carries with it the following information:

1. **The arc in the space-time graph** from which a part of the potential path represented by the label passes through. This information is used to put together the shortest path when the finish vertex has been reached;

2. **The previous label on the potential path** (the predecessor). This information is used to backtrack the labels to find the shortest path once the finish vertex has been reached;

3. **The resources consumed** in order to reach the vertex of the label, starting at the start vertex. This information is used to ensure no resources are exhausted. It is also used for dominance, i. e., to keep the number of potential paths small;

4. **The ordered train composition** of the arc, i. e., which train units are assigned to the arc and in what individual order. This information is used in the side constraints of the algorithm to check for train unit order feasibility and flexible space distribution for bicycles etc.

The inner workings of the two parts of the path finding algorithm are described in the following Sections 5.5.1 and 5.5.2.

5.5.1 Space-Time Graph Traversing

The first, outer part of the path finding algorithm is the space-time graph traversing part, shown as Algorithm 1.

Algorithm 1 starts its main for loop with the next vertex in topological order (Line 1). For each of the incoming arcs to this vertex (Line 2), a check is performed to see if the incoming arc is feasible (Line 3). This check is relayed to the space-time graph via the arc. If the arc it is not feasible, no processing of labels is occurring, and the algorithm is not proceeding further along that potential path. If the space-time graph responds that the arc is feasible, a loop over each of the labels of the from vertex of the incoming arc is started (Line 4).

Inside this loop, the inner, label processing part, is called, see Algorithm 2. Labels are only created if the side constraints are not violated. If not violated, the new label is added to the current vertex, otherwise no label is added.

Algorithm 1: Resource constrained shortest path label setting algorithm with side constraints for a space-time graph with arc resource consumption data and vertices sorted in topological order. The resource and side constraints are checked in Algorithm 2.

Input: The from vertex and the to vertex in the graph.

Output: The resource constrained shortest path between the given vertices, feasible with regard to all practical, railway oriented requirements.

```
1  foreach (toVertex in vertices) do
2      foreach (thisArc in toVertex.getIncomingArcs()) do
3          if (thisArc.isFeasible()) then
4              foreach (fromVertexLabel in thisArc.getFromVertex().getLabels()) do
5                  toVertex.add(getLabel(thisArc, fromVertexLabel));          /* See alg. 2 */
6              end
7          end
8      end
9      toVertex.getLabels().removeDominated() ;
10     if (toVertex.isFinishVertex() && toVertex.isReached()) then
11         processShortestPath(toVertex);                 /* Backtrack to set shortest path */
12         return shortestPathArcs;
13     end
14 end
15 throw NoPathFoundException;
```

Algorithm 1 then continues with the next incoming arc of the current to vertex. When all incoming arcs have been processed, the labels of the to vertex that are dominated are removed (Line 9).

If the current vertex is equal to the finish vertex, and this vertex has been reached, the shortest path is found by backtracking the labels starting with the label at the finish vertex having the least resource consumption for the path finding resource (in this case the negated additional net value).

5.5.2 Label Processing

The second, inner part of the path finding algorithm is the label processing part, shown as Algorithm 2.

The first part of Algorithm 2 relates to finding the ordered train composition of the extension of the current potential path. This part follows the hereby proposed principle of *(train) unit order flow conservation*, in which not only the inflow of train units to a vertex is conserved in the outflow (like in arc based multi-commodity flow models, as mentioned in Section 5.1.2), the train unit order is also conserved.

As such, line 1 calculates the train composition at the from vertex by coupling the train composition already assigned to the incoming arcs in the graph with the train composition of the predecessor label. This coupling is conducted using the topological information, the transition positions, as described in Section 5.3.2. As such the inflow order of train units to the from vertex is calculated for this particular potential path.

Next, line 2 sets a variable for the further reference to the train composition on the current arc consisting of train units already assigned to the arc in the space-time graph.

Line 3 then uses the order found in the inflow to the vertex in question to add the candidate

Algorithm 2: The label processing part of the path finding algorithm. This part of the algorithm makes sure that no resources are exhausted and that the train unit order and flexible space distribution for bicycles etc. is feasible.

Input: The current arc; The current label of the from vertex of the current arc; The candidate train composition (i. e., information on which train unit(s) the algorithm is currently finding a new trajectory for); The resources available for the path.

Output: The algorithm creates a new label if constraints are not violated, no label is created if they are.

```
1  thisFromVertexTrainComposition ← getTrainCompositionAt(thisFromVertex,
     fromVertexLabel);
2  thisArcOldTrainComposition ← thisArc.getTrainComposition();
3  thisArcNewTrainComposition ←
     thisFromEventTrainComposition.getPreserveOrderSumOf(
     candidateTrainComposition, thisArcOldTrainComposition);
4  if (thisArc instanceof TransitionPositioner) then
5      transitionPositioner ← (TransitionPositioner) thisArc;
6      if (!thisFromEventTrainComposition.canDecouple(thisArcNewTrainComposition,
         transitionPositioner.getDecouplingPosition()) ||
         !thisArcNewTrainComposition.hasLegalFlexibleSpaceDistribution()) then
7          return null;
8      end
9  end
10 thisConsumedResources ← resourcesPool.get();
11 thisConsumedResources.setSum(thisArc.getConsumption(),
     fromVertexLabel.getConsumedResources());
12 if (thisConsumedResources.exhaust(availableResources)) then
13     return null;
14 end
15 thisLabel ← labelPool.get();
16 thisLabel.set(thisArc, fromVertexLabel, thisConsumedResources,
     thisArcNewTrainComposition);
17 return thisLabel;
```

train composition to the train composition already assigned to the current arc in the space-time graph. As such, the outflow from the vertex in question is determined with the correct order on the current arc.

The algorithm then proceeds to reject cases where the inflow order and the outflow order are not compatible: Line 4 tests if the current arc is of a type where transitions occur. If so the current arc is type cast (line 5) to be able to query it for the decoupling transition position (in line 6). If decoupling cannot take place while preserving the inflow order, or, if the flexible space distribution of the ensuing train composition is not feasible, the algorithm terminates by not returning a new label (line 7).

Next (in line 10) a new resources object is retrieved from the pool [111]. This resources object is set to the consumed resources being the sum of the consumption of the resources on the incoming arc and the previously consumed resources of the current from label.

If the consumed resources exhaust the available ones, the algorithm also terminates without returning a label. If not, a label is retrieved from the label pool, set with relevant data and returned.

Note that the logic to determine feasibility of the order of the train units in the composition applies to decoupling only. This is because it is only in the process of decoupling that a train unit may obstruct the movements of another. A coupling process will always conserve the relative position of the individual train units. A feasible decoupling will also conserve the relative position of the train units involved, however, an infeasible coupling, if it could occur, would not.

5.6 Heuristic Framework

The fourth and last component of the integrated rolling stock planning model is the heuristic framework used to accept or reject candidate trajectories found with the path finding algorithm described in Section 5.5. The heuristic framework used is that of hill climbing [76].

The overall concept of the heuristic framework in the integrated rolling stock planning model is to remove a number of train unit trajectories from an existing rolling stock plan and then, one by one, to create a new trajectory and insert it into the plan.

By generating and inserting new train unit trajectories into the plan one at a time, it is assured that each new train unit trajectory is feasible in conjunction with the existing ones in the plan.

The objective function of the heuristic is described in Section 5.6.1, the inner workings of the hill climbing heuristic itself is described in Section 5.6.2, and Section 5.6.3 describes the flow of changes to the objective value in one iteration of the heuristic.

5.6.1 Objective Function

The objective function of the proposed heuristic is the *net value* of a rolling stock plan. The net value is defined as the benefit a rolling stock plan provides minus the costs for providing it and the penalties awarded for undesirable features. As such, the benefit and the costs plus penalties are competing terms in the objective function and a rolling stock plan may be improved by maximising the benefits and/or minimising the costs plus penalties.

The objective value may be calculated for the entire rolling stock plan, however, for performance reasons the net value of removing individual trajectories (the subtractional net value) and adding new ones (the additional net value) is used, since this involves fewer calculations.

The *iteration net value increase* z_Δ is thus calculated as the sum over all selected train units U^* of the subtractional net value $z^-(j_u^-)$ of each train unit trajectory j_u^- removed plus the sum over all selected train units U^* of the additional net value $z^+(j_u^+)$ of each train unit trajectory j_u^+ inserted (5.1). Note that the subtractional net value is calculated before removing each original train unit trajectory and the additional net value is calculated before inserting each new train unit trajectory.

$$z_\Delta = \sum_{u \in U^*} z^-(j_u^-) + \sum_{u \in U^*} z^+(j_u^+) \tag{5.1}$$

The calculation of z_Δ described above yields the same result as the difference in total net value for the entire rolling stock plan before and after an iteration, but involves fewer calculations.

The subtractional net value is calculated in (5.2) as the costs saved $c^-(a,u)$ minus the benefits lost $b^-(a,u)$ minus the penalties awarded $p^+(a,u)$ plus the penalties saved $p^-(a,u)$ as a result of subtracting train unit u from arc a .

$$z^-(j_u^-) = \sum_{a \in A_j} c^-(a,u) - b^-(a,u) - p^+(a,u) + p^-(a,u) \tag{5.2}$$

The additional net value is calculated in (5.3) as the benefits gained $b^+(a,u)$ minus the costs added $c^+(a,u)$ minus the penalties awarded $p^+(a,u)$ plus the penalties saved $p^-(a,u)$ as a result of adding train unit u to arc a .

$$z^+(j_u^+) = \sum_{a \in A_j} b^+(a,u) - c^+(a,u) - p^+(a,u) + p^-(a,u) \tag{5.3}$$

Benefits represent the fulfilment of an unfulfilled seat demand for each individual arc and depend upon seat demand by time interval $p \in P$ and train unit type $i \in I$. Benefits are gained if train units are added to arcs that represent revenue train services demanding additional seats. Benefits are lost if train units are subtracted from the same arcs, if this results in more unfulfilled seat demand.

Costs are incurred for energy, maintenance, infrastructure usage and personnel (train drivers and depot drivers) and depend upon service distance, arc type (revenue train services, train shunting operations etc.), time interval $p \in P$ and train unit type $i \in I$. Costs for energy and maintenance are added if train units are added to arcs that represent movements. Energy and maintenance costs are saved if train units are subtracted from the same arcs. For the case of infrastructure use and personnel, costs are only added for the first train unit added to the movement arc in question. These costs are also only saved when the train unit being subtracted is the last train unit assigned to the movement arc in question.

Penalties are awarded for unwanted features of the plan and depend upon time interval $p \in P$ and occurrence. A penalty for not having a revenue train service covered is awarded when the last train unit assigned to the arc in question is subtracted. The penalty is saved when the first train unit is added to the same arc again. A penalty is also awarded for unwanted shunting operations. Here the concept is the opposite: This penalty is awarded when the first train unit is added to the train shunting operation arc, and saved when the last one is subtracted.

Note that penalties can be both awarded and saved by both the subtractional as well as the additional operation, this is the reason for both $p^+(a,u)$ and $p^-(a,u)$ being present in both (5.2) and (5.3).

5.6.2 Hill Climbing Heuristic

The concept of removing one or more train unit trajectories from the plan and reinserting them one at a time, operates within a hill climbing heuristic framework in which all of the inserted new trajectories are either accepted (and kept in the plan) or rejected (and removed from the plan and old trajectories re-inserted, yielding a plan identical to before any modifications were performed).

Normally in heuristics, changes are evaluated **before** being inserted into the solution. However the modification scheme described here is necessary to ensure feasibility: One can not generate a second candidate trajectory without inserting the first one into the plan, because otherwise the plan cannot determine if the second candidate trajectory conflicts with the first one.

The selection of train unit trajectories to remove from the plan is conducted at random.

The heuristic component continues until a given stopping criterion is met. For the calculations given in Table 5.5 on page 88 this was given as a negative value of z_Δ for the past 5 minutes, equivalent to the convergence curve (see Figure 5.6 on page 91) flattening out.

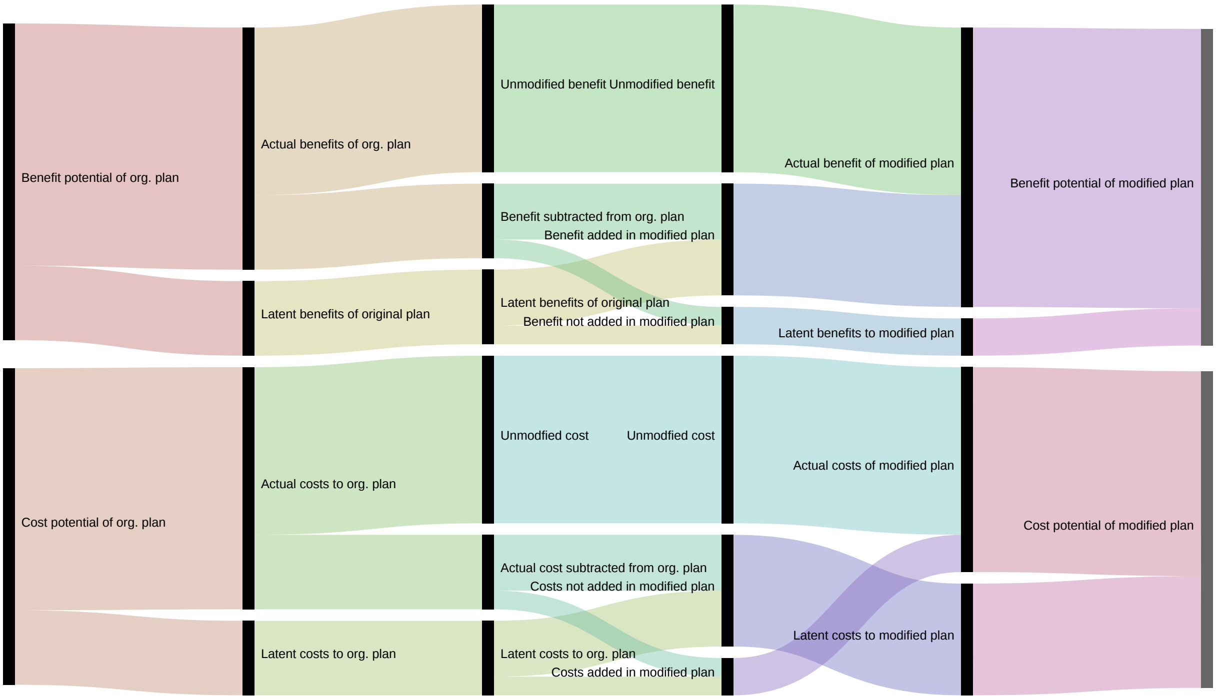

Figure 5.4: Sankey diagram of the flows for benefits and costs (including penalties) in the proposed heuristic.

5.6.3 Objective Value Flow

Figure 5.4 shows a Sankey diagram of the flows for benefits and costs (including penalties) in the proposed heuristic. The figure is read from left to right showing the flow of costs (bottom) and benefits (top) through one iteration of the heuristic. The term *actual benefit* is used to describe the benefit that is in the current plan. The term *latent benefit* is used to denote a benefit that is not in the current plan, but one that could be in the next iteration through the modifications of the heuristic. The *total benefit potential* is thus the sum of these two types of benefit. On the cost side the same terms apply. The general idea of the heuristic is to maximise the benefits by turning latent benefits into actual ones and to minimise the costs (including penalties) by turning actual costs into latent ones, thus maximising the total net value of the plan.

5.7 Experimental Results

The integrated rolling stock planning model proposed in this paper has been tested on a number of data instances. The purpose of the experiments has been to benchmark the performance of the heuristic with plans produced manually. The conditions were chosen so as to make the bench marking on as equal conditions as possible.

The integrated rolling stock planning model presented here has been implemented in the programming language Java 1.8 with approx. 15,000 lines of code. Apart from the library Joda-Time 2.8.2, only standard libraries have been used.

The tests were performed on a Dell PowerEdge T610 equipped with 16 Intel Xeon E5620 CPUs at 2.40 GHz and 16 GB RAM running Ubuntu Linux 14.04 LTS. Parallel processing was used in which each individual test was run in its own thread on one CPU, parallel to other tests.

5.7.1 Data Instances

The proposed heuristic has been tested on 15 different rolling stock plan data instances as shown in Table 5.5. All the data instances represent long-term circulation plans (as opposed to short-term train unit dispatching plans). Each individual data instance represents a particular date, e. g., 2012-10-19 and a particular weekday, e. g., Friday.

Most of the data in the instances are actual, real-world data. This includes infrastructure data, timetable data, passenger demand data and data on personnel on duty. How the individual parts of the real world data vary between instances is described in the following.

The same timetable is in effect from Monday to Friday, but a different one is used on Saturdays, and again a different one on Sundays. Night train services operate on mornings after Fridays and Saturdays. The timetable also differs between the years 2012 and 2014.

The depot driver duties differ by weekday. The reason for this is that the start up procedures on Monday morning are different from the ones on Tuesday, since there is a change of timetable between Sunday and Monday, but not between Monday and Tuesday.

In the data instances, the passenger demand is represented by running the DSB S-tog passenger prognosis model with the actual, measured passenger data for those days. This is possible because the data instances are in the past. In a real planning situation prognosis passenger demand would be used.

The instance of 2012-10-19 is special since it occurs during the autumn holiday and also represents an extraordinary plan with infrastructure maintenance works on a parallel, long distance railway line. For this reason this plan provides extra capacity on the one of the train service lines. (It has turned out that, in hindsight, this plan provides far too much seating capacity, which can be seen by how much the heuristic may increase its net value by removing this excess capacity again.) The other dates represent normal plans with no extraordinary features.

The train unit trajectories in the data instances are those from complete rolling stock plans produced manually by the planners. In the experiments, the infeasible train unit trajectories (if any) are removed prior to running the heuristic. The start and finish points in space and time of the original train unit trajectories are kept. As such, new trajectories constructed using the heuristic have the same origin and destination stations. This preserves the depot balance between the original and rolling stock plan when using the heuristic.

Since the current rolling stock planning procedures at DSB S-tog still involve some degree of manual work, data is not available for all aspects of manually produced rolling stock plans. Parts of the data instances not currently available include data on the train shuntings of the train units and the grouping of anonymous train units so as to determine the number of train units in the plan.

Firstly, the only information currently available on the movement of train units is the assignment of train units to train services. As such, the provided train unit trajectories have gaps in them. In order for the integrated rolling stock planning model to work, this information has to be generated artificially by *retrofitting* each manually produced plan to the data model described in Sections 5.3 and 5.4.

In some cases, all train unit trajectories can be retrofitted. However, as shown in Table 5.5 on page 88 in the column *# of infeasible trajectories*, in most cases, the retrofitting process is not able to retrofit all gaps in the supplied train unit trajectories. This is because the manually produced plans may not respect all of the practical, railway-specific requirements and thus can not be mapped onto the data model. (Recall that the data model is constructed so as not to allow violations of the requirements.) The infeasibilities of the manually produced plans may relate to depot track capacities, depot driver duties, depot driver time consumption and minimum turn-around times between consecutive train services. (See Appendix B.1 for more information about the retrofitting process.)

Some of these infeasibilities arise in the automated, step-by-step planing system currently in use at DSB S-tog. However, it is also common for the planners to use a variety of "dirty tricks" to improve the economic attractiveness of a rolling stock plan manually, or to replace intolerable infeasibilities from the automated planning system with "less intolerable" infeasibilities. Strictly speaking, these tricks are violations of the railway-specific requirements, but since they often save substantial costs they are accepted. The infeasibilities introduced by the planners are often used as a last resort and incorporate all the tacit knowledge of the planners. Therefore, it may indeed be very hard for any automated system to compete with plans having these mentioned infeasibilities in them.

Secondly, realistic data is also constructed for the grouping of anonymous train units in the plan. In the circulation planning phase of rolling stock planning, train units are anonymous since it may not be known which actual, physical train units may be available when the plan is to be commenced (some train units may e. g., be in unscheduled maintenance).

In the data instances used here, there is currently no information regarding whether two anonymous train units with non-overlapping train unit trajectories are actually intended to be supplied with rolling stock by assigning the same physical train unit in the train unit dispatching phase to both of them. This information is therefore generated artificially by coalescing train unit trajectories that do not overlap and that may be performed in sequence by connecting them at the intermediate depot.

This feature may make it harder for the integrated rolling stock planning model to find good solutions since it may have less train units (and thereby fewer degrees of freedom) to do so than the manual planners have. On the other hand, if the coalescing would not take place, there would be more anonymous train units than would fit in the actual depots, which would also mean missing retrofits and thus infeasibilities in the plan.

In the long-term circulation planning process of DSB S-tog, rolling stock plans are constructed for one week at a time, in which each day connects to the next. In this paper, however, the scope is on each individual day, not consecutive days. For this reason it does not make sense to restrict the service distance each individual train unit may travel before it has to undergo scheduled maintenance or refilling of consumables, since these events occur at intervals far greater than that of a day. If arbitrary values for the service distance limit would be included, the bench marking results would not be comparable.

This feature makes it slightly easier for the integrated rolling stock planning model to find good solutions since it can make an individual train unit roll a bit longer than it may roll due to e. g., scheduled maintenance. Experiments have been conducted to show that the model also performs well with the service distance limits in place.

All of the data instances used for the experiments represent the circulation planning phase of the rolling stock planning process. For this reason the remaining railway-specific requirements related to the short-term train unit dispatching phase are disregarded in the experiments.

These issues explained, the data used in the experiments represent a very close approximation to the real-life planning conditions.

Using the mentioned data instances, a typical space-time graph for a weekday has approx. 22,000 arcs and 13,000 vertices and approx. 16,000 arcs and 9,000 vertices for a Saturday or Sunday.

5.7.2 Convergence Characteristics of the Proposed Heuristic

The main driver of the heuristic is the objective function, the net value. The net value is calculated as the benefits of providing seats to passengers that demand them minus the costs of doing

Table 5.5: Overview of data instances and experimental results for the improvement of rolling stock plans using the integrated heuristic model with $k = 3$. The results given are based on $n = 10$ individual test runs of each data instance. *Mean net value gain* is the mean of the relative difference of the net value between the original plan (with infeasibilities in it, if any) and the modified one. Stopping criterion of the individual test run was feasibility and no net value gain obtained in the last 5 minutes of the calculation.

Instance meta data		Train units			Original characteristics						Modified characteristics						Performance characteristics						
Date	Weekday	# of type 1	# of type $\frac{1}{2}$	Total	Costs [kDKK]	Penalties [kDKK]	Benefits [kDKK]	Net value [kDKK]	# of infeasible trajectories	# of uncov. revenue train service arcs	Mean costs [kDKK]	Mean penalties [kDKK]	Mean benefits [kDKK]	Mean net value [kDKK]	Max # of infeas. trajectories	Max # of uncov. revenue train srv. arcs	Min net value gain [%]	Mean net value gain [%]	Max net value gain [%]	Min processing time [h:min]	Mean processing time [h:min]	Max processing time [h:min]	Max proc. time to feasibility [h:min]
2012-10-19	Fri	89	22	111	1,571	26	3,772	2,175	0	0	1,412	5	3,765	2,348	0	0	7.8	8.0	8.2	0:29	0:41	0:59	0:00
2014-03-31	Mon	90	27	117	1,538	28	4,445	2,879	1	0	1,494	9	4,424	2,922	0	0	1.2	1.5	1.6	0:21	0:41	1:00	0:00
2014-04-01	Tue	90	27	117	1,537	27	4,445	2,881	3	0	1,490	9	4,416	2,917	0	0	1.0	1.3	1.9	0:23	0:38	1:15	0:00
2014-04-02	Wed	90	27	117	1,537	31	4,445	2,877	3	0	1,489	9	4,417	2,919	0	0	0.8	1.5	1.7	0:24	0:44	1:15	0:02
2014-04-03	Thu	90	27	117	1,537	28	4,445	2,879	2	0	1,490	9	4,423	2,924	0	0	0.9	1.6	1.9	0:19	0:44	1:05	0:00
2014-04-04	Fri	90	28	118	1,544	30	4,492	2,917	3	0	1,486	10	4,471	2,975	0	0	1.3	2.0	2.4	0:31	0:40	0:49	0:04
2014-04-05	Sat	53	7	60	984	21	2,780	1,775	2	0	982	5	2,779	1,792	0	0	0.8	0.9	1.0	0:10	0:14	0:17	0:03
2014-04-06	Sun	52	11	63	980	19	2,166	1,167	0	0	955	2	2,162	1,205	0	0	3.1	3.3	3.7	0:11	0:17	0:32	0:00
2014-04-07	Mon	90	27	117	1,538	27	4,445	2,880	1	0	1,498	9	4,430	2,922	0	0	1.3	1.5	1.9	0:21	0:36	0:51	0:00
2014-04-08	Tue	90	27	117	1,538	27	4,445	2,880	3	0	1,490	8	4,416	2,918	0	0	0.8	1.3	1.7	0:25	0:48	1:15	0:06
2014-04-09	Wed	90	27	117	1,537	31	4,445	2,877	3	0	1,488	9	4,416	2,919	0	0	1.2	1.5	1.9	0:16	0:39	1:19	0:06
2014-04-10	Thu	90	27	117	1,538	27	4,445	2,880	1	0	1,490	9	4,421	2,922	0	0	1.2	1.4	2.0	0:22	0:41	1:13	0:00
2014-04-11	Fri	90	28	118	1,545	30	4,492	2,917	4	0	1,482	10	4,471	2,980	0	0	1.9	2.1	2.4	0:19	0:38	0:56	0:00
2014-04-12	Sat	53	7	60	983	21	2,780	1,775	2	0	983	5	2,780	1,792	0	0	0.9	1.0	1.0	0:09	0:12	0:16	0:00
2014-04-13	Sun	52	7	59	977	19	2,166	1,169	0	0	968	2	2,166	1,196	0	0	2.2	2.3	2.5	0:09	0:13	0:18	0:00

so and minus penalties for unwanted features of the plan.

The penalties are estimated as real monetary values. A very high specific penalty has been set for those aspects definitely unwanted, e. g., uncovered train services. A moderate specific penalty has been set for those aspects that are just undesirable, e. g., train shuntings across the main line tracks. Simple tests have shown that these estimates make the model perform well.

Experiments have justified the current settings for those parameters that are in the objective function. Another parameter that governs the behaviour of the heuristic is the number of train unit trajectories to pick and remove, k.

More than 2,000 test runs have been performed to analyse the behaviour of the model when varying model parameters. Some results are shown in Figures 5.5 and 5.6. The observations from the test runs are given in the following and characterised in relation to *initial gain* (how well the model converges early in the process - more is better), *maximum gain* (how much gain does the model achieve before the convergence curve flattens out - more is better), and *variance* (how results from different test runs with the same parameters vary - less is better).

Low values of k yield high initial gain, little variance, but only moderate maximum gain. Higher values of k yield low initial gain, higher variance but better maximum gain. However, beyond a threshold value of k, the performance seems to deteriorate generally.

We believe this behaviour is related to two characteristics of the heuristic: Firstly, if $k = 1$, no swapping of resources between train units can occur. From $k = 2$ and onward, swapping can occur, but for higher values of k, the greediness of the algorithm takes over, in the sense that the formerly inserted trajectories use the resources at the expense of the latter trajectories inserted, this yielding a lower net value gain in total.

Based on the observations from the test runs, the best performance of the proposed heuristic is achieved by selecting a moderate value for k, e. g., $k = 3$.

5.7.3 Obtained Results

As seen in Table 5.5, the experiments show that the heuristic used in the integrated rolling stock planning model is able to make all instances of typically infeasible rolling stock plans feasible. In most cases, feasibility is reached with a processing time less than 1 minute. In the worst case of the 150 test runs reported in Table 5.5, feasibility is reached within 7 minutes of processing time.

Furthermore, the heuristic is able to improve the economic attractiveness of all tested rolling stock plan instances with an average economic gain of 2%. The highest net value gain is achieved for the instance 2012-10-19. As mentioned in Section 5.7.1, this special instance represents an extraordinary plan providing (as it turns out) excessive, extra capacity on the one of the train service lines.

With a stopping criterion of no gain in 5 minutes the mean processing time is less than an hour for all instances. In no case is the processing time longer than 1 hour 20 minutes.

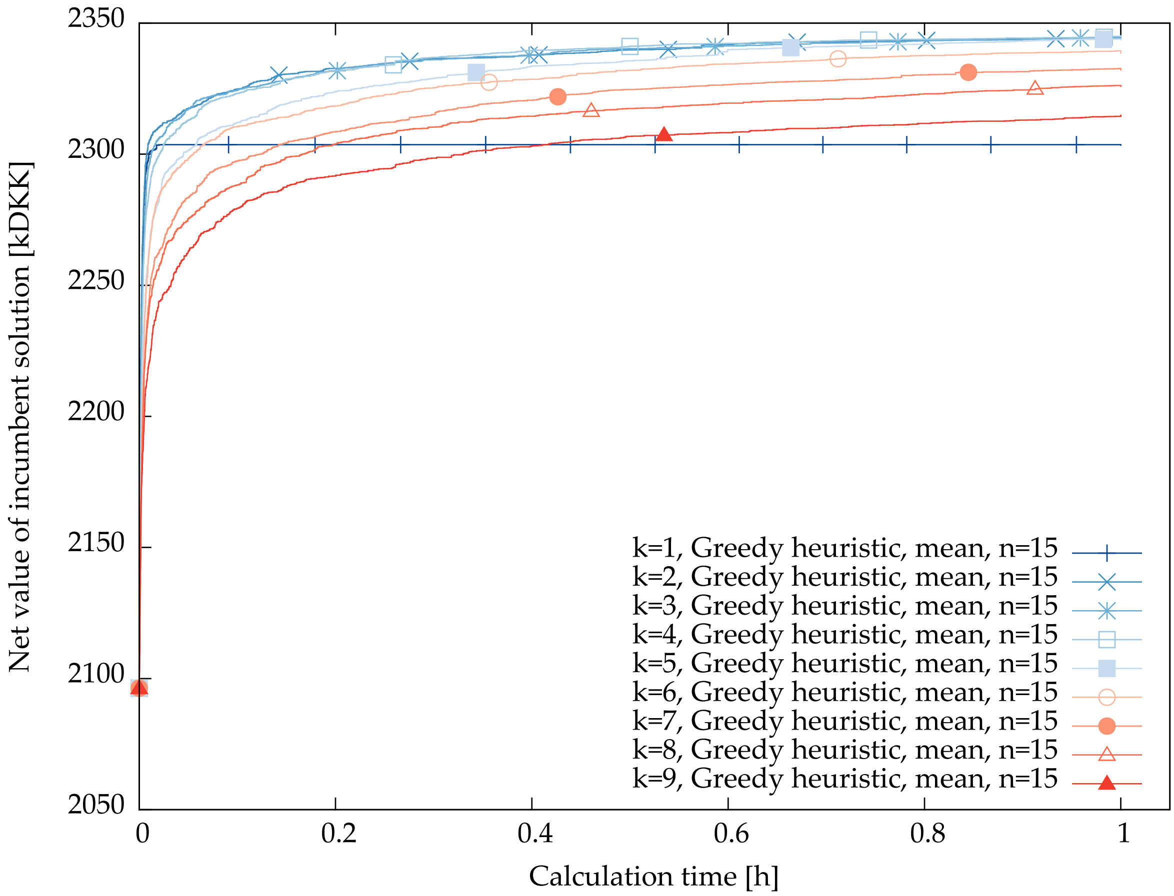

Figure 5.5: Convergence diagram for different values of k, i. e., the number of trajectories to remove and re-insert in one iteration. As may be seen, the best results are achieved with $k = 3$. Each line represents the mean of $n = 24$ test runs. Data is for 2012-10-19.

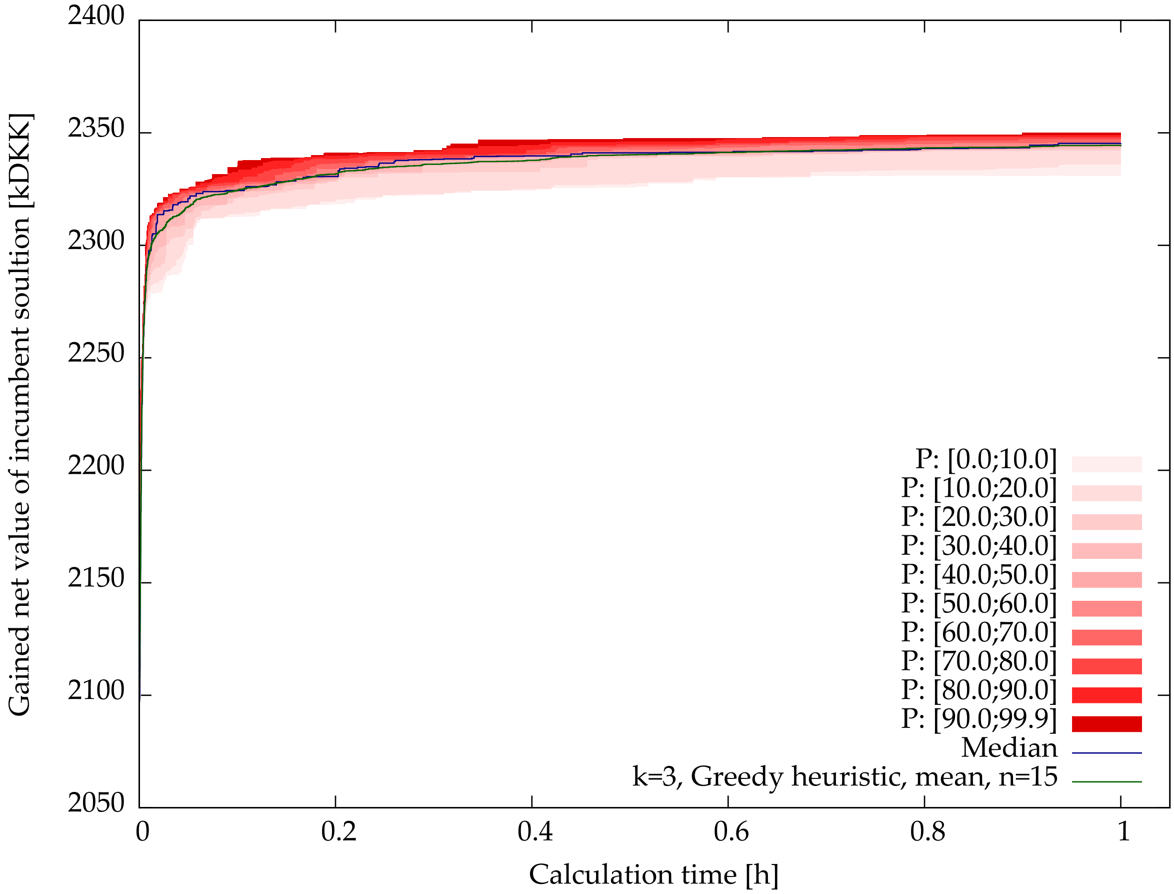

Figure 5.6: A convergence diagram showing the mean value and the median of the gain for 15 test runs with the given parameter settings. Also shown is the cumulative probability *P* in percent of getting a solution with a value less than a given gained net value. Data is for 2012-10-19.

5.8 Discussion

In implementing the proposed, integrated model, we have demonstrated that it is possible to integrate into one model processes in rolling stock planning that are normally solved in a step by step manner. We have also shown that the proposed *ordered flow conservation* principle works in practice. The work presented here is a case study on the specific real-world data and constraints of DSB S-tog, however, the proposed principles may be applied in general to other types of problems.

Furthermore, we have observed that the proposed heuristic operating on the integrated rolling stock planning model has good convergence characteristics in that it will make typical infeasible plans feasible within minutes of computation time and with an additional gain in economic value of 2% on average.

These features make the integrated rolling stock planning model highly suited to simplify and improve present semi-automatic or manual rolling stock planning procedures at DSB S-tog.

The integrated rolling stock planning model proposed in this paper uses a shortest path algorithm to find new candidate train unit trajectories. This fact makes the algorithm highly greedy, a feature we believe makes the algorithm very suitable for fast "trimming and grooming" of an existing rolling stock plan with excess seating capacity in it. However, the same greediness property may make it less suited for constructing a rolling stock plan from scratch, since the former found shortest train unit trajectories will be good at the cost of latter found ones. Further research may devise methods to overcome this limitation.

Extensive experiments have been conducted to replace the heuristic proposed here with a metaheuristic using Simulated Annealing [3] with an exponential cooling scheme and reheating. However, none of these experiments produced better convergence. Future research into other metaheuristic frameworks may be conducted to improve convergence.

The model proposed is primarily intended for the tactical planning scope. However, with few modifications, the model may also be used for the short-term train unit dispatching phase of rolling stock planning. Schemes for reinstating cancelled train services are currently not handled, nor is the objective in a disruption recovery setting to recover with a minimum of changes to the non-disrupted plan. Future research should cover these aspects, if the model is to be used in the train unit dispatching context.

A further integration of train unit routing could be built into the model by also modelling the individual platform tracks instead of modelling them as a whole.

Moreover, a concept for creating relevant non-revenue train services should be considered in the future. The model proposed here only uses the non-revenue train services in the original plan (if appropriate), however these non-revenue train services may not fit the modified plan very well, since they are specifically created to fit the original plan.

Even though the practical, railway-specific requirements that relate to the train unit dispatching phase have been implemented with the proposed, integrated rolling stock planning model, experiments with train unit dispatching plans have not been made yet. One major difference to the circulation plans used in the experiments is that the individual positions of the train units is given. This is because the individual, physical train units have a specific position at the time the plan is to be set in motion. In the long-term circulation planning phase, this position can be chosen. This difference somewhat limits the amount of candidate solutions for train unit dispatching.

Lastly, and perhaps most importantly, research should be conducted into the field of improving convergence by devising different schemes from which to choose train unit trajectories to remove as well as different schemes for inserting them into the plan.

Chapter 6

Net Value Upper Bound Calculation Models for Rolling Stock Planning

6.1 Introduction

As shown in the previous Chapter 5, it is possible to solve the rolling stock planning problem using the proposed greedy heuristic integrated rolling stock planning model. However, because of its heuristic nature, the algorithm neither guarantees the provision of an optimal solution, nor does it provide bounds on the deviation from the optimal solution. In order to provide these bounds, three different upper bound calculation models are formulated in this chapter.

From a practical point of view, an upper bound calculation model can be used to provide a quick approximation of the economic properties of a given scenario, e. g., for comparing the rolling stock costs and benefits of different, proposed timetables. For this purpose, an upper bound calculation model must be simple (or LP-relaxed) in order to have short processing times. An upper bound calculation model can also be used to quantify how close to the global optimum any given solution may lie. For this purpose, a more advanced upper bound calculation model is needed, so as to have a tighter bound. Naturally, this comes at the price of longer processing times.

The upper bound calculation models are designated A2, A4 and B10, referring to types A and B, and to how many constraint types they implement. Type A upper bound calculation models assign train composition types to arcs, whereas the one type B model assigns train unit types to arcs.

The purpose of formulating three different upper bound calculation models is to investigate the entire spectre of possible properties with regard to simplicity of formulation, practical implementation effort, processing time and tightness of bounds.

All upper bound calculation models are mixed integer linear programs. The models may be solved in both their mixed integer version (as MIPs) as well as in their linearly relaxed versions (as LPs).

The mathematical formulations of the three models are given in Section 6.2. Results from the numerical experiments are shown in Table 6.5, this includes a comparison to the heuristic results from Chapter 5. In Section 6.4 the upper bound calculation models are compared to each other and the perspectives for their use are discussed.

On a general note, the upper bound calculation models are applied to the whole rolling stock plan, i. e., to the full set of train units U, as opposed to the greedy heuristic described in Chapter 5, which is only applied to a subset U^* of the train unit trajectories in the plan at a time.

6.2 Mathematical Formulations

The upper bound calculation models consist of sets (with corresponding indices), parameters and decision variables as defined in Tables 6.1 to 6.4.

Since the benefits are all accounted for in each of the upper bound calculation models, and since only a subset of the costs and penalties are accounted for in each, the solution to any of the upper bound models for a given instance represents an upper bound on the optimal solution to the integrated real rolling stock planning problem.

6.2.1 Upper Bound Calculation Model A2

The first upper bound calculation model A2 is designed to be simple and fast and as such only to implement a few of the railway-specific requirements described in Chapter 3.

The objective of the model is to assign train composition types C to revenue train service arcs A_R, so that the total net value is maximised (6.1). Recall that train composition types are unordered combinations of train units by specific type, as shown in Table 3.5 on page 41. The value of the objective function is denoted z_{sup} since it is an upper bound (supremum) of the total net value of the rolling stock plan z taking all requirements into account. $x(c, a)$ denotes the binary decision variable for assigning composition type c to revenue train service arc a, and $b(c, a)$ the benefit, $c(c, a)$ the costs and $p_1(c, a)$ the penalties of doing so.

A decision variable $x(c, a)$ for a particular revenue train service arc is only created if it is at all feasible to assign the composition type to the arc in question based on train composition length and number of train units in the composition.

The assignment is subject to assigning exactly one non-empty composition type to each revenue train service arc (6.2), and not exceeding the number of available train units $n(i)$ by train unit type i in any time interval p (6.3), with I being the set of train unit types. $e(a, p)$ is a binary parameter assuming the value of 1 if arc a coincides with time interval p, and 0 otherwise, $n(i, c)$ is the number of train units of train unit type i in train composition type c. Equation (6.4) is the integrality constraint for the decision variable $x(c, a)$.

$$\max z_{sup} = \sum_{c \in C_1} \sum_{a \in A_R} \Big(b(c, a) - c(c, a) - p_1(c, a) \Big) \cdot x(c, a) \tag{6.1}$$

$$\sum_{c \in C_1} x(c, a) = 1 \qquad \forall a \in A_R \tag{6.2}$$

$$\sum_{c \in C_1} \sum_{a \in A_R} e(a, p) \cdot n(i, c) \cdot x(c, a) \leq n(i) \qquad \forall i \in I, \forall p \in P \tag{6.3}$$

$$x(c, a) \in \{0, 1\} \qquad \forall c \in C_1, \forall a \in A_R \tag{6.4}$$

Since all revenue train services will be covered by assigning non-empty train compositions to them in (6.2), no penalties for uncovered revenue train services need appear in the model formulation.

This upper bound calculation model is essentially a knapsack problem [8, Chapter 3] with added minimum assignment constraints (6.2), since it describes a number of items (in this case train units in the form of train composition types) to select (in this case to assign to revenue train services), this in order to maximise a given goal (in this case the net value), subject to maximum and minimum limits of the number of items (train units) that can be assigned.

The upper bound calculation model is neither an arc based nor a path based multi-commodity flow model. For this reason it neither takes into account the depot and side track capacities, nor

Table 6.1: Sets in the upper bound calculation models, their corresponding indices, domains and definitions, ordered alphabetically by symbol. Sets are in upper case, their indices use same lower case symbol without subscripts or superscripts.

Symbol	Description	Index, domain, definition
A	Arcs in the graph G, each arc going from one vertex to another. An arc corresponds to a time interval $p \in P$	$a \in A$; $a = (v_1, v_2)$ $v_1, v_2 \in V$
A_N	Arcs representing non-revenue train services	$a \in A_N \subset A$
A_P	Arcs representing train units undergoing parking	$a \in A_P \subset A$
A_R	Arcs representing revenue train services	$a \in A_R \subset A$
A_S	Arcs representing train shunting operations	$a \in A_S \subset A$
A_v^+	The set of incoming arcs to vertex v	$a \in A_v^+ \subset A$; $v \in V$
A_v^-	The set of outgoing arcs from vertex v	$a \in A_v^- \subset A$; $v \in V$
A_0^-	The set of all outgoing arcs from any source vertex $v_1 \in V_0$	$a \in A_0 \subset A$; $a = (v_1, v_2)$ $v_1 \in V_0$; $v_2 \in V$
C	Train composition types: The unordered combinations of train unit types coupled in train services or being parked. This also includes the empty train composition having no train units at all	$c \in C$
C_1	Train composition types having at least one train unit. This excludes the empty train composition having no train units at all	$c \in C_1 \subset C$
G	The directed and acyclic graph with vertices V and arcs A	$G = (V, A)$
I	Train unit types	$i \in I = \{\frac{1}{2}, 1\}$
P	All possible time intervals, a time interval being a sorted 2-tuple of point in time	$p \in P$; $p = (t_1, t_2)$ $t_1, t_2 \in T$; $t_1 < t_2$
P_D	Depot driver time intervals, i. e., the time intervals created by ordering and making unique the set of points in time (start and finish) from all train shunting operation arcs	$p \in P_D \subset P$
Q	Points in space (each depot track, each side track, all platform tracks [as a whole] at every station)	$q \in Q$
T	Points in time	$t \in T$
U	Individual train units currently available	$u \in U$
U^*	The set of train units selected for train unit trajectory removal and reinsertion for the heuristic described in Chapter 5	$u \in U^* \subset U$ $\lvert U^* \rvert = k$
V	Vertices in the graph G, each vertex being a *point in space, point in time* tuple	$v \in V$; $v = (q, t)$ $q \in Q$; $t \in T$
V_0	Source or sink vertices (i. e., with zero in-degree or out-degree)	$v \in V_0 \subset V$

Table 6.2: Parameters with symbols b to n in the upper bound calculation models, their domains and definitions, ordered alphabetically by symbol. All parameters have symbols in lower case. See Table 6.3 for parameters with symbols p to z.

Symbol	Description	Domain, definition
$b(a)$	The benefit (i. e., economic value) of providing one seat on arc a, proportional to the length of the time interval p of the arc	$b(a) \in \mathbb{R}_0^+$
$b(c, a)$	The benefit of assigning train composition type c to arc a	$b(c, a) \in \mathbb{R}_0$
$c(a)$	The train composition costs by having assigned a train composition of any length to arc a	$c(a) \in \mathbb{R}_0^+$
$c(c, a)$	The cost of assigning train composition type c to arc a	$c(c, a) \in \mathbb{R}_0$
$c(i, a)$	The train unit movement costs of assigning one train unit of type i to arc a	$c(i, a) \in \mathbb{R}_0^+$
$d(p)$	The number of depot drivers on duty in time interval p	$d(p) \in \mathbb{N}_0$
$e(a, p)$	Indicator parameter equal to 1 if arc a exists in time interval p, and 0 otherwise	$e(a, p) \in \{0, 1\}$
$e(a, v)$	Indicator parameter equal to 1 if arc a exists as using vertex v and 0 otherwise	$e(a, v) \in \{0, 1\}$
$l(a)$	The maximum length of train composition assigned to arc a	$l(a) \in \mathbb{R}^+$
$l(i)$	Length of train unit type i	$l(i) \in \mathbb{R}^+$
$l(u)$	Length of train unit u	$l(u) \in \mathbb{R}^+$
$n(a)$	The maximum number of train units in a train composition for train composition movement operations, i. e., for revenue and non-revenue train services and for train shunting operations	$n(a) = \{2 \mid a \in A_R\}$ $n(a) = \{3 \mid a \in A_N \cup A_S\}$
$n(i, c)$	Number of train units of train unit type i in train composition type c	$n(i, c) \in \mathbb{N}_0$
$n(i)$	Number of available train units of type i	$n(i) \in \mathbb{N}_1$

Table 6.3: Parameters with symbols p to z in the upper bound calculation models, their domains and definitions, ordered alphabetically by symbol. All parameters have symbols in lower case. See Table 6.2 for parameters with symbols b to n.

Symbol	Description	Domain, definition
$p_1(a)$	The penalty awarded for undesired train shunting operations. If train shunting operation arc $a \in A_S$ is in category “undesired” this is a positive real number, otherwise and for all other arcs $p_1(a) = 0$	$p_1(a) \in \mathbb{R}_0^+$
$p_1(c, a)$	The penalty awarded for undesired train shunting operations by assigning composition type c to arc a. If train shunting operation arc $a \in A_S$ is in category “undesired” and composition type c is non-empty, i. e., if $c \in C_1$, this is a positive real number, otherwise and for all other arcs $p_1(c, a) = 0$	$p_1(c, a) \in \mathbb{R}_0^+$
$p_2(a)$	The penalty awarded for uncovered revenue train services. If arc a is a revenue train service arc, i. e., if $a \in A_R$ this is a positive real number, otherwise and for all other arcs $p_2(a) = 0$	$p_2(a) \in \mathbb{R}_0^+$
$p_2(c, a)$	The penalty awarded for uncovered revenue train services by assigning composition type c to arc a. If train composition type c is empty, i. e., if $c \in C \setminus C_1$ and arc a is of type revenue train service, this is a positive real number, otherwise and for all other arcs $p_2(c, a) = 0$	$p_2(c, a) \in \mathbb{R}_0^+$
$s(a)$	The total seat demand of arc a	$s(a) \in \mathbb{N}_0$
$s(i)$	Perceived number of seats for train unit type i	$s(i) \in \{125, 300\}$
$s(u)$	Perceived number of seats for train unit u	$s(u) \in \{125, 300\}$
z	The total net value of a rolling stock plan as calculated by an integrated rolling stock planning model taking into account all railway-specific requirements	$z \in \mathbb{R}$
z_{sup}	The upper bound (supremum) of the net value z. z_{sup} as calculated using one of the upper bound calculation models	$z_{sup} \in \mathbb{R}$ $z_{sup} = \sup z$

Table 6.4: Decision variables of the upper bound calculation models, their domains and definitions, ordered alphabetically by symbol. All variables have lower case symbols.

Symbol	Description	Domain, definition
$f(a)$	Variable assuming the value 1 if arc a has no train units assigned to it, and 0 otherwise, it is the opposite of $g(a)$	$f(a) \in \{0, 1\}$
$g(a)$	Variable assuming the value 0 if arc a has no train units assigned to it, and 1 otherwise	$g(a) \in \{0, 1\}$
$x(c, a)$	Assign composition type c to arc a	$x(c, a) \in \{0, 1\}$
$x(i, a)$	Number of train units of type i to assign to arc a	$x(i, a) \in \mathbb{N}_0$ $x(i, a) \leq n(i)\ \forall a \in A$

the possibility or cost of positioning train units using non-revenue train services, nor the possibility, cost or penalty of train shunting operations, nor scheduled maintenance etc. at given service distance intervals. However, minimum turnaround times for train units between two train services is taken into account in the definition of the time intervals used, in that the minimum turnaround time is added to the finish time of each train service.

6.2.2 Upper Bound Calculation Model A4

The second upper bound calculation model is designed to take into account more railway-specific requirements (as described in Chapter 3) than the simple and fast A2 model described in Section 6.2.1. The model thus has additional flow conservation constraints, essentially making it a arc based multi-commodity flow model [8, Chapter 17].

Equivalent to the A2 model described in Section 6.2.1, $x(c, a)$ denotes the binary decision variable of assigning composition type c to arc a. In order to keep the number of decision variables small, a decision variable is only created if it is at all feasible to assign the composition type to the arc in question. A certain composition type may e. g., not be feasible due to train composition length limits on a particular arc. Moreover, empty train composition type variables $x(c, a), c \in C$ for arcs a are only created for the revenue train service arcs $a \in A_R$ since other types do not need empty train composition types for correct cost and constraint definition.

Equation (6.6) makes sure that every revenue train service arc has exactly one train composition type $c \in C$ assigned to it, empty or non-empty. All other constraints refer to non-empty train composition types $c \in C_1$.

Equation (6.7) makes sure that no more than one train composition type can be assigned to each arc that is not a revenue train service arc. When no non-empty train composition type is assigned to a given arc, this is equivalent to assigning an empty train composition (that is, not covering the arc), however, in this manner this need not be represented by a separate variable. Experiments have proven that in this case, the formulation with fewer variables leads to shorter processing times.

The objective (6.5) resembles that of the previous A2 model, however, since the A4 model can handle the case where no train units are assigned to revenue train service arcs, a penalty $p_2(c, a)$ for uncovered train service arcs must be warded. This penalty is positive for combinations of empty train compositions and revenue train service arcs. The penalty is zero for non-empty train compositions in every other arc combination.

$$\max z_{sup} = \sum_{c \in C} \sum_{a \in A} \Big(b(c, a) - c(c, a) - p_1(c, a) - p_2(c, a) \Big) \cdot x(c, a) \tag{6.5}$$

$$\sum_{c \in C} x(c, a) = 1 \qquad \forall a \in A_R \tag{6.6}$$

$$\sum_{c \in C_1} x(c, a) \leq 1 \qquad \forall a \in A \setminus A_R \tag{6.7}$$

$$\sum_{c \in C_1} \sum_{a \in A_0^-} n(i, c) \cdot x(c, a) \leq n(i) \qquad \forall i \in I \tag{6.8}$$

$$\sum_{c \in C_1} n(c, i) \left(\sum_{a \in A_v^+} x(c, a) - \sum_{a \in A_v^-} x(c, a) \right) = 0 \qquad \forall v \in V \setminus V_0, \forall i \in I \tag{6.9}$$

$$x(c, a) \in \{0, 1\} \qquad \forall c \in C, \forall a \in A \tag{6.10}$$

Equation (6.8) represents the outflow limit constraints over the set of all outgoing arcs from any source vertex A_0^-, making sure the available number of train units $n(i)$ by train unit type $i \in I$ is not exceeded.

Equation (6.9) represents the flow conservation constraints for non-source and non-sink vertices $V \setminus V_0$ and commodities (in our case train unit types I), making sure the flow on all outgoing arcs $a \in A_v^-$ for vertex v equals the flow on all incoming arcs $a \in A_v^+$ to the same vertex. The flow itself is calculated as the product between the decision variable $x(c, a)$ of assigning a particular composition type c to arc a and the number of train units by train unit type in that composition $n(c, i)$.

Equation (6.10) is the integrality constraint for the decision variable $x(c, a)$.

Equations (6.8) and (6.9) make the model equivalent to the general formulation of multi-commodity flows in [8], however, in model A4 a flow less than the total number of train units by type is also possible (6.8). When a flow less than the total number of available train units is occurring, this is equivalent to train units not being handled by the model. This option is included in the A4 model in order for its solutions to be comparable to those of the greedy heuristic described in Chapter 5.

6.2.3 Upper Bound Calculation Model B10

The third upper bound calculation model is designed to take into account as many of the railway requirements as possible while still being a pure mixed integer linear programming model.

Contrary to the previously described upper bound calculation models A2 and A4, the B10 model assigns a number of train units by train unit type to each arc. Like model A4, model B10 is a multi-commodity flow model.

$$\max z_{sup} = \tag{6.11}$$

$$\sum_{a \in A_R} b(a) \cdot s(a) - \sum_{a \in A_R} b(a) \cdot y(a) \tag{a}$$

$$- \sum_{i \in I} \sum_{a \in A_R \cup A_N \cup A_S} c(i, a) \cdot x(i, a) \tag{b}$$

$$- \sum_{a \in A_R \cup A_N \cup A_S} c(a) \cdot g(a) - \sum_{a \in A_S} p_1(a) \cdot g(a) \tag{c}$$

$$- \sum_{a \in A_R} p_2(a) \cdot f(a) \tag{d}$$

The objective of model B10 is to maximise the net value upper bound z_{sup}. z_{sup} is calculated by adding and subtracting six different terms, in (6.11) divided into four groups labelled (a) to (d). The groups each quantify the following:

(a) **The seat demand fulfilment benefit** quantifies the economic benefit achieved by the current solution. This is calculated as the difference of two terms. The first term quantifies the total seat demand fulfilment potential, that is, the benefit when all seat demand is met. $b(a)$ denotes the specific economic benefit of providing a single seat on arc a, and $s(a)$ the total seat demand for arc a. The first term is thus calculated by multiplying $b(a)$ with $s(a)$ and summing over all revenue train service arcs A_R. The second term quantifies the seat demand fulfilment benefit that is not achieved in the current solution. Since $y(a)$ is the seat slack variable capturing how much seat demand is not met in the current solution, the second term is thus calculated by multiplying $b(a)$ and $y(a)$ and summing over all revenue train service arcs A_R;

(b) **The train unit movement costs** quantify the variable costs that are incurred by the movement of the individual train units, covering energy and maintenance. It is calculated as the product of the specific train unit movement cost $c(i, a)$ of assigning one train unit of type i to arc a, and the decision variable $x(i, a)$ of actually doing so, and summed over all train unit types $i \in I$ and all train composition movement arcs, i. e., the union of revenue train services A_R, non-revenue train services A_N and train shunting operations A_S;

(c) **The train service costs and penalties** quantify the fixed costs and penalties incurred in the current solution if a train composition is formed and set in motion, regardless of with how many train units. The cost term is calculated as the specific cost for personnel and infrastructure $c(a)$, multiplied by the binary variable indicating if the arc is being covered $g(a)$, and summed over the union of arcs for revenue train services A_R, non-revenue train services A_N and train shunting operations A_S. The penalty term is calculated as the specific penalty for covering an unwanted shunting $p_1(a)$ multiplied by the variable indicating if the arc is being covered $g(a)$ (i. e., if the shunting is actually performed) and summed over train shunting operations A_S;

(d) **The penalties for uncovered revenue train services** quantify the penalties awarded for those revenue train services that were left uncovered in the solution. It is calculated as the product of the specific penalty $p_2(a)$ for not covering arc a and the binary variable $f(a)$ indicating if arc a is being left uncovered.

The upper bound calculation model B10 is subject to constraints (6.12) to (6.21) described in the following.

$$\sum_{a \in A_0^-} x(i,a) \leq n(i) \qquad \forall i \in I \tag{6.12}$$

$$\sum_{i \in I} x(i,a) \leq n(a) \qquad \forall a \in A_R \cup A_N \cup A_S \tag{6.13}$$

$$\sum_{i \in I} l(i) \cdot x(i,a) \leq l(a) \qquad \forall a \in A_R \cup A_P \tag{6.14}$$

$$\sum_{i \in I} s(i) \cdot x(i,a) + \sum_{a \in A_R} y(a) \geq s(a) \qquad \forall a \in A_R \tag{6.15}$$

$$\sum_{i \in I} x(i,a) - g(a) \geq 0 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{6.16}$$

$$\sum_{i \in I} x(i,a) - n(a) \cdot g(a) \leq 0 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{6.17}$$

$$g(a) + f(a) = 1 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{6.18}$$

$$\sum_{a \in A_S} e(a,v) \cdot g(a) \leq 1 \qquad \forall v \in V_A \cup V_D \tag{6.19}$$

$$\sum_{a \in A_S} e(a,p) \cdot g(a) \leq d(p) \qquad \forall p \in P_D \tag{6.20}$$

$$\sum_{a \in A_v^+} x(i,a) - \sum_{a \in A_v^-} x(i,a) = 0 \qquad \forall v \in V \setminus V_0, \forall i \in I \tag{6.21}$$

$$x(i,a) \in \mathbb{N}_0 \qquad \forall i \in I, \forall a \in A \tag{6.22}$$

$$y(a) \in \{0,1\} \qquad \forall a \in A_R \tag{6.23}$$

$$g(a) \in \{0,1\} \qquad \forall a \in A_R \cup A_N \cup A_S \tag{6.24}$$

$$f(a) \in \{0,1\} \qquad \forall a \in A_R \cup A_N \cup A_S \tag{6.25}$$

The sum of the flow on all outgoing arcs from source vertices $a \in A_0^-$ arcs must not exceed the available number of train units $n(i)$ by train unit type i (6.12).

All arcs representing train composition movements, i. e., arcs of the categories revenue train service arcs A_R, non-revenue train service arcs A_N and train shunting arcs A_S, these arcs must have assigned to them a number of train units less than or equal to the maximum number of train units $n(a)$ on a given arc a (6.13).

Arcs in the categories revenue train service arcs A_R and arcs representing train units undergoing parking A_P, those arcs must have assigned to them train compositions with a length less than or equal to the maximum assigned train composition length $l(a)$ for that arc a (6.14). $l(i)$ denotes the length of a train unit of type i.

The seat shortage slack variable $y(a)$ assumes the value of number of seats demanded but not provided, this for each revenue train service arc $a \in A_R$ (6.15). $s(i)$ denotes the perceived number of seats provided by a train unit of type i, $s(a)$ the total seat demand of arc a. (For dimensioning, DSB S-tog uses number of seats perceived by passengers rather than nominal number of seats, see Table 3.4 on page 40 for an explanation.)

Equations (6.16) and (6.17) define the binary variable $g(a)$ indicating if an arc is being covered by at least one train unit in the solution. The definition applies to the union of revenue train service arcs A_R, non-revenue train service arcs A_N and train shunting operation arcs A_S. Since $n(a)$ represents the upper bound on the number of assigned train units on arc a, (6.17) ensures that $g(a)$ can only assume values greater than the number of assigned train units on the arc divided by the upper bound of that number. This quotient lies in the interval [0;1]. Since $g(a)$ is binary, (6.16) yields the desired definition of $g(a)$, assuming the value 0 if no train unit is assigned to arc a and 1 otherwise.

Equation (6.17) represents a "big M" formulation. However, since M, in this case $n(a)$, is already as small as possible, the formulation can not be tightened additionally with *mixed 0-1 set valid inequality* as described in [113, Section 8.2]. Still, the Equations (6.16) and (6.17) expose a weak LP relaxation, since they will always result in fractional values if the objective function "pull" associated with assigning train units to the arc in question is negative. This "negative pull" is always the case for train shunting or non-revenue train service arcs, since these have no benefit, only costs and penalties. For revenue train service arcs that also have a benefit exceeding the costs and penalties, the objective function may issue a "positive pull" on the definition of $g(a)$, making the upper bound of become binding, yielding a correct value of $g(a)$ of 1.

The binary variable $f(a)$ indicating whether arc a is uncovered is defined as complementary to $g(a)$ in (6.18).

At most one train shunting operation is allowed following each train service arrival $v \in V_A$ or preceding each train service departure $v \in V_D$ for all train shunting operations A_S (6.19). The binary parameter $e(a, v)$ denotes if arc a exists as having vertex v.

When a train shunting operations are needed, (6.20) ensures that personnel (in the form of depot drivers) is available at all times, i. e., in all depot driver time intervals used by train shunting operations P_D. $e(a, p)$ is a binary parameter indicating if arc a exists in time interval p. $d(p, a)$ denotes the number of depot drivers on duty for time interval p at the station where train shunting arc a is occurring.

Finally (6.21) represents the flow conservation constraints stating that for all non-source and sink vertices $v \in V \setminus V_0$ and for all train unit types $i \in I$, the sum of the flow on all incoming arcs to that vertex A_v^+ must equal that of the flow on all outgoing arcs A_v^- from that vertex.

Equations (6.22) to (6.25) represent the integrality constraints for the decision variable $x(i, a)$, the slack variable $y(a)$, the covered variable $g(a)$ and its counterpart the uncovered variable $f(a)$.

Note that the requirement of parking train units on platform tracks is only partially implemented in model B10. Train units can be parked at platform tracks, but there is no constraint preventing trains parked there from being coupled or decoupled.

Since upper bound calculation model B10 is an arc based model, maximum service distance between scheduled maintenance is not modelled. Nor are conflicts relating to relative train unit order modelled.

6.3 Experimental Results

The three different upper bound calculation models presented here have been implemented in the programming language Java 1.8 with approx. 1,100 lines of code. The handling of input data was performed using the code already written for the greedy heuristic integrated rolling stock planning model described in Chapter 5.

To solve the linear and mixed integer linear programs, IBM ILOG CPLEX 12.6.1 has been used. Apart from the libraries Joda-Time 2.8.2, BTC ASCII Table 1.0 and Kryo 3.0.3, only Java

Table 6.5: Comparison of experimental results from the greedy heuristic with experimental results from the upper bound calculation models. The greedy heuristic results form Chapter 5 are with a stopping criterion of no gain in 5 minutes, which results in a mean processing time is less than an hour for all instances. All MIP models have been MIP hotstarted with an integer solution found by running the greedy heuristic from Chapter 5 on the original, manually constructed plan for the given instance for 60 minutes. *MIP upper bound value* is the best upper bound value [sic] of the upper bound calculation model, found at the processing time limit of 60 minutes, or the optimal value of the upper bound model, if found. *MIP relative upper bound* is the relative distance of the upper bound to the net value of the original, non-modified plan. *MIP gap* is the relative distance between the MIP upper bound value and the best integer solution found by the upper bound calculation model before the processing time limit is reached. A upper bound calculation model solution is considered optimal if this gap is strictly less than 0.05% points. Note that because of this definition of optimality in the numeric experiments, for some instances, the upper bound of the MIP A4 solution is slightly lower than that of the corresponding MIP B10 solution.

Instance metadata		Orig.	Heuristic (5)			A2 upper bound calculation model							A4 upper bound calculation model							B10 upper bound calculation model						
Date	Weekday	Net value [kDKK]	Min net value gain [%]	Mean net value gain [%]	Max net value gain [%]	LP upper bound value [kDKK]	LP relative upper bound [%]	LP processing time [min:s]	MIP upper bound value [kDKK]	MIP relative upper bound [%]	MIP gap [%]	MIP processing time [h:min:s]	LP upper bound value [kDKK]	LP relative upper bound [%]	LP processing time [min:s]	MIP upper bound value [kDKK]	MIP relative upper bound [%]	MIP gap [%]	MIP processing time [h:min:s]	LP upper bound value [kDKK]	LP relative upper bound [%]	LP processing time [min:s]	MIP upper bound value [kDKK]	MIP relative upper bound [%]	MIP gap [%]	MIP processing time [h:min:s]
2012-10-19	Fri	2,175	7.8	8.0	8.2	2,466	13.4	0:02	2,466	13.4	0.0	0:00:02	2,414	11.0	0:12	2,401	10.4	1.8	1:00:00	2,466	13.4	0:05	2,402	10.5	0.0	0:02:09
2014-03-31	Mon	2,881	1.2	1.5	1.6	3,120	8.3	0:03	3,120	8.3	0.0	0:00:03	3,036	5.4	0:18	3,019	4.8	2.4	1:00:00	3,151	9.4	0:04	3,019	4.8	0.0	0:58:41
2014-04-01	Tue	2,881	1.0	1.3	1.9	3,120	8.3	0:02	3,120	8.3	0.0	0:00:03	3,036	5.4	0:10	3,019	4.8	2.3	1:00:00	3,151	9.3	0:04	3,019	4.8	0.0	0:54:41
2014-04-02	Wed	2,879	0.8	1.5	1.7	3,120	8.3	0:02	3,120	8.3	0.0	0:00:03	3,036	5.4	0:09	3,019	4.8	2.6	1:00:00	3,151	9.4	0:04	3,019	4.9	0.0	0:28:33
2014-04-03	Thu	2,879	0.9	1.6	1.9	3,120	8.3	0:02	3,120	8.3	0.0	0:00:03	3,036	5.4	0:10	3,019	4.8	2.7	1:00:00	3,151	9.4	0:04	3,019	4.9	0.0	0:39:28
2014-04-04	Fri	2,919	1.3	2.0	2.4	3,174	8.7	0:02	3,174	8.7	0.0	0:00:03	3,101	6.2	0:11	3,080	5.5	2.5	1:00:00	3,209	9.9	0:04	3,081	5.5	0.0	0:35:33
2014-04-05	Sat	1,775	0.8	0.9	1.0	1,837	3.5	0:01	1,837	3.5	0.0	0:00:01	1,816	2.3	0:06	1,806	1.8	0.0	0:01:22	1,840	3.6	0:03	1,804	1.6	0.0	0:14:48
2014-04-06	Sun	1,167	3.1	3.3	3.7	1,264	8.4	0:00	1,264	8.4	0.0	0:00:01	1,244	6.6	0:05	1,235	5.8	0.2	1:00:00	1,259	7.9	0:03	1,231	5.5	0.0	0:06:51
2014-04-07	Mon	2,881	1.3	1.5	1.9	3,120	8.3	0:02	3,120	8.3	0.0	0:00:02	3,036	5.4	0:10	3,019	4.8	2.5	1:00:00	3,151	9.4	0:05	3,016	4.7	0.2	1:00:00
2014-04-08	Tue	2,881	0.8	1.3	1.7	3,120	8.3	0:03	3,120	8.3	0.0	0:00:03	3,036	5.4	0:14	3,019	4.8	2.5	1:00:00	3,151	9.4	0:05	3,016	4.7	0.1	1:00:00
2014-04-09	Wed	2,879	1.2	1.5	1.9	3,120	8.3	0:02	3,120	8.3	0.0	0:00:02	3,036	5.4	0:10	3,019	4.8	2.7	1:00:00	3,151	9.4	0:05	3,016	4.8	0.2	1:00:00
2014-04-10	Thu	2,881	1.2	1.4	2.0	3,120	8.3	0:02	3,120	8.3	0.0	0:00:02	3,036	5.4	0:10	3,019	4.8	2.5	1:00:00	3,151	9.4	0:05	3,016	4.7	0.2	1:00:00
2014-04-11	Fri	2,920	1.9	2.1	2.4	3,174	8.7	0:02	3,174	8.7	0.0	0:00:02	3,101	6.2	0:11	3,080	5.5	2.8	1:00:00	3,209	9.9	0:05	3,081	5.5	0.0	0:45:20
2014-04-12	Sat	1,775	0.9	1.0	1.0	1,837	3.5	0:01	1,837	3.5	0.0	0:00:01	1,816	2.3	0:05	1,806	1.7	0.4	1:00:00	1,840	3.6	0:03	1,804	1.7	0.0	0:12:13
2014-04-13	Sun	1,169	2.2	2.3	2.5	1,237	5.8	0:01	1,237	5.8	0.0	0:00:01	1,222	4.5	0:06	1,214	3.8	1.0	1:00:00	1,232	5.4	0:03	1,213	3.7	0.0	0:04:02

standard libraries have been used.

Numerical experiments have been performed for the three upper bound calculation with 15 different data instances. The data instances are identical to those described in Section 5.7.1 on page 86.

The upper bound calculation models have been solved both in their original MIP formulation as well as in their LP-relaxed counterparts. Results are presented in Table 6.5. Figures 6.1 and 6.2 show the results from the different upper bound calculation models and the results gained from the greedy heuristic from Chapter 5 compared to the net value of the original manual plan.

The tests were performed on a Dell PowerEdge T610 equipped with 16 Intel Xeon E5620 CPUs at 2.40 GHz and 16 GB RAM running Ubuntu Linux 14.04 LTS. CPLEX parallel processing was enabled.

6.4 Discussion

Table 6.6 shows an overview of different characteristics of the three upper bound calculation models. The characteristics are treated in the following.

6.4.1 Requirements Implementation and Implementation Effort

The upper bound calculation models have a percentage of requirements implemented ranging from 21% to 47% by count. See Table 6.7 for details of which requirements are implemented in which upper bound calculation model.

The number of constraint types implemented in the upper bound calculation models is already reflected in their individual designation.

As a metric for the implementation effort for the respective upper bound calculation models, number of lines of code is used. This is defined as total lines of code for each of the implemented, individual upper bound model classes including, in each case, their respective superclasses and any other domain specific classes they use. The A2 model is the simplest of the three upper bound calculation models leading to the least lines of code needed for its implementation. Since the A4 model assigns compositions rather than a number of individual train unit types (if any) to the arcs of the space-time graph, the implementation effort is also relatively low. Furthermore, model A4 is very similar in terms of the objective function and feasibility determination to the greedy heuristic described in Chapter 5, making it possible to just being "plugged in" this model to the existing framework. The B10 model has the most advanced formulation also yielding the highest effort in terms of lines of code needed for its implementation.

It is interesting to note that there is not proportionality between number of requirements implemented and implementation effort expressed as lines of code. Nor is there proportionality between number of constraint types and implementation effort expressed as lines of code. This has to do with the implementation overhead needed to make the models work, regardless of the inherent complexity they represent.

6.4.2 Tightness of Bounds

In Table 6.6, the tightness of bounds for the MIP case is indicated by the percentage of the number of railway-specific requirements the individual models implement.

Table 6.6: Overview of the characteristics of the proposed upper bound models. Each bar represents a relative measure of the characteristic in question, a full bar representing to a full degree, an empty bar not at all. Where applicable, bars represent characteristics that have been properly quantified. Where this is not possible, the bar size has been estimated.

	Model		
Characteristic	A2	A4	B10
Number of requirements implemented			
Number of constraint types in model			
Implementation effort (lines of code)			
MIP formulation tightness			
LP relaxation tightness			
MIP hotstart with uncovered revenue train services			
MIP hotstart with incomplete train unit trajectories			
Number of model variables			
MIP formulation processing time			
LP relaxation processing time			

For the LP relaxed case of the B10 model, the formulation is weaker than for models A2 and A4. This is due to the LP relaxation weakness of the definition of the covered variable $g(a)$ in Equations (6.16) and (6.17) on page 101. In the LP relaxed case, this weakness leads to an underestimation of the value of the covered variable, this in turn leading to a an underestimation of the covering cost and allowing the solution to use more depot driver resources than are actually available. Since models A2 and A4 do not have a covered variable but define the covering costs as part of their variable definition, these models do not expose the same LP relaxation weakness as the B10 model.

In the MIP case, the B10 formulation is tighter than both A2 and A4 formulations. In all cases model A4 is tighter than model A2.

Upper bounds are shown in Figures 6.1 and 6.2, showing the relative net value gain and the absolute net value, respectively. As may be seen from Figure 6.1 the greedy heuristic from Chapter 5 is able to improve the plan with a substantial order of at least 1/3 of the gap between the manual plan and the upper bound calculated with the B10 model, this for the best runs of the greedy matheuristic with number of runs $n = 50$.

Table 6.7 shows the number of railway-specific requirements implemented in the upper bound calculation model B10 and the greedy heuristic, respectively. As may be seen, the greedy heuristic implements way more requirements than the B10 model. Knowing that all of these requirements all carry a price tag, it may well be assumed that the results from the greedy heuristic lie very close to the optimum indeed.

As may be seen from Figure 6.2, the manually constructed plans for the weekend are already very close to the upper bound.

6.4.3 Processing Times

For faster processing, models A4 and B10 may be hotstarted with an existing MIP solution containing uncovered revenue train services. Model A2 has the covering of revenue train services implemented as a hard constraint, for which reason the mentioned hotstart can not be performed, since this would violate the constraint. However, since the A2 model is so simple, processing times are so low (just a few seconds) that an MIP hotstart is not really needed.

Contrarily, model A2 may be MIP hotstarted with incomplete train unit trajectories, since

Table 6.7: Requirements for rolling stock planning at DSB S-tog and the degree to which they are implemented in the upper bound calculation models. In the table, ▼ symbolises full requirement implementation in the heuristic part of algorithm, ○ partial implementation in the optimisation part of algorithm, and ● full implementation in the optimisation part of algorithm.

		Model			
Requirement Category	Requirement Detail	Greedy heuristic (Chapter 5)	Upper bound calculation model A2	Upper bound calculation model A4	Upper bound calculation model B10
Infrastructure	Adhere to track length capacities for parking	▼		●	●
	Handle order of train units in train composit.	▼			
	Use platform tracks for temporary parking	▼		○	○
	Use side tracks for temporary parking	▼		●	●
	Adhere to train control system rules	▼			
	Adhere to coupling and decoupling rules	▼			
	Keep train unit balance in depot over time	▼			
	Only one shunting per arrival/departure	▼			●
	Handle split depots and track usage rules	▼		●	●
Timetable	Assign train units to all revenue train services	▼	●	●	●
	Enable non-revenue services for positioning	▼		●	●
	Adhere to braiding & train service seq. rules	▼		●	●
Rolling Stock	Adhere to platform lengths by train line	▼	●	●	●
	Adhere to rules on # of train units per train	▼	●	●	●
	Handle train composition flexible space distr.	▼			
Passenger Demand	Provide seats according to demand	▼	●	●	●
Personnel on Duty	Perform shuntings only when personnel avail.	▼			●
Scheduled Maintenance	Get train unit to workshop within dist. limit	▼			
	Even out the flow of train units to workshop	▼			
Unscheduled Maintenance	Get train unit to workshop within time limit	▼			
Friction Sand	Get train unit to facility within distance limit	▼			
Exterior Cleaning	Get train unit to workshop within time limit	▼			
Graffiti Removal	Get train unit to workshop within time limit	▼			
Interior Cleaning	Allow time to clean train units	▼			
	Put newly cleaned train units into service	▼			
Winter Preparedness	Get train unit to facility within time limit	▼			
Exposure of Commercials	Expose commercials in certain regions	▼			
Surveillance Video Requests	Get train unit to workshop within time limit	▼			
Surface Foil Application	Get train unit to facility within time limit	▼			
Passenger Counting Equip.	Assign specific train unit to spec. train lines	▼			
Train Control System Equip.	Assign specific train unit to spec. train lines	▼			
Operating Costs	Minimise energy costs	▼	●	●	●
	Minimise maintenance costs	▼	●	●	●
	Minimise infrastructure usage costs	▼	●	●	●
	Minimise train driver costs	▼	○	●	●
	Minimise depot driver costs	▼		●	●

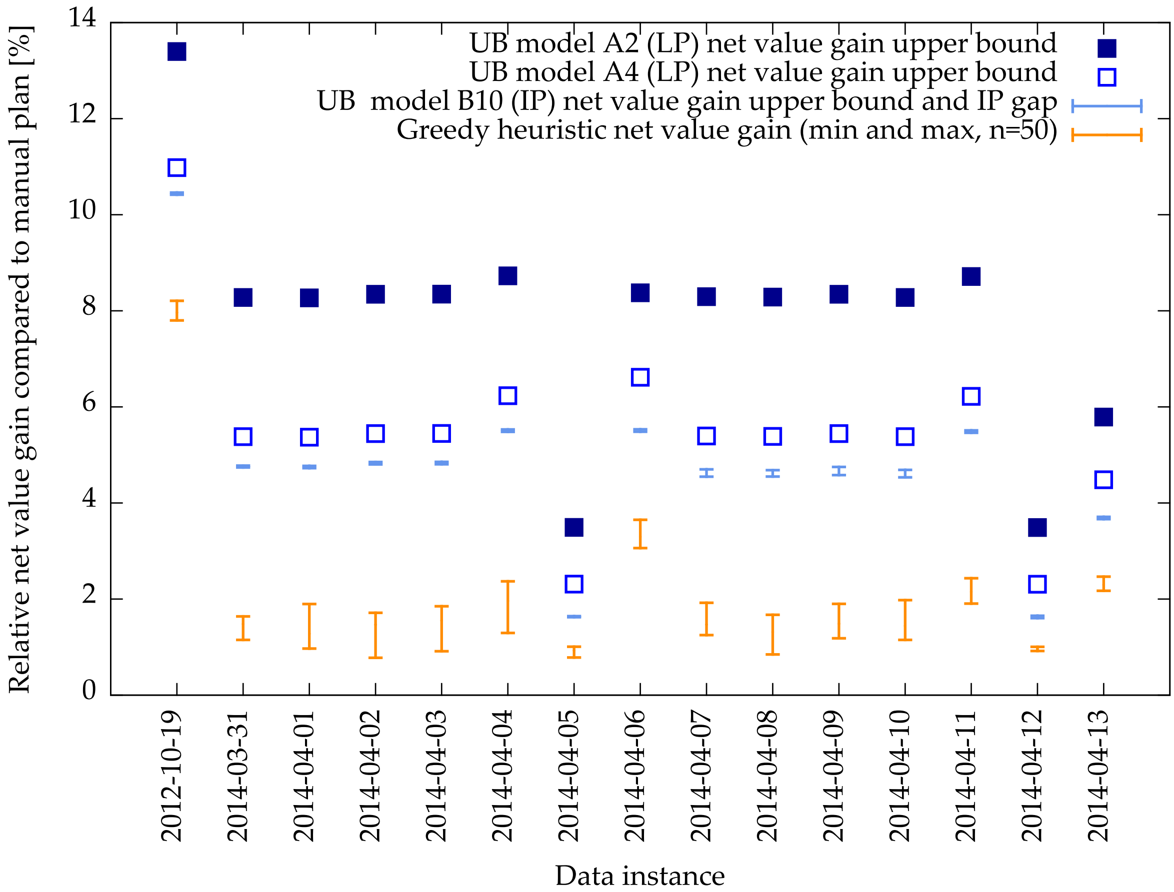

Figure 6.1: Relative net value gain over manual plan comparison for results from upper bound calculation models and greedy heuristic. The greedy heuristic was run with a stopping criterion of no gain in 5 minutes, resulting in a mean processing time is less than an hour for all instances.

it is not a flow model. Incomplete train unit trajectories may result from a failed *retrofitting* of trajectory information from the existing DSB S-tog rolling stock planning system into the data model used here (as described in Appendix B.1).

Models A4 and B10 are flow models with hard constraints for flow conservation. They can not be MIP hotstarted with solutions where the flow is not correct.

Models A2 and A4 use binary variables for train composition types. This is viable for model A2 having only the few revenue train service arcs in it, and consequently only a few feasible composition types on each arc. However, for the A4 model, this concept leads to a very large number of variables due to the large number of arcs (all from all categories). The B10 model has the same number of arcs but the binary composition variables are replaced by integer train unit type variables, leading to a substantial reduction in number of variables.

For some of the instances, the CPLEX solver can detect clique table members for the A4 model and perform clique merging in preprocessing, this shortening processing times. For others this is not the case. This underlines the fact that modern solvers may take different solution approaches based on small differences in the instance data sets leading to very different processing times.

Using the LP relaxed versions of the models, all instances can be solved in less than 20 seconds. Since the A2 upper bound calculation model is very simple and only contains very few variables, it can be solved in seconds also in the MIP case. The A4 model has substantially more variables, leading to longer solution times, especially for in the MIP case. For the A4

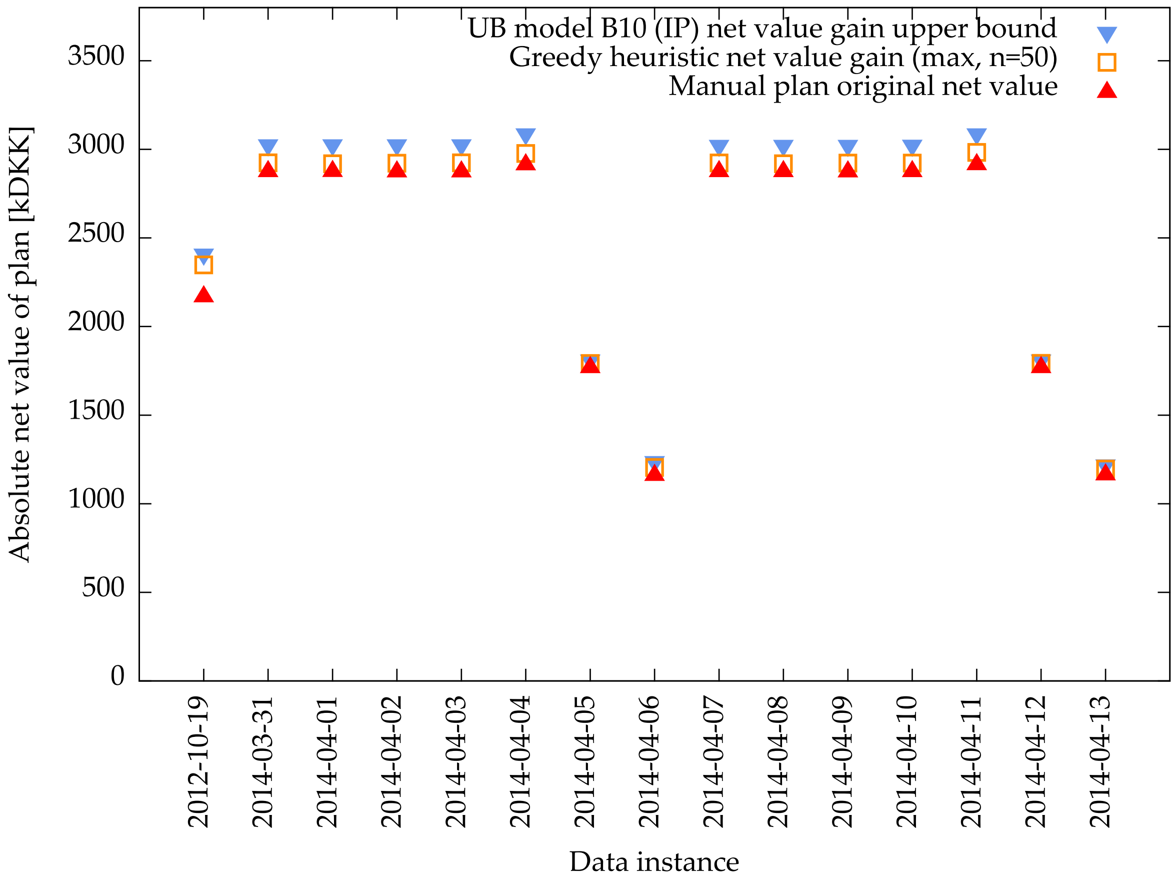

Figure 6.2: Absolute net value comparison for results from upper bound calculation model B10, the greedy heuristic and the original, manual plan. Note that the data points for the symbols are in their centres.

model only one of the 15 instances can be solved to optimality within the time limit of 1 hour. (Peculiarly enough, this instance can be solved to optimality in less than 2 minutes.) Model B10 has shorter LP relaxation solution times than A4, this is considered due to less variables. The solution times in the IP case for the B10 model are also shorter than those for the A4 model. For the B10 model, all but four of the 15 instances can be solved to optimality within the time limit of 1 hour. Of the instances that can be solved to optimality, the average processing time is just below 30 minutes.

As may be seen from Table 6.5, for some instances, the upper bound of the MIP A4 solution is slightly lower than that of the corresponding MIP B10 solution. This occurs when the B10 model has been solved to a MIP gap of just below 0.05 % (indicated as 0.0% in the table), whereas the A4 model has actually found a better solution but has not yet closed its gap. Using the branch-and-bound algorithm of the solver, the A4 MIP model may find a better solution faster than the B10 model, because it has a tighter LP relaxation. Because of the many variables of the A4 model, however, it takes a long time to close the gap of the A4 model. The B10 model has fewer variables and can close the gap faster.

6.4.4 Upper Bound Calculation Model Usage

As may be seen, each of the proposed upper bound calculation models have their distinct characteristics justifying their existence. In the MIP case, models A2 and B10 represent the *fast,* *simple and inaccurate* and the *slow, complex but more accurate* extremes. In the LP relaxed case, the B10 model is strongly inhibited by its weak covered variable definition. This makes the A4 model the tightest LP relaxed formulation in practice.

Deciding which upper bound calculation model type to build and use, the A2 model is suitable if you need a fast upper bound model but don't have much time to implement it. The A4 model is suitable if you need a fast model and have a bit more time to implement its tighter formulation. Finally, the B10 model is suitable if you need an accurate model and have lots of time to implement its complex workings and wait for it to process.

If even more accuracy is needed, the B10 model may be extended in future to also implement the requirement that no coupling or decoupling be performed for overnight parking at the platform. This can be done by adding a linear constraint in the same form as Equation (6.19) on page 101. However, it is not considered possible to implement additional, remaining constraints in a pure, integer linear programming context.

Chapter 7

A Branch-and-Price Matheuristic Integrated Rolling Stock Planning Model

7.1 Introduction

The current chapter is an attempt to improve the results from Chapter 5 by building a new and improved model from the previous model, this by replacing the combined greedy path finding algorithm and heuristics components with branch-and-price and matheuristic components to solve the identical problem. As such, the goal is to solve as many of railway-specific requirements in the optimisation part of the matheuristic algorithm, rather than in the heuristic part.

7.1.1 Overview of the Previous Model

The previous, heuristically based, integrated rolling stock planning model as described in Chapter 5 integrates all railway-specific requirements described in Chapter 3 and listed in Table 7.1. The previous model consists of four main components: The first two components combine to form a data model for the rolling stock plan, the last two components are algorithms applied to modify a given rolling stock plan to improve it. The four components of the previous model are:

1. **The combined timetable and infrastructure data model:** A space-time graph with extended arc and vertex attributes, describing the timetable, the infrastructure, passenger demand, personnel on duty, which train service has which train unit assigned to it and in which individual, relative order, etc. In the graph, arcs represent the possibility of a train unit to move in space and time or in time only. The vertices represent space and time events, that is, points in space and time where a train unit may perform different movements later on. As such, the topology of the graph plays an important role stating which activities are possible from an event. The graph exhibits feasibility and resources consumption features to the shortest path algorithm (see below). A schematic illustration of the principles in the space-time graph is shown in Figure 5.2 on page 78;

2. **The data model for train units**, interconnected with the space-time graph, describing the activities of the train units, e. g. which train unit is assigned to which train service. The data model also keeps track of which timetable lines the train unit in question may be assigned to. Start and finish space-and-time attributes may be set to control depot balance, scheduled maintenance and refilling of consumables. The train unit data model

Table 7.1: Overview of the requirements for rolling stock planning at DSB S-tog and to which degree they are implemented in the integrated models. In the table, ▼ symbolises full requirement implementation in the heuristic part of algorithm, ○ partial implementation in the optimisation part of algorithm, and ● full implementation in the optimisation part of algorithm. The consequences of the heuristically implemented railway-specific requirements for the branch-and-price matheuristic are discussed in Section 7.5.2.

Requirement Category	Requirement Detail	Greedy heuristic (Chapter 5)	Branch-and-price matheuristic
Infrastructure	Adhere to track length capacities for parking	▼	●
	Handle order of train units in train composit.	▼	▼
	Use platform tracks for temporary parking	▼	●
	Use side tracks for temporary parking	▼	●
	Adhere to train control system rules	▼	▼
	Adhere to coupling and decoupling rules	▼	▼
	Keep train unit balance in depot over time	▼	●
	Only one shunting per arrival/departure	▼	▼
	Handle split depots and track usage rules	▼	●
Timetable	Assign train units to all revenue train services	▼	●
	Enable non-revenue services for positioning	▼	●
	Adhere to braiding & train service seq. rules	▼	●
Rolling Stock	Adhere to platform lengths by train line	▼	●
	Adhere to rules on # of train units per train	▼	●
	Handle train composition flexible space distr.	▼	▼
Passenger Demand	Provide seats according to demand	▼	●
Personnel on Duty	Perform shuntings only when personnel avail.	▼	●
Scheduled Maintenance	Get train unit to workshop within dist. limit	▼	●
	Even out the flow of train units to workshop	▼	●
Unscheduled Maintenance	Get train unit to workshop within time limit	▼	●
Friction Sand	Get train unit to facility within distance limit	▼	●
Exterior Cleaning	Get train unit to workshop within time limit	▼	●
Graffiti Removal	Get train unit to workshop within time limit	▼	●
Interior Cleaning	Allow time to clean train units	▼	●
	Put newly cleaned train units into service	▼	●
Winter Preparedness	Get train unit to facility within time limit	▼	●
Exposure of Commercials	Expose commercials in certain regions	▼	●
Surveillance Video Requests	Get train unit to workshop within time limit	▼	●
Surface Foil Application	Get train unit to facility within time limit	▼	●
Passenger Counting Equip.	Assign specific train unit to spec. train lines	▼	●
Train Control System Equip.	Assign specific train unit to spec. train lines	▼	●
Operating Costs	Minimise energy costs	▼	●
	Minimise maintenance costs	▼	●
	Minimise infrastructure usage costs	▼	●
	Minimise train driver costs	▼	●
	Minimise depot driver costs	▼	●

also keeps track of the order of the individual train units relative to each other in order to secure feasibility of decoupling operations (see Figure 5.3 on page 78);

3. **A special-purpose resource constrained shortest path algorithm with side constraints** operating on the space-time graph. This algorithm is used to find new candidate train unit trajectories taking into account the maximum service distance a train unit may perform as a resource constraint. As a side constraint, the individual, relative position of the train unit in relation to the other train units in the space-time graph is handled, determining which movements the train unit can perform. The distribution of flexible space in the train composition is also handled as a side constraint. Since the weights on the arcs in the graph are set as the negated additional net value, the shortest path algorithm is operating to find the path between a start vertex and a finish vertex, having the largest additional net value. This is the path for which it is most advantageous to add a new train unit trajectory to the plan. By way of the algorithm constraints, this path is always feasible;

4. **A heuristic framework** to accept or reject the candidate train unit trajectories found using the components described above. The program flow of the heuristic framework is to remove a number of train unit trajectories from the original rolling stock plan and then, one by one, to create a new train unit trajectory and insert it into the plan. The newly inserted train unit trajectories are accepted if combined they produce an increase in the objective value, if not, they are all rejected, and the previous ones re-inserted.

7.1.2 Overview of the New Branch-and-Price Matheuristic Model

The new branch-and-price matheuristic integrated rolling stock planning model proposed in this chapter is based on the previous model, see overview above. The new model integrates every requirement mentioned in Chapter 3 and listed in Table 7.1. However, five of the listed requirements are not part of the optimisation steps of the matheuristic algorithm, they are adhered to by the outer heuristic. The practical impact of this property is treated in Section 7.5.2.

The new model consists of seven main components: The first two components constitute a data model for the rolling stock plan, the last five components are algorithms applied to modify a given rolling stock plan in order to improve it, specific to the new model. The seven components of the new model are:

1. **The combined timetable and infrastructure data model**, identical to the existing data model from Chapter 5;

2. **The data model for train units**, identical to the existing data model from Chapter 5;

3. **A special-purpose resource constrained path enumeration algorithm with side constraints**, identical to the existing path finding algorithm in Chapter 5;

4. **A mixed integer linear program** to find an optimal or near-optimal combination of new trajectories to re-insert into the plan. The mixed integer linear program is described in Section 7.2;

5. **A column generation framework** in which the mixed integer linear program is LP-relaxed and turned into a *restricted master problem*. The column generation framework is used to find new train unit trajectory candidates (columns) for the solution and is described in Section 7.3;

6. **A branch-and-bound framework** to turn fractional solutions found in the column generation process into integer ones. The branch-and-bound framework is described in Section 7.4;

7. **A (thus) matheuristic framework** to govern the whole solution process. The overall procedure of the matheuristic framework is to remove a number of train unit trajectories from the original rolling stock plan (Components 1 and 2 above) and then, using Components 3 to 6 (above) of the new model to decide which ones to choose, and insert those into the plan. The matheuristic framework is described in Section 7.5.

7.2 Mixed Integer Linear Program

The fourth component of the new integrated rolling stock planning model is the mixed integer linear program used in a branch-and-price framework to find an optimal set of train unit trajectories to re-insert into the rolling stock plan.

The mixed integer linear program is formulated as a generalised set partitioning problem [96] using the sets (with corresponding indices), parameters and decision variables as defined in Tables 7.2 to 7.4.

7.2.1 Objective Function

The objective is, in each iteration of the matheuristic (see Section 7.5), to assign train unit trajectories to train units so that the net value z is maximised (7.1).

$$\max z = \tag{7.1}$$

$$\sum_{a \in A_R} b(a) \cdot s(a) - \sum_{a \in A_R} b(a) \cdot y(a) \qquad \text{(a)}$$

$$+ \sum_{u \in U} \sum_{j \in J_u} \Big(r(j,u) - c(j,u) - p(j,u) \Big) \cdot x(j,u) \qquad \text{(b)}$$

$$- \sum_{a \in A_R \cup A_N \cup A_S} c(a) \cdot g(a) - \sum_{a \in A_S} p_1(a) \cdot g(a) \qquad \text{(c)}$$

$$- \sum_{a \in A_R} p_2(a) + \sum_{a \in A_R} p_2(a) \cdot g(a) \qquad \text{(d)}$$

$$- \sum_{a \in A_R \cup A_N \cup A_S} p_3 \cdot h(a) \qquad \text{(e)}$$

In order to model the actual cost structure of rolling stock planning and to enable the integration of all relevant constraints reflecting the railway-specific requirements at DSB S-tog, and to enable solution by the branch-and-price method, the net value z in (7.1) is calculated by adding and subtracting eight different terms, in (7.1) divided into five groups labelled (a) to (e). The groups each quantify the following:

(a) **The seat demand fulfilment benefit** quantifies the monetary benefit achieved by the current solution, calculated as the difference of two terms. The first term calculates the total seat demand fulfilment potential, i. e., the benefit if all seat demand is fulfilled. With $b(a)$ denoting the specific monetary benefit of providing a single seat on arc a, and $s(a)$ denoting the total seat demand for arc a, the first term is calculated by multiplying $b(a)$ and $s(a)$ and summing over all revenue train service arcs A_R. The second term quantifies the seat demand

Table 7.2: Sets of the matheuristic model, their corresponding indices, domains and definitions, ordered alphabetically by symbol. Sets are in upper case, their indices use same lower case symbol without subscripts or superscripts.

Symbol	Description	Index, domain, definition
A	Arcs in the graph G, each arc going from one vertex to another. An arc corresponds to a time interval $p \in P$	$a \in A$; $a = (v_1, v_2)$ $v_1, v_2 \in V$
A_j	The arcs of train unit trajectory j	$a \in A_j \subset A$
A_N	Arcs representing non-revenue train services	$a \in A_N \subset A$
A_P	Arcs representing train units undergoing parking	$a \in A_P \subset A$
A_Q	Train service sequence arcs connecting train service arrival events with the next possible departure event	$a \in A_Q \subset A$
A_R	Arcs representing revenue train services	$a \in A_R \subset A$
A_S	Arcs representing train shunting operations	$a \in A_S \subset A$
G	The directed and acyclic graph with vertices V and arcs A	$G = (V, A)$
J	All train unit trajectories currently in the mixed integer linear program. A train unit trajectory is an ordered set of arcs representing the movement in space and time of a train unit in the graph G	$j \in J$ $j = a_1, a_2, ..., a_{\|j\|}$ $a \in A$
J_u	The set of train unit trajectories for train unit u	$j \in J_u \subset J$
P	All possible time intervals, a time interval being a sorted 2-tuple of point in time	$p \in P$; $p = (t_1, t_2)$ $t_1, t_2 \in T$; $t_1 < t_2$
P_D	Depot driver time intervals, i. e., the time intervals created by ordering and making unique the set of points in time (start and finish) from all train shunting operation arcs	$p \in P_D \subset P$
Q	Points in space (each depot track, each side track, all platform tracks [as a whole] at every station)	$q \in Q$
T	Points in time	$t \in T$
U	Individual train units currently available	$u \in U$
U^*	The set of train units selected for train unit trajectory removal and re-insertion	$u \in U \subset U$ $\|U\| = k$
V	Vertices in the graph G, each vertex being a (point in space, point in time)-tuple	$v \in V$; $v = (q, t)$ $q \in Q$; $t \in T$
V_A	Vertices representing the arrival of train services (revenue and non-revenue)	$v \in V_A \subset V$
V_D	Vertices representing the departure of train services (revenue and non-revenue)	$v \in V_D \subset V$

Table 7.3: Parameters of the matheuristic model, their domains and definitions, ordered alphabetically by symbol. All parameters have symbols in lower case.

Symbol	Description	Domain, definition
$b(a)$	The specific benefit (i. e., economic value) of providing one seat on arc a, proportional to the length of the time interval p of the arc, see Chapter 5	$b(a) \in \mathbb{R}_0^+$
$c(j,u)$	The train unit movement costs of assigning train unit trajectory j to train unit u	$c(j,u) \in \mathbb{R}_0^+$
$d(p,a)$	The number of depot drivers on duty in time interval p for the station relating to shunting arc a	$d(p) \in \mathbb{N}_0$
$e(a,j)$	Indicator parameter equal to 1 if arc a exists in trajectory j, (i. e., if $a \in A_j$) and 0 otherwise	$e(a,j) \in \{0,1\}$
$e(a,p)$	Indicator parameter equal to 1 if arc a exists in time interval p, and 0 otherwise	$e(a,p) \in \{0,1\}$
$e(a,v)$	Indicator parameter equal to 1 if arc a exists as using vertex v and 0 otherwise	$e(a,v) \in \{0,1\}$
k	The number of train units to select in order to remove and re-insert their trajectories	$k \in \mathbb{N}_1$ $k = \lvert U^* \rvert$
$l(a)$	The maximum length of train composition assigned to arc a	$l(a) \in \mathbb{R}^+$
$l(u)$	Length of train unit u	$l(u) \in \mathbb{R}^+$
n	Number of test runs performed on each data instance for algorithm performance testing	$n \in \mathbb{N}_1$
$n(a)$	The maximum number of train units in a train composition for revenue and non-revenue train services and train shunting operations	$n(a) = \{2 \mid a \in A_R\}$ $n(a) = \{3 \mid a \in A_N\}$ $n(a) = \{\max(n'(a), 1) \mid$ $a \in A_S\}$
$n'(a)$	The number of train units assigned to a given arc in the original, manual plan	$n'(a) \in \mathbb{N}_0$
$n_1(a)$	Lower bound on $n(a)$ used in flow branching, see Section 7.4.2	$n_1(a) \in \{0,1,2\}$
$n_2(a)$	Upper bound on $n(a)$ used in flow branching, see Section 7.4.2	$n_2(a) \in \{1,2,3\}$
$p(j,u)$	Penalty awarded for better depot track utilisation when assigning train unit trajectory j to train unit u	$p(j,u) \in \mathbb{R}^+ \mid a \in A_S$ $p(j,u) = 0 \mid a \in A \setminus A_S$
$p_1(a)$	The penalty awarded for undesired train shuntings. If train shunting operation arc $a \in A_S$ is in category "undesired" this is a positive real number, otherwise and for all other arcs $p_1(a) = 0$	$p_1(a) \in \mathbb{R}_0^+$
$p_2(a)$	The penalty awarded for uncovered revenue train services. If arc a is a revenue train service arc, i. e., if $a \in A_R$ this is a positive real number, otherwise and for all other arcs $p_2(a) = 0$	$p_2(a) \in \mathbb{R}_0^+$
p_3	The penalty awarded for using an artificial variable $h(a)$ in the solution of the restricted master problem, i. e., the infeasibility penalty	$p_3 \in \mathbb{R}^+$
$r(j,u)$	Reward given for better robustness when assigning train unit trajectory j to train unit u	$r(j,u) \in \mathbb{R}^+ \mid a \in A_Q \cup A_R$ $r(j,u) = 0$ otherwise
$s(a)$	The total seat demand of arc a	$s(a) \in \mathbb{N}_0$
$s(u)$	Perceived number of seats for train unit u	$s(u) \in \{125, 300\}$
z	The net value, calculated as benefits and rewards minus costs and penalties, see (7.1), page 113	$z \in \mathbb{R}$

Table 7.4: Decision variables of the matheuristic model, their domains and definitions, ordered alphabetically by symbol. All variables have lower case symbols.

Symbol	Description	Domain
$g(a)$	Variable assuming the value 0 if arc a has no trajectories (i. e., train units) assigned to it, and 1 otherwise, see Equation (7.6) on page 118	$g(a) \in \{0, 1\}$
$h(a)$	Artificial variable to ensure it is possible to find a feasible solution in the first iteration of the column generation process, even though no trajectories may have been generated that use the arc a, see Equation (7.6) on page 118	$h(a) \in \mathbb{N}_0$ $h(a) \leq n(a)$
$x(j, u)$	Assign train unit trajectory j to train unit u, see Equation (7.5) on page 118	$x(j, u) \in \{0, 1\}$
$y(a)$	Slack variable capturing how much seat demand is not met on arc a, see Equation (7.5) on page 118	$y(a) \in \mathbb{N}_0$

fulfilment benefit that is not achieved in the current solution. With $y(a)$ being the seat slack variable capturing how much seat demand is not met in the current solution, the second term is calculated by multiplying $b(a)$ and $y(a)$ and summing over all revenue train service arcs A_R. See Appendix B.2.2 for details on quantifying $b(a)$;

(b) **The train unit movement costs, penalties and rewards** quantify the variable costs, penalties and rewards that are incurred by the movement of the individual train units. The cost is for energy and maintenance, the penalties in order to better utilise depot track capacity and rewards are for better robustness. The term is calculated as the specific train unit movement reward $r(j, u)$ minus the specific train unit movement cost $c(j, u)$ minus the specific train unit movement penalty $p(j, u)$, all of assigning train unit trajectory j to train unit u. This sum is then multiplied by the decision variable $x(j, u)$ of actually assigning train unit trajectory j to train unit u, and summed over all train units $u \in U$ and all train unit trajectories for each of those train units $j \in J_u$;

(c) **The train composition movement costs and penalties** quantify the fixed costs and penalties incurred in the current solution if a train composition is formed and set in motion, regardless of with how many train units. The cost term is calculated as the specific cost for personnel and infrastructure $c(a)$, multiplied by the binary variable indicating if the arc is being covered in the current iteration $g(a)$, and summed over the union of arcs for revenue train services A_R, non-revenue train services A_N and train shunting operations A_S. The penalty term is calculated as the specific penalty for covering an unwanted shunting $p_1(a)$ multiplied by the variable indicating if the arc is being covered $g(a)$, i. e., if the shunting is actually performed, and summed over train shunting operations A_S;

(d) **The penalties for uncovered revenue train services** quantify the penalties awarded for those revenue train services that are left uncovered in the solution. It is calculated by two terms, the first term being the total penalty if all revenue train service arcs A_R were uncovered, with $p_2(a)$ denoting the specific penalty for not covering arc a. The second term quantifies the penalties saved for revenue train service arcs A_S that are actually covered in the solution, $g(a)$ being the binary variable indicating if arc a is covered.

(e) **The penalties for having artificial variables in the solution** calculated as the specific penalty p_3 of having an artificial variable in the solution, multiplied by the binary artificial variable itself $h(a)$ and summed over the union of arcs for revenue train services A_R, non-revenue train services A_N and train shunting operations A_S. The artificial variable is defined to ensure that there will always be a feasible solution to the restricted master problem for the column generation framework (see Section 7.3) so as to always be able to finish its first iteration. The artificial variable $h(a)$ is thus defined using a *big M* formulation [22].

The current model formulation is exposed to symmetry and fractionality. The conditions under which these properties occur in the model formulation will be discussed in Section 7.7. In order to investigate and alleviate the fractional and symmetric properties of the current model formulation, apart from accommodating the same costs as the heuristic model of Chapter 5, the objective function can now accommodate rewards and penalties, Equation (7.1), label (b). Two versions of the objective function are thus evaluated. Firstly, a new and enhanced objective function is evaluated, addressing fractionality and symmetry while providing better depot track utilisation and greater robustness. Secondly, the standard objective function from Chapter 5 is evaluated. The properties of the objective functions will be compared in Section 7.6.

Setting rewards and penalties to zero, $r(j,u) = p(j,u) = 0, \forall j \in J, \forall u \in U$ in Equation (7.1), label (b), makes the objective function of the mixed integer program equivalent to the standard objective function from Chapter 5. The enhanced objective function uses positive values for rewards and penalties as described in the following. The cost definition is the same in the standard and enhanced objective functions.

In order to better utilise depot tracks, each train shunting arc in the space-time graph carries a penalty in the enhanced objective function. The greater the index of the depot track associated with the depot shunting arc, the greater the penalty on that arc. This penalty is summed over all arcs for each train unit trajectory and used in the objective as $p(j,u)$. This scheme leads to a better "filling up" of the individual depot tracks so as to prevent half-length train units taking up the last space which would actually fit a full-length train unit. It also alleviates symmetry since train unit trajectories using parallel depot tracks but being otherwise identical will have different objective coefficients.

In order to provide solutions that are more robust to delays, each arc in the space-time graph connecting an arrival with a departure in the same train service sequence is rewarded in the enhanced objective function. The rationale for this is to promote re-using train unit(s) in a train service sequence, since the operation of turning around a train composition at a terminal station is more robust than changing it. This also diminishes fractionality in the branch-and-price framework, since train shunting arcs (where split flow may occur) will not be favoured in the solution.

Also for the purpose of robustness, in the enhanced objective, each revenue train service arc is rewarded proportional to the seat demand of the train service. The rationale is that when positioning train units using revenue train services, those train services which have a high passenger seat demand should be favoured to those with a low demand. In this way, seat demand fluctuations beyond the already assigned train composition capacities can be accommodated with a greater likelihood, reducing the likelihood of delays occurring due to overcrowding. These rewards also lead to less symmetry in the branch-and-bound framework since different train unit trajectories all relating to the same positioning of a train unit will have different objective coefficients in the situation where seat demand has already been met.

The rewards on the individual arcs in each train unit trajectory are summed as $r(j,u)$ and used in the objective.

With regard to the benefits, the enhanced objective function is identical to the standard objective function, however the seat demand data for the numerical experiments has been perturbed for those train services that do not have seat demand data. Previously, in the no-data case, the seat demand was set to the average seat demand, in the current data it is set to a uniformly distributed random value close to the average. This is done to alleviate symmetry.

The penalties and rewards in the enhanced objective function have been calibrated experimentally to yield a net value as close as possible to the standard one from Chapter 5 while still providing the above mentioned, desired effects. The differences may be seen by comparing Tables 7.6 and 7.7 on page 130 and on page 131.

7.2.2 Constraints

The mixed integer linear program is subject to constraints (7.2) to (7.10) described in the following.

$$\sum_{j \in J_u} x(j,u) = 1 \qquad \forall u \in U \tag{7.2}$$

$$\sum_{u \in U} \sum_{j \in J_u} e(a,j) \cdot x(j,u) + h(a) \leq n(a) \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.3}$$

$$\sum_{u \in U} \sum_{j \in J_u} e(a,j) \cdot l(u) \cdot x(j,u) \leq l(a) \qquad \forall a \in A_R \cup A_P \tag{7.4}$$

$$\sum_{u \in U} \sum_{j \in J_u} e(a,j) \cdot s(u) \cdot x(j,u) + \sum_{a \in A_R} y(a) \geq s(a) \qquad \forall a \in A_R \tag{7.5}$$

$$\sum_{u \in U} \sum_{j \in J_u} e(a,j) \cdot x(j,u) - n(a) \cdot g(a) + h(a) \leq 0 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.6}$$

$$\sum_{u \in U} \sum_{j \in J_u} e(a,j) \cdot x(j,u) - g(a) + h(a) \geq 0 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.7}$$

$$g(a) \leq 1 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.8}$$

$$\sum_{a \in A_S} e(a,v) \cdot g(a) \leq 1 \qquad \forall v \in V_A \cup V_D \tag{7.9}$$

$$\sum_{a \in A_S} e(a,p) \cdot g(a) \leq d(p,a) \qquad \forall p \in P_D \tag{7.10}$$

$$x(j,u) \in \{0,1\} \qquad \forall j \in J, \forall u \in U \tag{7.11}$$

$$y(a) \in \mathbb{N}_0 \qquad \forall a \in A_R \tag{7.12}$$

$$g(a) \in \{0,1\} \qquad \forall a \in A_R \tag{7.13}$$

$$h(a) \in \mathbb{N}_0 \qquad \forall a \in A \tag{7.14}$$

All train units U must have exactly one train unit trajectory assigned to them (7.2). J_u denotes the train unit trajectories for train unit u currently in the mixed integer linear program. This is a generalised upper bound (GUB) constraint of the generalised set partitioning problem. The space-time graph contains so called *wormhole arcs* that directly connect the station source and sink vertices for a particular train unit. Following a wormhole arc is the equivalent to removing a particular train unit from the problem all-together. For each train unit, a train unit trajectory using the wormhole arc is constructed and added to the mixed integer linear program. This ensures that constraint (7.2) can always be satisfied in the column generation process (see Section 7.3).

All arcs representing train composition movements, i. e., arcs of types revenue train service arcs A_R, non-revenue train service arcs A_N and train shunting arcs A_S, these arcs must have assigned to them a number of train units less than or equal to the maximum limit $n(a)$ (7.3). The binary parameter $e(a, j)$ indicates if arc a exists as a part of train unit trajectory j. The artificial variable $h(a)$ is added to ensure that there will always be a feasible solution to the restricted master problem for the column generation framework (see Section 7.3). If the trajectories that are currently in the mixed integer program cannot ensure feasibility of this constraint, the model can choose to include the variable $h(a)$ in the solution, this at penalty p_3 as defined in the objective function, and feasibility is ensured.

Arcs used by the set of train unit trajectories in the categories revenue train service arcs A_R or arcs representing train units undergoing parking A_P, those arcs must have assigned to them train compositions with a length less than or equal to the maximum assigned train composition length for that arc $l(a)$ (7.4). $l(u)$ denotes the length of train unit u. Since any individual track space in excess of the length of the maximum integer number of train units (of any train unit type) cannot be used, $l(a)$ is set to this maximum length rather than the actual length of the given track represented by the arc. This is equivalent to performing a *mixed integer rounding cut* as described in [113, Section 8.7].

The seat shortage slack variable $y(a)$ assumes the value of number of seats demanded but not provided, and is defined for each revenue train service arc used by the set of train unit trajectories A_R (7.5). $s(u)$ denotes the perceived number of seats provided by train unit u, $s(a)$ the total seat demand of arc a. DSB S-tog operates with number of seats perceived by passengers rather than nominal number of seats, see Section 3.4 for an explanation.

Equation (7.6) defines the binary variable $g(a)$ indicating if an arc is being covered by at least one train unit in the solution. The definition applies to the union of all arcs of the following categories: Revenue train service arcs A_R, non-revenue train service arcs A_N and train shunting operation arcs A_S. Since $n(a)$ represents the upper bound on the number of assigned train units on arc a, (7.6) makes sure that $g(a)$ can only assume values greater than the actual number of assigned train units on the arc divided by the upper bound of that number. This quotient lies in the interval [0;1]. Since $g(a)$ is binary, (7.6) yields the desired definition of $g(a)$, assuming the value 0 if no train unit is assigned to arc a and 1 otherwise.

In order for the mixed integer linear program to function in a branch-and-price context, Equations (7.7) and (7.8) need also be defined. These ensure that the definition of the covered variable $g(a)$ is defined as tightly as possible when the mixed integer linear program is LP-relaxed. In addition, it makes sure that all column generation framework relevant constraints also have dual variables that can be used in the subproblem, see Section 7.3. If the upper bound on covered variable $g(a)$ would only be set using a variable upper bound in the solver, no dual information would be available and non-basic variables at their upper bounds would possibly occur, see [22, Chapter 2]. This would compromise the calculation of the reduced cost (see Section 7.3).

Equation (7.6) represents a "big M" formulation. However, since M, in our case $n(a)$, is already chosen as small as possible, the formulation can not be additionally tightened by adding a *mixed 0-1 set valid inequality* as described in [113, Section 8.2].

Still, Equations (7.6) to (7.8) do pose a weak LP relaxation: They result in fractional values of $g(a)$ if the objective function "pull" for assigning train units to the given arc is negative. This "negative pull" is the case for train shunting or non-revenue train service arcs: These have no benefit, only costs and penalties. For revenue train service arcs that have a benefit exceeding the costs and penalties, the objective function may issue a "positive pull" on $g(a)$, making the upper bound become binding, yielding a correct value of $g(a)$ of 1.

This weakness leads to an underestimation of the value of the covering variable, this in turn leading to an underestimation of the covering cost and allowing the use of more depot driver resources than actually available.

In order to alleviate this weakness, $n(a)$ is set to 1 for all train shunting arcs that have a number of train units strictly less than 2 assigned to them in the original, manual rolling stock plan. This is equivalent of only permitting shunting operations with two or more train units when they actually occur in the manual plan. In the vast majority of cases, this provides a much tighter definition of $g(a)$ leading to substantially less fractionality in the branch-and-price framework.

The artificial variable $h(a)$ is also added to Equations (7.6) and (7.7) as a way of adhering to these constraints regardless of the value of $g(a)$, this at the penalty p_3 as defined in (7.1).

A maximum of one train shunting operation is allowed following each train service arrival $v \in V_A$ or preceding each train service departure $v \in V_D$ for all train shunting operations A_S (7.9). The binary parameter $e(a, v)$ denotes if arc a exists as having vertex v.

When a train shunting operation is required, (7.10) ensures that enough personnel is available at all times, i. e., in all depot driver time intervals used by train shunting operations P_D. $e(a, p)$ is a binary parameter indicating if arc a exists in time interval p. $d(p, a)$ denotes the number of depot drivers on duty for time interval p at the station where train shunting arc a is occurring.

Equations (7.11) to (7.14) are the integrality constraints for the decision variable $x(j, u)$, the slack variable $y(a)$, the covered variable $g(a)$ and the artificial variable $h(a)$, respectively.

In its present formulation, the mixed integer linear program does not prevent coupling or decoupling from being performed for overnight parking at the platform. This requirement could be implemented by adding a linear constraint in the same form as Equation (7.9) on page 118, but has not been attempted here. If a solution to the mixed integer linear program violates this requirement it will be rejected as described in Section 7.5.2.

7.3 Column Generation Framework

The fifth component of the new integrated rolling stock planning model is the column generation framework.

Column generation is a method applied for solving linear programs with a very large number of variables [45, 75, 113]. The general idea is not to add all variables (columns) initially but iteratively using dual information to determine new columns that can improve the current solution. The linear program we want solve is designated the *master problem*. We then consider the corresponding, so called *restricted master problem* which contains all the constraints (rows) of the master problem, but only a subset of its variables (columns). In addition to the restricted master problem we have a *subproblem* (also called the *pricing problem*) which we use to find new candidate variables (columns) to put in the restricted master problem to improve its objective value. This is done iteratively. When the subproblem is no longer able to find more columns that can improve the solution, we have an optimal solution for the restricted master problem, which is also an optimal solution for the master problem.

In our case the restricted master problem is the LP-relaxation of the mixed integer linear program described in Section 7.2. Since the variables in this problem correspond to train unit trajectories, the subproblem must produce new train unit trajectories that are candidates for improving the solution of the restricted master problem. In order for this to work, we need to find train unit trajectories by way of their *reduced cost*, i. e., if they have the potential to improve the solution to the restricted master problem. We find these candidate train unit trajectories using the space-time graph G, and the resource constrained shortest path algorithm with special side constraints described in Section 5.5 on page 79. However, we need to update the weights in the graph so as to reflect the reduced cost.

For a given linear program, if its primal formulation is given by (7.15), its dual is then given by (7.16) and the reduced cost vector $\hat{\mathbf{c}}$ can be calculated by (7.17), where $\mathbf{y}$ is the dual solution vector [87].

$$\max\{z = \mathbf{c}^{\mathrm{T}}\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}\} \tag{7.15}$$

$$\min\{w = \mathbf{b}^{\mathrm{T}}\mathbf{y} \mid A^{\mathrm{T}}\mathbf{x} \geq \mathbf{c} \wedge \mathbf{y} \geq \mathbf{0}\} \tag{7.16}$$

$$\hat{\mathbf{c}} = \mathbf{c} - \mathrm{A}^{\mathrm{T}}\mathbf{y} \tag{7.17}$$

Whereas (7.17) generally describes the reduced cost vector, in our case the reduced cost of an individual arc is calculated by its "original cost" minus the sum of the matrix coefficient for the constraint corresponding to the arc multiplied with the dual value of the constraint. The original cost is in our case the sum of the train unit variable rewards, costs and penalties $r(j,u) - c(j,u) - p(j,u)$ in (7.1), label (b).

The constraints for number of shuntings per arrival and departure (7.9) are vertex oriented rather than arc oriented, however, since we are only allowing one train shunting for each train service arrival and departure we can use the dual variable values found for the vertices for all of the train shunting arcs relating to the same vertex. Since only one of the arcs can be chosen at a time, we are not subtracting the same dual value more than once, and the pricing scheme will not be compromised.

The generalised upper bound constraints (7.2) are also not arc oriented, but train unit oriented. The dual values of these constraints must therefore be related to the individual train unit trajectory as a whole and added to the individual trajectory rather than to any of its arcs.

All other constraints are arc oriented.

Note that since multiple constraints may relate to the same arc, the dual reduction term $\mathrm{A}^{\mathrm{T}}\mathbf{y}$ must be calculated for all constraints relating to the given arc.

The reduced cost of a train unit trajectory is thus calculated as its original cost (in our case the cost plus penalties minus rewards) minus the dual reduction for the generalised upper bound constraint for the train unit in question minus the sum of all the dual reductions for all the arcs in the train unit trajectory.

Since we are maximising our objective (7.1), we will be searching for candidate train unit trajectories with a **positive** reduced cost. If no additional train unit trajectories having a positive reduced cost can be found, the current optimal solution to the restricted master problem is also an optimal solution to the master problem.

Since we are solving an LP-relaxed version of the mixed integer linear program, we will most likely get fractional solutions. As will be seen in Section 7.4 we then use a branch-and-bound framework to turn the solutions found using column generation into integer ones.

7.4 Branch-And-Bound Framework

The sixth component of the new integrated rolling stock planning model is the branch-and-bound framework [113]. Used in combination with the column generation framework described in Section 7.3 the two frameworks constitute a *branch-and-price* framework [13, 45].

The purpose of the branch-and-bound framework is to force the LP-relaxed solutions to the restricted master problem found in the column generation framework to become integer. In other words, the framework is used to "unsplit" the split flow of each train unit occurring in the LP-relaxed version of the mixed integer linear program.

The branch-and-bound framework consists of a branch-and-bound tree, two types of branching schemes with corresponding branch entity selection criteria, and a node priority queue as described in the following Sections 7.4.1 to 7.4.4.

7.4.1 Branch-And-Bound Tree

In the branch-and-bound framework, a tree consisting of nodes connected by edges is used (for a visualisation, see Appendix A.4). The tree originates at a root node, has a branching factor of 2 and all edges pointed towards the root node, making it a *rooted, binary, ordered, in-tree* [40].

Integer solutions to the LP-relaxed restricted master problem are created by the process of branching, i. e., by, in each node, creating two new nodes and adding them to the branch-and-bound tree as children of the current node.

Each of the nodes represents the restricted master problem with fixed bounds on one (or more) of its fractional variables. The bounds on the variables may be fixed directly or indirectly through constraints. Each child inherits the fixed bounds of its parent. The root node represents the original restricted master problem without fixed bounds to any variables.

In each of the nodes in the tree, the corresponding restricted master problem, with fixed bounds, is solved using the column generation framework described in Section 7.3. Since we are maximising our objective (7.1), the found LP solution represents an upper bound on the integer solution for that node.

If an integer solution is found as an upper bound for a given node, this is a candidate for the best integer solution found so far (i. e., the best lower bound) and we need not branch further from that node, we are *pruning by optimality*. If the upper bound of a node is less than or equal to the best integer solution found so far (the best lower bound), no further branching needs to be performed on this node, since a better solution is not contained in the search space the node represents. This is *pruning by bound*. Also, when no solution can be found in a node, no further branching is needed from that node. This is *pruning by infeasibility*.

7.4.2 Branching Schemes

The integrated matheuristic rolling stock planning model uses two different branching schemes called *flow branching* and *constraint branching* as described in the following. An overview of characteristics is given in Table 7.5.

Flow branching

Flow branching is performed on the total flow of train units across an arc. In order for this to work, the maximum count constraint (7.3) is reformulated as a *range* with upper and lower bounds $n_1(a)$ and $n_2(a)$ on the flow (7.18). These bounds can be set directly in the solver.

$$n_1(a) \leq \sum_{u \in U} \sum_{j \in J_u} e(a, j) \cdot x(j, u) \leq n_2(a) \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.18}$$

The current, total flow of train units over arc a, which is the value to branch upon, is given by the expression between the two inequality signs in (7.18). This value can be queried directly in the solver as the “activity value” of the constraint, calculated as the vector product between the constraint vector and the solution vector.

Nodes created in this branching scheme are either *force floor nodes* or *force ceiling nodes*, in which the lower bound $n_1(a)$ and the upper bound $n_2(a)$ of the flow on the arc is set to the

Table 7.5: Overview of the characteristics of the branching schemes used. Notes to table: 1) Ceiling nodes are considered to be on 1-branches, flow nodes are considered to be on 1-branches if their upper bound (UB) is strictly greater than 0; 2) Flow nodes are considered to be on 0-branches if their upper bound (UB) equals zero; 3) If the arc on which flow branching is applied is not reachable by other train units, an individual affection is achieved; 4) May affect other train units through constraints in restricted master problem.

	Branch type	Branching scheme: Flow branching	Branching scheme: Constraint branching
Node type names	1	Ceiling; Floor, UB>0 [1]	Force through
	0	Floor, UB=0 [2]	Force around
Affects individual train unit	1	○[3]	●
	0	○[3]	●
Affects multiple train units	1	●	○[4]
	0	●	○[4]
Ensures integer flow on arc	1	●	
	0	●	
Ensures integer trajectory on arc	1		●
	0	●	●
Does set covered variable bounds in master problem	1	●	
	0	●	
Does set constraints in subproblem	1		●
	0		●
Does set constraints in master problem	1	●	
	0	●	
Does set variable bounds in master problem	1		●
	0		●
Does set dual prices in subproblem	1	●	●
	0	●	●
Complexity of setting process	1	$O(1)$	$O(\|V\|)$
	0	$O(1)$	$O(1)$

ceiling and floor value of current total flow, respectively:

$$n_1(a) = \left\lceil \sum_{u\in U} \sum_{j\in J_u} e(a, j) \cdot x(j, u) \right\rceil \tag{7.19}$$

$$n_2(a) = \left\lfloor \sum_{u\in U} \sum_{j\in J_u} e(a, j) \cdot x(j, u) \right\rfloor \tag{7.20}$$

For example, if the value of total flow over the arc in question is 1.83, the floor node will enforce an upper bound of 1 on the flow across the arc, whereas the ceiling node will enforce a lower bound of 2, effectively eliminating fractional flow values between 1 and 2.

The bounds are enforced on already existing constraints (that is, their ranges). For this reason, no special implementation is needed in the subproblem: When the bounds change in the master problem, this will be reflected by the dual values used to calculate the reduced cost in the subproblem. This reduces implementation effort.

The flow branching scheme described here is identical to the scheme proposed in [110], however, in our case, the branching constraint need not be added to the problem, since it is already present in the problem formulation in the form of a count constraint on each arc (7.3).

By branching on the total flow over an arc, determining whether this flow should be less than or equal to 0 or greater than or equal to one, we are implicitly also branching on the covered variable $g(a)$ as defined in (7.6), (7.7) and (7.8). Recall that the covered variable assumes the value of 0 if no train units are assigned to the arc, and 1 otherwise.

Conditions by which the covered variable $g(a)$ will be fractional have been given in Section 7.2.2. The further condition may arise, where all trajectories in a solution are integer, but there are still fractional covering variables. In this case it is still necessary to make the fractional covering variables integer in order to not underestimate the value of the objective function. In this case the same branching scheme is used as if the trajectory was fractional and in the interval]0;1[.

In a solution with all-integer trajectories, and only fractional covering variables which have zero cost and do not play a part in any constraints, no further branching is needed. In this case the objective value would not change even though the fractional covered variables would become integer.

The fixing of covered variables is performed using bounds on existing constraints, no special implementation is needed in the subproblem: When the bounds change in the master problem, this will be reflected by the dual values used in the subproblem.

The constraint (7.8) is thus reformulated as a range, with the bounds set as in (7.21) for the floor node, and as in (7.22) for the ceiling node. This technique enables branching on the covered variables without a separate scheme, without adding constraints to the problem and without changing the inequalities.

$$0 \leq g(a) \leq 0 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.21}$$

$$1 \leq g(a) \leq 1 \qquad \forall a \in A_R \cup A_N \cup A_S \tag{7.22}$$

Constraint branching

The second branching scheme in the branch-and-bound framework uses constraint branching as described in [95] and applied also in e. g., [96, 92]. Constraint branching works by identifying a pair of constraints for which a set of columns are covering both constraints and for which there exists a set of columns having a fractional sum of coefficients (also called *sum of fractions*). The idea is then to force that the two constraints must be covered together in the one-branch, and must not be covered together in the zero-branch.

For the case where vertices in the sub-problem graph G have out-degree 2, constraint branching is strongly integerising (at least in a set partitioning problem with a binary A matrix and right hand side b), since each branch will then have unique subsequence for the vertex in question.

In our case constraint branching is performed in that a particular train unit is either forced through an arc in the space-time graph G or forced around it (given that the from-vertex of that arc has been reached). This corresponds to the creation of *force-through nodes* and *force-around nodes* in the branch-and-bound tree, respectively.

In order for the constraint branching scheme to work in a column generation framework, the force-through branch is created implicitly: Not by forcing the resource constrained shortest path algorithm to use a given arc but rather by making no alternatives should the from node of the given arc be reached.

As such, a force-through node is created by the following operations:

- In the subproblem: Disallow the train unit in question to use all other arcs having:
  - the same from-vertex as the arc in question;
  - the same to-vertex as the arc in question.
- In the master problem: Set the upper bounds of the variables of all train unit trajectories using disallowed arcs to zero.

A force-around node is created by:

- In the subproblem: Disallowing the train unit in question to use the arc in question.
- In the master problem: Setting the upper bounds of the variables of all train unit trajectories using disallowed arcs to zero.

This branching scheme works by forcing split flow occurring at the from-vertex of the arc in question to not be split. The branching scheme is necessary in order to be able to branch on solutions with all-integer arc flow and all-integer covering variables, but with fractional trajectories performing a *crossover*. This situation occurs e. g., when two train units have four train unit trajectories, each with the value 0.5. If two train unit trajectories from two different train units follow the same arc, the arc flow is integer, but the trajectories are still fractional. This situation needs to be solved by branching since the two train units may be of different types with hence different properties. See Appendix Figure A.16 on page 173 for a visualisation of a crossover.

7.4.3 Branching Entity Selection

Once a particular branching scheme has been invoked, it must be determined by which entity the branching is to occur. Flow branching branches on arcs, constraint branching on a (train unit, arc)-tuple. Branching is only performed for arcs with corresponding covered variables, see (7.6) and (7.7), since these constitute the minimal set of arcs where the splitting of train unit flow may occur. Which entity to branch on is determined as follows:

1. **Flow branching arc selection** is performed by selecting the train unit trajectory solution columns being either fractional by themselves or having fractional covered variables. From this selected set, the train unit trajectory solution column with the largest variable value is then selected. From the arcs in its trajectory, the ones with covered variables are selected. These are then sorted by a) descending absolute reduced cost and then by descending sum of fractions. The first arc is then chosen.

The rationale for the sorting is a) to branch on the arc where the largest change in objective value is likely to occur (the reduced cost is used as a proxy), and b) to branch on the arc where most of the fractional flow occurs, so as to influence as many trajectories as possible with the branching in order to make them less fractional.

2. **Constraint branching (train unit, arc)-tuple selection** is performed by selecting any fractional train unit trajectory. This determines the train unit in the (train unit, arc)-tuple. Next, any other train unit trajectory with the same train unit as the first one is selected. The two train unit trajectories are then compared to get the first arc on each of the trajectories that is not part of the other trajectory, i. e., by which from vertex the spit occurs. The first one of these two arcs being an arc with a covered variable is then chosen as the arc in the (train unit, arc)-tuple. The rationale for this is simplicity: This type of branching, as a consequence of how branching schemes are chosen, occurs seldom and most often deep in the branch-and-bound tree.

7.4.4 Node Priority Queue

In the process of branching, nodes are not only added to the branch-and-bound tree, they are also added to a single-ended priority queue [40] in order to determine for which node the upper bound is to be calculated as the next.

This priority queue has a priority ordering where nodes are processed using two different ordering schemes, each for their own phase:

1. **Depth first search:** Initially, nodes are processed by descending level (depth), then by branching scheme (in the order enumerated in Section 7.4.2). Within each branching scheme, ceiling and floor nodes are processed in the order of proximity to the flow value. Force-through nodes are processed before force-around nodes. The rationale is to dive into the branch-and-bound tree so as to find an integer feasible solution as quickly as possible. This in turn is in order to establish a good lower bound which the algorithm can then use to prune later processed nodes;

2. **Best first search:** As soon as the first integer node has been found, the priority ordering changes to a best first scheme in which nodes with the highest (best) upper bound are processed first. If there is a tie, nodes are processed in the order of branching scheme (in the order enumerated in Section 7.4.2). The rationale is then to find the best solution as fast as possible by using the upper bound as a sign of direction.

7.5 Matheuristic Framework

The seventh and last component of the new integrated rolling stock planning model is the matheuristic framework used to govern program flow of the new model.

The overall concept of the heuristic framework in the new model is the same as in the previous model: Select k number of train units U^* and remove their respective train unit trajectories from an existing rolling stock plan. Then generate new train unit trajectories and re-insert them into the plan. The way the new train unit trajectories are generated differs between the two models, the previous model using a greedy sequential resource constrained shortest path algorithm with side constraints, the new model utilising the previously described combined branch-and-price framework (see Sections 7.3 and 7.4).

For reasons of simplicity, the selection of train unit trajectories to remove from the plan is conducted at random. The objective value of the matheuristic framework is the net value z from the mixed integer linear program (see Section 7.2.1 on page 113).

The underlying heuristic used is that of hill climbing [76]. Since the original train unit trajectories are also included in mixed integer linear program, the new solution found is always at least just as good at the previous one. This is equivalent of saying that the iteration net value increase is always non-negative. The underlying hill climber heuristic framework thus never has to decide whether or not to accept a solution, every solution is accepted.

However, since train unit order conflicts may occur (see Section 7.5.2), a roll-back mechanism to undo the changes performed in the individual iteration is still needed.

7.5.1 Program Flow

The branch-and-bound framework works inside a matheuristic framework, in which only parts of the plan are modified. For this reason all variables not having anything to do with the k number of modified train units need to be fixed initially, so as to keep them unchanged in the solution. This occurs prior to the creation of the root node in the branch-and-bound tree.

The branch-and-bound algorithm then works by creating the root node for the matheuristic iteration in progress, which is the restricted master problem without any fixed bounds. The root node is then added to the node priority queue. As long as there are nodes in the priority queue, the following steps are then performed:

1. Get next node from node priority queue;
2. Fix variable or constraint bound(s) according to branching scheme for current node and recursively for all ancestor nodes in the branch-and-bound tree;
3. Calculate node IP objective value upper bound using column generation;
4. Process branching:
    - If the restricted master problem is infeasible, or if its objective value is less than or equal to the best lower bound, don't branch on this node, return;
    - Else, if there is fractional flow anywhere, or if there are fractional covering variables that have a non-zero objective coefficient or are part of constraints, invoke branching scheme 1: Arc flow branching;
    - Else, if there are fractional trajectories, invoke branching scheme 2: Constraint branching;
    - Else, the current node is the new best node is found, don't branch further, return;
    - Select branching entity (described in Section 7.4.3) according to invoked branching scheme;
    - Create new nodes and add them to the branch-and-bound tree and to the node priority queue.
5. Release the variable bound(s) and/or constraint bound(s) for the current node and all of its ancestors recursively up the branch-and-bound tree.

After this procedure, variables not having anything to do with the k number of modified train units U^* (that have previously been fixed) are released again.

The hierarchy of branching schemes has been chosen on the grounds of descending restrictiveness, so as to have branching schemes that exert a lot of restrictions near the root node.

Note that the primal/dual nature of the matheuristic linear program is an advantage in a branch-and-price framework. Due to the dimensions of our problem, the solver usually solves the root node using the dual simplex method. Whenever the restricted master problem is solved to optimality in any node of the branch-and-bound tree, both a primal and a dual optimal solution exists. Adding more columns to the linear program as part of the column generation process makes the dual solution infeasible but the primal solution remains feasible. For this reason the solver can use the existing primal basis to find the next optimal solution rather than starting all over.

If any non-zero artificial variable is detected in the solution, the solution is considered infeasible. The presence of artificial variables in the solution may occur when the branching is forcing a train unit trajectory through parts of the space-time graph where there is no space for an additional train unit, either by count or by length, or where there are no depot drivers available for train shuntings.

7.5.2 Potential Train Unit Order Conflicts

The column generation framework takes into account conflicts arising regarding train unit order, coupling and decoupling, and flexible space distribution. However, the resource constrained shortest path algorithm only prevents conflicts from occurring that may arise between the train unit trajectory that is being constructed and the existing train unit trajectories in the rolling stock plan. It does not take into account conflicts that may arise between trajectories found individually using the path finding algorithm.

Should conflicts like these occur, the conflicts are found in the process of inserting the chosen train unit trajectories into the graph. If an attempt is made to insert train unit trajectories into the graph that are mutually incompatible, the space-time graph component will throw an exception indicating that a conflict has been detected.

For this reason, when an instance of the mixed integer linear program has been solved with the branch-and-price framework in an iteration of the matheuristic and conflicts have been detected in the solution, the changes of this iteration are rolled back and a new iteration is started.

Experiments have shown that train unit order conflicts occur more often for higher values of k than for lower values. This is as expected: For higher values of k there are more trajectories handled outside the space-time graph, trajectories between which train unit order can be violated. For small values of k, train unit order is widely taken care of in the space-time graph by the resource constrained shortest path algorithm with special side constraints. Train unit order conflicts occur in approx. 1.5% of all iterations for $k = 3$, in approx. 2.3% for $k = 6$ and in approx. 4.8% for $k = 9$.

Experiments with solving small instances in the form of individual train service lines for particular days have been performed using the enhanced objective function. The smallest instance involves all 5 train units serving the F line for the Sunday 2014-04-05. This instance was solved to optimality in the branch-and-price context, however, the solution was afterwards rejected for having a train unit order conflict for Hellerup station at side track 15 between 00:39 and 00:47. The instance of line H for Friday 2012-10-19 with all 9 train units was also solved to optimality in the branch-and-price step. However, this solution also had in it a train unit order conflict at Farum station, depot track 10 between 09:22 and 15:23. An attempt to solve line F for Monday 2014-03-31 with all 13 train units resulted in a huge branch-and-bound tree and the experiment was terminated before the instance was solved to completion.

For reasons mentioned above, five of the railway-specific requirements in Table 7.1 are marked as not handled in the optimisation steps of the matheuristic. The requirements are implemented in so far as to prevent violations occurring in the solution by rejecting the solution if they occur, they do not play a role in the optimisation process of the branch-and-price framework.

7.6 Numerical Experiments

The new integrated rolling stock planning model proposed here is tested on various data instances. The purpose of the experiments has been to benchmark the performance of the new matheuristic model against the previous heuristic model and these against plans produced manually. The conditions have been selected as to make the benchmarking as fair as possible.

The new components of the integrated rolling stock planning model presented in this chapter have been implemented in the programming language Java 1.8 with approx. 4,000 lines of code. Almost the entire 15,000 line code base of the previous heuristic model has been reused, the total size of the code base thus reaching approx. 19,000 lines of code. These figures do not include unit test cases and visualisation functionality also used for the current chapter.

To solve the linear program, IBM ILOG CPLEX 12.6.1 has been used. Apart from the libraries Joda-Time 2.8.2, BTC ASCII Table 1.0 and Kryo 3.0.3, only Java standard libraries have been used.

The tests were conducted on a Dell PowerEdge T610 equipped with 16 Intel Xeon E5620 CPUs at 2.40 GHz and 16 GB RAM running Ubuntu Linux 14.04 LTS.

7.6.1 Data Instances

The models are tested on 15 different rolling stock plan data instances as shown in Tables 7.6 to 7.8. All instances are long-term circulation plans (as opposed to short-term train unit dispatching plans). The instances are complete rolling stock plans produced manually by the planners. Each data instance represents a date, e. g., 2012-10-19 and a weekday, e. g., Friday.

Most of the data in the instances are real-world data, this includes infrastructure data, timetable data, passenger demand data and data on personnel on duty. How the individual parts of the real world data vary between instances is described in the following:

An identical timetable is in effect from Monday to Friday, but a different one is used on Saturdays, and again a different one on Sundays. On mornings after Fridays and Saturdays, night train services operate. The timetable is different between years 2012 and 2014.

The depot driver duties differ by each weekday, since start up procedures on Monday mornings are different from the ones on Tuesday: There is a change of timetable between Sunday and Monday, but not between Monday and Tuesday.

In the data instances, passenger demand is found by running the DSB S-tog passenger prognosis model with the actual, measured passenger data for those days. This is possible because the data instances are in the past. In a realistic planning situation prognosis passenger demand would be used.

The 2012-10-19 instance is special as it represents both the autumn holiday and also extraordinary conditions with infrastructure maintenance works on a parallel, long distance railway line. This plan thus provides extra seating capacity on the one of the train service lines. The other instances represent normal plans with no extraordinary features.

The train unit trajectories in the data instances are those from complete rolling stock plans produced manually by the planners. In the experiments, any infeasible train unit trajectories are removed prior to running either the heuristic or matheuristic.

The space-and-time start and finish points of the original train unit trajectories are kept, new trajectories have the same origin and destination stations. This preserves the depot balance.

Table 7.6: Short processing time numerical experiments for the greedy heuristic and the branch-and-price matheuristic using the **enhanced** objective function. k denotes the number of train unit trajectories to modify, t processing time. The parameter settings for columns with labels (A) to (C) are explained in detail in the text.

Instance meta data		Train units			Original plan characteristics						Modified plan characteristics (A)				(B)		(C)	
Date	Weekday	# of type 1	# of type $\frac{1}{2}$	Total	Costs [kDKK]	Benefits [kDKK]	Penalties [kDKK]	Rewards [kDKK]	Net value [kDKK]	# of infeasible trajectories	Net value gain, matheuristic, first integer, $k = 5$, $t = 1$h [%]	# of iterations = # of branch-and-bound trees	# of nodes created in branch-and-bound trees	# of columns generated in column generation framework	Net value gain, greedy heuristic, $k = 3$, $t = \infty$ [%]	# of iterations	Net value gain, greedy heuristic, $k = 3$, $t = 1$h [%]	# of iterations
2012-10-19	Fri	89	22	111	1,571	3,769	489	705	2,414	1	12.7	81	1,013	7,588	10.0	81	14.5	48,368
2014-03-31	Mon	90	27	117	1,538	4,445	494	760	3,173	0	5.3	68	1,776	7,039	3.7	68	6.9	64,308
2014-04-01	Tue	90	27	117	1,537	4,445	490	760	3,177	2	4.4	54	1,536	8,507	2.9	54	7.0	63,644
2014-04-02	Wed	90	27	117	1,537	4,445	494	760	3,174	2	4.0	52	1,340	7,832	3.2	52	6.9	62,459
2014-04-03	Thu	90	27	117	1,538	4,445	494	760	3,174	1	4.9	59	1,623	7,122	2.6	59	6.8	62,963
2014-04-04	Fri	90	28	118	1,544	4,492	496	758	3,210	3	5.1	57	1,367	8,765	3.8	57	8.0	61,585
2014-04-05	Sat	53	7	60	984	2,780	297	399	1,899	2	9.5	149	873	19,937	8.7	149	10.2	85,469
2014-04-06	Sun	52	11	63	980	2,166	307	331	1,209	0	17.9	162	1,500	29,009	15.6	162	18.5	100,146
2014-04-07	Mon	90	27	117	1,538	4,445	493	760	3,174	0	5.1	59	1,539	8,577	4.1	59	6.9	65,921
2014-04-08	Tue	90	27	117	1,538	4,445	492	760	3,175	2	3.9	67	1,851	10,070	2.1	67	6.9	68,036
2014-04-09	Wed	90	27	117	1,537	4,445	496	760	3,171	3	4.7	77	2,117	7,665	3.2	77	7.1	68,533
2014-04-10	Thu	90	27	117	1,538	4,445	493	760	3,174	1	4.3	48	1,278	8,326	2.4	48	7.0	63,953
2014-04-11	Fri	90	28	118	1,545	4,492	498	759	3,208	4	5.3	75	1,807	8,623	3.6	75	8.0	66,269
2014-04-12	Sat	53	7	60	983	2,780	296	400	1,901	2	9.2	109	717	18,254	8.2	109	10.0	85,973
2014-04-13	Sun	52	7	59	977	2,166	296	330	1,223	0	16.0	106	926	19,057	14.2	106	16.5	84,108

Table 7.7: Short processing time numerical experiments for the greedy heuristic and the branch-and-price matheuristic using the **standard** objective function. k denotes the number of train unit trajectories to modify, t processing time. The parameter settings for columns with labels (A) to (C) are explained in detail in the text.

Instance meta data		Train units			Original plan characteristics						Modified plan characteristics (A)				(B)		(C)	
Date	Weekday	# of type 1	# of type $\frac{1}{2}$	Total	Costs [kDKK]	Benefits [kDKK]	Penalties [kDKK]	Rewards [kDKK]	Net value [kDKK]	# of infeasible trajectories	Net value gain, matheuristic, first integer, $k = 5$, $t = 1h$ [%]	# of iterations = # of branch-and-bound trees	# of nodes created in branch-and-bound trees	# of columns generated in column generation framework	Net value gain, greedy heuristic, $k = 3$, $t = \infty$ [%]	# of iterations	Net value gain, greedy heuristic, $k = 3$, $t = 1h$ [%]	# of iterations
2012-10-19	Fri	89	22	111	1,571	3,769	26	0	2,172	1	5.0	32	942	7,260	4.3	32	8.1	47,567
2014-03-31	Mon	90	27	117	1,538	4,445	26	0	2,881	0	0.7	47	1,653	7,316	0.4	47	2.0	53,195
2014-04-01	Tue	90	27	117	1,537	4,445	26	0	2,882	2	0.4	51	1,783	9,076	0.0	51	1.7	46,104
2014-04-02	Wed	90	27	117	1,537	4,445	28	0	2,880	2	0.4	40	1,408	8,762	0.1	40	2.0	53,261
2014-04-03	Thu	90	27	117	1,538	4,445	28	0	2,879	1	0.3	34	1,220	9,086	0.0	34	2.1	57,350
2014-04-04	Fri	90	28	118	1,544	4,492	28	0	2,919	3	0.5	49	1,671	9,583	1.0	49	2.4	60,034
2014-04-05	Sat	53	7	60	984	2,780	21	0	1,776	2	1.0	173	1,107	21,884	0.9	173	1.1	76,966
2014-04-06	Sun	52	11	63	980	2,166	19	0	1,167	0	3.2	167	2,193	28,698	2.0	167	4.0	96,627
2014-04-07	Mon	90	27	117	1,538	4,445	26	0	2,881	0	0.4	30	956	7,500	0.3	30	1.9	54,641
2014-04-08	Tue	90	27	117	1,538	4,445	26	0	2,881	2	0.4	46	1,292	9,301	−0.4	46	1.6	65,229
2014-04-09	Wed	90	27	117	1,537	4,445	28	0	2,880	3	0.1	45	1,529	8,075	−0.6	45	1.7	62,436
2014-04-10	Thu	90	27	117	1,538	4,445	26	0	2,881	1	−0.1	39	1,361	8,454	−0.4	39	1.8	50,923
2014-04-11	Fri	90	28	118	1,545	4,492	27	0	2,920	4	0.0	58	1,996	9,060	0.2	58	2.4	50,644
2014-04-12	Sat	53	7	60	983	2,780	21	0	1,776	2	1.0	140	944	20,418	0.6	140	1.1	66,102
2014-04-13	Sun	52	7	59	977	2,166	19	0	1,169	0	2.2	147	1,367	19,723	1.8	147	2.6	76,455

Table 7.8: Long processing time numerical experiments for the greedy heuristic, the branch-and-price matheuristic and a hybrid thereof, using the **standard** objective function. k denotes the number of train unit trajectories to modify, t processing time. The parameter settings for columns with labels (D) to (I) are explained in detail in the text.

Instance meta data		Train units			Original plan characteristics						Modified plan characteristics					
											(D)	(E)	(F)	(G)	(H)	(I)
Date	Weekday	# of type 1	# of type $\frac{1}{2}$	Total	Costs [kDKK]	Benefits [kDKK]	Penalties [kDKK]	Rewards [kDKK]	Net value [kDKK]	# of infeasible trajectories	Net value gain, greedy heuristic, $k = 3, t = 36$h [%]	Net value gain, greedy heuristic, $k = 3, t = 48$h [%]	Heuristic gain from $t = 36$h to $t = 48$h [%]	Net value gain, hybrid, $k_1 = 3$, $t_1 = 36$h, $k_2 = 5, t_2 = 12$h [%]	Difference greedy vs. hybrid gain, $t = 48$h [%-points]	Relative difference greedy vs. hybrid gain, $t = 48$h [%]
2012-10-19	Fri	89	22	111	1,571	3,769	26	0	2,172	1	8.31	8.31	0.00	8.31	0.00	0.04
2014-03-31	Mon	90	27	117	1,538	4,445	26	0	2,881	0	2.38	2.38	0.00	2.46	0.08	3.54
2014-04-01	Tue	90	27	117	1,537	4,445	26	0	2,882	2	2.31	2.39	0.08	2.34	−0.04	−1.77
2014-04-02	Wed	90	27	117	1,537	4,445	28	0	2,880	2	2.44	2.46	0.02	2.48	0.02	0.93
2014-04-03	Thu	90	27	117	1,538	4,445	28	0	2,879	1	2.62	2.62	0.00	2.66	0.04	1.46
2014-04-04	Fri	90	28	118	1,544	4,492	28	0	2,919	3	2.75	2.75	0.00	2.79	0.04	1.35
2014-04-05	Sat	53	7	60	984	2,780	21	0	1,776	2	1.17	1.17	0.00	1.21	0.04	3.61
2014-04-06	Sun	52	11	63	980	2,166	19	0	1,167	0	3.97	3.97	0.00	4.02	0.05	1.28
2014-04-07	Mon	90	27	117	1,538	4,445	26	0	2,881	0	2.41	2.43	0.03	2.49	0.05	2.25
2014-04-08	Tue	90	27	117	1,538	4,445	26	0	2,881	2	2.43	2.43	0.00	2.45	0.02	0.82
2014-04-09	Wed	90	27	117	1,537	4,445	28	0	2,880	3	2.27	2.30	0.04	2.36	0.05	2.28
2014-04-10	Thu	90	27	117	1,538	4,445	26	0	2,881	1	2.13	2.18	0.06	2.19	0.01	0.41
2014-04-11	Fri	90	28	118	1,545	4,492	27	0	2,920	4	2.96	3.01	0.05	3.00	−0.01	−0.29
2014-04-12	Sat	53	7	60	983	2,780	21	0	1,776	2	1.15	1.15	0.00	1.17	0.02	1.63
2014-04-13	Sun	52	7	59	977	2,166	19	0	1,169	0	2.70	2.70	0.00	2.73	0.02	0.89

Current rolling stock planning procedures at DSB S-tog still involve some degree of manual work. For this reason data is not available for all aspects of manually produced rolling stock plans. Parts of the data not currently available include: Data on train shuntings and the grouping of virtual (anonymous) train units so as to determine the number of train units in the plan. These data are artificially *retrofitted*, for details, please refer to Appendix B.1.

At DSB S-tog, in the long-term circulation planning process, rolling stock plans are constructed for one week at a time. In this chapter, however, the scope is on each individual day, not consecutive days. Due to this short planning time horizon, restrictions as to service distance are omitted in the experiments, if arbitrary values for the service distance limit would be included, the benchmarking results would not be comparable. Experiments with the previous greedy heuristic model (see Chapter 5) show that the model performs well with the service distance limits in place.

Since the data instances used in the experiments relate to the circulation planning phase of rolling stock planning, the remaining railway-specific requirements related to the short-term train unit dispatching phase are omitted in the experiments.

These conditions explained, the data used in the experiments represent a very close approximation to the real-life planning conditions.

With the data instances described above a typical space-time graph for a weekday has approx. 22,000 arcs and 13,000 vertices and approx. 16,000 arcs and 9,000 vertices for a Saturday or a Sunday.

7.6.2 Obtained Results

For the purpose of performance comparison, in the following, the term *iteration effectiveness* is used to denote the average objective value gain per iteration and the *time effectiveness* the average objective gain per iteration when running the algorithms for a period of time.

The numerical experiments presented in Tables 7.6 to 7.8 have been conducted using both the greedy heuristic, the branch-and-price matheuristic and a hybrid thereof. Experiments have been conducted with both the enhanced and the standard objective function and with short processing times of 1 h and long ones of 48 h. The latter time frame is comparable to the processing times of the existing automated circulation planning system of DSB S-tog. For all experiments presented here, the branch-and-bound search was terminated when the first integer node was found or when more than 50 nodes where created in the branch-and-bound tree. Moreover, in order to alleviate degeneracy in the column generation framework, the column generation process was stopped if there was no change in objective value for 2 min. Of the parameter settings tested experimentally, this setting was the one with the most favourable time efficiency. As such, the branch-and-price step does not necessarily solve the problems in the individual matheuristic iterations to optimality in the presented experiments. For an overview of other model design and calibration decisions, see Appendix A.5 on page 174.

Table 7.6 shows the 1 h short processing time results for the enhanced objective function. The columns below the label (A) show the net value gain, number of iterations, number of nodes and columns created, from running the branch-and-price matheuristic on the original, manual plan with parameter settings $k = 5$ and $t = 1$ h. As may be seen, the matheuristic works well and is able to improve the net value of the original plan by an average of approx. 7.5% across all 15 data instances with 1 h processing time. There is some variation in the number of matheuristic iterations performed, number of nodes and columns created. This is related to the number of train units in the plan, i. e., to the day type. The iteration effectiveness is approx. 2,300 DKK/iteration and the time effectiveness approx. 50 DKK/s for the first hour of processing.

The next columns, labelled (B), show characteristics from running the greedy heuristic on the original, manual plan with parameter settings $k = 3$ and using the same number of iterations as in the columns labelled (A). This yields an average gain of approx. 5.9% across all instances, which is 1.6%-points lower than for the branch-and-price matheuristic. This demonstrates the branch-and-price matheuristic superiority regarding iteration effectiveness.

The columns labelled (C) also show characteristics for the greedy heuristic, but this time given $t = 1$ h to process. This yields an average net value gain of about 9.4% which is 2.1% points better than the branch-and-price matheuristic. This demonstrates the greedy heuristic superiority with regard to time effectiveness. As may be seen, a substantial number of iterations can be performed compared to the branch-and-price matheuristic with identical processing time limits. The iteration effectiveness is approx. 3 DKK/iteration and the time effectiveness approx. 65 DKK/s for the first hour of processing.

Thus, the branch-and-price matheuristic has an approx. 660 factor better iteration effectiveness than the greedy heuristic, whereas the greedy heuristic has an approx. 1.3 factor better time effectiveness than the branch-and-price matheuristic for the first hour of processing original, manual plans and using the enhanced objective function. Using the standard objective function from Chapter 5, with results as shown in Table 7.7, these factors are approx. 400 and 2.8. Thus, both the time and iteration efficiency of the branch-and-price matheuristic relative to the greedy heuristic is more favourable using the enhanced objective function. This is as expected, since the enhanced objective function diminishes symmetry and fractionality for the branch-and-price matheuristic and has little or no influence on the performance of the greedy heuristic.

Negative net value gains can be seen in Table 7.7. These occur when infeasible trajectories in the original, manual plan have been removed from the plan and the net value drop by doing so has not been compensated for by the increase in net value from the greedy heuristic or matheuristic because of the limited processing time.

Comparing Tables 7.6 and 7.7 the branch-and-price matheuristic can perform approx. 11% more iterations in the first hour of processing using the enhanced objective function than when using the standard one. This is as expected since integer solutions may be found more quickly if there is less fractionality and symmetry. Less fractionality and symmetry has also been confirmed by investigating a large number of branch-and-bound trees using the enhanced and the standard objective functions, respectively.

The number of nodes and columns generated using the respective objective functions are approx. the same. This is also as expected, since the only difference in this context is that integer nodes are found faster using the enhanced objective function than using the standard.

Looking at Table 7.8, this table shows experimental results using longer processing times up to 48 h. Column (D) shows the net value gain in % of running the greedy heuristic for 36 h, column (E) for 48 h, column (F) the absolute difference in % between these two. A zero value of column (F) thus indicates that the convergence curve of the greedy heuristic has flattened out. As may be seen, the convergence curve for the greedy heuristic has flattened out for 9 of the 15 instances after 36 h of processing.

Next, column (G) shows the net value gain in % of the hybrid algorithm of first running the greedy heuristic for 36 h and then the branch-and-price upon that for 12 h. Column (H) shows the difference between columns (E) and (G) in %-points. As shown, the hybrid algorithm is able to outperform the greedy heuristic by a small margin for all instances except two. These two are characterised by not having their greedy heuristic convergence curve flattened out at 36 h. These differences in gains may seem small, and relative to the overall net value of the respective plans they are small, about 0.03%-points on average. However, in absolute economic terms of the objective, the best difference is that of 2014-03-31, 0.08%-points, equivalent to a net value gain of 2,300 DKK/day. Based on the experimental results presented here, the overall net value economic gain from using the hybrid algorithm in favour of just the greedy algorithm amounts to approx. 450,000 DKK/year, a somewhat substantial sum.

Column (I) shows the difference relative to the gain achieved by the greedy heuristic at 36 h, i. e., the gain of the hybrid algorithm relative to the gain of the greedy heuristic. As may be seen, the hybrid algorithm is able to achieve an additional gain of approx. 1.3 % in 48 h total, relative to the gain already achieved at 36 h by the greedy heuristic.

7.7 Discussion

Section 7.7.1 discusses the convergence characteristics of the matheuristic. Properties regarding symmetry and fractionality are treated in Sections 7.7.2 and 7.7.3.

7.7.1 Branch-and-Price Matheuristic Convergence

Figure 7.1 shows the effect on varying the model parameter k for the branch-and-price matheuristic model with stopping criterion “stop at first integer”. As may be seen the best time effectiveness for the 6 h processing time frame shown is given for $k = 6$, with time effectiveness decreasing with increasing value of $k > 6$.

It is interesting to note that in Chapter 5 the best value of k for the greedy heuristic was found at $k = 3$, while for the matheuristic the best value found here is 6. This underlines the fundamental differences between the two methods with regard to e. g., greediness.

Figure 7.1 suggests that, for best time efficiency, k should be chosen according to how much time is available for processing. If there is less than 1 h available, a value of $k = 3$ should be chosen, whereas higher values of k may be chosen if more processing time is available.

Experiments of varying k for the branch-and-price matheuristic for rolling stock plans that have been preprocessed with the greedy heuristic for 24 h or longer have been conducted. Under these conditions, the results showed that time effectiveness is small and with little variation between different settings of $k \in \{6, 7, 8, 9, 10, 11, 12\}$. This may suggest that the selection criteria by which to choose the k train unit trajectories is more important than the value of k itself under these conditions. Train unit trajectories are currently chosen at random.

Figure 7.2 shows the variation of $n = 8$ branch-and-price matheuristic runs with regard to median and average values, and cumulative probability of getting a solution with a value less than the given gained net value.

7.7.2 Symmetry in the Mixed Integer Linear Program

Symmetry is a highly undesired property in a branch-and-bound context [43, 84]. Symmetry occurs when a multitude of different feasible solutions to an optimisation problem have equivalent objective values. This may lead to a multitude of equivalent solutions that each needs to be evaluated even after an optimal solutions is found. In the current matheuristic integrated rolling stock planning model, symmetry is undesired because it prevents the pruning of the branch-and-bound tree. The symmetry in the mixed integer linear program is present as a result of the following conditions:

1. **Named train units:** The change in objective value for assigning a given train unit to a given arc in the space-time graph is not dependent on the individual, named train unit itself, but dependent on its train unit type, i. e., dependent on how many seats can be

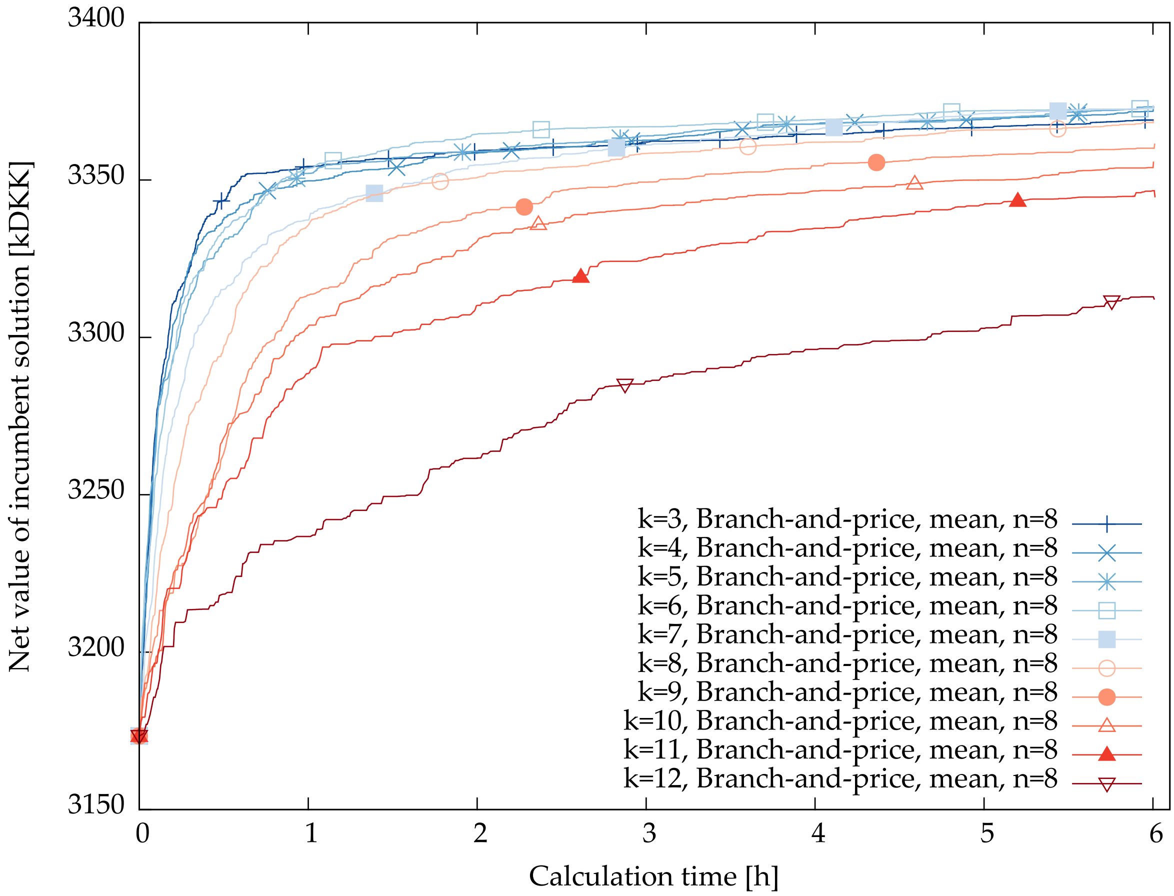

Figure 7.1: Convergence diagram for different values of k for the branch-and-price matheuristic for runs with stopping criterion "stop at first integer", i. e., stop after one depth-first dive into the branch-and-bound tree, n is the number of test runs per instance. Data instance is 2014-03-31.

supplied and at what cost. This is a cause of symmetry, in that two named train units of the same train unit type can be interchanged (partially or completely) in solutions without any change in objective value;

2. **Parallel tracks:** Multiple, in space and time parallel depot tracks, side tracks or platform tracks belonging to the same station and having available space at the same time. If the net value is defined as being equal for these tracks, this leads to many possible train unit trajectories through the space-time graph, all having the same net value. Strictly speaking, the conditions that prevail in the real-world data instances do not produce symmetry, but what could be called *quasi-symmetry*: If there are other train units parked e. g., at the mentioned parallel depot tracks, these train units may in some cases prevent the movement of others, meaning that symmetric trajectories will not be generated. However, this distinction only makes it harder to avoid the symmetry since the symmetric solutions cannot easily be identified and ignored;

3. **Revenue positioning:** The positioning of train units performed using revenue train services that already have train units assigned to them to supply the demanded number of seats. Since the seat demand has already been met, no additional benefit will be achieved by assigning additional train units to these revenue train services. This leads to a multitude of possible trajectories to choose from for the revenue positioning, all having the same net value, since the cost is the same;

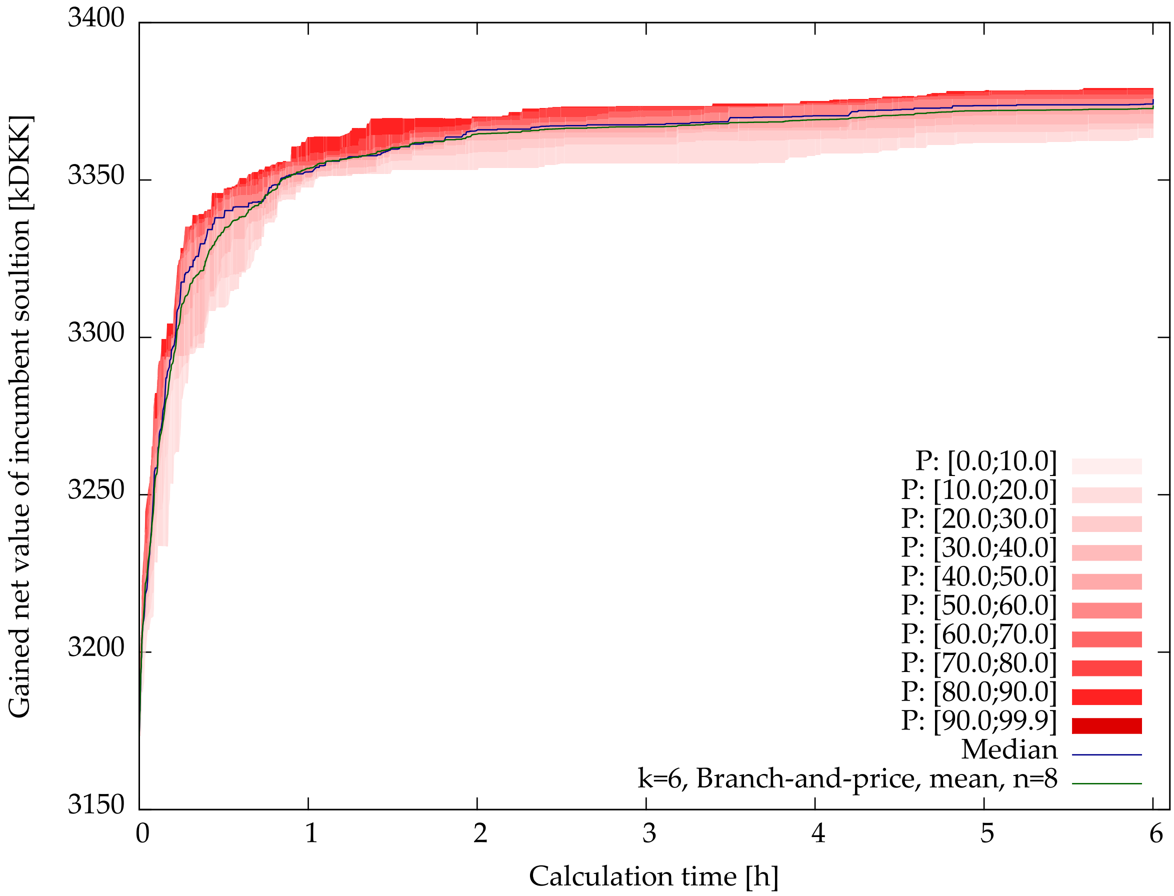

Figure 7.2: A convergence diagram showing the mean value and the median of the gain for $n = 8$ test runs with the given parameter settings. Also shown is the cumulative probability P in percent of getting a solution with a value less than a given gained net value. Data instance is 2014-03-31.

4. **Non-revenue positioning:** The positioning of train units using non-revenue train services. Since all of the non-revenue train services have the same cost going from A to B this may also lead to symmetric train unit trajectories. Luckily, the timetable data instances used only contain few non-revenue train services. However, as mentioned in Section 5.8 a method for creating more, or even all, relevant, non-revenue train services should be considered in the future, at which time symmetry will need to be considered.

In the branch-and-price matheuristic model, the symmetry type mentioned as item 1 in the list above is addressed by the flow branching scheme. Since flow branching is branching on arc flow, it does not distinguish between symmetric solutions performed by different train units. This way, this type of problem-inherent symmetry does not affect branching and pruning when the flow branching scheme is invoked, however, it is not addressed in the constraint branching scheme. Nevertheless, the latter does not seem to be problematic, since the constraint branching scheme is only seldom invoked.

The symmetry types mentioned as items 2 to 4 in the list above are addressed by the robustness and depot utilisation enhancements in the enhanced objective function already described in Section 7.2.1. The enhanced objective function makes problems with a setting of $k > 3$ generally tractable, problems that were otherwise not generally tractable due to branch-and-bound trees of excessive size.

7.7.3 Fractionality of Solutions to the LP-Relaxed Restricted Master Problem

Fractionality of the restricted master problem solution is another undesired property in a branch-and-price context. Based on an analysis of a large number of branch-and-bound trees and the correspondingly occurring fractional flow in the nodes, we have observed that fractionality occurs especially when there is great variation in the seat demand on a train service sequence. This leads to the splitting up of the should-be single train unit trajectory into a multitude of fractional train unit trajectories. The fractional train unit trajectories are being generated so that the seat demand may be met exactly as it occurs, as it were by a lot of smaller, fractional train units. The analysis was conducted using the visualisation tools described in Appendix A.4 on page 166.

The problem of fractionality is addressed by the robustness enhancements of the enhanced objective function already described in Section 7.2.1. The analysis of a large number of fractional flow diagrams (see Appendix A.4.2 on page 170) has shown that the fractionality of the restricted master problem solution diminishes using the enhanced objective function.

7.8 Conclusions and Further Research

An integrated rolling stock planning model based on matheuristics has been designed, implemented and tested. It has been shown that the matheuristic model can take into account all railway-specific requirements, while at the same time handling the vast majority of requirements in the optimisation part of the algorithm. Used in conjunction with the greedy heuristic from Chapter 5 the two methods can achieve a small objective value gain, not achievable using the individual methods by themselves. However, the implementation effort to reach this small net value gain is substantial: 4,000 extra lines of highly complex code to produce an average extra net value gain of 0.03%-points.

For these reasons and others, the current state of the integrated matheuristic model warrants further research. Future research may look into the effect of adding flow branching by train unit type, rather than for all train unit types at the same time. Flow branching by train unit type would be a branching scheme somewhere in-between the described flow branching and constraint branching schemes, and would eliminate the need for the two existing branching schemes.

Future research may also look into a branching scheme on depot drivers, i. e., on Equation (7.10) on page 118. This scheme would be analogous to the flow branching scheme described in 7.4.2 and in [110], however, it would force the sum of covering variables to ultimately become integer, thus working on the variables of the problem in an aggregated manner.

Needless to say, there is of course a multitude of other low-level branching strategies into which further research may be conducted in order to improve the time effectiveness of the proposed branch-and-price matheuristic.

With regard to high-level branching strategies, [6] review *pseudo cost branching, strong branching,* hybrids thereof and propose *reliability branching*. Based on these reviews, we have conducted experiments using an alternative, high-level branching strategy with strong branching. Experiments for both full and partial strong branching were conducted. Examples are shown in the appendix in Figures A.11 and A.12 on page 168. These experiments show that, in our case, while full strong branching branch-and-bound trees are generally smaller, more processing time is consumed than using the normal non-strong branching strategy. This is because, all in all, more nodes need to be processed. For $k = 5$ for the instance 2012-10-19, 6%

less nodes were observed in the branch-and-bound trees, while 27% more nodes needed to be processed, yielding a 24% higher processing time for the strong branching strategy than for the normal non-strong strategy. Choosing a partial strong branching strategy did not improve algorithm performance beyond the normal non-strong strategy. Based on these experimental results it may be assumed that both full and partial strong branching may be more useful in a context of one deep branch-and-bound tree than in a matheuristic context with lots of (hopefully) shallow trees. Further research may uncover if variations of the branching strategies mentioned in the start of the paragraph may yield better algorithm performance.

[83, 84] deal with ways to handle symmetry, including *orbitopal fixing* and *orbital branching*. Further research may be conducted into using these methodologies for handling the symmetry in the current problem. Moreover, a heuristic could be devised working somewhat along the same principles as orbital branching by way of dynamically detecting when symmetry is occurring and consequently performing a much more aggressive variable fixing on the zero-branch. This would also be somewhat similar to performing a *beam search* of the branch-and-bound tree (see [97] for a review and a machine scheduling application).

Further research should be conducted into better ways of handling the covered variable $g(a)$ in the current formulation or into ways to not include it at all. Especially the fractionality of this variable is considered problematic.

Especially for high values of k, a *tailing off effect* presumed due to degeneracy is observed in the column generation step of the model. A lot of iterations are needed to prove optimality without any change in objective value. Further research into the implementation of constraint aggregation [98, 49, 48], stabilisation [45, 47] and row-reduced column generation [44] may be conducted to alleviate this issue.

As the branch-and-price algorithm progresses, more and more columns are added to the restricted master problem in the column generation process of each node in the branch-and-bound tree. For this reason, the processing time for solving the restricted master problem in the individual iteration increases with the number of nodes processed. Future research may be conducted into whether it may prove useful to remove columns that have not been in the basis for some time.

The branch-and-price matheuristic model has been analysed using a performance profiler when run. Profiling has showed that approx. 70% of the CPU time is used by the CPLEX solver, whereas 5% is used by the resource constrained shortest path finding algorithm with special side constraints. The remaining time is used by a multitude of other processes of the matheuristic model, the vast majority of which each use less than 0.5 % of the CPU time. Based on these findings it is seems reasonable to suggest that future research into improving branch-and-price algorithm performance should be directed at limiting the number of nodes to be processed by the solver in the branch-and-bound tree or into means to make the solver be able to solve each node faster as already mentioned above.

More general ideas for future research are presented in Chapter 9.

Part III

Perspectives on Integrated Rolling Stock Planning

Chapter 8

Discussion

This chapter discusses the results gained and the lessons learned from the five different rolling stock planning models developed for this thesis. The discussion takes its offset in the industrial and scientific goals described in Section 1.3 on page 18. The general characteristics of the implemented models are compared in Section 8.1. In Section 8.2 some probabilistic observations regarding the convergence characteristics of the greedy heuristic and the branch-and-price heuristic are discussed. Finally, the main conclusions for the design, implementation, test and use of the models are given in Section 8.3.

8.1 Comparison of Model Characteristics

To which degree the different models implement the railway-specific requirements described in Chapter 3 is shown in Table 4.1 on page 61. Other characteristics of the models are shown in Table 4.2 on page 62.

The greedy heuristic model implements all of the railway-specific requirements. The branch-and-price matheuristic implements all requirements, but some requirements are not handled in the optimisation part, only in the heuristic part of the algorithm. The upper bound models implement less requirements.

Greedyness can be a cause of models yielding suboptimal solutions. Greedyness is present in the greedy heuristic (thus the name), other models do not exhibit any aspect of greediness at all.

k-optimality indicates to which degree a model is solved to optimality for the k selected train unit trajectories in each iteration (for the heuristic and matheuristic models) or in each model run (for the upper bound calculation models). The upper bound calculation models are all solved to optimality. The branch-and-price matheuristic can be solved to optimality in each iteration, however, this is not time-effective in a matheuristic context, for which reason the branch-and-bound tree search is terminated as soon as the first integer solution is found. The latter yields better time efficiency. The greedy heuristic does not solve each iteration to k-optimality, since by its sequential nature, the firstly found train unit trajectory is the best for itself, but not necessarily in conjunction with the $k - 1$ other train unit trajectories.

By providing an existing solution (either feasible or in some cases also infeasible), models can be hotstarted for better model performance. All models except the upper bound calculation model A2 can be hotstarted with a rolling stock plan containing infeasibilities in the form of uncovered revenue train services. All models except the upper bound calculation models A4 and B10 can be hotstarted with a plan containing infeasibilities in the form of incomplete train unit trajectory data.

As described in Section 7.7.2 on page 135, the branch-and-price matheuristic model exposes to four types of symmetry: In relation to named train units, parallel depot tracks, revenue and non-revenue positioning. Only the first symmetry type is handled by the branching scheme, the other types may be alleviated using an enhanced objective function. Using the enhanced objective function the branch-and-price matheuristic is able to find 11% more integer solutions by time than using the standard, non-enhanced objective function. Upper bound calculation models A4 and B10 also expose the latter three types of symmetry, and none of the types are handled by the models. Symmetry is considered a major challenge for the time-effective performance of these models. No symmetry has been identified for the remaining models.

Due to the formulation, the upper bound calculation model A4 exhibits a very high model-variable cardinality. As such, it typically contains in the order of 130,000 variables, whereas the A2 model only typically contains 6,500 and the B10 model 21,000 variables. The variable cardinality of the branch-and-price matheuristic models is strongly dependent on the parameter k, and other model parameters, but typically lies below a few thousand. The cardinality for the A4 upper bound calculation model is regarded a performance challenge.

The term *iteration effectiveness* is used to denote how well with regard to objective value gain a model performs in each iteration (or model run for the upper bound calculation models). *Time effectiveness* is how much objective value gain a model can achieve by time.

Iteration effectiveness is high for all upper bound calculation models and the branch-and-price matheuristic since these are solved to optimality (or near-optimality when the search is abandoned after first integer solution found). The greedy heuristic exposes a much lower iteration effectiveness since many iterations are rejected due to no objective value increase. In 15 experimental runs each of 1 h, using a the standard objective function, the iteration effectiveness of the branch-and-price matheuristic was a factor of 660 better than that of the greedy heuristic.

However, the greedy heuristic model is by far the fastest of all the models. Its feature of low iteration effectiveness is thus counteracted by speed, making the greedy heuristic model highly time-efficient. The time effectiveness of the greedy heuristic is a 2.8 factor better than that of the branch-and-price matheuristic on average for the 15, 1 h experimental runs.

The simple upper bound calculation model A2 has a similar high time effectiveness relative to the other upper bound calculations models, however, this is achieved at a lower degree of requirements integration.

8.2 Heuristic and Matheuristic Algorithm Convergence

An interesting difference between the greedy heuristic and the branch-and-price matheuristic (when run to optimality in each iteration) is the value to which the models converge after many iterations.

The greedy heuristic converges to an objective value at which no further gain can be achieved by removing k number of train unit trajectories, generating new ones and reinserting them. New train unit trajectories are generated greedily and sequentially.

The branch-and-price matheuristic converges to an objective value at which no k train unit trajectories can be selected and changed so as to achieve an objective value gain. As such, in theory, if it was tractable to solve an instance of the integrated rolling stock planning problem with $k = |U|$ using the branch-and-price matheuristic, this would yield the global optimal solution to the problem (provided that no train unit order conflicts would arise).

The justification for using both the greedy heuristic and the branch-and-price matheuristic as means of improving a rolling stock plan lies in that the convergence values of both methodologies are relatively tight lower bounds on the global optimal objective value. As demonstrated in Chapter 6, this is a fair assumption.

In the numerical experiments, the k number of train units U^* to select for modification is chosen at random. However, there is a finite number of ways k number of train units can be selected from the set of all train units U for the respective methodologies.

The greedy heuristic uses a sequential approach in which the selection order matters. Thus the number of ways train k train unit trajectories can be chosen form the total number of train units $|U|$ is given by the partial, k-permutation $P(|U|, k)$, (8.1).

$$P(|U|, k) = \frac{|U|!}{(|U| - k)!} \tag{8.1}$$

For the branch-and-price matheuristic, order does not matter (if each iteration is solved to optimality). Thus, the number of ways train k train unit trajectories can be chosen is given by the k-combination, $C(|U|, k)$ (8.2).

$$C(|U|, k) = \binom{|U|}{k} = \frac{|U|!}{k!(|U| - k)!} \tag{8.2}$$

If, contrarily to the numerical experiments conducted, the k number of train unit trajectories were to be chosen one-by-one, a complete enumeration of all possible ways could be conducted in the number of iterations equal to the partial, k-permutation $P(|U|, k)$ for the case of the greedy heuristic, and for the k-combination $C(|U|, k)$ for the case of the branch-and-price matheuristic. As such, in a one-by-one selection scheme, the partial, k-permutation is a lower bound on the number of iterations needed to ensure the convergence value has been reached for the greedy heuristic, whereas the k-combination is a lower bound for the branch-and-price matheuristic. The k-permutation and k-combination are lower bounds since these represent the maximum number of iterations that has to be conducted with no gain to prove that the convergence value has been reached. Table 8.1 shows numerical values for these combinatorial characteristics.

It is interesting to note that the lower bound on the number of iterations needed to reach the level of convergence for the greedy heuristic is $k!$ higher that of the branch-and-price matheuristic. As such it seems reasonable to assume that the convergence value may be reached in fewer iterations by the branch-and-price matheuristic in practice. Furthermore, it is reasonable to assume that convergence value of the branch-and-price matheuristic is at least as high as that of the greedy heuristic, since the objective value of the branch-and-price matheuristic is always greater than or equal to the greedy heuristic.

This underlines the primary properties of the respective methodologies looking at the individual, heuristic or matheuristic iteration: The greedy heuristic is “really fast but often bad” and branch-and-price matheuristic is “really good but always slow”. However, the processing speed of the greedy heuristic is by way the more important one of the properties, leading to its higher time efficiency. For a quantitative comparison, refer to Section 7.6.2 on page 133.

Based on the values in the Table 8.1 it is considered tractable to run the greedy heuristic until its convergence value for $k = 3$ has been reached. Based on the experiments from Chapter 7, it would take approx. 25 h to prove that the convergence level has been reached for the greedy heuristic with $k = 3$, if the convergence level was reached. Due to its much longer processing time, the branch-and-price matheuristic is only considered practically tractable to run until its convergence value for $k = 2$. For lack of time resources neither has been attempted here.

Another consequence of the difference in processing time between the greedy heuristic and the branch-and-price matheuristic arises from the number of iterations that can be performed for a given time period. Each iteration selects a combination of k train unit trajectories, and since the greedy heuristic can perform many iterations, it can try out many different combinations of

Table 8.1: Overview of combinatorial characteristics for the selection of k number of train units to modify from the set of U train units, with $|U| = 117$. The number of ways k train units can be selected in the greedy heuristic is the partial, k-permutation $P(|U|, k)$ and in the branch-and-price matheuristic this is the k-combination $C(|U|, k)$. The quotient between $P(|U|, k)$ and $C(|U|, k)$ is $k!$, as shown in the last column.

k	$P(\|U\|, k)$	$C(\|U\|, k)$	$k!$
1	117	117	1
2	13,572	6,786	2
3	1,560,780	260,130	6
4	177,928,920	7,413,705	24
5	20,105,967,960	167,549,733	120
6	2,251,868,411,520	3,127,595,016	720
7	249,957,393,678,720	49,594,720,968	5040

the k train unit trajectories. Since many combinations can be tried out, the greedy heuristic eventually finds those combinations that yield an increase in objective value. The branch-and-price matheuristic, on the other hand, because it is much slower, cannot try out as many combinations. As such it seems reasonable to assume that using a “more intelligent” selection scheme rather than just at random may improve the performance of the branch-and-price matheuristic. Future research may investigate this assumption.

8.3 Main Conclusions

The following main conclusions relate to the industrial and scientific goals described in Section 1.3 on page 18.

It has been shown in Chapter 5 that it is possible to design and build an integrated rolling stock planning model taking into account all the railway-specific requirements of DSB S-tog. The model is implemented with a greedy heuristic and uses the novel *(train) unit order conservation* principle, implemented as special side constraints to a resource constrained shortest path algorithm. The fully functional, implemented integrated rolling stock planning model has been tested extensively on 15 real-world, manually constructed rolling stock plan data instances. When run on these instances, the greedy heuristic achieves an average economic gain of approx. 2% with processing times in all cases less than 1 hour 20 minutes. Moreover, the greedy heuristic can make feasible typically infeasible rolling stock plans in a matter of minutes of processing time.

In Chapter 6 three different net value upper bound calculation models have been designed, implemented and tested. The net value upper bound calculation models implement the railway-specific requirements to a varying degree and consequently expose different properties regarding tightness of bounds and processing times. The net value upper bound model having the highest degree of requirements integration adheres to 47% of the requirements by count. Using this tightest net value upper bound calculation model, it is shown that the greedy heuristic is able to gain approx. 1/3 of the relative gap between the net value of the original, manual plans and the net value upper bound. This gain by the greedy heuristic is regarded substantial, since the difference in requirements integration by count is 53% between the greedy heuristic and the upper bound calculation model with the tightest upper bound. Moreover, it is shown, that in most cases, the net value of the original, manual plans already lie close to the upper bound. This is in particular the case for weekend instances that are typically planned manually with less overhead for robustness than weekday plans.

Finally, in Chapter 7 a branch-and-price matheuristic integrated rolling stock planning model is designed, implemented and tested. It is shown that this type of model is able to adhere to all railway-specific requirements while still handling the vast majority of requirement in the optimisation part of the matheuristic algorithm. The matheuristic can solve small instances to optimality. Used in conjunction with the greedy heuristic as a hybrid, it can produce solutions with a small extra gain, a gain not achievable using the individual methods by themselves.

The cost of the rolling stock operation at DSB S-tog lies in the hundreds of million DKK per year. Based on the experiments conducted, the potential benefit of a real-world application of the models to DSB S-tog is estimated to be in the order of several million DKK per year. Moreover, a substantial benefit can be achieved by way of automating the current, manual planning procedures, enabling planners to invest more creativity and meticulousness into the planning process, since being liberated from manual planning procedures. For the reasons mentioned, DSB S-tog is eager to proceed with the real-world application of the models developed in this thesis.

Chapter 9

Future Research

Naturally, the research conducted in relation to this thesis has uncovered many new and interesting directions for further research. Firstly, in Section 9.1 we look at alternative methodologies to solving the integrated rolling stock planning problem. Secondly, a topic for further research involves the prospect of integrating the processes circulation planning and train unit dispatching, processes that are now treated separately at DSB S-tog. A brief outlook to this is presented in Section 9.2. Lastly, in Section 9.3 we are asking ourselves the fundamental question if we have solved the right problem with the integrated rolling stock planning model and its corresponding railway-specific requirements.

9.1 Alternative Solution Methodologies

9.1.1 Flow Decomposition Theorem Applied to Arc Flow Upper Bound Model Solution

As reported in Chapter 5, the implemented greedy heuristic integrated rolling stock planning model works best as an improvement heuristic on an existing rolling stock plan, as opposed to as a construction heuristic on an empty plan.

An idea to remedy this would be to use any flow oriented upper bound model from Chapter 6 to update the time space-time graph with limits on the number of train units by type allowed on the arcs. The number of train units by type on each arc in the found upper bound model solution is used as the limit. The *flow decomposition theorem* [8, Chapter 3] states that train unit trajectories can then be created that fulfil this flow. However, this would disregard the distance requirements of the train unit trajectories. Nevertheless, if we were to order the train units by ascending distance allowed before maintenance, and use the resource constrained shortest path to find new trajectories to each train unit in that order in the modified graph, chances are that we will only violate distance constraints occasionally. The flow limits can then be released and distance-infeasible trajectories can then be made feasible using the heuristic from Chapter 5.

9.1.2 More Accurate Submodels

The integrated rolling stock planning models described in this thesis operate on passenger seat demand figures measured by the weighing mechanism in the individual train units. These figures are processed using an advanced statistical prognosis model, see Section 3.4. In the rolling stock planning models proposed in this thesis, each train service has assigned to it the maximum dimensioning passenger count occurring along the entire train service. The dimensioning passenger count is corresponding to a given comfort level of the statistical prognosis model. In reality, however, the passenger count fluctuates a lot between individual stations. For this reason it may be worth looking into more accurate ways of quantifying the benefit term in the net value objective function, this by looking at the time and space distribution of passenger seat demand all along the entire individual train service, rather than just using the maximum passenger seat demand value for the train service.

Moreover, the energy consumption calculation used in the models described in this thesis is very simple. Since the energy costs represent roughly 25% of the total operational costs at DSB S-tog, it may be worth looking into methods to quantify the energy consumption more accurately, e. g., [74, 63].

9.1.3 Adaptive Large Neighbourhood Search Metaheuristic

As described primarily in Chapters 5 and 7, the integrated rolling stock planning models implemented for this thesis have a choice of heuristic methodologies and parameters with innumerable combinations of settings. The most prominent parameter is the number of train unit trajectories to choose and modify, k, and the influence of this parameter on model performance has been analysed.

However, other parameters or methodologies may not have been investigated as thoroughly. Most important among these is perhaps the selection methodology for choosing the k train unit trajectories for modification. It may well turn out that the selection methodology can be improved, e. g., by using tournament selection in favour of the current random selection. A simple tournament selection method would be to select $k + l$ train unit trajectories, and then choosing only the k train unit trajectories with the lowest net value for modification. A more advanced tournament selection method would be to select $k + l$ train unit trajectories and then choose the m train unit trajectories with the highest net value and the n ones with the lowest, with $k = m + n$. Clearly, a very large number combinations for selection methodologies and selection parameters k, l, m, n exist.

It may well turn out that one selection methodology and parameter combination is suitable early in the process of improving a rolling stock plan, whereas another may be the better suited later in the process. Apart from this timely variation, there may of course also be a variation from data instance to data instance. It may be difficult to adapt to these variations in a static scheme of preset parameter or methodology combinations.

The principle of the *adaptive large neighbourhood search (ALNS)* metaheuristic as proposed originally by [94] involves defining a number of competing heuristic methodologies (including their parameters settings). In each iteration of the ALNS metaheuristic a heuristic methodology is then chosen with a probability proportional to its historic performance. In this way the ALNS metaheuristic is able to adapt to different instance characteristics and to the variation in different heuristic methodologies effectiveness over time.

For this reason the ALNS metaheuristic seems well suited for implementation in the integrated heuristic or matheuristic rolling stock planning models proposed here. Other model parameters or heuristic methodologies than the ones mentioned here may also be calibrated adaptively using ALNS, including the hybrid interaction between the greedy heuristic described in Chapter 5 and the branch-and-price matheuristic one described in Chapter 7.

This idea may be taken to its logical extreme *hyper heuristics*, in which the adaptation can also include some aspect of memory of past solutions for different data instances like in [101].

This suggestion for further research may be taken as a general advice to think parameter calibration into model design from the beginning, rather than having to perform a lot of time consuming manual parameter calibration later.

The heuristic and matheuristic methods proposed in this thesis are well-suited to be implemented using parallel processing in a future ALNS metaheuristic framework. In a parallel setup, multiple threads may each operate their own heuristic/matheuristic instance with specific parameter settings (e. g., of k). Each thread would work on a separate instance of the rolling stock plan. Once any thread finds an improvement all other threads are halted. The improvement is then applied to the plan and new heuristic/matheuristic instances generated according to their historic performance in the ALNS framework, each working in their own thread with their own new copy of the rolling stock plan.

9.1.4 Proposal for A Generic Integrated Rolling Stock Planning Model

As demonstrated, primarily in Chapter 5, it is possible to build an integrated rolling stock planning model that adheres to all identified railway-specific requirements for the case of DSB S-tog and that performs well under realistic planning conditions.

Still, this model is specifically built to handle the precise requirements of only one suburban railway operator, DSB S-tog. While the main requirements (e. g., passenger demand, infrastructure, timetable, etc.) are more or less the same for all railway operators, other requirements (e. g., train control system movement rules) may well be very specific to DSB S-tog. The aim of this thesis has been to prove it is possible to build an integrated rolling stock planning model that works for the requirements given. For this reason, the developed integrated rolling stock planning model is to some degree specific. As such, it will probably require some effort to adapt the developed integrated rolling stock planning model to the requirements of other railway operators (including the non-suburban railway operation of DSB).

One way to alleviate this would be to generalise the concepts of the developed integrated rolling stock planning model so as to be able to accommodate a wider variety of railway-specific requirements. Such a generic integrated rolling stock planning model is envisioned to be highly component oriented, so as to let it adhere to new requirements by plugging in new components built separately and independently. For generalisation, the space-time graph may be constructed so as to handle user-defined resources on its arcs and vertices, resources that are consumed either by time or distance or in some other user-defined way. These resources may then be replenished in user-defined ways on arcs and/or vertices in the graph. Moreover, the resources may then be coupled to the objective function in a user defined way. For instance, the energy consumption could be modelled in this scheme, coupled to its cost. The wear and tear of the rolling stock may be modelled as a generic maintenance resource, incurred by distance, coupled to its cost, and replenished at the workshop. The replenishment would then provide the service distance that the train unit can consume before having to go into the workshop for maintenance again.

This generic concept would also make it easier to implement correctly a complicated objective function. The implementation of the objective function has been particularly error-prone in the developed integrated rolling stock planning models due to its highly complex nature (see e. g., Equation (7.1) on page 113), and due to the lack of an implementation methodology developed for generality, and therefore not sufficiently transparent.

A specific technology well suited to enable this generic concept is that of *lambda expressions* or *anonymous functions*, now available in many fourth-generation programming languages including Java, Python and C++.

One very important aspect when building a generic integrated rolling stock planning model is model performance. It may require a considerable effort to make a generic model run time-effectively in every configuration, much more than is required for a specific model as the one implemented, even though that effort has also been substantial.

On a note, it has also been a main development goal to implement the models proposed in this thesis component oriented, see Appendix C for details. However, some compromises have had to be made due to available time resources and in order to achieve a high-performance implementation.

9.2 Circulation Planning and Train Unit Dispatching Process Integration

The idea of splitting the process of rolling stock planning in circulation planning and train unit dispatching at DSB S-tog is probably historically rooted. At the time when planning was conducted using pencil and paper, this may well have been a very good idea, since the circulation plans may have been produced well in advance thus giving the planners the needed time to produce them. Also the structure of the planning can be kept so simple it is easy to conduct, and to inform the involved parties when done.

In the age of automatic planning tools, however, one should ask this question: *Does it make sense to make a detailed long term circulation plan stretching a whole month into the future, when reality demands changes to train unit dispatching only a few hours into the plan?*

The posit is that if the processes for both circulation planning and train unit dispatching may be automated and integrated into each other, a new, possibly simpler way of performing rolling stock planning may be conducted. In this new form there is no need for the abstraction of virtual train units and the planning horizon may be very short (perhaps even real time). As such, there will be no need for the manual dispatching step of assigning physical train units to virtual ones.

We propose the term *adaptive rolling stock planning* to describe this new process, Adaptive meaning that the rolling stock plans are produced when needed and continually adapted to the changes in prevailing conditions: Infrastructure breakdowns, technical problems with the train units, delays, etc. The general idea is to *do the best with what we have*.

In fact, recent research in related models for airline crew scheduling shows that the cost of a plan may be reduced with up to 9% if the *pairing* step is integrated with the *rostering* step [98]. The pairing aggregation step has been performed up until now to reduce complexity of the airline crew scheduling problem. The pairing step in airline crew scheduling deals with combining flight duties to form a *tour of duty* starting and ending at a given crew base. The tours of duty are specific for a given crew type and as such anonymous. In the following rostering step, actual crew members are assigned to the anonymous tours of duty, together with stand-by duties, training, time off, etc. When paring and rostering are integrated, the individual flight duties are assigned directly to the crew members.

For rolling stock planning, the analogy to an integrated crew scheduling problem would be to *directly* assign physical train units to individual train services, as opposed to constructing aggregated train services and then assigning these to the virtual train units, and after that assigning physical train units to the virtual ones.

Another aspect is the plan duration. Traditionally, rolling stock planning at DSB S-tog is conducted with a plan duration of 7 to 18 days, typically. Recent research in related topics [18] (covering hump yard sorting and scheduling), however, shows that there is only a little gain in producing a combined plan for a longer period as opposed to producing individual, daily plans for each of the days in the given period and letting the finishing conditions of each day form the starting conditions for the calculation of the next.

Speaking in analogies one could say that only a few problems are "bulldozed" into the future by doing so, and if disruptions are occurring anyway, which they are, the effect of making a plan with a long duration is negligible.

One potential difficulty in integrating rolling stock planning and train unit dispatching processes may occur if only parts of the different processes are automated. If the rolling stock planning process is automated and plans are produced on a very short time horizon, these plans may prove very difficult to handle manually in the case of a disruption, since the plans may not necessarily contain the structures built into the manual plans today making them easy to handle manually.

Investigations must show if the realisation of these ideas of integration (to what ever degree) may prove beneficial to DSB S-tog and other railway operators.

9.3 Are We Solving the Right Problem?

We have seen, primarily in Chapter 5, that we can solve the problem of rolling stock planning in an integrated manner, taking all of the railway-specific requirements into account. This may all be very well, but we also need to ask ourselves the question, then, if we have solved the right problem?

Maybe some of the requirements are so costly to adhere to that their justification as requirements is doubtful? This leads to the following question: *What are the costs of adhering to the different requirements?* Given an answer to this question, new questions arise: *Are we trying to solve a problem in the rolling stock planning department that would much better be solved elsewhere in the railway organisation?* Or: *Are we using operations research, optimisation and information technology to support planning processes that are designed to make manual planning easy?* Or: *Do we try to adhere to business rules that have lost their meaning, because planning need not be performed with pencil and paper any more?* In other words, are we utilising information technology to its full potential or are we using it to extend the life of manual planning?

In the following Sections 9.3.1 to 9.3.4 we look at different examples related to these questions.

An interesting perspective is the fact that the design and implementation of future integrated rolling stock planning models with fewer requirements would be much less complicated. However, it may not be known how this would influence the processing times of future models, since the solution space may increase with fewer requirements. Then again other methods may be used if the requirement set is different.

9.3.1 Flexible Space Distribution

At DSB S-tog, as described in Section 3.3 on page 39, when train units of type $\frac{1}{2}$ are used in the southern end of train compositions of type $1\frac{1}{2}$, there is no flexible space at this end of the train composition. This may lead to delays since passengers with bicycles may need extra time to find a carriage with flexible space to enter after the arrival of the train service. One could say that the cause for the delay is passenger behaviour, which again is caused by their lack of information. Note that delays of the mentioned kind do not occur due to lack of physical resources, but merely due to the fact that passengers lack the information of where to find these resources.

For DSB S-tog, this problem is presently solved by enforcing a business rule for rolling stock planning stating that train units of type $\frac{1}{2}$ must be in the northern end of train compositions of type $1\frac{1}{2}$. The problem is thus solved, not by providing passengers with the information as to where to find resources, but to evenly distribute resources so passengers need not look for them.

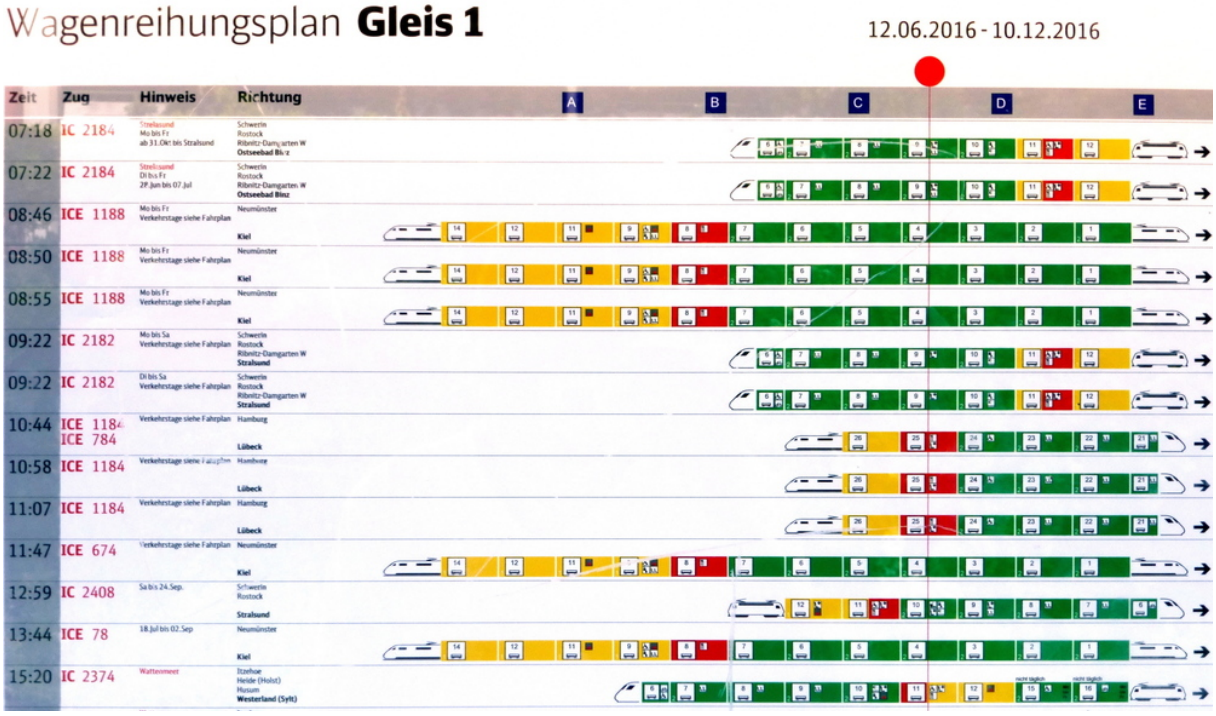

Figure 9.1: An example of paper based passenger information: A Deutsche Bahn *Wagenreihungsplan* poster from Hamburg-Harburg station, July 2016, showing the train compositions of the individual train services departing from a particular track, in this case track 1. These information posters are printed on paper and put up next to the track in question [42].

Figure 9.2: An example of a modern, electronic passenger information system: The new platform overhead information screens of Schweizerische Bundesbahnen (SBB) at Zürich HB station, February 2017. This is the electronic way of conveying the same information as in Figure 9.1. The difference to the paper poster is of course that this information can be updated instantly, as shown on the lower part of the screen where a track change is announced.

Apart from limiting the number of possible train compositions (and thereby the number of feasible train unit trajectories), this business rule makes it harder to fully utilise the available depot capacity, as described in detail in Appendix D.1.2.4.

As shown in Chapter 5 it is possible to construct feasible rolling stock plans that adhere to this business rule. However, this presumably comes at a cost. Further research should be conducted into quantifying the actual cost of this business rule, e. g., by scenario analysis, and comparing this cost to the delay cost it is supposed to limit.

In addition to this, alternative measures to limit this particular type of delay should be considered. It may well turn out to be beneficial to physically rebuild the train units of type $\frac{1}{2}$ to also feature a flexible space in the southern end.

Furthermore, an even better solution may be to promote the better use of the already existing electronic passenger information systems such as platform displays to inform about the train composition. This is already practised by some operators, as seen in Figure 9.2.

9.3.2 Platform Track Usage Rules

For DSB S-tog, as described in Section 3.1 on page 33, business rules stating which track to use for specific station, train service line and direction combinations are in place in order to make it easier for regular passengers to find the train service of their choice.

This, however, also comes with a price tag, paid for in the rolling stock plan, since depending on the track topology of a station this may limit the access to and from depot tracks. Recall that some depot tracks in some stations are only reachable from some platform tracks. This in turn may make it harder to perform the necessary train services or train shuntings so as to minimise the operating costs.

Research should be conducted into the quantification of the cost of this requirement, e. g., by scenario analysis. Following that, it should be examined if it would be better to omit this requirement from rolling stock planning all-together, and instead provide passengers with the information they need on a daily basis via the already existing passenger information systems on the stations.

9.3.3 Passenger-Unfriendly Train Shuntings

For DSB S-tog, as described in Section 3.1 on page 33, certain - one might call them “passenger-unfriendly” - train shuntings are disallowed: A business rule disallows to decouple a train composition into two parts and letting the part that is going to continue as a revenue train service depart before the other part has been shunted into the depot. This is to prevent passengers getting confused when parts of the train composition are in service, and other parts not. Again this has to do with passenger behaviour due to lack of information.

Just as for the case described in Section 9.3.1, apart from limiting the amount of possible train services and train shuntings, this business rule also makes it harder to fully utilise the available depot capacity, as described in detail in Appendix D.1.2.4 on page 194.

The train shunting operation currently disallowed would have to be performed after the revenue train service has departed and for this reason a depot driver may then not have time to perform another train shunting for the next train service if this train shunting is to occur before the next revenue train service departs.

Nevertheless, the price tag of this requirement should also be quantified and it should be examined if the departure procedures involving train drivers and depot drivers could be changed. Moreover, it should be examined if existing passenger information systems can be used to convey the necessary information to passengers, so as to omit this requirement from rolling stock planning all-together.

9.3.4 Cyclicity

The plan duration for the rolling stock plans treated in this thesis has been one day only. For this reason the work has not dealt with the requirement of cyclicity: That rolling stock plans should be the same (or at least similar) from day to day. However, this requirement is very much in effect at DSB S-tog and presumably also in the vast majority of all other railway operators world-wide.

The primary justification for the requirement is of course to make it easier for regular passengers to find the train service of their choice. Secondarily, railway operators have requirements like these in place in order to make it easy on their staff - some may even claim this to be the primary justification.

As described in Section 5.1.2 on page 64, [20, 21] solve the weekly rolling stock planning problem for the entire German ICE high speed fleet. This work is without question state-of-the-art operations research. However, we need to ask ourselves the question, if we should apply state-of-the-art operations research methods to solve problems with requirements that have a presumably very high price tag on them, rather than question the requirements themselves?

Figure 9.1 shows a graphic representation of one of the requirements to the work of [20, 21]. This is a Deutsche Bahn (DB) paper poster showing the train compositions for the train services departing from a given track. The requirement is that each particular train service should have the same train composition every day.

This requirement forces the railway operator to perform a "one size fits all" type of rolling stock planning, providing the same number of seats every day, regardless of passenger demand fluctuations. With this requirement in place, it is not possible for the railway operator to perform rolling stock planning based on the actual passenger prognosis for a particular day, and to accommodate passenger demand fluctuations by providing a varying number of seats.

The alternative would be to omit the cyclicity requirement from rolling stock planning and instead inform the passengers of the train composition in effect on the day of their travel using the existing passenger information systems, like platform displays. Some operators are already adapting to this, see Figure 9.2.

This is perhaps the prime example of how information conveyance technology may be much better at solving problems related to passenger behaviour than rolling stock planning will ever be. This underlines the importance of the question of where in an organisation a problem is best solved.

As may be apparent by the following question, other industries may be further ahead with regard to information conveyance technology: *Can you find a paper poster in any airport showing all the departures of a given airline, with departure time, destination airport, departure gate and aircraft type?* Those days are long gone. Then why do we, in the age of information technology, apparently still need paper poster type requirements in the railway industry?

Appendix

Appendix A

Visualisation Tools for Rolling Stock Planning

This appendix presents different visualisation tools for rolling stock planing. All but one of the tools have been specifically developed for the models presented in this thesis.

A generic tool for visualising timetable oriented data relating to rolling stock plans as space-time diagrams is presented in Appendix A.1. The circulation diagrams that can be printed with the existing, manual rolling stock planning system at DSB S-tog are treated in Appendix A.2. Next, the interactive visualisation tool, developed for the timetable and infrastructure data model (the space-time graph), is presented in Appendix A.3. Next, two interactive visualisation tools developed for the analysis of the progress of the branch-and-price algorithm are presented in Appendix A.4. Finally, an attempt to draw the causality of specific features regarding model architecture of the branch-and-price matheuristic model is shown in Appendix A.5.

These visualisation tools have played an essential role in the design, implementation and testing of the rolling stock planning models presented in this thesis. Without these visualisation tools, it would not have be possible to assert the correctness of the developed models, their results or to calibrate model parameters.

Apart from the already existing tool to draw circulation diagrams, the tools presented here have been implemented in Java 1.8. The tools all function by reading data from existing models and by writing diagrams to standard graphics formats using custom-made light-weight graphics objects. Standard open source tools are then used for rendering, navigation etc.

The visualisation tools have been developed strongly inspired by “Fundamental Principles of Analytical Design” proposed by Edward Tufte in [108]. As such, the tools have been developed to show *comparisons*, to show *causality, mechanism, structure* and to *explain*. Attempts have been made to show *multivariate* data with as many variables at a time as practically possible, while also *integrating evidence* by combining different types of *content:* Diagrams, numbers, and explanatory text together. Moreover, it has been a main goal that the visualisation tools provide actual *documentation* for what is shown, including metadata.

A.1 Space-Time Diagrams

In order to be able to visualise timetable oriented data relating to a rolling stock plan, a generic space-time diagram visualisation tool has been developed. Space-time diagrams (space-time graphs) are long known in the railway industry, first use for timetabling is attributed to French engineer Ibry before 1885 [79].

The developed space-time diagram visualisation tool accepts a timetable as key input, describing the movement of train services between key stations in space and time. Value data to be visualised in different colours on the train services may be continuous or discrete. The tool can construct corresponding colour ramps and legends based on user parameter input. See Figures A.1 to A.7 on pages 157–163 for examples of diagrams generated with the tool. In the landscape orientation, time is plotted on the x axis and space on the y axis. Data sets that may be visualised using this tool include data on:

- Seat demand (Figure A.1);
- Seat supply (Figure A.2);
- Seat surplus (Figures A.3 and A.4);
- Train composition demand (Figure A.5);
- Train composition supply (Figures A.6 and A.7).

Moreover, the individual trajectory of each train unit may also be shown on a number of these space-time diagrams. Data sets not suited for drawing in this type of diagram include data on cross-line, non-revenue train services and data related to individual depot tracks.

In the examples on the following pages the page layout is such that when reading this document in a portable document format (PDF) reader, the space-time diagrams are positioned at the exact same position on each page for easy comparison by page-forward, page-backward movements. Figures A.3 and A.4 and Figures A.6 and A.7, show seat surplus and train composition supply diagrams before modification (with the greedy heuristic from Chapter 5) and after modification, respectively. As may be seen, the greedy heuristic is able to reduce seat surplus and seat deficit by changing the train composition supply.

The space-time diagram tool writes data to a scalable vector graphics (SVG) and standard open source SVG rendering tools are used to render the diagrams.

These types of diagrams have been widely used to visualise and analyse the different data instances and the results of the heuristic and matheuristic models.

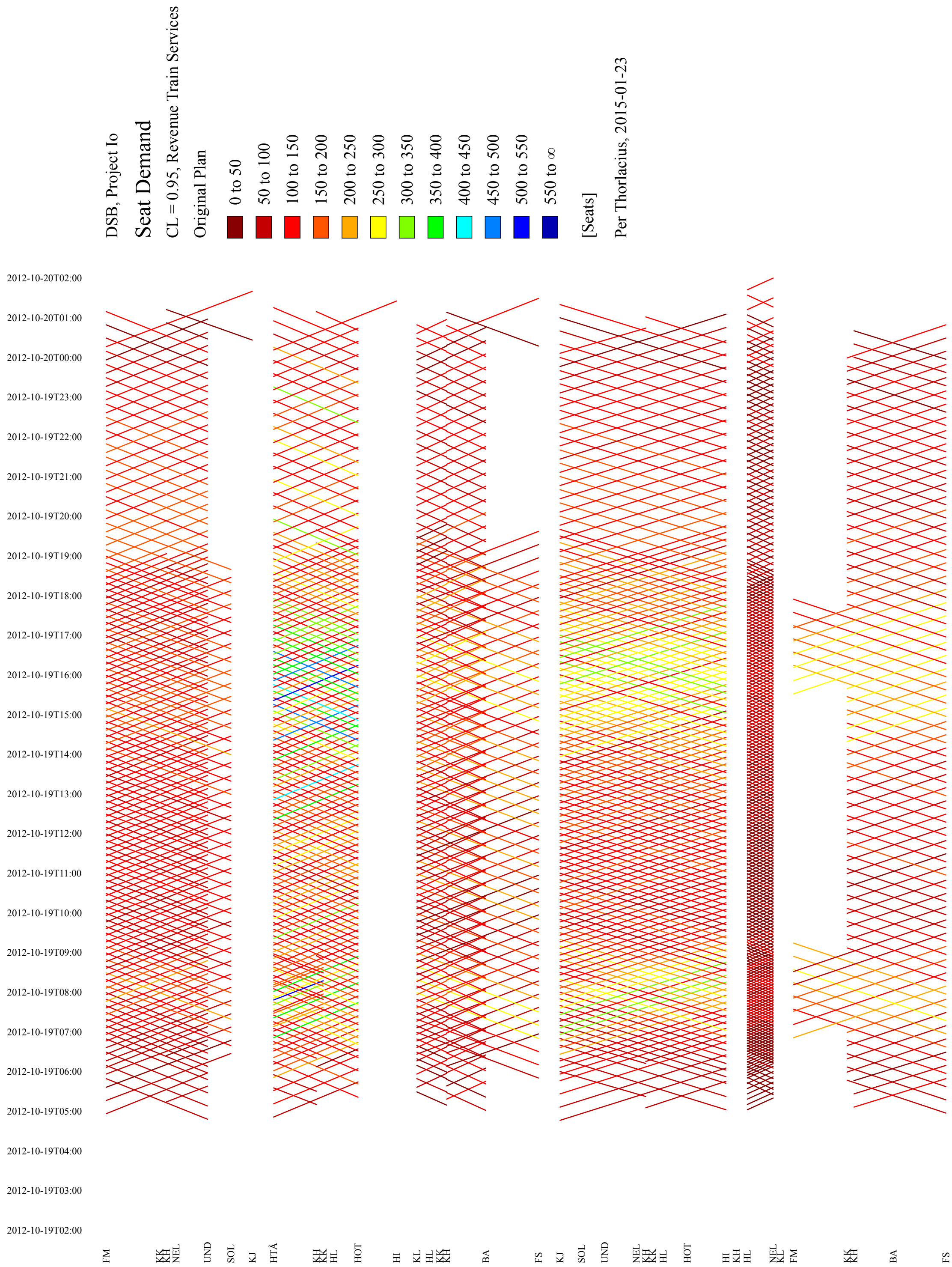

Figure A.1: Example of a seat demand space-time diagram. The data instance shown is 2012-10-19 and refer to a comfort level (CL) of 95% in the statistical prognosis model.

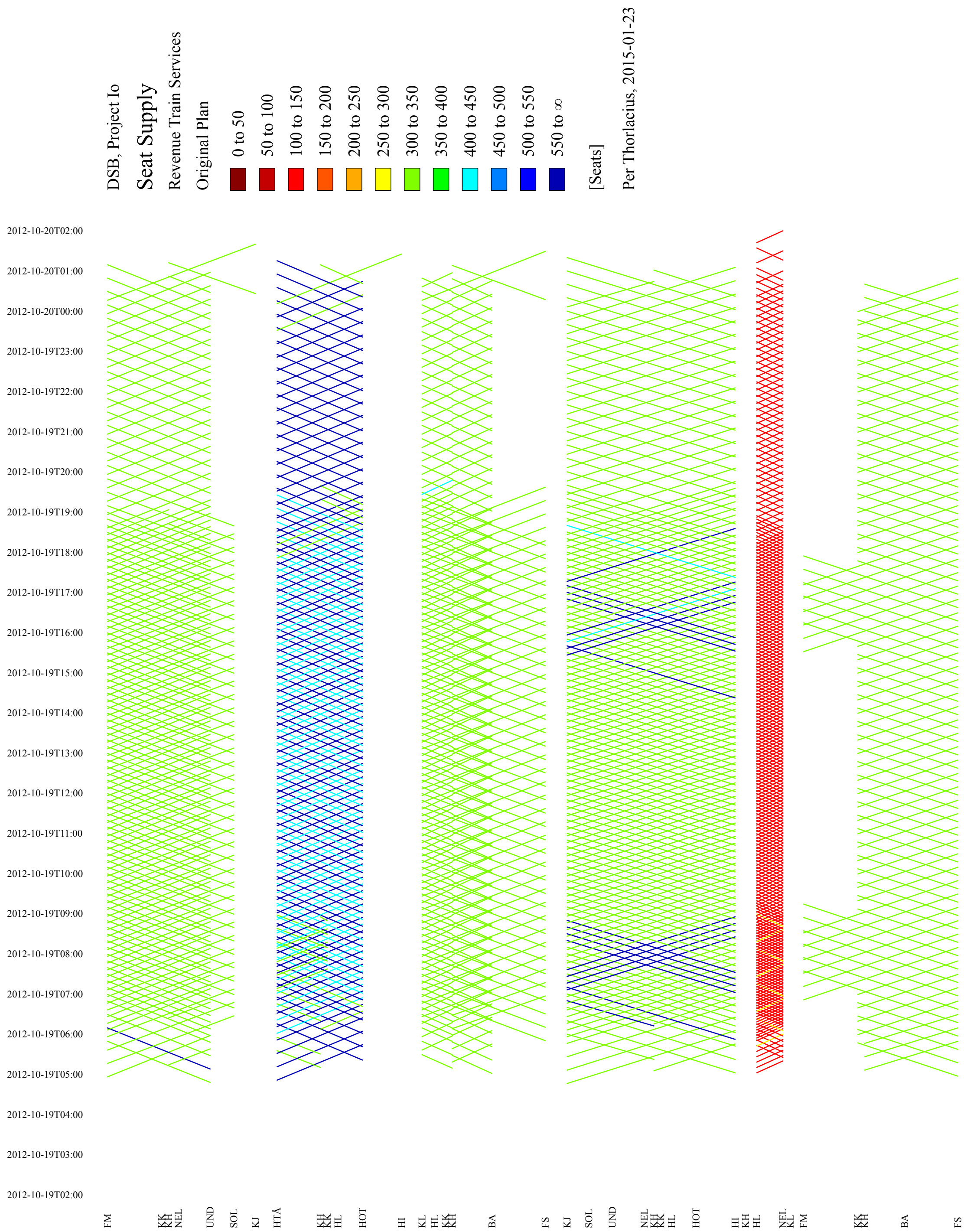

Figure A.2: Example of a seat supply space-time diagram. The data instance shown is 2012-10-19.

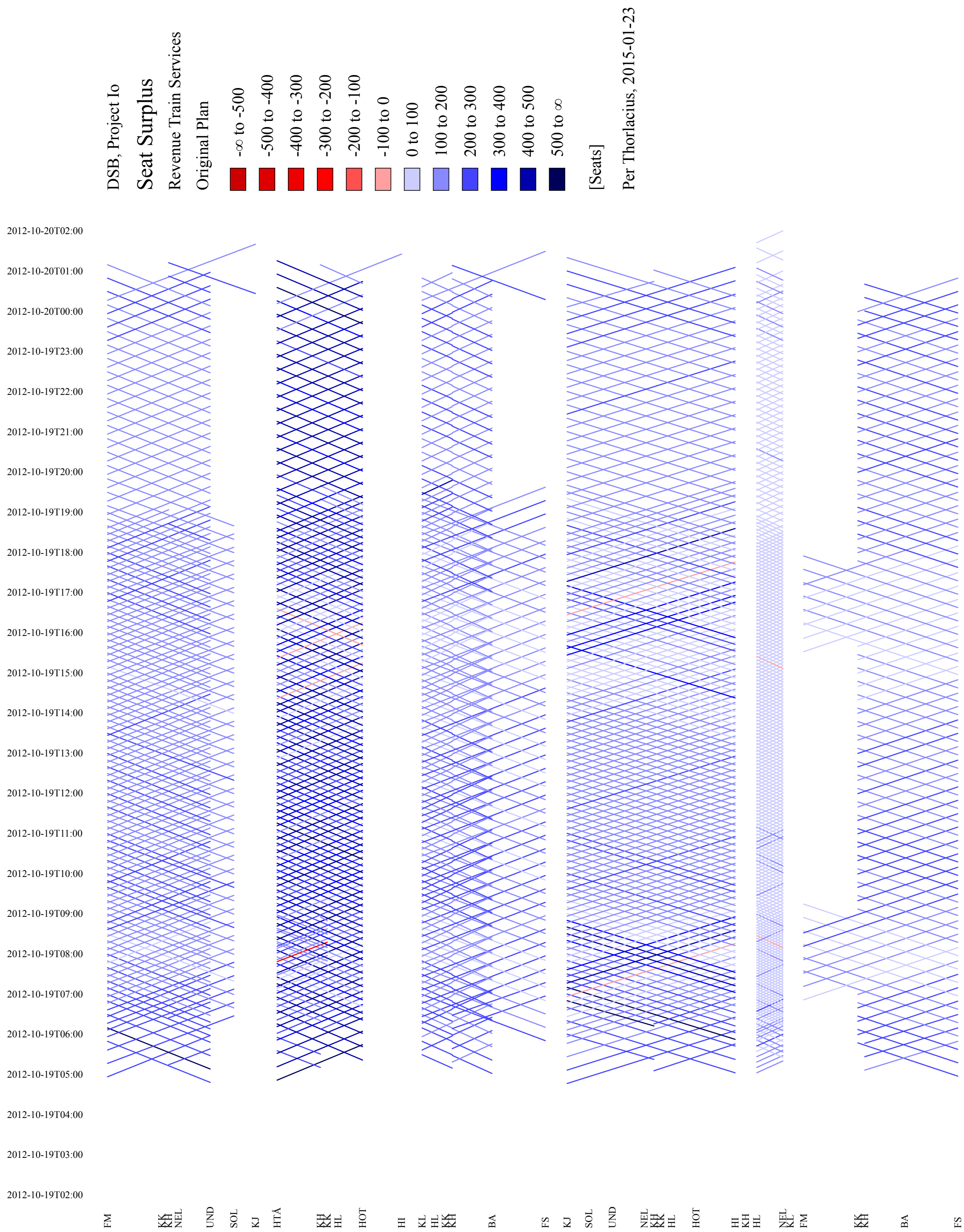

Figure A.3: Example of a seat surplus space-time diagram for an original plan. The data instance shown is 2012-10-19. Demand data refer to a comfort level (CL) of 95%.

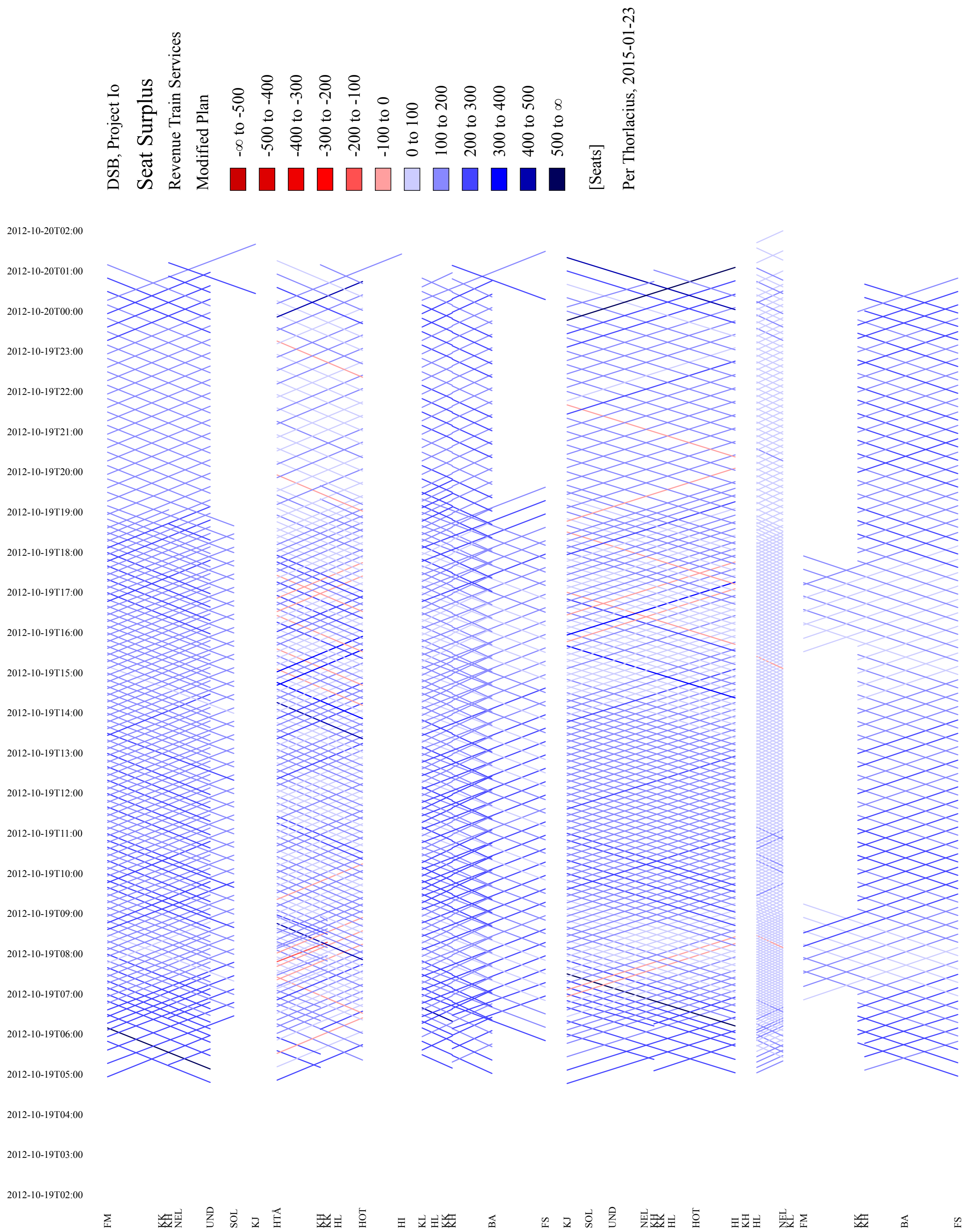

Figure A.4: Example of a seat surplus space-time diagram for a modified plan. The data instance shown is 2012-10-19. Demand data refer to a comfort level (CL) of 95%.

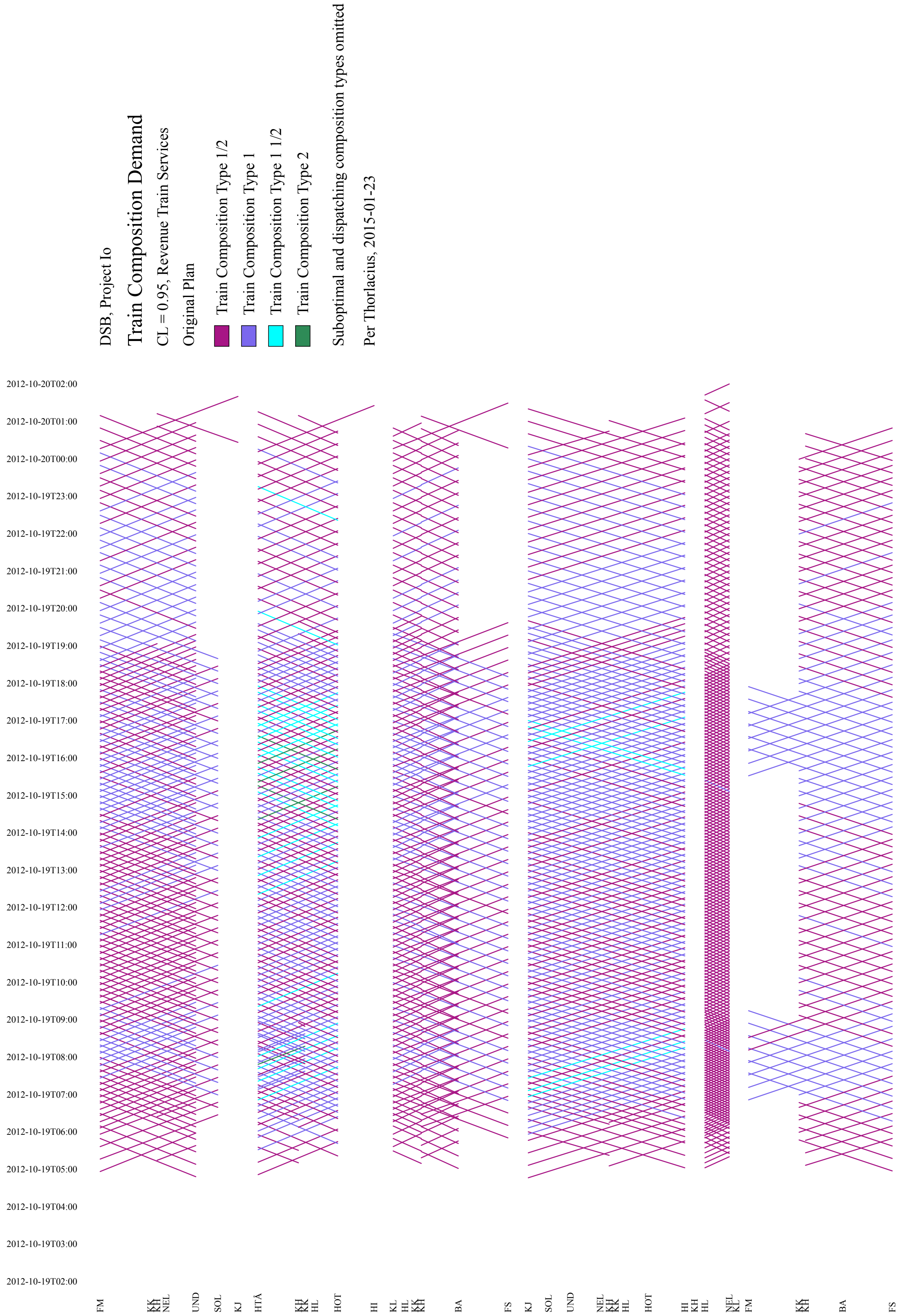

Figure A.5: Example of a train composition space-time diagram. The data instance shown is 2012-10-19. Demand data refer to a comfort level (CL) of 95%.

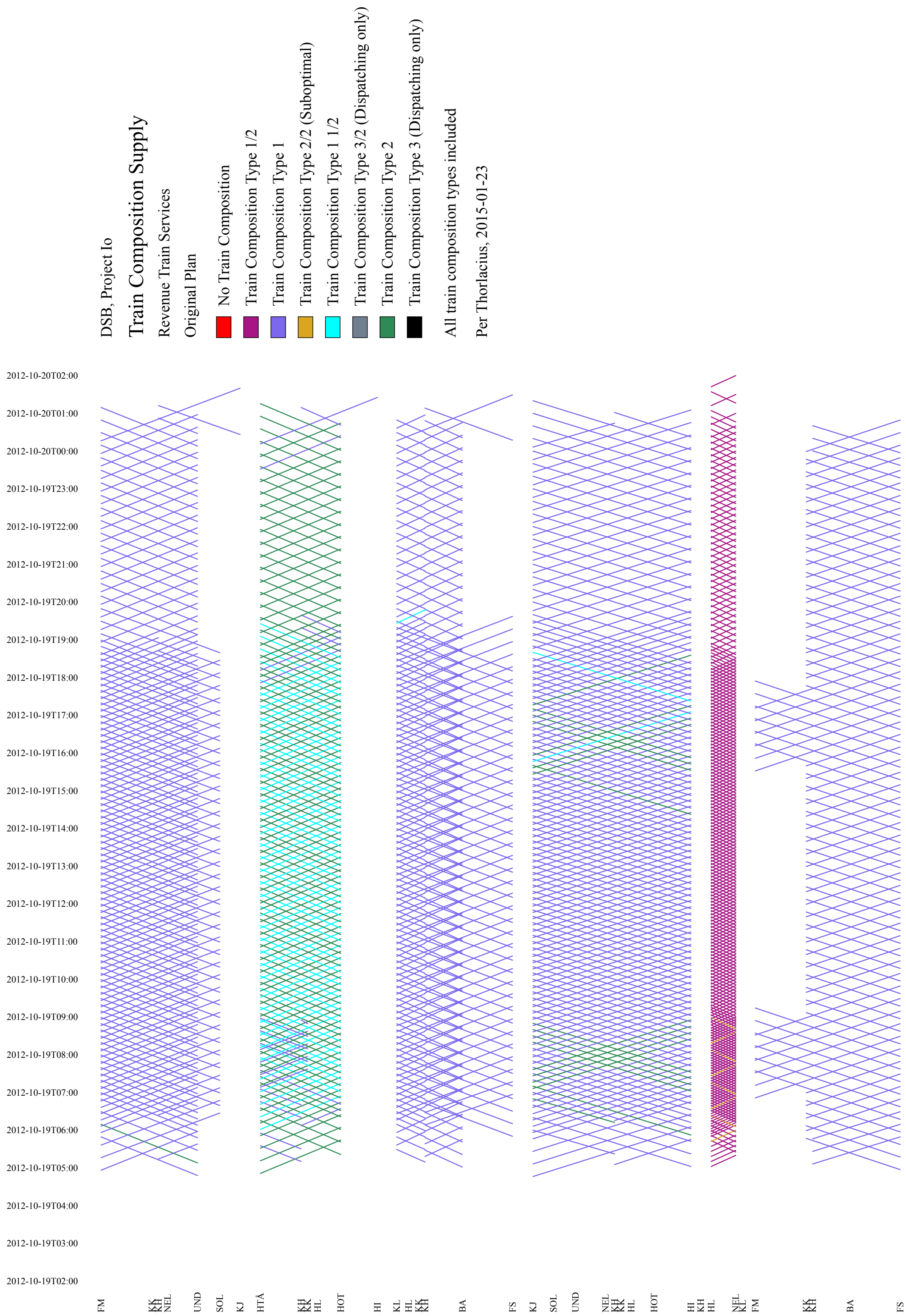

Figure A.6: Example of a train composition supply space-time diagram for an original plan. The data instance shown is 2012-10-19.

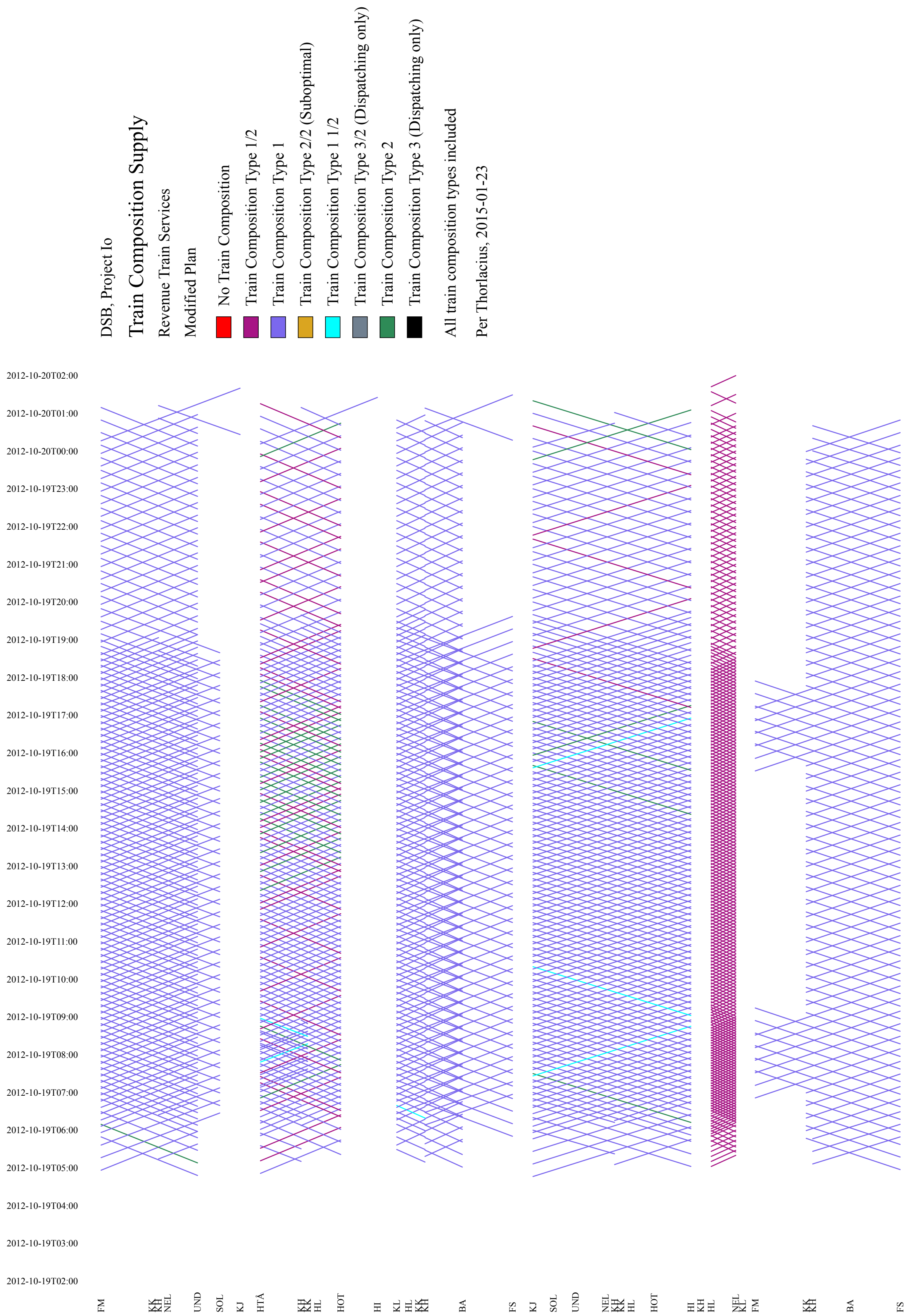

Figure A.7: Example of a train composition supply space-time diagram for a modified plan. The data instance shown is 2012-10-19.

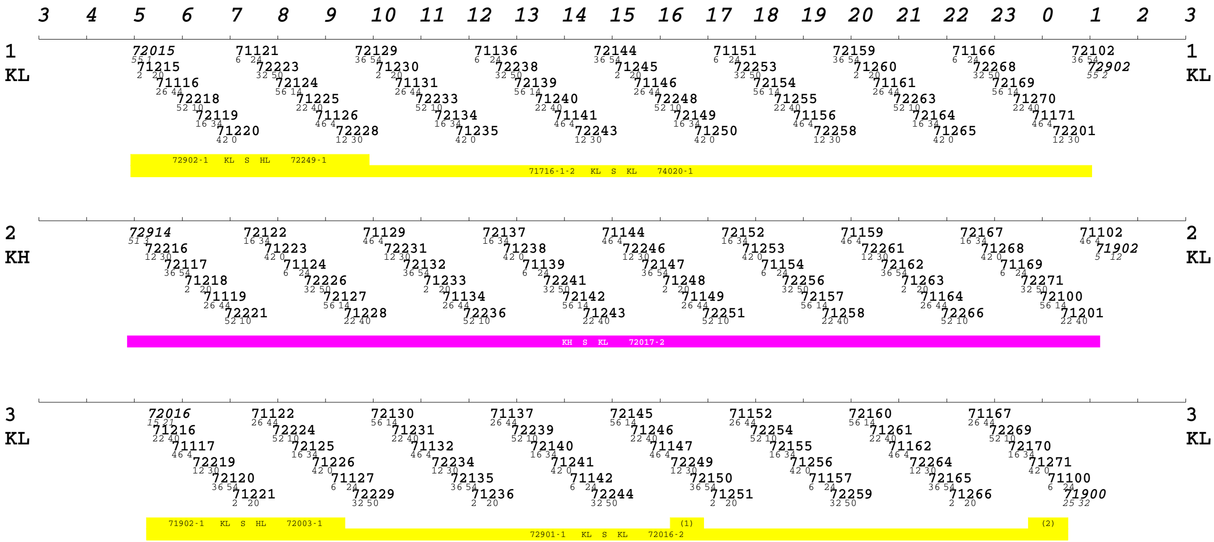

Figure A.8: Example of a DSB S-tog rolling stock circulation diagram for Line F for weekdays in the week starting with Monday 2014-03-31, as printed by the current manual rolling stock planning system at DSB S-tog. See Figure 2.2 on page 28 for an explanation of the diagram layout.

A.2 Circulation Diagrams

This type of diagram is drawn from the existing DSB S-tog manual rolling stock planning system (described in Chapter 2). This type of diagram serves as the main rolling stock plan visualisation tool in the planning process at DSB S-tog. It is also used to communicate rolling stock plans between departments.

See Figure A.8 and Figure 2.2 on page 28 for examples. The latter figure also provides an explanation of the details for this diagram type.

Note that there is a substantial difference in information density on circulation diagrams and space-time diagrams. Where the entire DSB S-tog timetable with all train services for an entire day can fit in one page in a space-time diagram (e. g., Figure A.1 on page 157), to display the same timetable as a circulation diagram, 26 pages would be needed. This is firstly related to the fact that there is somewhat more and more accurate information on a circulation diagram, secondly that most of this information is in the form of text, whereas on a space-time diagram the information is plotted as coloured graphic lines with space-time coordinates.

A.3 Infrastructure and Timetable Data Model Interactive Diagrams

In order to visualise the infrastructure and timetable data model (the space-time graph, described in Section 5.3 on page 73), an interactive visualisation tool has been developed. The tool is writing *dot* files for graph layout and rendering using the open source Graphviz graph visualisation package. An example including explanation is shown on Figure A.9 on the next page.

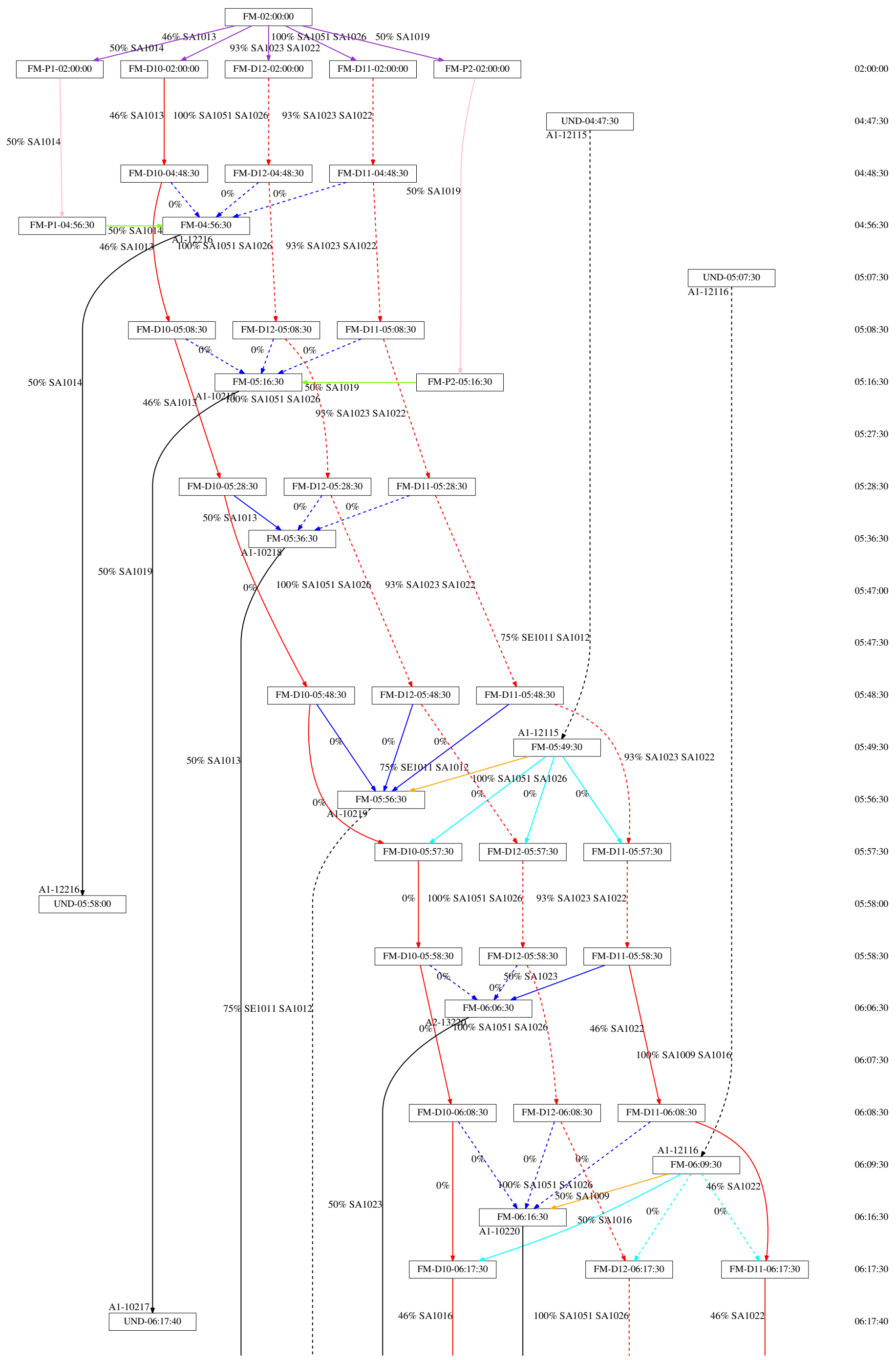

Figure A.9: A infrastructure and timetable data model interactive diagram example for real-world data: A view of a part of the space-time graph for the start up situation at Farum station (FM) in the morning for the rolling stock plan for 2014-03-31. The time increases from top to bottom, the (non-linear) time line is shown to the right. Vertices from the space-time graph are drawn as boxes, arcs as lines. The different types of arcs are coloured as follows: Graph flow arcs from the station source (violet); Parking at a depot track (red); Parking at a platform track (pink); Shunting from depot to platform (blue); Transition from being parked at a platform track to departing as train service (green); Revenue train services (black); Shunting from platforms to depot (light blue); Turning around from one train service to the next (orange). Solid arcs can accommodate more train units, dashed ones can't. Percentages show how much length capacity is used. Labels refer to platform (e. g., P1) or depot track numbers (D11), train lines (A1), train service numbers (12216) and virtual train unit IDs (SA1013).

This type of diagram has been essential to assert the correctness of the space-time graph topology (Section 5.3.1 on page 74), the correctness of the resource constrained shortest path algorithm with special side constraints (Section 5.5 on page 79), and the correctness of the procedures for retrofitting existing data (Appendix B.1 on page 176).

A typical space-time graph has approx. 22,000 arcs, a number way to large to make any sense to draw in one diagram. For this reason the user must choose which station(s) and time periods from which to construct the diagram. When the tool has created the diagram file, navigation (pan, zoom, topological movement, query) is then performed using the Graphviz graph visualisation package tools.

A.4 Branch-and-Price Algorithm Progress Interactive Diagrams

The branch-and-price algorithm can be a highly complicated solution method for solving mixed integer linear programs. In order to assert the correctness of the algorithm and in order to facilitate the exploration of different branching schemes and their effects on the inner workings of the algorithm, two interconnected visualisation tools have been developed, one for the branch-and-bound tree (presented in the following Appendix A.4.1) and one for the fractional flows for each node in the branch-and-bound tree (presented in Appendix A.4.2).

Like the tool for visualising the space-time graph, both of the branch-and-price visualisation tools work by writing *dot* files for graph layout, rendering and navigation (pan, zoom, topological movement, query) using the open source Graphviz graph visualisation package tools.

The visualisation tools work together in that the user can interactively click on any node having a fractional solution in the branch-and-bound tree diagram, and the fractional flow diagram for that solution is then shown.

The diagrams are constructed on-the-fly as the branch-and-price algorithm progresses and may as such be navigated when the algorithm is running.

A.4.1 Branch-and-Bound Tree Diagrams

Branch-and-bound tree diagrams are drawn as shown in example Figures A.10 to A.12, with an explanation on the former figure.

Other branch-and-bound tree visualisation methods are known from literature: A very simple visualisation is presented in [6], this using a tree structure as the only means. The branch-and-bound tree in [6] fans out equally in one dimension and is drawn by branching level in the other dimension. [100] demonstrates the visualisation features of a commercial solver. In these visualisations, the branch-and-bound tree also fans out equally in one dimension, but the other dimension is used to represent the objective value change. This highlights where in the tree the good branching decisions are taken, however this may lead diagrams taking up a lot of space.

The visualisation tool developed here differs from [6, 100] mainly by being much more detailed (showing more information), and from [100] by allowing a much more compact visualisation, this by including the objective value change (in this case, the drop) as numeric digits rather than as a dimension on the secondary axis of the diagram.

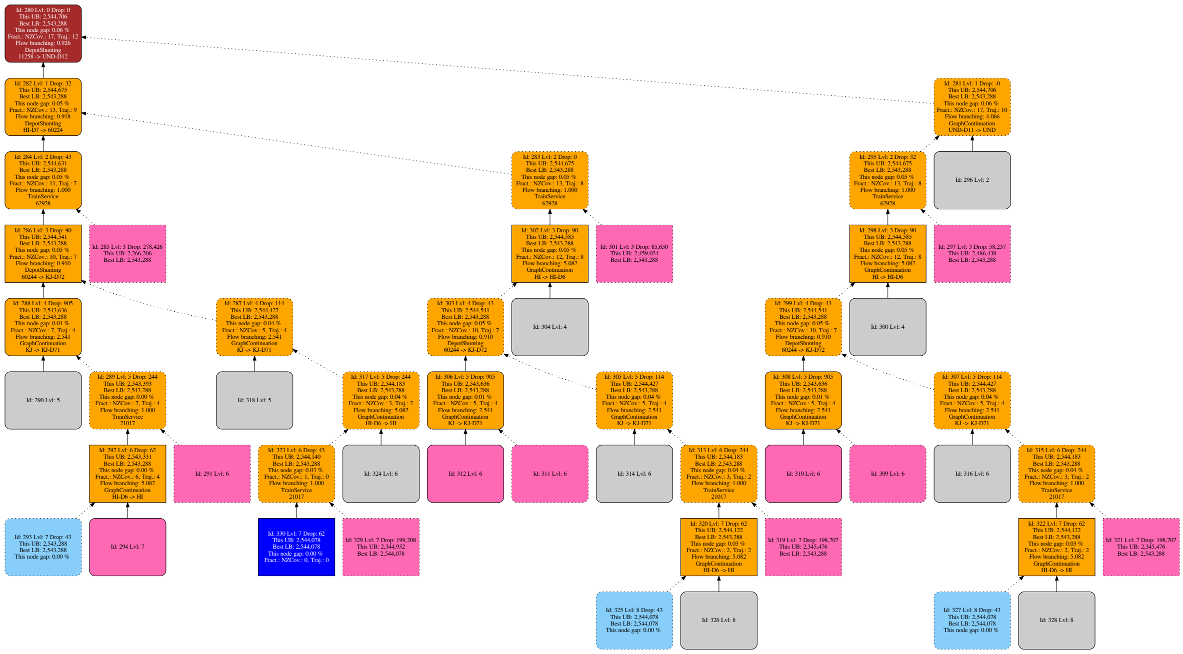

Figure A.10: An example of a branch-and-bound tree diagram showing a small branch-and-bound tree with nodes and edges between nodes. The diagram shows the root node (brown), nodes pruned by infeasibility (grey), nodes pruned by bounds (pink), fractional nodes within bounds (orange), integer nodes that have improved the current best lower bound (dark blue), integer nodes with same value as current best lower bound at the time of processing (light blue). The shapes of the nodes show branching type of parent node: Flow branching nodes (straight sides), constraint branching nodes (slanted sides, rarely occurring, not shown in example). Flow branching nodes are subdivided into the ones performed on fractional trajectories (rounded corners) and the ones performed on fractional covered variables (sharp corners). Solid edges and node outlines represent ceiling or force-trough nodes, respectively. Dotted edges and node outlines represent floor or force-around nodes, respectively. Solid edges are always vertical. Edges are ordered left-to-right by their order of processing. Thus, for two nodes with the same parent, the node to the left has been processed first. Each node includes information regarding its level, and if available: The drop in objective value from its parent, its objective upper bound, the current best lower bound, the gap between upper and lower bound, the number of fractional, non-zero coefficient, covered variables, the number of fractional train unit trajectories, the branching scheme, the branching entity type and attributes, etc.

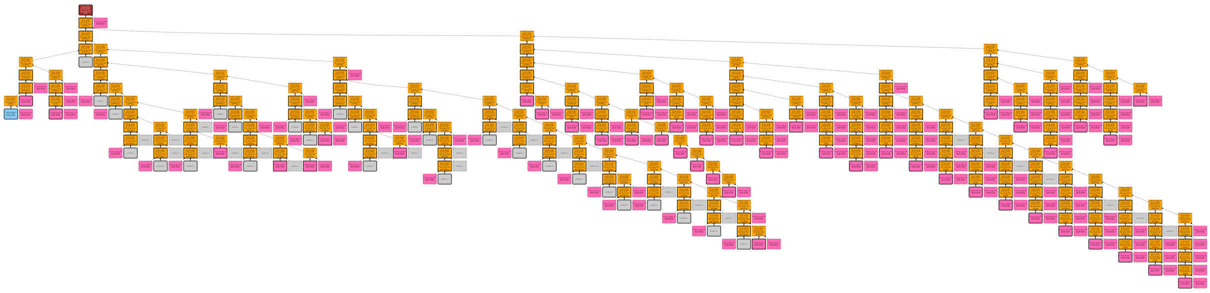

Figure A.11: An example of a branch-and-bound tree diagram showing a branch-and-bound tree constructed using the normal, non-strong branching scheme from Chapter 7. This example is using the objective function defined in Chapter 5. Recall that this objective function exposes unwanted symmetric properties. The effects of symmetry can be seen on the unbalanced series of branches ending at the lower right part of the figure. This series of unbalanced branches occurs on the zero-side of each branching (downward to the left of each node), in which there is little or no drop in objective value due to symmetry.

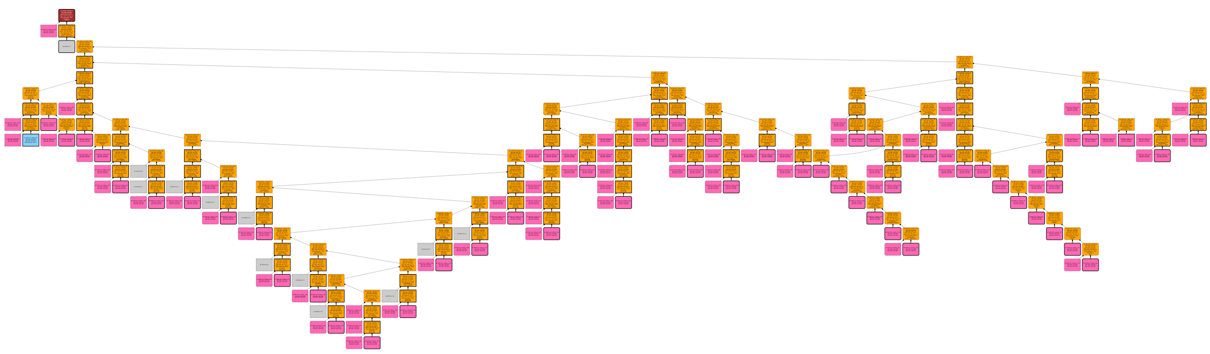

Figure A.12: An example of a branch-and-bound tree diagram showing an example branch-and-bound tree constructed using full strong branching. This example is also using the objective function defined in Chapter 5, which has unwanted symmetric properties. As may be seen, the branch and bound tree contains less nodes than the one in Figure A.11 above, and is also more balanced. However, in our case, all in all, substantially more nodes are processed in the strong branching scheme since all possible children nodes are generated and solved for each branching operation. This is done to chose the branching operation generating nodes that may be pruned or otherwise have the largest drop in objective value. For this reason, in our case, strong branching is less time efficient than non-strong branching.

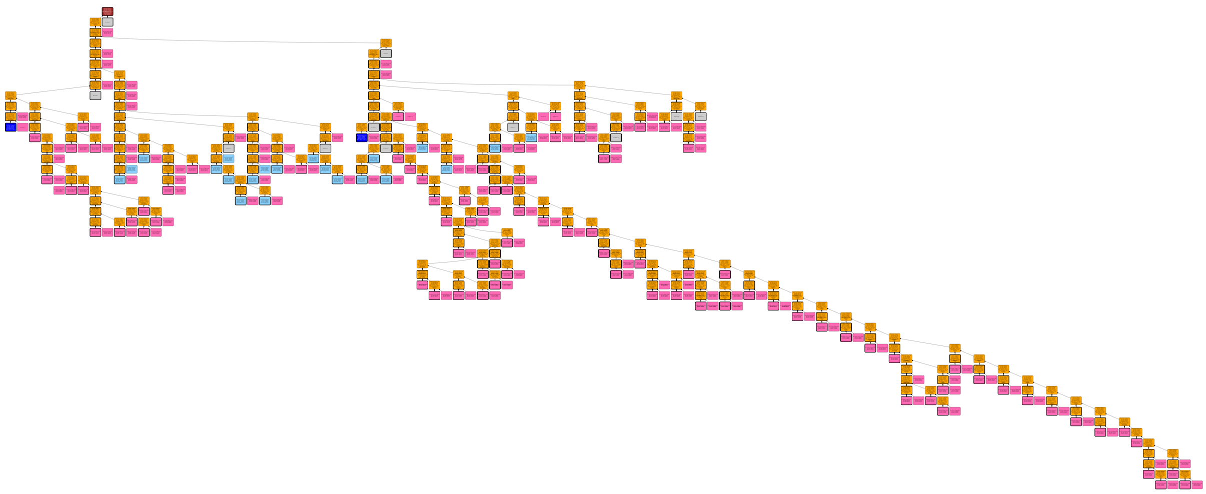

Figure A.13: An example of a branch-and-bound tree diagram showing the branch-and-bound tree for the solution of Line H on the instance 2012-10-19 using the enhanced objective function. The fractional flow diagram corresponding to the root node in this diagram is shown in Figure A.15 on page 172. As may be seen, the branch-and-price algorithm finds a new incumbent best integer solution two times, firstly in the leftmost part of the tree (blue), secondly in the middle of the tree. As may be seen, a lot of equally good integer solutions are also found (light blue). This indicates some symmetry in the problem even in the case of the enhanced objective function.

A.4.2 Fractional Flow Diagrams

Fractional flow diagrams are drawn as shown in examples in Figures A.14 to A.16 on pages 171–173, with an explanation on the former figure.

The fractional flow diagrams are processed using a specially developed coalescing algorithm so as to render only the relevant nodes in the graph on the diagram. Nodes where no split flow is occurring are omitted, and their corresponding arcs are coalesced. This yields a much more compact diagram only showing relevant aspects of otherwise vast amounts of data.

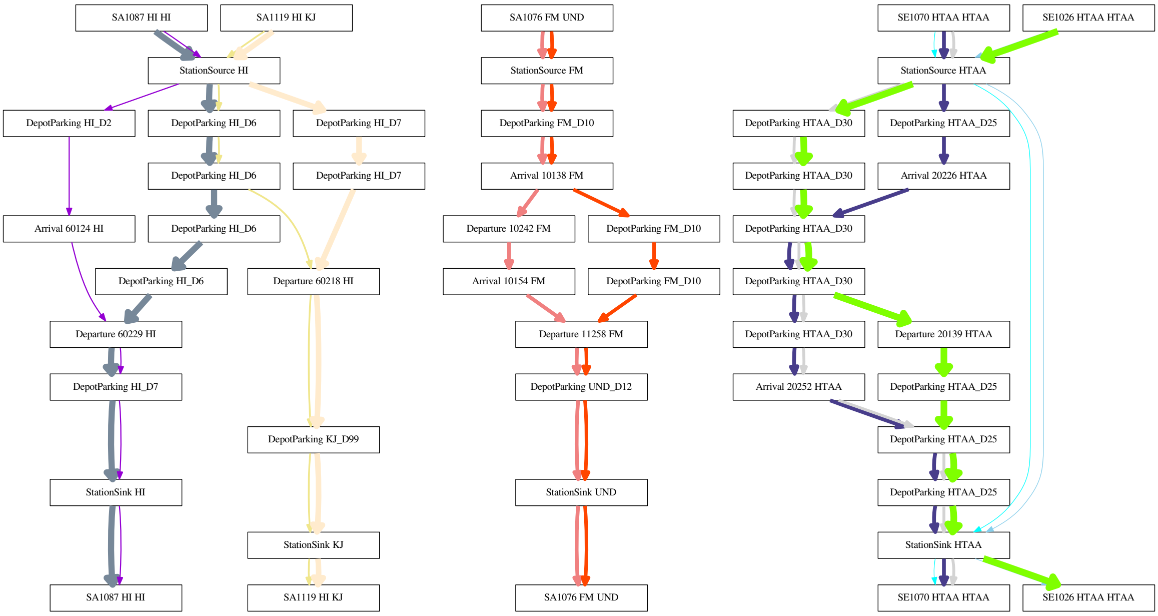

Figure A.14: An example of a fractional flow diagram with few fractional flows. Boxes represent either virtual train units (top and bottom rows) or space-time events (vertices) from the space-time graph (all other rows). Space-time events are e. g., train service arrivals or departures, or the start and finishing events of train shuntings. Edges represent the flow of train units between space-time events, a flow that is taking place on the arcs of the space-time graph. The width of each edge is proportional to the fractional decision variable value, i. e., thick edges represent decision variables values close to 1, thin edges decision variables close to 0. Only train units with fractional flows are drawn on the diagram. Edges to nodes where no splitting is occurring are coalesced for diagram compactness. Beginning at the upper left box on the figure, this represents the virtual train unit SA1087 starting its train unit trajectories at Hillerød station (HI), ending also in Hillerød (HI). From that box two edges leave, a thick grey one and a thin violet one, both going to the station source node of Hillerød Station (HI). According to the generalised upper bound constraint (Equation (7.2) on page 118), the sum of the variable values for these two edges is 1. At the source node of Hillerød Station (HI) a splitting of the flow is occurring, sending the violet trajectory to depot track 2 and the grey to depot track 6. From this point on, the flow of the virtual train unit SA1087 is fractional, split over two trajectories. The thick gray trajectory stays at depot track 6, while the thin violet trajectory serves train service 60124. Both trajectories join to serve train service 60229 from Hillerød (HI), from which point on the flow on the arcs of the space-time graph is thus integer. While the gray fractional trajectory of virtual train unit SA1087 was parked at depot track 6 in Hillerød (HI) it was joined by a fractional trajectory from the virtual train unit SA1119, coloured yellow. This trajectory left the depot track to serve train service 60218 together with the remaining khaki coloured trajectory of virtual train unit SA1119, from which point on the flow for that virtual train unit is also integer.

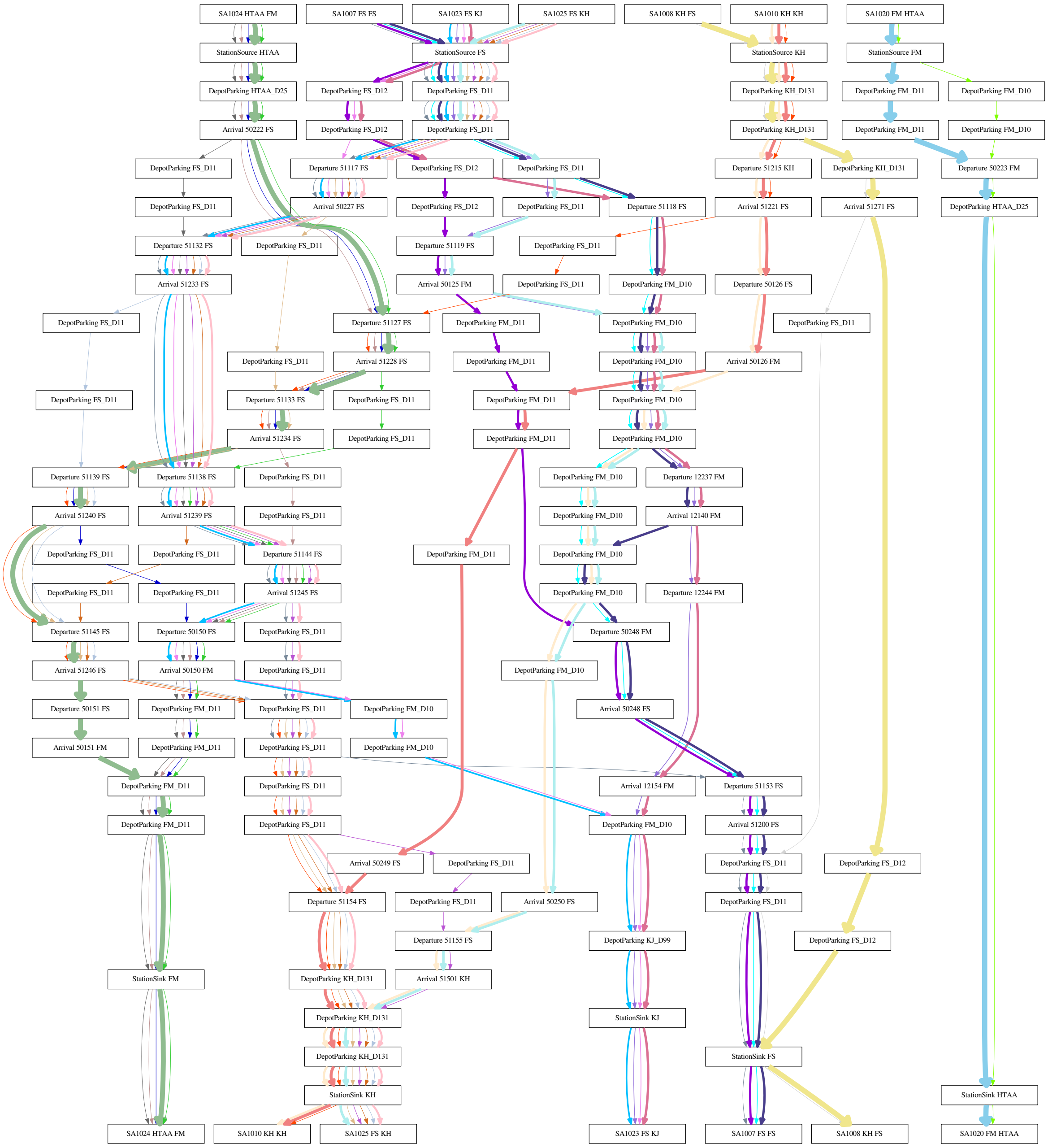

Figure A.15: An example of a fractional flow diagram with many fractional flows. This is the fractional flow diagram for the root node of solving the entire Line H in one iteration of the branch-and-price matheuristic, the branch-and-bound tree diagram of which is shown in Figure A.13 on page 169. The data instance is 2012-10-19, in which Line H has 11 train units. As may be seen, 7 of these have fractional solutions in the root node of the branch-and-bound tree. What can be seen on the figure is the effect of each train unit being split into fractional pieces so as to fulfil passenger demand on each train service exactly as it occurs.

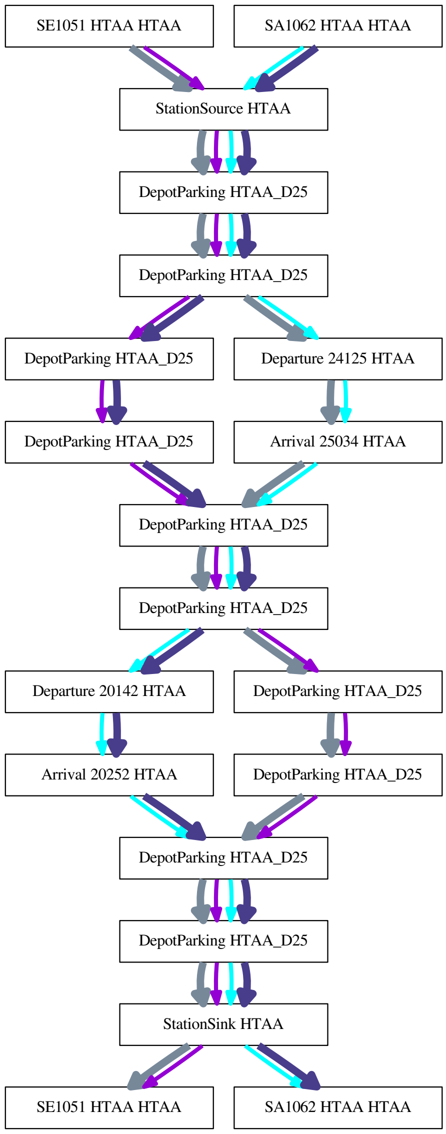

Figure A.16: An example of a fractional flow diagram with crossover trajectories. Two different types of train unit types participate in the crossover: While the total train unit flow of all arcs in the space-time graph is integer, it still consists of a sum of fractions from different train unit types that cannot be split. It is because of the occurrence of crossovers like this one that the second branching scheme, constraint branching (see Section 7.4.2 on page 124), is necessary.

A.5 Causal Diagram for Model Design Decisions

Designing the architecture of an optimisation oriented mathematical model to handle a large number of requirements will inevitably involve trade-offs, e. g., between model capabilities and processing time. In order to clarify the causal relationship between model architecture design decisions and their effects, an attempt was made to visualise these in *causal diagram* form [85, 58].

Figure A.17 shows the causal diagram for the integrated matheuristic rolling stock planning model presented in Chapter 7. Edges represent a directed cause-and-effect relationship between two "events", represented by boxes. Orange boxes are model architecture design decisions, gray boxes derived effects of these decisions and light blue boxes the desired overall goals of the modelling process. Each box is marked by a "+" on top symbolising an increase (or a decision to do something), and a "-" on the bottom symbolising a decrease (or a decision to refrain from doing something). For example, a decision to increase *C7. Penalty for train shuntings* would result in a decline in *B1. Number of train shuntings in trajectories* and so on. Causal relationships are drawn recursively from the three desired goals of having a short total processing time of the model (A1), a high probability of plan net value improvement (A2) and a highly integrated and realistic model (A3). As may be seen, some decisions influence the desired goals in both directions, e. g., *C8. Number of train units to modify: k*, where an increase in *k* adds to the desired goal of having a high probability of net value improvement (A2), whereas a decrease in the value of *k* adds to the desired goal of a low total processing time (A1), and vice versa. For an overview of other causal relationships in integer programming models, see [69].

Figure A.17 was used in the process for determining model architecture for the branch-and-price matheuristic model described in Chapter 7 by providing an overview of the consequences of design decisions.

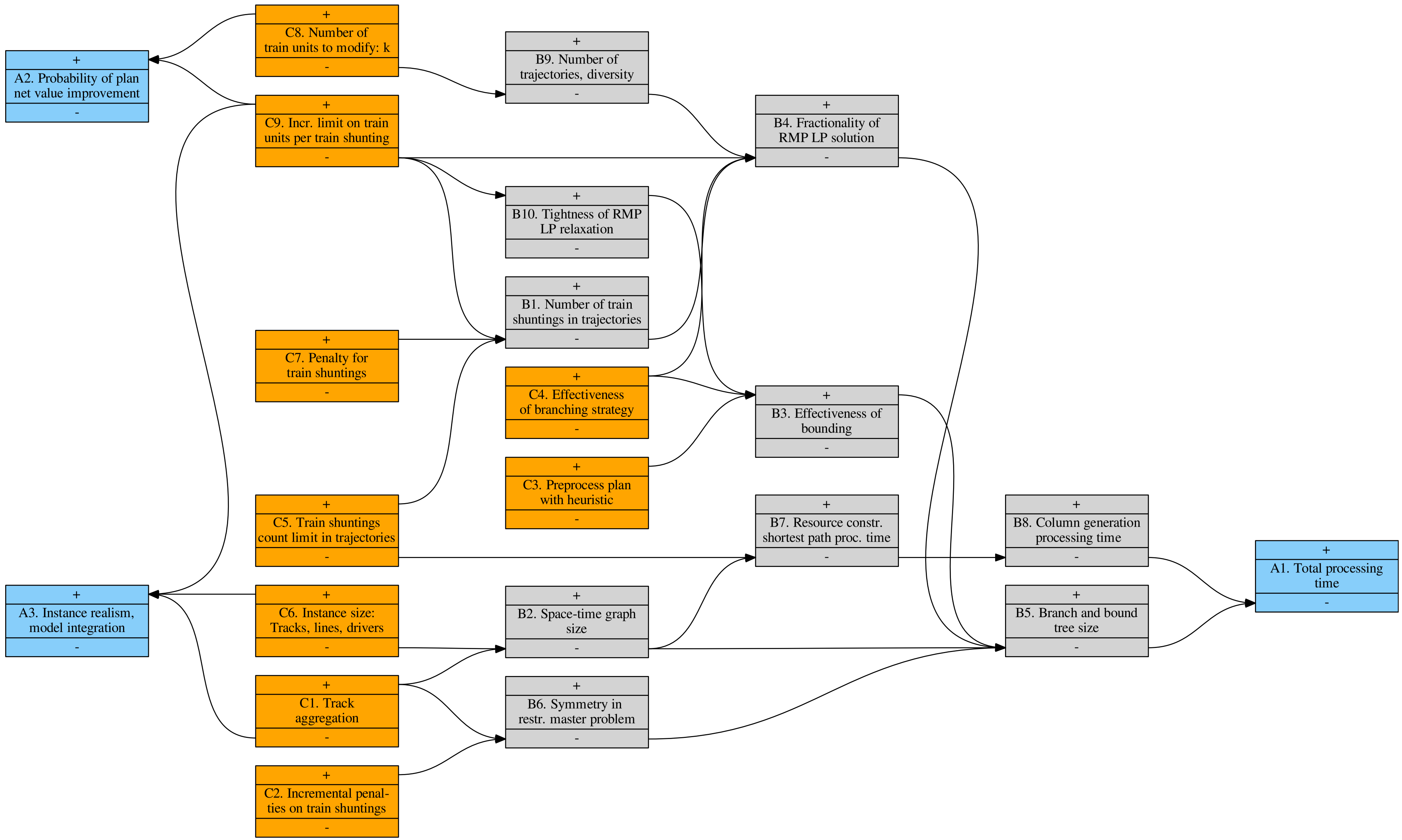

Figure A.17: A causal diagram for the integrated matheuristic rolling stock planning model presented in Chapter 7. Edges represent a directed cause-and-effect relationship between two “events”, represented by boxes, see Appendix A.5 for further explanation.

Appendix B

Auxiliary Methodologies for Rolling Stock Planning

This appendix contains descriptions of the auxiliary methodologies applied in order to implement the rolling stock planning models presented in this thesis. Firstly, in Appendix B.1, the procedure to retrofit incomplete train unit trajectory data into the integrated data model for rolling stock planning is described. This involves a modified version of the Welsh-Powell algorithm and an conflict graph. Next, in Appendix B.2 it is described in detail how the cost and benefit elements of the objective function (the net value) are defined and quantified. In Appendix B.3 a detailed description of the *(train) unit order flow conservation principle* (as stated in Chapter 5) is presented.

B.1 Retrofitting Incomplete Train Unit Trajectories Data

The current rolling stock planning procedures at DSB S-tog still involve some degree of manual planning. For this reason, data is not available in database form for all aspects of the rolling stock plan data instances. One aspect to which data is currently not available is train shuntings of the individual train units, the available information only relates to the assignment of train units to train services. Each train unit trajectory thus has gaps in it between train services. The information to close these gaps (if at all possible) has to be generated artificially by *retrofitting* each original plan to the data model described in Chapter 5. This retrofitting process is described in detail in the following.

The retrofitting process utilises a conflict graph as known from [55, 53, 72, 46] and a graph vertex minimum colouring algorithm inspired by the Welsh-Powell algorithm [65]. The Welsh-Powell algorithm was chosen as basis algorithm due to its low complexity, the algorithm does not necessarily yield the minimum number of colours.

For the purpose of explaining the retrofitting we define the term *retrofit*. A retrofit describes a gap in a train unit trajectory. If the retrofit gap can be filled, we have a successful retrofit, if not a failed retrofit. Each retrofit has a train composition assigned to it, that is, the collection of train units that need to use the retrofit to close the same gap in their train unit trajectories.

A retrofit can occur between to successive train services in a train service sequence. In that case, the retrofit gap can be filled with a train service sequence arc from the space-time graph.

A retrofit can also occur between two train services that are not in the same train service sequence. In that case the gap can be filled by a collection of arcs from the space-time graph that connect the two train services through a given depot track or side track. This collection of arcs will then consist of:

1. A train shunting arc from the train service arriving at the platform and into the depot track or side track;

2. A number of depot track or side track parking arcs;

3. A train shunting arc from the depot track or side track to the train service departing from the platform.

Naturally, retrofits occur only at a station, not between them.

The retrofitting process starts by registering all retrofits and the train units that use them. This is performed by looping across all train service arcs in the space-time graph. For each train service arc another loop is initiated across all train units. If the train service arc is contained in the assigned train unit trajectory of the train unit, an iterator is created to point just before this train service arc in the train unit trajectory. (An iterator is a pointer to a position in a collection, in this case the collection of arcs that constitute the train unit trajectory.)

If there is no previous train service arc to the iterator, we are at the very beginning of the trajectory. A search for existing retrofits from the station source to the from-station of the first train service arc is then conducted. If no retrofit is found among the existing ones, a new one is created and added to the collection of existing ones. The current train unit is added to this retrofit.

Next, if there exists a next train service arc to the iterator, we are in between train services. Again a search for existing retrofits from the to-station of the arriving train service and the to the from-station of the departing train service arc is conducted. If no retrofit is found in the existing ones, a new one is created and added to the collection of existing ones. The current train unit is added to this retrofit.

Else, if there exists no next train service to the iterator, we are at the end of the train unit trajectory. This is handled in exactly the opposite way as when we are at the beginning.

All retrofits have now been registered. If multiple train units use the same retrofit, they are registered in the train composition of that retrofit.

The next step is to register conflicting retrofits. Two retrofits are conflicting if they represent movements of train compositions on the same track that are overlapping, e. g., if train composition 1 (belonging to retrofit 1) enters a given depot track before train composition 2 (belonging to retrofit 2), and train composition 1 needs to leave before train composition 2 has left. In that case train composition 2 is blocking the movement of train composition 1, and they are conflicting.

Conflicts are registered by comparing each retrofit to all others (that have not yet been compared to that retrofit). When comparing two retrofits, they are in conflict if all of the following criteria are met:

- They occur at the same station;
- They need to occur at a depot track or side track;
- They have overlapping time intervals;
- Their time intervals are not identical;
- Their time intervals are not contained in each other.

When a conflict is detected this is registered in both retrofits by setting pointers to the other in each of them. These pointers then represent the edges in the conflict graph. For a conflict graph example, see Figure B.1.

The list of identified retrofits is then sorted by descending number of conflicts (the vertex degree in the conflict graph), then by descending train composition length. The rationale for

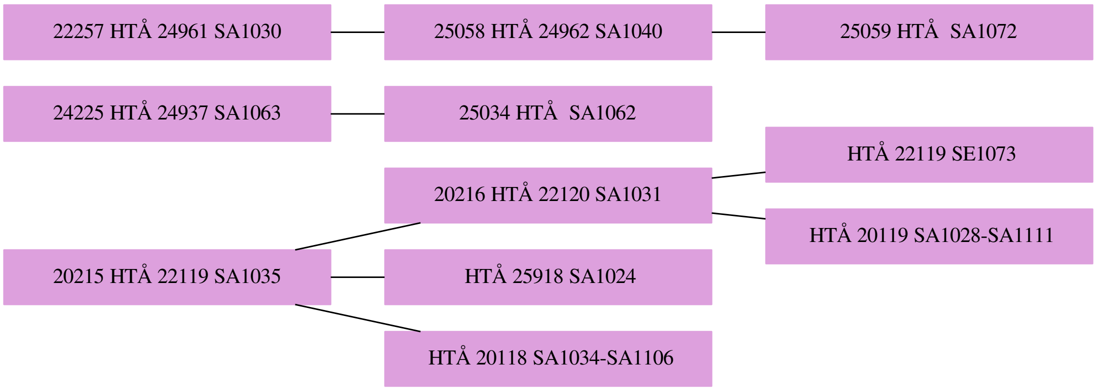

Figure B.1: An example of a conflict graph for the retrofitting process for the depot of Høje Tåstrup (HTÅ) for the data instance 2012-10-19, a Friday. In the graph, vertices represent retrofits, edges conflicts between them. The bottom centre retrofit represents the first shunting out from the depot (HTÅ) of the day, this to train service 20118 with the train composition consisting of virtual train units SA1034 and SA1106. This retrofit is in conflict with the one above to the left, which represents the retrofit of the two train services 20215 and 22119 (that are not in the same train service sequence) for the train composition consisting of virtual train unit SA1035. The two mentioned retrofits are in conflict because the former retrofit train shunting with SA1034-SA1106 is to take place after SA1035 has arrived with train service 20215 but before SA1035 has departed again with train service 22119. For this reason, these two retrofits cannot use the same depot track. The latter retrofit is one of the two having the greatest number of conflicts. In the retrofitting process, one of these retrofits is accordingly retrofitted as the first one.

doing this is that by sorting on the descending vertex degree, we are following the Welsh-Powell algorithm by starting to colour (in our case, to assign segments) those vertices (in our case, retrofits) in the conflict graph with the largest number of conflicts first. By sorting by descending train composition length we are retrofitting those train compositions that take up a lot of track space first, in order to utilise the available track space as good as possible. This is analogous to putting stones of different sizes into a bucket: If we start by putting in the big stones, we can fill the bucket more, because we can then put the small stones in between the big ones.

Each retrofit is then retrofitted by finding the shortest path between the from and to space-time events (e. g., arrival and subsequent departure). If the retrofit has registered conflicts, the shortest path segments found in the space-time graph are then blocked for the use by the train compositions of the conflicting retrofits. This way two conflicting retrofits cannot use the same segments in the space-time graph.

This blocking procedure is necessary because at this time the individual order of train units cannot be known because of the gaps in the trajectories. With gaps it is not known how the train units are moving relative to each other. (If it could be known, we could just rely on the train unit order conservation principle, see Appendix B.3.)

Once the retrofit gap segments (the shortest path segments) have been found, they are added to the trajectory. The segments are also assigned to the train units in the train composition in the space-time graph.

In some cases, all train unit trajectories can be retrofitted using the procedures described above. However, as shown e. g., in Table 5.5 on page 88 in the column *# of infeasible trajectories*, in most cases, the retrofitting process fails for a just a few of the train unit trajectories. The is because the original, manually produced plans may not respect all of the railway-specific requirements themselves. It is common for the planners to use a variety of "dirty tricks" to improve the economic properties of a rolling stock plan manually, or to replace intolerable infeasibilities from the existing, automated circulation planning system with "less intolerable" infeasibilities. Strictly speaking, these tricks are violations of the railway-specific requirements, only justified by the fact that they may save otherwise substantial costs. The infeasibilities introduced by the planners are often used as a last resort and incorporate all their tacit knowledge. For these reasons the trajectories cannot be mapped onto the data model.

For the 15 data instances used in this thesis, there are missing retrofits in less than 2% of all trajectories when retrofitting the original plan into the data model. In the numerical experiments performed in this thesis, the train unit trajectories that were not possible to retrofit correctly are removed from the model. As the different algorithms progress (e. g., the greedy heuristic from Chapter 5), the train units will eventually get assigned to them new train unit trajectories that do not violate the railway-specific requirements.

For other methods to detect and prevent conflicts like the ones described, see [53].

B.2 Rolling Stock Plan Net Value Calculation

This section describes in detail the cost and benefit elements of the net value objective function for the rolling stock planning models developed in this thesis. The penalties and awards are a matter of model calibration and are not treated here.

B.2.1 Rolling Stock Operating Cost Types and Structure

As may be seen from Table B.1, seven different types of operating costs are relevant for DSB S-tog as a suburban passenger train operator. The different types of costs are proportional to different factors of the operation, and are, as such, incurred as stated in the table and described in the following.

All train units in a train composition need primary acceleration energy to reach cruising speed upon departure from the origin station, regardless of whether the train units perform revenue or non-revenue positioning. However, since a non-revenue positioning train service is not stopping at intermediate stations, no subsequent acceleration energy is needed. The energy needed to sustain cruising speed is proportional to the distance each train unit moves, regardless of positioning type.

The maintenance costs are also proportional to the distance each train unit moves, and as such, regardless of positioning type.

The fee for using the railway infrastructure is only levied for revenue train services. As such, there is no extra infrastructure cost for non-revenue positioning. There is also no extra infrastructure cost for revenue positioning, because the fee has already been paid for the train service that is involved in the revenue positioning, since this train service runs with passengers according to the given timetable. In this sense, one could say that the infrastructure costs are solely determined by the given timetable.

Similarly, the train driver duty costs are strongly dependent on the given timetable since each train service must have a train driver. If a decision is made to perform a revenue positioning, this will not induce additional driver duties, since the train service already has a train driver. However, deciding to perform a non-revenue positioning may (under certain conditions) require

Table B.1: Overview of rolling stock operating cost types for a suburban passenger railway operator like DSB S-tog. *Primary acceleration energy* is the energy needed to accelerate the train composition to cruising speed upon departure from its origin station. Analogously, *subsequent acceleration energy* is the acceleration energy needed at subsequent stations. *Speed sustainment energy* is the energy needed to sustain cruising speed.

Type of cost	Proportional to	Incurred by revenue positioning	Incurred by non-revenue positioning
Primary acceleration energy	# of train units of type	Yes	Yes
Subseq. acceleration energy	# of train units of type at station	Yes	No
Speed sustainment energy	# of train units of type service dist.	Yes	Yes
Maintenance	# of train units of type service dist.	Yes	Yes
Infrastructure	Revenue train service distance	No	No
Train driver duties	Engine drivers hired	No	Possible
Depot driver duties	Depot drivers hired	Possible	Possible

that another train driver be hired to drive the train composition to be positioned (if the number of train drivers hired and on duty is not sufficient to perform this task).

The costs related to the depot drivers are proportional to the number of depot drivers hired. Under certain conditions there may not be enough depot drivers hired and on duty to perform the train shuntings that may be required. In this case, additional depot drivers must be hired, thus inducing a higher cost.

The models developed for this thesis use a simplified cost structure, as shown in Table B.2, in which the different energy terms from Table B.1 are simplified and no distinction between revenue and non-revenue train services is made with regard to energy consumption. Moreover, the cost for train drivers and depot drivers is calculated based on an hourly rate. This rate is estimated based on the average monthly wage a train driver or a depot driver receives, multiplied by the fraction of the time they are currently undertaking driving of train services or of train shuntings. The energy usage for train shuntings is considered negligible.

B.2.2 Quantifying Benefits of the Rolling Stock Operation

The main purpose of any rolling stock plan is to provide seats for the conveyance of passengers for revenue. As such, a passenger railway operator may look at the act of providing a seat to a passenger demanding one as a way of generating revenue, i. e., providing a benefit.

In the following, a method is proposed to calculate the benefit of providing a seat to a passenger demanding it. The economic specific seat demand fulfilment benefit b_t is defined as the benefit of providing one demanded seat for one unit of time. The value for b_{st} is estimated from the stated preference time penalty for having no seat t_s, the specific value of time for commuters v_t and the duration of an average travel $\hat{t}$, Equation (B.1).

$$b_{st} = \frac{t_s v_t}{\hat{t}} \tag{B.1}$$

For the case of DSB S-tog and the city of Copenhagen, $t_s = 5.8$ min., $v_t = 1.21$ DKK/min and $\hat{t} = 16$ min, yielding a value for b_{st} of 0.44 DKK/min [82].

Table B.2: Overview of the unit costs for the rolling stock operation at DSB S-tog as used in the models in this thesis. Note that the infrastructure usage costs are proportional to the revenue train km and as such independent of train unit type composition and non-revenue positioning. Figures are from 2013. Note that the energy and maintenance costs for type $\frac{1}{2}$ train units are a bit higher than half of those of the type 1 train units.

Unit Cost Type	Unit	Cost for train unit type 1	Cost for train unit type $\frac{1}{2}$	Cost for train service/shunt.
Maintenance	DKK/Train unit km	12.58	7.40	-
Energy	DKK/Train unit km	6.80	4.40	-
Infrastructure	DKK/Rev. train srv. km	-	-	0.0475
Train drivers	DKK/Train service h	-	-	427
Depot drivers	DKK/Shunting h	-	-	314

This value is roughly equivalent to the actual specific revenue gained by DSB S-tog for conveying an average passenger by means of ticket sales.

In the branch-and-price matheuristic model from Chapter 7, the specific monetary benefit $b(a)$ of providing a single seat on arc a is used in Equation (7.1) on page 113. $b(a)$ is calculated as shown in Equation (B.2) below, $t_1(a)$ being the departure time of the revenue train service represented by arc $a \in A_R$, and $t_2(a)$ being the arrival time.

$$b(a) = b_{st} \cdot (t_2(a) - t_1(a)) \tag{B.2}$$

B.3 The Unit Order Flow Conservation Principle

Due to scientific article space restrictions in Section 5.4.2 on page 77, only an abbreviated explanation to the train unit order conservation principle is included there. What follows is the full explanation.

The train unit data model keeps track of the order of the individual train units relative to each other. In the following, the logic for coupling and decoupling train compositions in relation to the order of the train units will be explained. This logic is specific to the business and train control system rules currently in effect at DSB S-tog.

The simple explanation is this: *At the platform, train units are being coupled and decoupled in the direction that is facing the depot. At the depot, train units are being coupled and decoupled in the direction facing the platform.* This is illustrated in Figure 5.3 on page 78.The formal explanation follows.

A *train composition* is the ordered sequence of one or more train units coupled together. At DSB S-tog, train compositions consisting of one or two train units may be assigned to revenue train services. Non-revenue train services may consist of train compositions with up to three train units provided the total length limit is not exceeded. Train compositions of more than two train units may be formed when parking train units at a depot (and thereby coupling them).

The result of the coupling of two train compositions is a new train composition made out of all the units of the two original train compositions. The result of the decoupling of a train composition is two new train compositions made from the train units of the original train composition. The relative order of the train units remains the same in couplings and decouplings.

Each coupling and decoupling involves one *train shunting* and can either take place at the *platform track* or at the *depot track* or *side track*. The train shunting is the movement either bringing the one train composition to the other for coupling or bringing the decoupled train composition away from the remaining train composition after decoupling. As such, a train shunting always has a depot track or side track as a starting point and a platform track as its finishing point or vice versa.

Current business and train system control rules at DSB S-tog state that when a decoupling takes place at the platform, the train composition moving away must move to the depot track or side track. It may not be assigned to a train service. Furthermore, when a coupling is to take place at a platform, the train composition moving in to couple must come from the depot track or side track. It may not come from the main line, i. e., from a train service.

The term *platform train composition* is used to denote the train composition in the operation that is facing the platform. Similarly, the *depot train composition* is the train composition that is facing the depot.

The term *relative position* denotes how an object (platform track, depot track, side track, train composition, train unit) is oriented relative to another[1]. For DSB S-tog, the relative position can be either *North* or *South*. For example, the relative position of a depot track to a platform track at its corresponding station may be South, meaning that to reach the depot track from the platform track, train units must move towards the South[2]. Equivalently, a train unit may have the relative position South of another train unit. At the same time, this also means that the other train unit has the opposite relative position, i. e. North, of the first one.

The individual train units in a train composition have a relative position to each other. With the proposed definition, the relative position of train units in compositions is conserved in all feasible coupling and decoupling operations.

In the following example, couplings and decouplings are envisioned at a depot station, however, the principle is the same for at side track station.

Picture the **coupling** of two train compositions at a **platform track**, one being a platform train composition, the other being a depot train composition. The situation before the depot train composition is shunted in from the depot is depicted in Figure 5.3d on page 78, the situation after coupling in Figure 5.3b. After coupling, the original depot train composition will have the relative position to the original platform train composition (in the new train composition) equal to the relative position of the depot to the platform (in other words, equal to the relative position of where the train shunting **started**). If, like in the example in Figure 5.3, the relative position of a depot track is to the South of a platform track, then the original depot train composition being shunted in to the platform on this station will have the relative position South to the original platform train composition in the new train composition. This is the result of the transition between the situations from Figure 5.3d to Figure 5.3b.

In the case of a **coupling** taking place at a **depot track**, the relative position of the platform train composition (which is the one undergoing movement in the operation) will also be the same as the relative position of the place from which the shunting **started**, in this case the platform track. If, like in the example above, the relative position of a depot track is to the South of a platform track, then the relative position of the platform track is to the North of the

[1]Note that the relative orientation of a train unit to the train composition to which the train unit belongs does not make sense.

[2]Note that this does not mean that a depot track is located to the South of a platform track in the geographic sense. In the case of Hillerød, for instance, depot track #6 has the relative position South to the platform track #3, however, geographically, the depot track is located to the North of the platform track. The explanation is this: Depot track #6 is reached from platform track #3 by moving the train composition South into track #119 (hence its relative position), and then changing direction of movement towards North, moving parallel to platform track #3 and beyond it to reach depot track #6. See Figure 3.1 on page 35 for the map of the infrastructure.

depot track. The original platform train composition after coupling will then have the same relative position to the original depot train composition as the platform track has to the depot track, which in this case is North. This is the result of the transition between the situations from Figure 5.3b to Figure 5.3d.

In the case of a **decoupling** taking place at a **platform track**, only the train units at the same relative position to the others in the original train composition as the relative position of the depot track to the platform track may be decoupled to form the new depot train composition to be shunted into the depot. This is equivalent to the relative position of where the shunting **ended**. If, like in the example above, the relative position of a depot is to the South of a station, then a new depot train composition to be decoupled from the original one can only be formed by train units that all have the relative position South to the remaining train units in the original train composition. This is the result of the transition between the situations from Figure 5.3b to Figure 5.3c and from Figure 5.3b to Figure 5.3d.

In the case of a **decoupling** taking place at a **depot track**, only the train units at the same relative position to the others in the original train composition as the relative position of the station to the depot may be decoupled to form the new platform train composition to be shunted to the platform track. This is also equivalent to the relative position of where the train shunting **ended**. If, like in the example above, the relative position of a depot is to the South of a station, then the relative position of the station is to the North of the depot, then a platform train composition can only be formed by train units that all have the same relative position North to the remaining train units in the original train composition. This is the result of the transition between the situations from Figure 5.3c to Figure 5.3b and from Figure 5.3d to Figure 5.3b.

The ordering of train units in a train composition can be found by sorting the individual train units according to their relative positions.

Appendix C

Implementation Details and Metrics

This chapter gives a brief overview of some of the practical details relating to the software implementation of the models described in this thesis. In the following, the term *software application* is used to denote the entire software implementation needed to make the models work. The application bears the name *Io* after the innermost Galilean moon of planet Jupiter. Letters I and o are also the initials in *integrated optimisation*, *input/output*, and symbolise the binary numbers 1 and 0.

C.1 Software Application Architecture and Design

The software application for the models described in this thesis is designed as an off-line, standalone, file-based input-output application. The application is built to high modularity with implemented classes distributed across packages as seen in Table C.1. Packages are named in reversed URL order, according to convention. The overall functionality of the application is provided by the following key packages:

- **dk.dsb.io.util.input**, providing functionality to parse data from many different formats needed to build a complete rolling stock plan;
- **dk.dsb.io.entity**, providing the rolling stock plan entity data model;
- **dk.dsb.io.model**, providing the mathematical models (heuristics, upper bound calculation models, matheuristics) used to modify the entity data model;
- **dk.dsb.io.util.output**, providing functionality to output results in the form of graphs, diagrams, etc., and in the form of complete rolling stock plan readable to the input package.

A typical use of the application involves running classes from the key packages above in sequence. No graphical user interface has been implemented, user control is exercised through parameter files. Extensive logging features have been implemented, including structured, textual logging to the console and to user defined files, and logging to graphical formats as described in Appendix A.

The application is developed using the object oriented programming paradigm with low-level architectural design decisions conducted according to principles described in [93]. Wide use was made of the following, other programming paradigms:

- Encapsulation;
- Inheritance (including multiple inheritance and polymorphism);
- Abstraction (data and control);

- Generic typing (parametrisation);
- Anonymous functions (lambda expressions);
- Streams;
- Recursion;
- Unit testing (including the use of mock objects);

In the following, selected architectural considerations for the application are described according to software design patterns as originally proposed by [56]. In addition to this, Figures C.1 and C.2 show example class diagrams for selected classes from the application. Class diagrams are drawn to the unified modelling language (UML) standard [1]. The diagrams are drawn by an automated, open source, reverse engineering tool and show a varying degree of detail.

Creational software design patterns applied in the implementation of the application include:

- **Abstract factory pattern**, e. g., used to parse and instantiate different types of segment objects (representing arcs in the space-time graph). These objects are handled generally as objects of an abstract superclass, but are instantiated by factory classes as concrete subclasses. A UML class diagram of the class hierarchy for some of these class are shown in Figure C.1;

- **Builder pattern**, e. g., to construct an entire rolling stock plan based on a multitude of parameters;

- **Factory method pattern**, e. g., to perform late and repeated instantiation of objects based on previously instantiated factories passed as arguments;

Structural software design patterns applied include:

- **Bridge pattern**, widely applied to decouple class abstractions form their implementation;

- **Composite pattern**, widely applied making a single entity capture the combined properties of a collection of other objects, e. g., the class TrainUnits capturing the plural properties of TrainUnit objects, as shown in Figure C.1;

- **Flyweight pattern**, used for capturing immutable properties as a type, e. g., TrainUnitType, declared as a Java 1.8 *enum* type (a language specific feature encapsulating this pattern in the programming language itself);

Behavioural software design patterns applied include:

- **Proxy pattern**, e. g., applied to control the access to random numbers so as to be able to globally switch from deterministic pseudo-random random number generation (for debugging) to non-deterministic random number generation (for numerical experiments). The proxy pattern has also been used to handle object pools [111] so as to recycle instantiated objects to prevent time-consuming repeated object instantiation each time a new object is required;

- **Chain-of-responsibility pattern**, e. g., used to recursively force the bounds of all nodes above a given node in the branch-and-bound tree (see Section 7.5.1 on page 127). The same pattern is applied to release the bounds again once the computation of the upper bound for the given node has been performed. This may be seen on Figure C.2 for methods *force()* and *release()*;

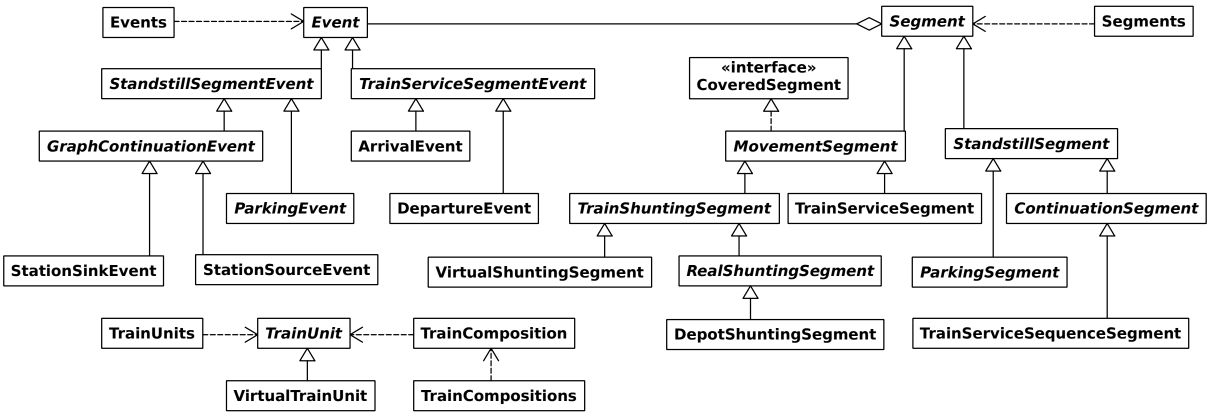

Figure C.1: UML class diagram showing class hierarchies for Segment, Event and TrainUnit classes. Simplified for compactness, no method names shown.

- **Command pattern**, e. g., in implementing different behaviour in the different branching schemes, independent of their instantiation. This may also be seen on Figure C.2;
- **Iterator pattern**, widely applied both in the form provided by the programming language and in custom forms implemented to perform special traversing of data structures, e. g., the traversing of train unit trajectories to determine the order of TrainUnit objects in a TrainComposition, i. e., in implementing the train unit order conservation principle (see Appendix B.3);
- **Mediator pattern**, e. g., to isolate the linear integer program from the arcs and vertices of the space-time graph, while still retaining their model/real-world relationships;
- **Memento pattern**, to perform rollback of changes made to a rolling stock plan when it can be determined that these changes are unwanted;
- **State pattern**, to handle different behaviour based on a state. The use of this pattern, however, turned out somewhat cumbersome. In this concrete case, a better solution would probably have been plain subclassing;
- **Strategy pattern**, applied to handle different parsing behaviour based on whether the input data for a rolling stock plan are in circulation plan form or a train unit dispatching plan form;
- **Template pattern**, widely applied to define overall, abstract behaviour that subclasses or clients can refine. Several language specific features to support this pattern are provided, including generic typing, default interface methods and lambda expressions, all of which have been widely used.

The **singleton pattern** has been deliberately avoided, as have static variables. This has been in order to prevent multiple threads interfering with each other and/or to avoid synchronised methods with intrinsic locks.

Multi-threading was only used to a small extent, this for the execution of numerical experiments. No multi-threading was implemented in the models (heuristic, bound models, matheuristic) themselves, however the multi-threading features of the commercial solver were used. Multi-treading was implemented using a completely isolated set of objects for each tread, with the only exception being a "result object" created by the parent thread and handed over to one child. The result object is used by the child thread to return the result of its calculation to the parent thread. This is sort of the reverse functionality of the **observer pattern**, because the parent thread is being informed of the termination of the child thread, at which point the result is available for further processing.

A strict coding standard was adhered to, partially controlled by automated procedures. Continuous refactoring according to [54] was conducted throughout the implementation process. Without strict coding standards and continuous refactoring it was deemed impossible to maintain and provide additional features to a code base eventually reaching approx. 28,600 lines.

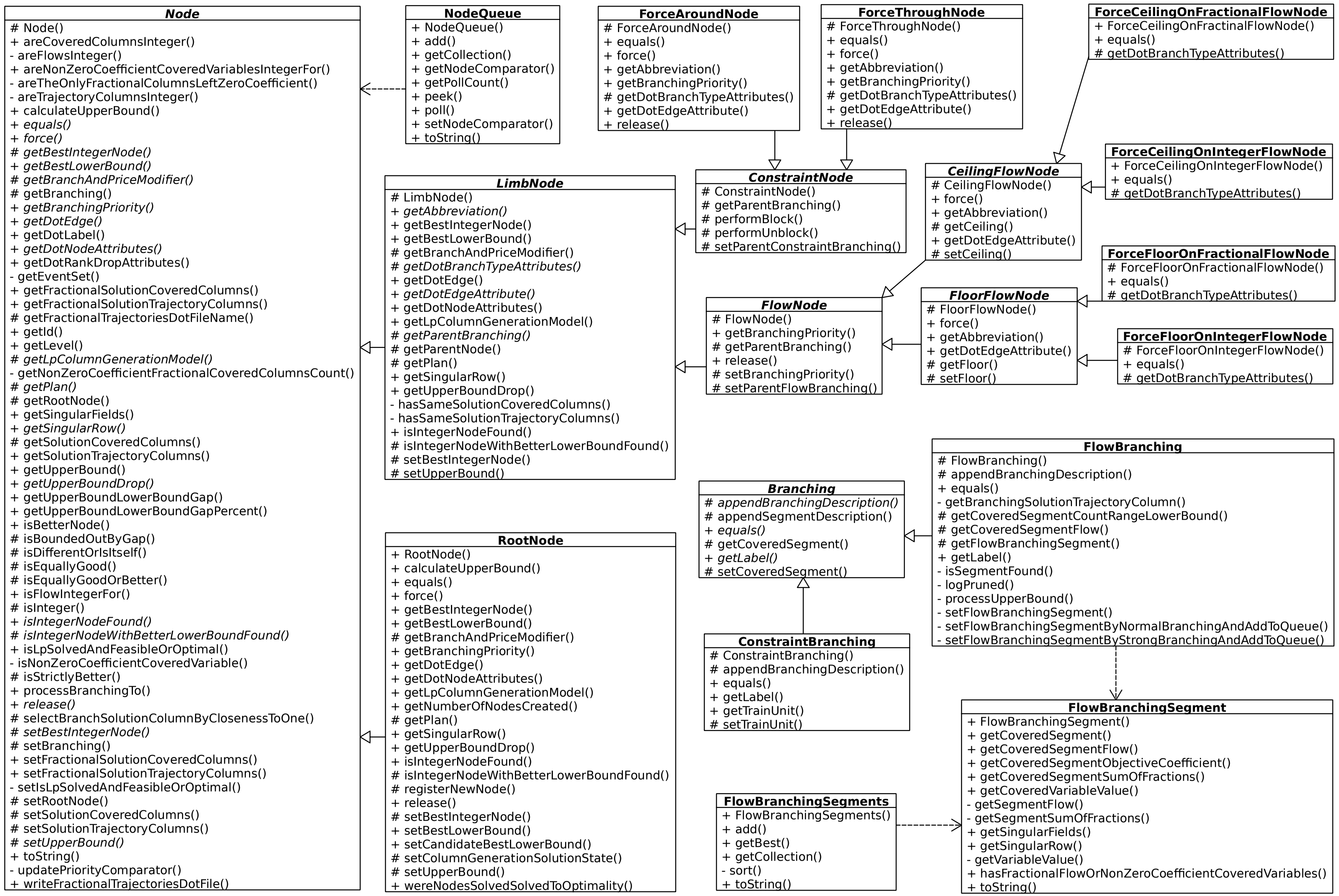

Figure C.2: UML class diagram showing the node and branching class hierarchies, the node queue and branching helper classes, including method names. These are all the classes in package dk.dsb.io.model.branching.

C.2 Software Platforms Used

The application has been implemented in the programming language *Java 1.8* using the integrated development environment (IDE) *Eclipse 4.4.1 Luna*. For version control, *git 1.9.1* was used.

Extensive memory footprint and run time performance profiling was conducted using *Java Virtual VM 1.3.8* and *Eclipse Memory Analyzer 1.6.0*.

For the purpose of unit testing, *JUnit 4.11* has been used, with test mock-up object creation provided by *Mockito 1.9.5*.

All dates and times in the model have been handled by base classes provided by *Joda-Time 2.8.2*. These classes have been used to provide railway-specific day-of-operation classes in which the date changes at 03:00h, not 00:00h.

All linear programming models and integer linear programming models have been implemented using *CPLEX 12.6.1*.

For the purpose of storing to disk plans that have been modified, and for restoring them later, the object serialiser *Kryo 3.0.3* has been utilised. Specific serialisers were provided by *Kryo-Serialisers 0.37*.

For rendering of Graphviz plots, *dot 2.36* was used, for navigation *xdot 0.6*. For SVG rendering *eog 3.10.2* was used and for SVG conversion to PDF *rsvg-convert 2.40.2*. *BTC-AsciiTable* 1.0 was used for structured textual model output to the console and to files. For reverse engineering Java code to UML diagrams *Umbrello 2.20.3* was used.

Software model development as well as numerical experiments were conducted on machines running *Ubuntu Linux 14.04 LTS*.

Table C.1: Overview of code metrics, contents and example classes by package for the models implemented in this thesis.

Package	# of classes	# of lines code	Contents	Example classes
dk.dsb.io	2	356	Main class, global parameters	Io, Parameters
dk.dsb.io.entity.event	12	557	Space-time event classes	ArrivalEvent, DepartureEvent
dk.dsb.io.entity.events	4	229	Space-time event class collections	TrainServiceSegmentEvents
dk.dsb.io.entity.movement	21	2,019	Train unit movement classes	TrainService, Trajectory
dk.dsb.io.entity.personnel	4	431	Depot driver classes	DepotDriverDuty
dk.dsb.io.entity.plan	2	761	Rolling stock plan classes	Plan, PlanType
dk.dsb.io.entity.segment	19	2,111	Space-time segment (arc) classes	Segment, TrainServiceSegment
dk.dsb.io.entity.segments	7	1,099	Space-time segment collections	Segments, ParkingSegments
dk.dsb.io.entity.station	3	493	Station related classes	Station, Stations, Distances
dk.dsb.io.entity.track	6	264	Track related classes	DepotTrack, PlatformTrack
dk.dsb.io.entity.tracks	4	295	Track class collections	SideTracks, PlatformTracks
dk.dsb.io.entity.trainunit	10	1,429	Train unit classes and collections	VirtualTrainUnit, TrainUnits
dk.dsb.io.model.branching	19	1,893	Branch-and-bound classes	RootNode, ForceAroundNode
dk.dsb.io.model.column	13	681	Linear model column classes	SolutionTrajectoryColumn
dk.dsb.io.model.enumerator	11	404	Trajectory enumerator classes	SequentialShortestPathEnumerator
dk.dsb.io.model.graph	9	844	Space-time graph classes	Label, SpaceTimeGraph
dk.dsb.io.model.heuristics	4	445	Heuristic flow control classes	ModifiableSolution
dk.dsb.io.model.mip	12	2,999	Mixed integer linear progrm. classes	LpColumnGenerationModel
dk.dsb.io.model.modifier	9	921	Heuristic modifier classes	ColumnGenerationModifier
dk.dsb.io.model.row	11	389	Linear model row classes	TrainUnitTypeRowAssociator
dk.dsb.io.util.application	2	231	Utilities for CPLEX and threads	CplexHelper, RunHelper
dk.dsb.io.util.data	7	827	Utilities for data manipulation	Comparators, SortedUniqueList
dk.dsb.io.util.file	4	316	File handling utilities	FileHelper, XmlWriter
dk.dsb.io.util.input	7	1,461	Parsers for input data	PlanReader, SegmentsFactory
dk.dsb.io.util.number	6	358	Number manipulation utilities	NumberHelper, Statistics
dk.dsb.io.util.output	12	1,436	Graphical output utilities	Plot, SvgHelper, DotGraph
dk.dsb.io.util.selector	7	259	Selection utilities	RandomTrainUnitsSelector
dk.dsb.io.util.string	6	660	String manipulation utilities	StringHelper, Table, Key
dk.dsb.io.util.time	9	664	Date and time manipulation	DayOfOperation, TimeInterval
test.dk.dsb.io.entity	8	799	Unit tests for entity packages	TrainCompositionTest
test.dk.dsb.io.model	5	237	Unit tests for model packages	ConvergenceLogTest
test.dk.dsb.io.runnables	19	2,033	Runnable, non-unit tests	PriorityQueueTest
test.dk.dsb.io.util	12	729	Unit tests for util package	NumberHelperTest
Total	286	28,630		

C.3 Software Application Code Metrics

Table C.1 shows an overview of code metrics, contents and example classes for the application.

As may be seen, 286 Java classes have been implemented to make the models presented in this thesis work. Packages with a large number of classes include dk.dsb.io.entity.movement, dk.dsb.io.entity.segment, dk.dsb.io.entity.branching which reflect large data entity class hierarchies. Package test.dk.dsb.io.runnables also has a large number of runnable tests. The average class size is approx. 100 lines of code.

Approx. 28,600 lines of code have been written to make the application work. If printed on A4 paper, this would amount to somewhere around 570 pages, more than $2\frac{1}{2}$ times the amount of pages in this thesis.

The individual package with the largest number of lines of code is the dk.dsb.io.mip package containing the classes for implementing the mixed integer linear programs. The fact that this package is the largest, reflects the condition that a lot of programming is required to keep track of the relations between CPLEX model objects (constraints, variables, etc.) and real-world data entity objects (in this case train services, depot drivers etc.). By convention, CPLEX mostly references model objects by their integer index. However, this way of referencing is error prone since indices may change when the model changes. Moreover, this way of referencing is not strongly typed, it is not typed at all. A way to get abound this issue would be to allow objects in CPLEX to be instantiated with user defined, strongly typed, parametrisised pointers to other objects. We can only hope for this feature in future releases.

Appendix D

Repairing Infeasible Rolling Stock Plans

At DSB S-tog, circulation planning is currently performed in an automated circulation planning system as the separate subprocesses *composition planning, rotation planning* and *depot planning*. As explained in Chapter 2, these subprocesses are executed sequentially. Due to the large number of practical, railway oriented requirements that a rolling stock plan needs to take into account, and the sequential way in which it is conducted in the automated circulation planning system, it may prove difficult to produce a rolling stock plan that is feasible with regard to the requirements as well as being economically attractive.

This appendix looks at the infeasibilities thus arising. Appendix D.1 describes the types of infeasibilities that may occur in a rolling stock plan and Appendix D.2 the manual procedures used by the planners to remedy them.

D.1 Characteristics of Infeasible Rolling Stock Plans

To this date, all plans created with the existing, automated circulation planning system required manual processing to ensure feasibility. The number of infeasibilities in a number of arbitrarily chosen plans varied between 7 and 43. When the existing, automated circulation planning system is not able to produce a feasible rolling stock plan, this is in all cases related to the limited parking space available in the depots. Appendix D.1.1 looks hat the types of infeasibilities that may occur and Appendix D.1.2 their detailed causes.

D.1.1 Types of Infeasibilities

D.1.1.1 Train Shunting Infeasibilities

This type of infeasibility occurs, e. g., when an arriving train composition needs to be split up and each resulting train composition driven into the depot separately. This would require more than one train shunting, which is disallowed as a business rule. In addition, more than one train shunting may need that more depot drivers be hired. This type of infeasibility can also occur in the other direction, when two train units from separate depot tracks need to be coupled at a platform track. Moreover, an infeasibility of this type can also occur when the existing, automated circulation planning system is not able to park an arriving train composition at the depot at all (not even if split).

D.1.1.2 Too Many Depot Driver Duties Required

When the existing, automated circulation planning system is not able to perform the needed train shuntings in the depot, additional (fictitious) depot driver duties are added, and the system performs another attempt at solving the problem. However, the depot duties created may not be possible to cover with the number of depot drivers hired.

Another aspect of this problem is that the new duties created by the system are subsequently utilised for other purposes than just the task that triggered the extra duty. This makes the subsequent manual processing to secure feasibility (with regard to number of depot drivers) much more complicated, since the tasks are now equally distributed among the depot drivers (real as well as fictitious).

A further aspect of this problem is also related to the sequential nature of the automated circulation planning system. In the second step (*rotation*, see Section 2.1.2 on page 30), the number of depot drivers on duty is only taken into account, not that some train shuntings in some depots may take a longer time than others, thus requiring more depot driver duties or not utilising them fully.

D.1.1.3 Maintenance Service Distance Not Set According to Fleet Characteristics

The existing, automated circulation planning system does not take into account that the physical train units show very different characteristics as to how long the train units can actually be in service before having to go into maintenance (see Figure 3.9 on page 47). Some train units are fresh out of the workshop and may drive all the way up to the service distance limit. Other train units have been running for some time and may need to go into the workshop for maintenance soon.

The existing, automated circulation planning system assigns the full service distance limit to all virtual train units in the plan at the beginning of the plan, rather than a distribution of service distances. As such, the system only guaranties that a physical train unit fresh out of the workshop may perform the actual plan (as one virtual train unit). A physical train unit close to the service distance limit will need to be reassigned to another virtual train unit in order to reach the workshop on time.

This is not a strict infeasibility, however it does make it harder for the planners to assign physical train units to the virtual train units in the train unit dispatching phase, since train units that soon need to go into maintenance may have to be reassigned.

D.1.1.4 Not Enough Train Units

Under certain conditions, the existing circulation planning system produces a plan in which more rolling stock train units are needed than are actually available. This is similar to the infeasibility type of too many depot drivers required (Appendix D.1.1.2). This type of infeasibility may occur as a result of functionality built into the system in the rotation phase to prevent the possible later occurrence of more than one train shunting prior to departure or upon arrival (Appendix D.1.1.1). Due to the complex parameter settings of the existing, automated circulation planning system it may prove difficult for planners to avoid this type of infeasibility all-together.

D.1.2 Causes for the Current Infeasibilities

D.1.2.1 Individual Steps of the Planning Process not Integrated

The root cause of the problem with creating infeasible plans lies in the fact that the different steps in the existing model are not integrated. As described in Chapter 2, all the economic decisions lie in the first step of the existing, automated circulation planning system (the *composition planning*, how long trains should be) and most of the constraints in the last step (the *depot planning*, how trains should be parked).

As a consequence of the lack of integration, some of the railway-specific requirements are modelled as soft constraints in the existing, automated circulation planning system. These requirements thus have a penalty when violations are occurring. However, if in a plan, violations are occurring that are not possible to handle in the real world, this is in fact an infeasible plan. The discrepancy between the model and the real world thus leads to the plans generated with the existing system being infeasible.

Other aspects which the system does not handle efficiently or which may lead to problems due to the lack of integration are described in the following sections.

D.1.2.2 Split Depots

Split depots are depots at stations where some of the depot tracks may only be reached from some of the platform tracks. Split depots may prove problematic when track usage rules interfere. Trains arriving at a particular platform track (as stated in the track usage rules), may not be driven into the part of the depot that is not reachable from that track (and where there may be parking capacity).

The same goes the other way: A train service that is to depart from a particular platform track, as stated by the track usage rules, may not be constituted from train compositions parked at the part of the depot from which that particular platform track may not be reached.

D.1.2.3 Night Trains Arriving in the Depot for Cleaning in the Morning

This problem arises when night trains arriving for cleaning are parked in front of the already cleaned train units. By doing so, they obstruct the passage of the cleaned train units that are to be put into service in the morning. The existing system does not take cleaning into consideration.

D.1.2.4 Half-Length Train Units Taking Up Full-Length Train Unit Space

This problem occurs when a number of depot tracks can accommodate each an integer number of full-length train units of type 1 , but when the last full-length section of each of the two depot tracks is used to park a half-length train unit of type $\frac{1}{2}$ on each. In this case, a full-length train unit type 1arriving at the depot may only be parked after one of the half-length train units has been shunted internally from one track to another. A business rule disallowing internal shunting currently prohibits this.

Due to the present requirements for the rolling stock planning of DSB S-tog (see Chapter 3), the lastly mentioned problem is currently occurring frequently at Farum station, one of the terminal stations on the A Line. Since there is no depot at Solrød Strand (one of the other two terminal stations), no train units may be coupled or decoupled there. From a passenger demand point of view, trains running on the A Line should be of composition type $1\frac{1}{2}$ in the rush hour and of composition type 1 at other times. Due to a business rule to provide flexible space at both ends of the train, the half-length train unit must always be at the northern end. Train composition movement rules demand that no train unit in revenue train service may couple to a train unit anywhere. A business rule states that no empty train unit may be left behind at a platform track by a revenue train service, the empty train unit later being driven into the depot. Since the depot at Farum station is located to the south of the station, no train unit in the northern end of a train may thus be neither coupled nor decoupled. The only way to change the train composition on the A Line is thus to drive the entire train composition of $1\frac{1}{2}$ into the depot, thus creating the problem.

D.2 Instruments to Make Infeasible Rolling Stock Plans Feasible

When problems of the types mentioned in Appendix D.1.1 occur, planners must currently repair the plans manually. When trying to make an otherwise infeasible rolling stock plan valid, the instruments described in the following sections may be used.

Note that under certain circumstances, the instruments mentioned may not be available. For example, depot space may be already be exhausted, or the required number of depot drivers may not be on duty (or hired).

D.2.1 Instruments Related to Timetabling

When facing the lack of parking space or depot driver duties with a number of arriving train units in a train at a depot, instruments related to timetabling may be used to remedy this.

It may be possible to perform non-revenue positioning to another depot station where parking space and depot driver duties may still be available.

It is also possible to perform non-revenue positioning in the opposite situation, that is, from another depot and to the origin station of a train service. In deed, that is required to solve an unparked train composition infeasibility, since the train units will need to return to the station where the infeasibility occurred to be put into service according to plan again. Otherwise other changes to the plan will be necessary.

Decisions to perform non-revenue positioning relate to timetabling in the sense that they are decisions as to which train services to run.

When planning non-revenue positioning, one should observe that the *last train* in the day arriving at its terminal station should always enter the depot of this station and not continue to another depot as non-revenue positioning. This is in order to keep the tracks free for maintenance works upon timetable finish. As such, all problems need to be solved at each terminal station before the last train arrives at that station.

Due to the infrastructure requirements of DSB S-tog, the train services on the circular F Line perform non-revenue positioning to and from Klampenborg station, the depot for this line. The train services on the F Line are as such exempt form the last train rule mentioned above.

Note that from a combinatorial point of view, a very large number of non-revenue train services are possible.

D.2.2 Instruments Related to Rotation Planning

When problems of the types mentioned in Appendix D.1.1 occur, the planners may also try to repair the plans by using instruments related to the rotation phase of the rolling stock planning process.

This includes manually locking selected turnarounds in the rotation phase of the planning process in the existing, automated circulation planning system. This is equivalent of saying: *This virtual train unit must continue as a part of this train service*.

Instruments may also include the decoupling of virtual train units of the train at a depot visited earlier, resulting in seating capacity not being offered to the passengers. The opposite is also possible: Decoupling later and providing excess seating capacity. The same goes for the coupling of virtual train units, not just the decoupling as mentioned above.

In the event of there not being enough train units to perform a rolling stock plan (see Appendix D.1.1.4), the planners must decide which train services should have assigned to them a smaller number of virtual train units in order to meet train unit balance.

The instruments mentioned may also lead to changes to the rotation as to achieve that the train units arrive at the terminal stations in a different order than before, thus enabling otherwise infeasible movements in or out of the depot.

The execution of both the rotation and the depot planning steps may then be repeated with the before mentioned locks in place. With luck, this may make the automated circulation planning system able to produce both rotation and depot plans that are feasible. Alternatively, only the execution of the depot step is repeated. In both cases, manual repair intervention is conducted inside the otherwise automatic planning process.

However, using the mentioned instruments, one should show caution in not just moving the problem to another depot.

Sometimes, if the manual changes to the rotation are few and are easily propagated manually into the depot planning phase, the execution of the rotation and depot planning steps in the existing system need not be repeated. In this case manual repair intervention is performed after the parts of the process that have been automated (and not in between automated parts).

D.2.3 Instruments Related to Depot Planning

When problems of the types mentioned in Appendix D.1.1 occur, the planners may also try to repair the plans by using instruments related directly to the depot planning phase of the rolling stock planning process. This may prove the most intuitive and simple, since it is in this phase that the actual requirements determining feasibility of the plan exist. Instruments are described in the following.

D.2.3.1 Multiple Train Units to Platform, Depart as Different Train Services

If a plan is infeasible because it is demanding too many depot driver duties (as described in Appendix D.1.1.2) in the morning, planners may chose to remedy this using the following instrument: Drive multiple train units from the depot to the platform track in the morning (thus only requiring one depot driver), and letting them depart as different non-revenue positioning train services. Note that a business rule would prevent this operation for revenue train services.

The opposite operation in the evening is also permitted as long as the arriving train services are non-revenue. However, with the current train control system, the train driver is required to call the train control centre manually in order to obtain permission to perform the operation, making it unpractical and prone to delays.

D.2.3.2 Return to Platform from Depot after Cleaning

If depot parking space is exhausted in the evening, one may provide more space in the depot by driving train units that have already been cleaned back to the platform track for overnight parking. This may of course only be conducted if there are available platform tracks and available depot driver duties to perform the operation.

Furthermore, it may not be desirable to have train units parked at the platform tracks because these tracks may be easily accessible for graffiti painters in the night time.

Currently, this operation is only performed on stations Hillerød, Farum, Køge and Frederikssund.

D.2.3.3 Shunt Internally to Fully Utilise Capacity and Achieve Desired Train Unit Order

If the problem of a half-length train unit taking up full-length train unit space is occurring (as described in Appendix D.1.2.4) or if the train units are parked in an unfortunate order (as for instance the one described in Appendix D.1.2.3), shunting internally in the depot may be performed to remedy these problems. This may also remedy infeasibilities of more than one train shunting upon arrivals and prior to departures, and make way for departing train units blocked by other train units.

The internal shunting operation will of course only be possible if there are depot driver duties available. When there is not much traffic, the main line may be used in the shunting operation.

D.2.3.4 Use Side Tracks for Temporary Parking

At Hellerup Station, two side tracks may be used to temporarily park train units between rush hours. Note the distinction between side tracks and depot tracks. Side tracks have no other purpose than parking and have no cleaning facilities. Side tracks may only be used for day-time parking.

At present, DSB S-tog has side tracks available for parking at Hellerup station. These tracks are used to park train units not in use on the F Line between rush hours.

References

[1] Unified Modeling Language, UML 2.4.1 Superstructure Specification. *Object Management Group*, (August):1–786, 2011.

[2] Danmarks Statistik - Statistics Denmark. *http://www.dst.dk/en*, 2016.

[3] Emile Aarts, Jan Korst, and Wil Michiels. Simulated Annealing. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Search Methodologies: Introductory Tutorials in Optimization and Decis. Support Techniques*, pages 187–210, 2005.

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

[5] 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, feb 2004.

[6] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.

[7] Ravindra K. Ahuja and Claudio B. Cunha. Network Models in Railroad Planning and Scheduling. In *Tutorials in Operations Research, Informs*, pages 54–101. Informs, 2005.

[8] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: Theory, Algorithms and Applications*. 2014.

[9] Ravindra K. Ahuja, James B. Orlin, Dushyant Sharma, and Larry A. Shughart. Solving Real-Life Locomotive-Scheduling Problems. *Transportation Science*, 39(4):503–517, 2005.

[10] Arianna Alfieri, Rutger Groot, Leo Kroon, and Alexander Schrijver. Efficient Circulation of Railway Rolling Stock. *Transportation Science*, 40(3):378–391, aug 2006.

[11] Ranga Anbil, John J. Forrest, and William R. Pulleybank. Column Generation and the Airline Crew Pairing Problem. *Documenta Mathematica*, pages 677–686, 1998.

[12] Jonathan F. Bard and Hadi W. Purnomo. Preference scheduling for nurses using column generation. *European Journal of Operational Research*, 164(2):510–534, 2005.

[13] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46(3):316–329, 1998.

[14] Cynthia Barnhart, Peter Belobaba, and Amedeo R. Odoni. Applications of Operations Research in the Air Transport Industry. *Transportation Science*, 37(4):368–391, nov 2003.

[15] Cynthia Barnhart, Ellis L. Johnson, Ranga Anbil, and Levent Hatay. A Column-Generation Technique for the Long-Haul Crew-Assignment Problem. In T. A. Leachman and Ciriani R. C., editors, *Optimization in Industry*, pages 8–23. John Wiley and Sons, 1994.

[16] Jeroen Beliën and Erik Demeulemeester. On the trade-off between staff-decomposed and activity-decomposed column generation for a staff scheduling problem. *Annals of Operations Research*, 155(1):143–166, 2007.

[17] Berkay Beygo. *Improvement of Decision Support System for Rail Depot Planning*. Master of science thesis, Chalmers University of Technology, Gothenburg, Sweden, 2011.

[18] Markus Bohlin. Personal communication. *Swedish Institute of Computer Science, SICS*, sep 2013.

[19] Ralf Borndörfer, Markus Reuter, and Thomas Schlechte. A Coarse-To-Fine Approach to the Railway Rolling Stock Rotation Problem. Technical report, 2014.

[20] Ralf Borndörfer, Markus Reuther, Thomas Schlechte, and Steffen Weider. A Hypergraph Model for Railway Vehicle Rotation. *OpenAccess Series in Informatics (OASIcs)*, 20:146–155, 2011.

[21] Ralf Borndörfer, Markus Reuther, Thomas Schlechte, and Steffen Weider. Integrated Optimization of Rolling Stock Rotations for Intercity Railways. Technical report, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2012.

[22] Hax Bradley. Chapter 2 - Solving Linear Programs. *Applied mathematical programming*, (1):38–75, 1977.

[23] Berit D. Brouer, J. Fernando Alvarez, Christian E. M. Plum, David Pisinger, and Mikkel M. Sigurd. A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design. *Transportation Science*, (July), 2013.

[24] Peter Brucker, Johann Hurink, and Thomas Rolfes. Routing of Railway Carriages. *Journal of Global Optimization*, pages 313–332, 2003.

[25] Jens O. Brunner and Günther M. Edenharter. Long term staff scheduling of physicians with different experience levels in hospitals using column generation. *Health Care Management Science*, 14(2):189–202, 2011.

[26] Sofie Burggraeve, Simon Henry Bull, Pieter Vansteenwegen, and Richard Martin Lusby. Integrating robust timetabling in line plan optimization for railway systems. *Transportation Research Part C (in print)*, 2016.

[27] Valentina Cacchiani, Alberto Caprara, and Paolo Toth. A Fast Heuristic Algorithm for the Train Unit Assignment Problem. In Daniel Delling and Leo Liberti, editors, *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 25 of *OpenAccess Series in Informatics (OASIcs)*, pages 1–9, Dagstuhl, Germany, 2012.

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

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

[30] Gabrio Caimi, Leo Kroon, and Christian Liebchen. Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 2016.

[31] Alberto Caprara, Matteo Fischetti, and Paolo Toth. Modeling and Solving the Train Timetabling Problem. *Operations Research*, 50(5):851–861, 2002.

[32] Alberto Caprara, Matteo Fischetti, Paolo Toth, Daniele Vigo, and Pier Luigi Guida. Algorithms for railway crew management. *Mathematical Programming*, 79(1-3):125–141, 1997.

[33] Alberto Caprara, Metteo Fischetti, and Paolo Toth. A Heuristic Method for the Set Covering Problem. *Operations Research*, 47(5):730–743, 1999.

[34] Alberto Caprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. Passenger Railway Optimization. In *Handbook in OR &MS, Vol. 14*, volume 14. Elsevier, 2007.

[35] Marielle Christiansen, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen. Chapter 4 Maritime Transportation. *Handbooks in Operations Research and Management Science*, 14(C):189–284, 2007.

[36] Marielle Christiansen, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483, 2013.

[37] Jean-François Cordeau. A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34:133–149, 2000.

[38] Jean-François Cordeau, François Soumis, and Jacques Desrosiers. Simultaneous Assignment of Locomotives and Cars to Passenger Trains. *Operations Research*, 49:531–548, 2001.

[39] Jean-François Cordeau, Paolo Toth, and Daniele Vigo. A survey of optimization models for train routing and scheduling. *Transportation science*, 1998.

[40] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms, Second Edition*. The MIT Press, Cambridge Massachusetts, 2001.

[41] COWI. External Costs of Transport. 3rd Report - Total external costs of road and rail transport in Denmark. *Danish Ministry of Transport*, (July), 2004.

[42] Bernd Dammann. Photograph reproduced with kind permission. 2016.

[43] J. Desrosiers and M. E. Lübbecke. Branch-Price-and-Cut Algorithms. *Wiley Encyclopedia of Operations Research and Management Science*, pages 1–18, 2010.

[44] Jacques Desrosiers, Jean Bertrand Gauthier, and Marco E. Lübbecke. Row-reduced column generation for degenerate master problems. *European Journal of Operational Research*, 236(2):453–460, 2014.

[45] Jacques Desrosiers and Marco E. Lübbecke. A primer in column generation. *Column Generation*, 3:1–32, 2005.

[46] Jacob Dirksen. Parking and Litra Assignment for S-tog. Technical report, Technical University of Denmark, DTU Management Engineering, 2010.

[47] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237, 1999.

[48] Issmail Elhallaoui, Guy Desaulniers, Abdelmoutalib Metrane, and François Soumis. Bi-dynamic constraint aggregation and subproblem reduction. *Computers and Operations Research*, 35(5):1713–1724, 2008.

[49] Issmail Elhallaoui, Daniel Villeneuve, François Soumis, and Guy Desaulniers. Dynamic Aggregation of Set-Partitioning Constraints in Column Generation. *Operations Research*, 53(4):632–645, 2005.

[50] European Railway Agency. Railway safety performance in the European Union 2014. 2014.

[51] 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.

[52] Matteo Fischetti, Domenico Salvagnin, and Arrigo Zanette. Fast Approaches to Improve the Robustness of a Railway Timetable. *Transportation Science*, 43(3):321–335, 2009.

[53] Peter Foens. *Desicion Support for Depot Planning in the Railway Industry*. PhD thesis, Technical University of Denmark, 2006.

[54] Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts. Refactoring: Improving the Design of Existing Code. pages 1–337, 1999.

[55] 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, may 2005.

[56] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns - Elements of Reuseable Object-Oriented Software. Number February, pages 1–429, 1995.

[57] Giovanni Luca Giacco, Andrea D'Ariano, and Dario Pacciarelli. Rolling Stock Rostering Optimization Under Maintenance Constraints. *Journal of Intelligent Transportation Systems*, 18(1):95–105, jan 2014.

[58] Sander Greenland and Babette Brumback. An overview of relations among causal modelling methods. *International journal of epidemiology*, 31(5):1030–1037, 2002.

[59] Mattias Grönkvist. Accelerating column generation for aircraft scheduling using constraint propagation. *Computers & Operations Research*, 33:2918–2934, 2006.

[60] Jørgen Thorlund Haahr, Richard Martin Lusby, Jesper Larsen, and David Pisinger. A Branch-and-Price Framework for Railway Rolling Stock Rescheduling During Disruptions. Technical report, Technical University of Denmark, Dept. of Management Engineering, 2014.

[61] Ali E. Haghani. Rail Freight Transportation: A Review Of Recent Optimization Models For Train Routing And Empty Car Distribution. *Journal of Advanced Transportation*, 21(2):147 – 172, 1987.

[62] Frank Harary. *Graph Theory*. Addison Wesley, 1969.

[63] J. Hickman, D. Hassel, R. Joumard, Z. Samaras, and S. Sorenson. Methodology for Calculating Transport Emissions and Energy Consumption. *Transport Research Laboratory for The European Commission*, 1999.

[64] Dennis Huisman, Leo Kroon, Ramon M. Lentink, and Michiel J. C. M. Vromans. Operations Research in passenger railway transportation. *Statistica Neerlandica*, 59(4):467–497, nov 2005.

[65] Thore Husfeldt. Graph colouring algorithms. In L. W. Beineke and Robin J. Wilson, editors, *Topics in Chromatic Graph Theory*, chapter XIII, pages 277–303. Cambridge University Press, 2015.

[66] Stefan Irnich and Guy Desaulniers. Shortest Path Problems with Resource Constraints. In *Column Generation*, pages 33–65. Springer, 2005.

[67] Krishna C. Jha, Ravindra K. Ahuja, and Güvenç Sahin. New Approaches for Solving the Block-to-Train Assignment Problem. *Networks*, 51(1):48–62, 2008.

[68] Christian Vad Karsten, Berit Dangaard Brouer, Guy Desaulniers, and David Pisinger. Time constrained liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 2015.

[69] Ed Klotz and Alexandra M. Newman. Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science*, 18(1):18–32, 2013.

[70] Leo Kroon and Matteo Fischetti. Crew Scheduling for Netherlands Railways "Destination: Customer". *ERIM Report Series Research In Management*, 2000.

[71] Leo Kroon, Dennis Huisman, Erwin Abbink, Pieter-Jan Fioole, Matteo Fischetti, Gábor Maróti, Alexander Schrijver, Adri Steenbeek, and Roelof Ybema. The New Dutch Timetable: The OR Revolution. *Interfaces*, 39(1):6–17, feb 2009.

[72] Leo Kroon, Ramon M. Lentink, and Alexander Schrijver. Shunting of Passenger Train Units: An Integrated Approach. *Transportation Science*, 42(4):436–449, nov 2008.

[73] Jes Larsen. Photograph reproduced with kind permission. *Territorium Photography APS*, 2012.

[74] E. B. G. Lindgreen and S. C. Sorenson. Simulation of energy consumption and emissions from rail traffic. Technical report, 2005.

[75] M.E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, pages 1007–1023, 2005.

[76] Sean Luke. *Essentials of Metaheuristics*. George Mason University, 2014.

[77] R. Lusby, A. Dohn, T. Range, and J. Larsen. A column generation-based heuristic for rostering with work patterns. *Journal of the Operational Research Society*, 63:261–277, 2011.

[78] Henning Makholm. A collection of track maps for Danish railways, track map A, 2012.

[79] Etienne-Jules Marey. *La Méthode graphique dans les sciences expérimentales et principalement en physiologie et en médecine*. G. Masson, Paris, 1885.

[80] Gábor Maróti and Leo Kroon. Maintenance routing for train units: The interchange model. *Computers & Operations Research*, 34(4):1121–1140, apr 2007.

[81] Morten N. Nielsen, Bjørn Hove, and Jens Clausen. Constructing periodic timetables using MIP - A case study from DSB S-train. *International Journal of Operational Research*, 1(3):213–227, 2006.

[82] Otto Anker Nielsen. Personal communication. *Technical University of Denmark, DTU Transport, Dept. of Transport*, aug 2013.

[83] James Ostrowski. Orbital Branching. *Mathematical Programming*, Volume 126(1):147–178, 2011.

[84] James Ostrowski, Miguel F Anjos, and Anthony Vannelli. Symmetry in Scheduling Problems. pages 1–17, 2010.

[85] Judea Pearl. Causal Diagrams for Empirical Research. *Biometrika Trust, Oxford University Press*, 82(4):669–688, 1995.

[86] Marc Peeters and Leo Kroon. Circulation of railway rolling stock: A branch-and-price approach. *Computers & Operations Research*, 35(2):538–556, feb 2008.

[87] Marc Pfetsch. Multicommodity Flows and Column Generation. In *Ganzzahlige Optimierung im Öffentlichen Verkehr*. Zuse Institute Berlin, 2006.

[88] Matias S. Rasmussen, Richard M. Lusby, David M. Ryan, and Jesper Larsen. Subsequence Generation for the Airline Crew Pairing Problem. Technical Report May, Technical University of Denmark, DTU Management Engineering, 2011.

[89] Andreas Razen. Einsatz-Management von Loks und Lokpersonal im Güterverkehr der SBB Cargo. Technical report, Institut für Operations Research, ETH Zürich, Zürich, 2004.

[90] N. J. Rezanova. *The train driver recovery problem - Solution method and decision support system framework*. PhD thesis, Technical University of Denmark, DTU Management Engineering, 2009.

[91] Natalia J Rezanova. Line Planning Optimization at DSB. *Proceedings of the 13th Conference on Advanced Systems in Public Transport (CASPT) 2015. Erasmus University*, 2015.

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

[93] Arthur J. Riel. *Object-Oriented Design Heuristics*. Addison Wesley, 1996.

[94] Stefan Røpke and David Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4):455–472, 2006.

[95] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. *Computer Scheduling of Public Transport*, pages 269–280, 1981.

[96] David M. Ryan. The Solution of Massive Generalized Set Partitioning Problems in Aircrew Rostering, 1992.

[97] Ihsan Sabuncuoglu, Yasin Gocgun, and Erdal Erel. Backtracking and exchange of information: Methods to enhance a beam search algorithm for assembly line scheduling. *European Journal of Operational Research*, 186(3):915–930, 2008.

[98] Mohammed Saddoune, Guy Desaulniers, Issmail Elhallaoui, and François Soumis. Integrated Airline Crew Pairing and Crew Assignment by Dynamic Constraint Aggregation. *Transportation Science*, 46(1):39–55, feb 2012.

[99] Anita Schöbel. Line planning in public transportation: Models and methods. *OR Spectrum*, 34(3):491–510, 2012.

[100] Louise K. Sibbesen. *Mathematical models and heuristic solutions for container positioning problems in port terminals*. PhD thesis, 2008.

[101] Kevin Sim, Emma Hart, and Ben Paechter. A Lifelong Learning Hyper-heuristic Method for Bin Packing. *Evolutionary Computation*, 23(1):37–67, 2015.

[102] René Strandbygaard. Photograph reproduced with kind permission. 2004.

[103] Per Thorlacius. Rolling Stock Planning at DSB S-tog: Processes, Cost Structures and Requirements. Technical Report. *DTU Management Engineering*, pages 1–32, 2015.

[104] Per Thorlacius and Jens Clausen. Scheduling of inspectors for ticket spot checking in urban rail transportation. *Trafikdage På Aalborg Universitet*, pages 1–14, 2010.

[105] Per Thorlacius, Jesper Larsen, and Marco Laumanns. An integrated rolling stock planning model for the Copenhagen suburban passenger railway. *Journal of Rail Transport Planning and Management*, 5(4):240–262, 2015.

[106] Trafikministeriet. Kontrakt mellem Trafikministeriet og DSB S-tog a/s om S-togstrafikken udført som offentlig service i perioden 2005-2014, 2004.

[107] Transportministeriet. TEMA2015 - Et værktøj til beregning af transporters energiforbrug of emissioner i Danmark (Transportation Energy and Emmision Calculation Model of the Danish Ministry of Transport). Technical report, Transportministeriet, Cowi, 2015.

[108] Edward R. Tufte. *Beautiful Evidence*. Graphics Press, 2006.

[109] Balachandran Vaidyanathan, Ravindra K. Ahuja, Jian Liu, and Larry A. Shughart. Real-life locomotive planning: New formulations and computational results. *Transportation Research Part B: Methodological*, 42(2):147–168, feb 2008.

[110] François Vanderbeck and Laurence A. Wolsey. An exact algorithm for IP column generation. *Operations Research Letters*, 19(4):151–159, 1996.

[111] Alexandre Sieira Vilar. What's Object Pooling All About? *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(8), 2000.

[112] Maya Voegeli. Integrated Timetabling and Passenger Assignment for Capacitated Public Transportation Networks. *Masters thesis*, pages 1–50, 2014.

[113] Laurence A. Wolsey. *Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, 1998.

Some men write their lives to save themselves from ennui, careless of the amount they inflict on their readers.

Others write their personal history, lest some kind friend should survive them, and, in showing off his own talent, unwittingly show them up.

Others, again, write their own life from a different motive – from fear that the vampires of literature might make it their prey.

I have frequently had applications to write my life, both from my countrymen and from foreigners. Some caterers for the public offered to pay me for it. Others required that I should pay them for its insertion; others offered to insert it without charge. One proposed to give me a quarter of a column gratis, and as many additional lines of eloge as I chose to write and pay for at ten-pence per line. To many of these I sent a list of my works, with the remark that they formed the best life of an author; but nobody cared to insert them.

I have no desire to write my own biography, as long as I have strength and means to do better work.

Charles Babbage, "Passages from the Life of a Philosopher", 1864