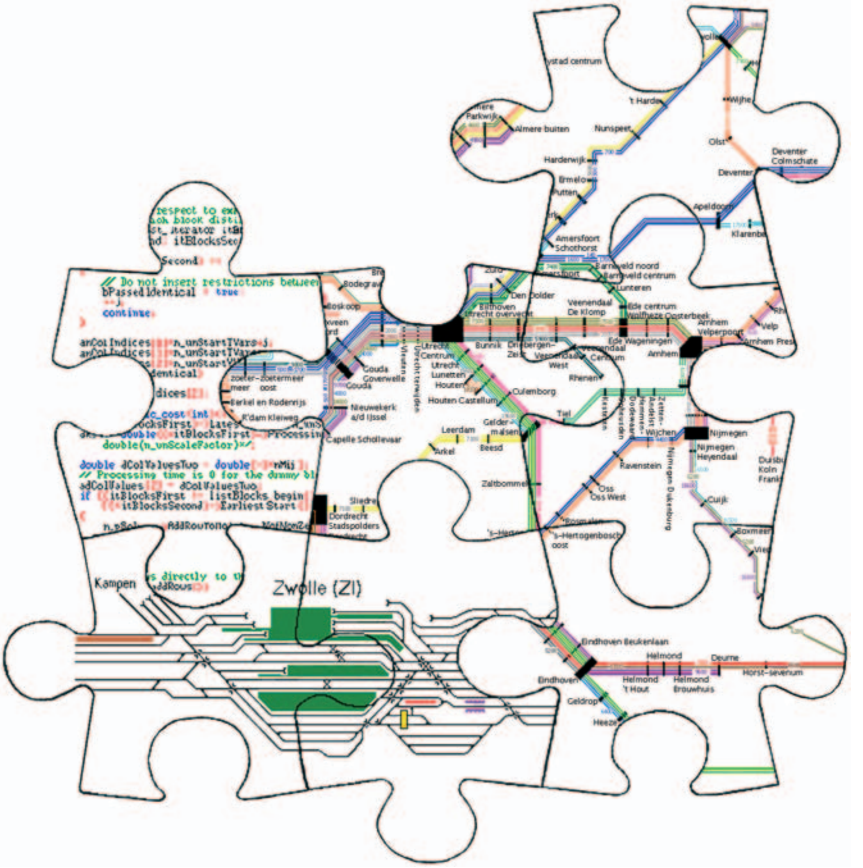
RAMON M. LENTINK

Algorithmic Decision Support for Shunt Planning



Algorithmic Decision Support for Shunt Planning

Ramon Martijn Lentink

Algorithmic Decision Support for Shunt Planning

Algoritmische Beslissingsondersteuning voor Rangeerplanning

Proefschrift

ter verkrijging van de graad van doctor aan de
Erasmus Universiteit Rotterdam
op gezag van de
rector magnificus
Prof.dr. S.W.J. Lamberts
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

vrijdag 10 februari 2006 om 11:00 uur

door

Ramon Martijn Lentink

geboren te Apeldoorn

Promotiecommissie

Promotoren:	Prof.dr. L.G. Kroon
	Prof.dr.ir. J.A.E.E. van Nunen
Overige leden:	Prof.dr. A.P.M. Wagelmans
	Prof.dr. A. Schrijver
	Prof.dr.ir. J.W.M. Bertrand

Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus University Rotterdam
Internet: http://www.erim.eur.nl
ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM Ph.D. series Research in Management 73
TRAIL Thesis series T2006/2, The Netherlands TRAIL Research School

ISBN-10: 90-5892-104-2
ISBN-13: 978-90-5892-104-8
Design: B&T Ontwerp en advies www.b-en-t.nl
Print: Haveka www.haveka.nl

Cover Illustration: Bearene Laferte

Acknowledgements

Richard Freling guided the first steps in my professional career. His enthusiasm, his skills of applying Operations Research to real-life problems, and his views on life in general still inspire me deeply. Daily, I use his tips and tricks for solving all kinds of problems. After an illness of 5 months, Richard passed away in January 2002 at the age of 34. I will remember his sincere interest in others and his positive attitude. I am grateful for his help in writing this thesis.

Furthermore, I would like to thank my promotors Leo Kroon and Jo van Nunen. Leo, your patience in reading each draft of many manuscripts and your eye for detail have been recognized before by many, and have improved this thesis as well. Moreover, I appreciate the freedom you gave me to pursue research directions in which I was interested. Jo, you are the best manager I have encountered. You often surprised me with your genuine interest in peoples lives and you are a great motivator. More specifically, I appreciate your efforts to clear the path for my part-time Ph.D. research, while your comments greatly improved the overall structure and coherency of this thesis.

The strive for high-quality at the Rotterdam School of Management and the Erasmus Research Institute of Management is clearly present. Therefore, I am proud to receive my Ph.D. from Erasmus University. Furthermore, my department there was a pleasant environment to work on this thesis. Besides discussions on other interesting topics, my colleagues also helped me with many small steps in the preparation of the thesis. Jacqueline, as my first roommate and fellow inhabitant of Gouda, I thank you for the numerous discussions on a broad range of serious and less serious subjects and for your comments on a draft of this thesis. Moreover, I also appreciate the fact that the you often drove me around when I ruptured my ankle ligaments. João, as my second roommate, I thank you for interesting me in the international focus of our department.

In addition, I would like to thank my paranoias. Dennis, I thank you for many things including carefully reading an entire preliminary version of this thesis, accompanying me to several conferences at beautiful places, working together on some projects and papers, and for being a good friend in general. Michiel, I am grateful for our joint journey from Masters to Ph.D.'s. In this respect, I enjoyed our trips to AMORE conferences (the train trip to Konstanz!), your amazing knowledge of railway related facts, and all kinds of railway discussions on Fridays. Moreover, I thank you for your comments on an earlier draft and sharing you experiences on several procedures.

I would like to take the opportunity to thank the other members of my Ph.D. committee for devoting their effort to reading this thesis: Lex Schrijver, Will Bertrand, Albert Wagelmans, René de Koster, René Jorna, Gerrit Timmer, and Steef van de Velde.

In order to understand the shunt planning problem, numerous people at NS Reizigers explained relevant practical aspects to me. Bert, I enjoyed your enthusiasm in our discussions as well as your open mind to the advanced support to which this thesis might lead. Moreover, from a more scientific point of view, work by and discussions with Marinus, Pieter-Jan, and Gábor improved my understanding.

A valuable aspect of scientific work is cooperation with others in writing publications. The work of Richard Freling, Leo Kroon, Dennis Huisman, Albert Wagelmans, Pieter-Jan Fioole, Cor van 't Woudt, Michiel Vromans, and Lex Schrijver has considerably improved our papers and therefore I thank you all. I am especially grateful for the efforts of those, who helped us win the first student paper competition in railways.

During my Ph.D. trajectory, ORTEC provided the opportunity to remained part-time employed. Especially my colleagues at the Aviation and Traffic Management department provided nice practical projects, providing the right professional perspective for my research. Thank you Ard, Franka, Gerbrand, Jethro, Paul, and Stefan for your interest in my research and the pleasant distractions. Thank you Joaquim for your comments on a draft of the thesis.

In my private life, many have helped me to change my mind from this thesis whenever necessary (especially Daniël, Erik, Johan, Mark, Maarten, and my fellow futsallers), while also many showed their interest in my research (including Erick, Frits, Henny, Nicky, Niels and Pauline, Patrick, Peter Bosgoed, and Peter and Kirsten). Thank you!

De laatste woorden in dit voorwoord wil ik graag in het Nederlands schrijven. Deze richt ik eerst aan Cynthia, Karen en Richard, en Mienke en Herman: bedankt voor jullie interesse en ondersteuning in de afgelopen jaren. Nu dit project is afgerond zie ik er naar uit om een deel van de vrijgekomen tijd samen met jullie te besteden. Pap en mam, zonder jullie voortdurende steun zou ik waarschijnlijk nooit aan dit proefschrift begonnen zijn. Dat het afronden ervan is gelukt, is dus ook voor een deel jullie prestatie. Mam, met name jouw doorzettingsvermogen inspireert mij. Tenslotte natuurlijk Bearene: ondanks dat jij het meeste voordeel hebt gehad van mijn onderzoek, heb jij zeker ook de meeste nadelen ervan ondervonden. Als het onderzoek goed ging was ik bijna alleen maar bezig met mijn onderzoek, terwijl ik behoorlijk humeurig kon zijn als het wat minder ging. Met veel plezier denk ik nu aan onze toekomst, waarin we snel onze gezamenlijke dromen gaan vervullen.

Ramon Lentink
Rotterdam, February 2006

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Shunting

This thesis describes algorithmic decision support for railway planners who create operational plans for shunting processes at shunt yards. These processes involve parking of rolling stock, routing of rolling stock over the local railway infrastructure, and cleaning and maintenance of rolling stock. Plans for these processes are called `shunt plans`. The planning process related to these shunting processes is called `shunt planning`.

Operational shunt planning is a critical part of the planning process of a railway passenger operator. Because it is highly sensitive to changes in previous planning processes, like the rolling stock circulation and the timetable for the trains, it is performed as late as possible, resulting in a high time pressure on operational shunt planning. Moreover, due to predicted increases in railway passenger traveling, more rolling stock will be required and efficient shunt planning will become even more crucial. Defining advance decision support for shunt planners to speed up shunt planning is challenging, since different planners apply different problem solving strategies. Moreover, the shunt planning problems these planners face are difficult problems. Similar problems also occur for different modes of transport, such as buses and trams. This chapter discusses these applications. In addition, we state the aim and research questions of the thesis. After introducing the relevance and demarcation of the thesis, the chapter concludes with an outline of how we work towards the aim of our research.

1.1 SHUNTING IN RAILWAY PASSENGER TRANSPORTATION

A typical characteristic of passenger transportation in general are the `rush hours`, where the demand for transportation peaks. These peaks occur on weekdays between approximately 7:00 and 9:00 in the morning and between 16:00 and 18:00 in the afternoon.

Within these rush hours, all rolling stock of a passenger railway operator is used, except for the rolling stock in maintenance. However, outside these rush hours, an operator usually has a surplus of rolling stock. The idle rolling stock can be parked at a shunt yard to be able to fully exploit the main physical infrastructure for railway passenger services or freight transportation. Typically, some parking is required between the morning and the afternoon rush hours, while nearly all rolling stock needs to be parked during the night. Of course, this parking requires routing train units from their arrival platforms to shunt tracks and from shunt tracks to their departure platforms.

During its stay at a shunt yard, rolling stock generally undergoes several processes, e.g. internal and external cleaning of the rolling stock, but also maintenance checking and small repairs.

Operating these local processes is called `shunting`. It arises for all kinds of public transport: trams, buses, trains, and also for parking taxis. In general, shunting is strongly influenced by the characteristics of the local shunt yard, the local management, and the operational policies of the yards. In this thesis, we focus on railway shunt planning of `train units`, which are automotive and bi-directional sets of railway carriages. As opposed to train units, railway carriages require a locomotive in order to move.

The rolling stock is generally partitioned into several families, each with its own characteristics. Examples are double-deck train units, and train units with faster acceleration and deceleration. One specific family typically consists of two types. A specific type within a specific family is discerned from the other type in the same family by its number of carriages. Examples of the characteristics of a type are its length and its seating capacity for first class and second class passengers. Substantial differences exist between different rolling stock families.

The railway passenger services at a certain station can be seen as a sequence of arrivals and departures of trains. It is common that this sequence is given as a timetable, with planned times of the arrivals and departures.

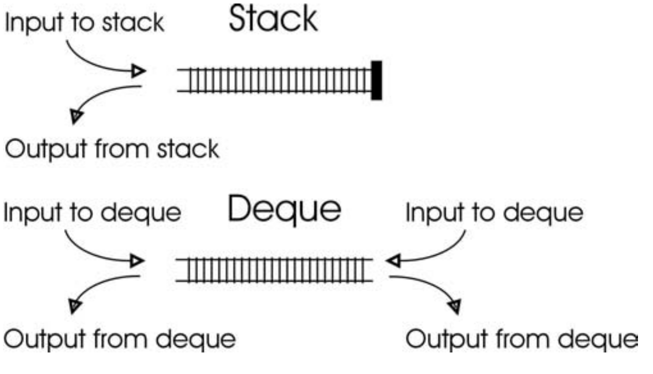


Figure 1.1: Two configurations of a shunt track.

The shunt yard consists of a set of tracks at which rolling stock can be parked. Each track has a stack or a deque configuration, depicted in Figure 1.1. A track with a stack configuration will be called a LIFO track and a track with a deque configuration will be called a free track. At a LIFO track, rolling stock can only enter and leave the track from one side. Note that rolling stock must be able to move bi-directionally if it is to be parked at a LIFO track without a locomotive. At a free track, rolling stock can enter and leave from both sides of the track. In this case, it is possible to arrive from the left side and depart to the right side and vice versa, which is called a general deque.

Important aspects of shunt planning are:

- Conflicts between routes over the physical station infrastructure.
- Requirements for rolling stock to undergo certain processes, like cleaning and maintenance checks, during its stay.
- Rolling stock of a certain type blocking the arrival or departure of other rolling stock of a different type.
- Preferences and restrictions for parking rolling stock at certain tracks and via certain sides, in case of free tracks.
- Robustness providing flexibility regarding small disturbances of e.g. the arrival or departure sequence.
- Efficiency, which means that a minimum amount of resources is required for operating the plans.

Approximately 30 out of 380 Dutch stations have shunt yards. For these stations, shunt plans are created manually, which requires much effort and time. At Netherlands Railways Passengers (in Dutch: Nederlandse Spoorwegen Reizigers, NSR) approximately 130 planners out of 350 are working on operational shunt planning. The creation of shunt plans for a timetable requires approximately 4 months of throughput time and is typically based on the previous timetable. In addition, modifications of a shunt plan require up to several days [Fioole, 2003].

Shunt planning is one of the final elements of the complete planning process underlying a railway system: every modification of the timetable or the rolling stock circulation will require modifications of shunt plans at some stations. Therefore, it is important that these processes have a minimum throughput time and that they can be carried out as closely as possible to the end of the overall logistic planning process. Postponing shunt planning decreases the uncertainty regarding input data. In turn, the improved quality of the data results in less operational effort for NSR. Moreover, less modifications to the plans during the operations are required, which improves the service to the passengers.

1.2 PLANNING PROBLEMS FOR PASSENGER RAILWAYS

This paragraph describes how shunt planning fits in the overall planning process of a railway system with passenger services. In Figure 1.2, the typical planning process of a railway system for passengers is depicted. All elements in the figure are aimed at satisfying the demand for passenger transportation in a profitable manner. The overall planning process is a very complicated process. Therefore the process is decomposed, while relations between subprocesses are taken into account. For example, the timetable resulting from the timetabling process typically needs approval from an infrastructure manager. After this approval, one can continue the planning process with the subsequent subprocess: the scheduling of rolling stock. It is possible that a subprocess is infeasible given the plans of previously planned subprocesses. In such a case, changes in one or more plans of previous subprocesses are required. In this paragraph, we discuss the subprocesses in a little more detail.

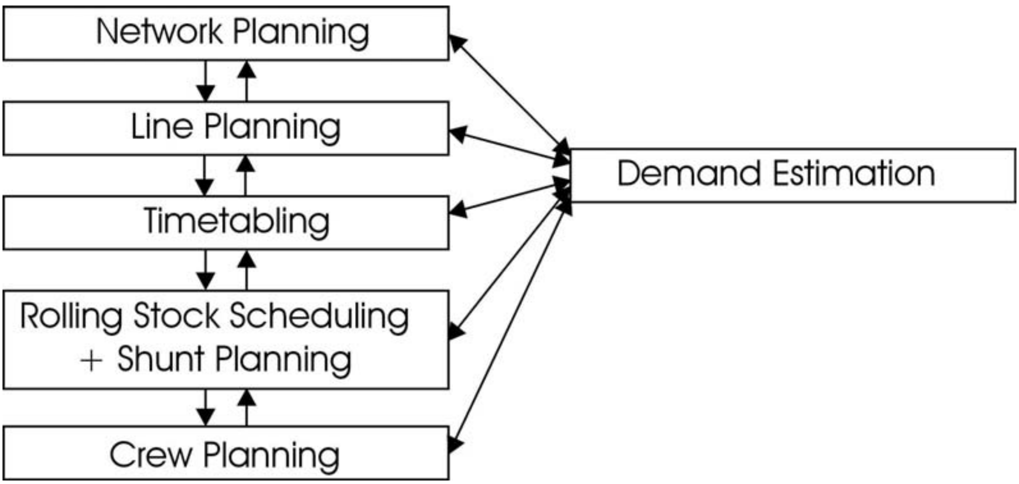


Figure 1.2: The typical planning process of a passenger railway operator.

Network planning focuses on the design of the network that will be used for operating the public transport services. This network includes stations, yards as well as tracks. The network is in general historically grown and owned by the government. Therefore, politics has a strong influence on the design of the network, since it sets the budgets for maintenance and expansion of the network.

The line planning problem determines origins and destinations, and frequencies of lines in the network. In addition, it determines which origins and destinations will be served by a direct connection.

In timetabling, one tries to assign arrival and departure times to the services that will be operated on all railway lines. These arrival and departure times have to meet certain restrictions, such as travel times, and dwell times at a platform for letting people board and alight. This results in a set of trips, called the timetable.

Rolling stock scheduling assigns rolling stock to the timetabled services, thereby also deciding on the configurations of the trains for specific trips. The configuration of a train consists of an ordered number of train unit types. The goal of this process is to determine a good mix of efficiency of the rolling stock circulation on one hand, and comfort for the passengers on the other. Shunt planning focuses on planning the local shunting processes at different stations, including maintenance planning of the rolling stock.

Crew planning combines trips into cyclic anonymous rosters for crews. The crew plan needs to comply with laws and union regulations. Typically, crew planning is decomposed even further. In the first phase of this decomposition, daily shifts are generated independently of the crew. These anonymous duties are combined into anonymous cyclic rosters in a second phase. Another possibility is the combination of these duties into a set of personalized rosters for crews. Freling et al. [2004] provide a comparison of a decomposed approach generating personalized rosters with an integrated approach. They conclude that the quality of the anonymous duties have a significant impact on the quality of the set of personalized rosters and on the number of tasks not scheduled by a dedicated algorithm.

Obviously, these planning processes require the estimated demand for railway passenger transportation as input. Furthermore, feedback from planning processes to the demand estimation as well as to earlier planning processes is of interest. For instance, a high quality of the offered services will probably result in more people using railway transportation, and thus in an increased demand and more revenues for railway transportation. Moreover, increased chances on encountering aggressive travellers on a train result in more conductors working on this train and therefore demand estimation and crew planning are related. Finally, the estimated demand is also influenced by external factors, such as fuel prices and governmental policies, with respect to e.g. road transport and physical railway infrastructure.

The main sources for demand estimation are ticket sales, passenger counts, and passenger interviews. Moreover, a model for studying policy scenarios regarding national mobility is also available [Adviesdienst Verkeer en Vervoer website, 2005]. Finally, in the near future a new ticketing and fare collection system will be introduced in the Netherlands. This system is based on smart cards and will provide a wealth of information. This information can be used among others to improve passenger demand estimation and pricing strategies [Li, 2005].

In Figure 1.2, "optimal" solutions in a higher planning process might severely restrict lower planning processes, resulting in sub-optimal solutions for the whole system.

Therefore, one would like to take into account the objectives and restrictions of the lower level planning processes, while planning the higher level processes. Usually, experienced planners are good at considering such issues, typically in an implicit manner. In addition, a solution is only optimal given a certain objective function, and even defining a proper objective for real-life problems can be very complicated.

During the operations, disturbances are likely to occur and plans might need adjustments. In such a case, a robust plan will require small adjustments for maintaining a fairly high quality. Therefore, some measures of robustness can be taken into account in the different elements of the planning process. Simulation of the operations is typically used to determine the robustness of plans.

More elaborate discussions on these planning problems can be found in Bussieck et al. [1997] and in Goossens [2004].

1.3 SETTING THE SCENE

After the overview of the planning process, we continue with a short description of current developments in the European railway industry. Rail transport within the European Union (EU) is subject to change, caused by a changing business environment. We introduce EU-15 as the 15 member states of the European Union before the enlargement of the union in May 2004. The member countries of EU-15 are: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, Portugal, Spain, Sweden, the Netherlands and the United Kingdom. In the period 1990 - 2001, the total railway network of the EU-15 countries decreased by 7% in length. This is caused for a large part by significant reductions of the large networks in Germany, France and Spain. In the same period, the total size of the highway network for the same countries increased by more than 30% in length. If we account for the significant growth in the number of cars per 1000 inhabitants, the difference in the developments of the network sizes is striking. In 2000, the total number of railway passenger kilometers of the EU-15 countries increased by 10% compared to 1990 on these significantly smaller networks. Growth in passenger transportation by passenger cars and buses and coaches shows similar figures, while passenger air transportation increased by 79% in this period. The share of railway passenger transportation remains fairly stable. Concluding, we see that although the railway networks are being used more intensively, the share of passenger railway transportation of the total passenger transportation is not increasing. The data of this analysis are publicly available from the Statistical Office of the European Communities [2004].

In 1991, the European Commission issued the policy directive 91/440/EEC. This directive requires separate organizations to perform management of the physical infrastructure on one hand and transport on the other hand. Similar changes have been introduced for other transport modes and result from the ideas of open markets and free movement of people and goods within the EU. In general, the government takes care of the physical infrastructure, while the operators are private organizations. Now, nearly fifteen years later, many European railways are still far away from a fully liberalized railway system [European Commission, 2001].

International competition is complicated due to technical operational differences between countries, such as power supply and the width of the trains and tracks [Heimerl, 1997]. Moreover, quite strict regulations apply regarding authorizations of train crews for specific rolling stock and lines. These regulations result in another obstacle for smooth international railway operations. Recently, Ministers of Transport from the countries of the European Union agreed upon a European drivers license for train drivers, which might reduce such obstacles in the future.

The Dutch railway network is heavier utilized than any other Western-European network in terms of number of train kilometers per kilometer of network [Poort, 2002]. However, this high utilization does not necessarily result in low performance regarding punctuality of the train services. Despite the Dutch public opinion, this Dutch performance in Western Europe was in 2000 only exceeded by the Swiss performance [NS (Nederlandse Spoorwegen), 2000]. In contrast, Dutch railway passengers have had some years with relatively bad punctuality.

In line with similar developments for the European Union, Dutch passenger traffic is expected to increase significantly in the near future. In order to serve this higher demand for railway passenger transportation, several measures are studied and/or implemented to increase the capacity of the railway system. These measures include:

- Changes to the railway system, for example by introducing a timetable similar to a metro timetable with frequent services.
- A new safety system for trains based on high-tech ICT.
- Extensions of the railway infrastructure. Examples are the high speed line from Amsterdam via Rotterdam to Brussels and a new dedicated freight line, the Betuwe Route, resulting in less cargo trains on railway infrastructure shared with passenger trains.

1.4 PLANNING PROCESS AT NSR

This paragraph describes the planning process of NSR and is largely based on Prins [1998], Peeters [2003], and Vromans [2005]. Although these authors focus on the planning process at NSR, largely the same processes can be found at other railway operators. An overview of the planning process at NSR is given in Figure 1.3. Huisman et al. [2005b] and Kroon [2001] discuss how Operations Research supports these planning processes. The core of the timetable has been fixed over the last decade. Shifts in market demand and altered physical infrastructure resulted in small changes in the timetable. Therefore, the previous plan typically serves as a good starting point for the new plan.

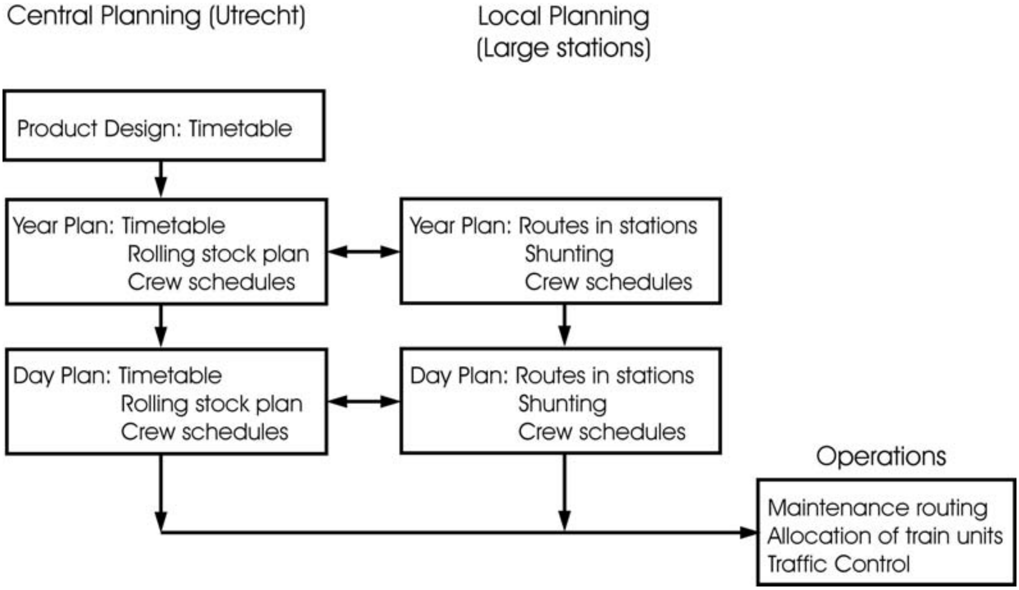


Figure 1.3: The planning process at NSR.

The planning at NSR is separated between the central planning department in Utrecht and the local planning departments at 5 locations throughout the country. Centrally, a rough national timetable is worked out for one hour during the Product Design phase. Such a rough timetable is called a One-Hour Timetable (OHT). Besides the general OHT, alternative OHTs are created for the rush hours and the evenings and weekends, with lower demand for passenger transportation. A OHT describes arrival and departure times for train services at the stations in the network. The OHTs are centrally checked for feasibility within some stations. In addition, platforms are centrally assigned to the arrivals and departures. This results in preliminary Basic Platform Assignments, one for each OHT at each checked station.

In the Year Plan phase, Central Planning creates a Weekly Plan, including a 24*7-timetable, rolling stock schedules and crew schedules. Local Planning checks the feasibility at the stations in detail. Moreover, they fill in the local logistic details, such as shunting, and they can also propose adjustments to the timetable.

The Day Plan phase handles daily adjustments of the Weekly Plan. For every single day of the year, adjustments to the Weekly Plan are necessary. These adjustments are caused for example by additional train services for events and infrastructure maintenance. The central planning department creates a rough Daily Plan and local details are handled by the local planning departments.

Finally, the Daily Plan is handed over at least 36 hours before the operations. It is handed over to the Rail Traffic Control department at Prorail, the Dutch infrastructure manager, and its NSR counterpart.

NSR intends to redesign its planning process. The goal of this redesign is to reduce the throughput time of the planning process. This should be achieved among others by virtually integrating the central and local planning departments, thereby removing current iterations in the planning process between these departments. A blueprint of the new planning process can be found in Reinartz and Fassaert [1999].

1.5 SHUNT PLANNING AT NSR

This paragraph gives some insight into the current shunting process at NSR. We discuss the role of shunt planning at the different levels of planning within NSR. Moreover, we pay attention to the operational shunting processes and we conclude with some notes on environmental aspects of shunting.

1.5.1 Shunting in the Netherlands

Nearly all passenger train services in the Netherlands are operated by train units. The remaining passenger train services are operated by a locomotive with a number of carriages. Between 15 and 20 different types of train units exist, which belong to about 8 families. The upper part of Figure 1.4 shows an example of a train unit, while the lower part depicts the middle carriage of the unit. This example unit belongs to the InterCity Rolling Stock family, which is translated from "InterCity Materieel (ICM)" in Dutch. Units of this family mostly run on intercity services. Train units belonging to the same family can be combined to form longer trains, taking into account certain restrictions on the length of the resulting train. Typically, not all types appear at a specific station.

Figure 1.4: An example of an ICM train unit with 3 carriages (ICM_3) and one carriage [Pijpers, 2004].

A station with a shunt yard is station Zwolle, depicted in Figure 1.5. Zwolle is located in the northeastern part of the Netherlands. Figure A.1 on page 202 presents a map of the Dutch railway network.

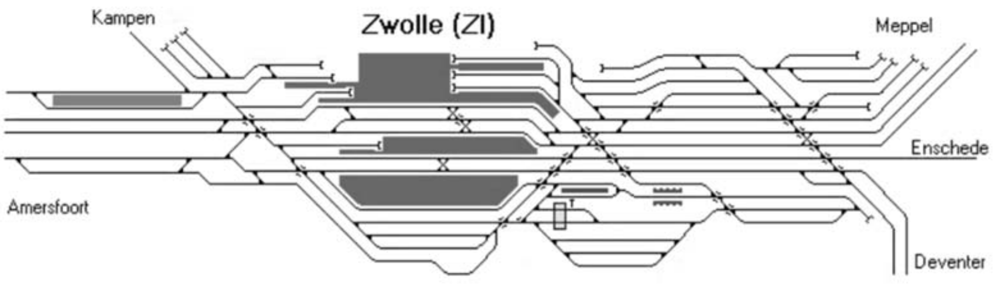


Figure 1.5: The layout of the railway station Zwolle, based on ZEEGERS [2004].

In order to satisfy the increased demand for railway transportation during the morning and afternoon rush hours, NSR operates more and longer trains during these periods than normal. This results in a differentiated demand for train units at a specific line throughout the day for the passenger train services. Especially at night almost no units are required for passenger services. Therefore, NSR operates just a few night train services. Most of the train units are parked at the shunt yard of the end station of the last trip. Logically, they should start their operation in the next morning from the same station. Exceptions occur when, for example, the capacity of the nearby shunt yard is insufficient for parking all train units that have their last trip at the corresponding station. In this case, the train units are repositioned to another shunt yard in the railway network. This is undesirable since it results in additional use of resources, such as energy, crews and infrastructure capacity.

Besides NSR, several other parties are involved in shunt planning:

- ProRail is responsible for providing sufficient capacity, reliability and safety of the Dutch railway infrastructure. ProRail evaluates the shunt plans. If positive, it assigns infrastructure capacity for operating these plans. ProRail especially focuses on the capacity of the main railway infrastructure.

- NedTrain is in charge of maintenance and cleaning of rolling stock. The plans of NedTrain for these processes should be coordinated with the shunt plans of NSR. In particular, NSR plans parts of these processes at some shunt yards, caused by large relations between the different processes.

- Cargo operators operate freight trains through railway stations. The detailed routes of these trains through the stations are typically determined by planners of NSR. If the cargo operators would plan their own trains, the planning processes would become too complicated and would require too much time and effort.

- Other passenger operators. These operators mainly operate train services at the ends of the Dutch railway network. Coordination of processes is required at parts of the network which are used by more than one operator. In the Netherlands, only a small fraction of the network is used by multiple passenger operators.

Essentially, there are two types of shunting problems. The first type occurs during the day. In this period, few train units are parked at shunt yards and the station infrastructure is heavily used by timetabled passenger train services. Sufficient capacity is available for parking, while the capacity for routing train units to and from shunt yards is scarce. During the night, the situation is the other way around and the second type of shunting problems occurs. Then, capacity for parking train units is scarce, since many train units need to be parked. In this period, routing capacity is usually sufficiently available because there are no timetabled services.

Figure 1.6 shows the number of arriving and departing carriages in train units on a typical Tuesday at station Zwolle in 2000. More specifically, from Tuesday 5:00 until Wednesday just after midnight. Analyzing the figure, we see that in the morning many more carriages depart than arrive. The same happens to a lesser extent before the afternoon rush hours. This means that this difference has to be supplied from the shunt yard. The opposite occurs after the morning and afternoon rush hours, when more carriages arrive than depart. These carriages need to be parked at the shunt yard. The absolute value of the difference between the number of arrivals and departures in a certain time period is therefore a good indication of the number of shunting activities at a station during this time period. Although these carriages arrive and depart in train units, we chose to depict the number of carriages because it gives a better insight into the different capacities of the deployed rolling stock for passenger transportation throughout the day.

1.5.2 Levels of Shunt Planning

The creation of shunt plans plays an important role at the strategic, tactical and operational level of planning. At the strategic level, one might be interested in e.g. changes in the layout of a shunt yard. Increasing fleet sizes might require extensions of shunt yards, which bring about large costs and a long lead time. The tactical level concentrates on global checks of several capacity measures, such as parking capacity, routing capacity, and the availability of shunting crew. Decisions on this level focus on the distribution of rolling stock over the shunt yards available to a railway operator. In addition, the focus is also on the required capacity of shunting crews at the shunt yards. Both at the strategic and the tactical level, environmental issues play an increasingly important role. Especially, the permitted noise levels for shunt activities during the night are set more and more tightly by the government. At the operational level, detailed plans are created, describing the exact location of rolling stock at the yard, when it should be parked there and who should park it there. In addition, detailed plans for routing and cleaning of rolling stock are created during this phase. Of course, during the operations,

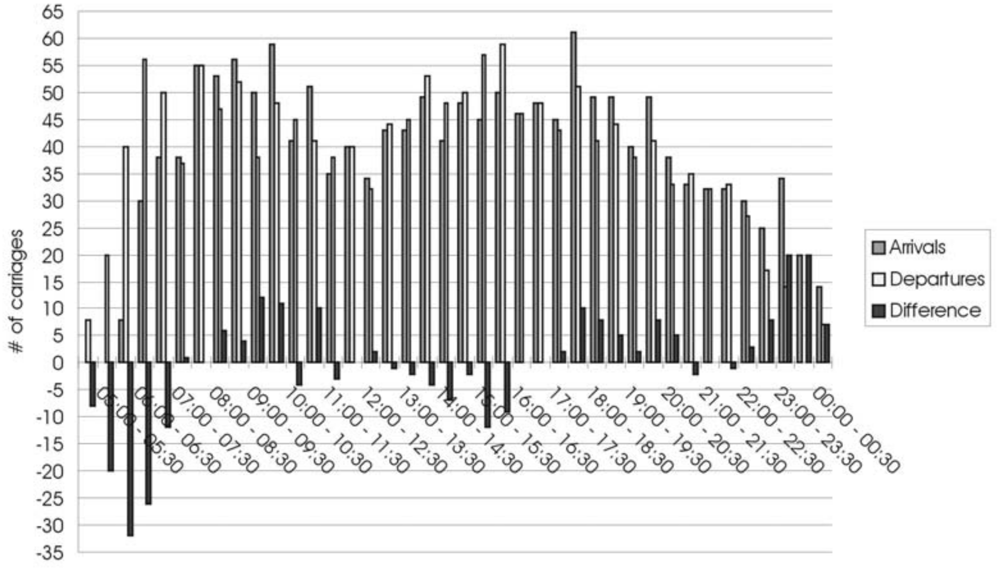


Figure 1.6: The arriving and departing train units during a typical Tuesday at station Zwolle.

disturbances and delays are likely to occur and changes in the operational plans need to be handled in a real-time setting.

The goal at the operational level is to plan the shunting processes in such a way that the railway operations can start up in the morning as smoothly as possible, while certain restrictions with respect to these processes are met. This is related to the fact that shortly after the start of the railway operations in the morning, the demand for railway transportation is at its maximum for the day. In addition, this specific aim adds to the robustness and stability of the railway operations. It is well known that, especially during the morning, disturbances might very well propagate throughout the network and throughout the day. In turn, increased robustness and stability of the railway operations improve the quality of the railway processes. Moreover, temporarily parking idle train units at shunt yards enables NSR to use the main railway infrastructure more efficiently.

1.5.3 Overview of Shunting Processes

Several processes are part of the shunting of rolling stock. An overview of the most important ones is given in Figure 1.7.

The parking of train units is far from trivial because in general parking capacity is scarce. In addition, the choice to park a train unit on a particular shunt track has several implications. First, when train units at a shunt track are of different types, then the

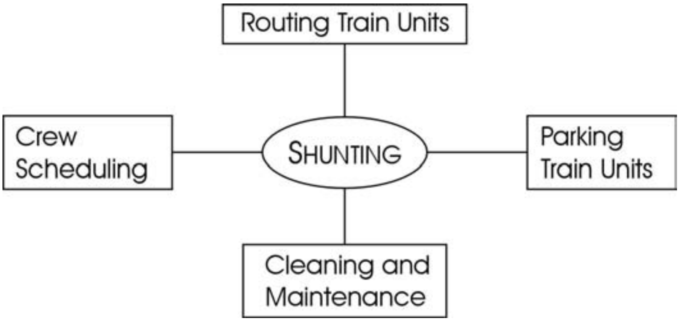


Figure 1.7: Several processes involved in shunting.

order of the train units is important. Obstruction of arriving or departing train units by other units is not allowed. Second, this choice restricts the possible routes between the platforms and the shunt tracks. Third, crews have to be available to carry out the resulting shunt activities within certain time intervals. Finally, certain routes and shunt tracks are preferred over others by shunt planners. Here, a track is preferred if it is located close to the platform tracks, or if it is rarely used for other purposes, e.g. for through train services or for temporary parking of rolling stock.

Routing of train units through a station takes place from specific arrival platforms to the shunt yard and back to specific departure platforms. When a platform track is available, it is possible to leave a train unit for a certain period of time on this platform after arrival or to park it there some time before departure. This introduces some flexibility with respect to the timing of the routing. Additional routing could be necessary for other processes, such as cleaning at dedicated tracks. Of course, the routes of the different train units should neither conflict with each other nor with the routes of the through train services, or other infrastructure reservations, such as track maintenance. A through train service continues passenger service after a short dwell time. Note that the routes of through train services have been decided upon in previous planning processes and are fixed.

Train units should be cleaned both externally and internally on a regular basis, which typically takes place at a so-called train-wash or along dedicated cleaning platforms. Both of these are available at a restricted number of stations. Moreover, it is mandatory that train units are checked every 48 hours for defects. Such checks can be performed at some stations within the network.

The tasks that result from the routing of train units and the coupling and decoupling of them, have to be assigned to shunting crews. These crews are local shunting crews at a given station. Typically, members of the local shunting crews have different qualifications, for example qualifications to drive trains over station infrastructure. Crews performing cleaning and maintenance activities normally do not drive trains. These qualifications restrict the crews which can perform a certain task.

1.5.4 Environmental Issues related to Shunting

As mentioned before, the most important environmental issue that plays a role in shunting passenger train units is the maximum noise level the processes are allowed to make. If the processes are too noisy, citizens complain to ProRail. Possibly, ProRail orders the operator to reduce the amount of noise during these operations or, more precisely, the amount of disturbance of the noise. Sometimes, inventive solutions for such problems can be found. For example, at some stations rolling stock can be parked in such a way that it absorbs some of the noise for citizens living nearby the station or shunt yard.

Watson and Sohail [2003] conclude that reducing railway noise levels in the United Kingdom involves large costs and, therefore, European legislation restricting these levels will have a large impact on the railway operations. Specialized design of trains, tracks and other structures might result in such reductions [Crockett and Carlisle, 2003].

Moreover, national legislation is gaining impact. As an example, we mention the influence of such legislation on railway freight operations. In the Netherlands, legislation concerning stations through which hazardous goods are transported is being prepared. This legislation would prohibit ‘vulnerable’ structures, such as housing, in some range around the station. Note that this legislation conflicts with efforts to improve the quality and looks of stations and surroundings. In turn, these improved quality and looks have positive effects on the perception of safety of the railway passengers [Kuenen, 2003].

1.6 DECISION SUPPORT FOR SHUNT PLANNERS

In the current practice, shunt planners create their plans with pencil, paper and eraser. An information systems is used for recording and communicating the created shunt plans. The information systems contains hardly any functionality for detecting or resolving conflicts. In order to improve this practice, NS Reizigers initiated the project “Intelligent Shunting” (in Dutch: Rangereren INTELligent, RINTEL).

RINTEL consists of two streams of research. The first stream of research focuses on the development of quantitative models and algorithms supporting the creation of shunt plans. Besides the research described in this thesis, the work of Haijema et al. [2005] belongs to this stream. This work will be discussed in more detail in Chapter 7.

The second stream of research puts the shunt planner in the center of attention. Here, researchers made an extensive and detailed analysis of the tasks a shunt planner performs. Typically, tasks consist of subtasks, which can be broken down to even smaller pieces of work. Van Wezel and Barten [2002] introduce an example of such an analysis for changing a shunt plan because a track goes out of service to receive maintenance. Subsequently, Van Wezel et al. [2003] report on a larger but similar research project.

The first result from this extensive analysis is the observation that planners work in an iterative fashion: if the planner is stuck, he backtracks a step in his solution process and redefines the starting situation for further analysis. In this practice, it is valuable that there is a possibility to easily retrieve previous partial solutions.

The second result from the analysis found by Van Wezel and Jorna [2004] is that planners extensively use several types of algorithms. These types include sorting, selecting, searching, and evaluating. Since the problem solving process differs for different planners, automation of such algorithms results in components that can be combined in different manners by different planners.

The goal of this second stream of research is to facilitate the interaction between planners and computerized algorithms. Van Wezel and Jorna [2004] state that the chances on actual usage of a decision support system increase because the structure of the developed algorithms resembles the structure a planner would use. Moreover, they conclude that the computerized algorithms do no have "the risk of adopting all non-optimal habits" of planners.

1.7 OTHER APPLICATIONS OF SHUNTING

In scientific literature, we found shunting applications of train scheduling [Tomii et al., 1999; Tomii and Zhou, 2000], tram scheduling [Winter, 1999; Blasum et al., 2000; Winter and Zimmermann, 2000], and bus scheduling [Gallo and Di Miele, 2001; Hamdouni et al., 2004].

For both buses and trams, rolling stock can be partitioned into families similarly to train units. Examples of such different families are buses with a boosted engine and trams qualified for the transport of physically handicapped people.

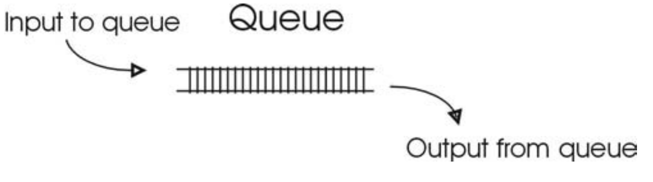


Figure 1.8: A queue configuration of a shunt track.

LIFO tracks are typically used for parking train units and trams, see Figure 1.1 on page 2. These tracks can only be entered and left via one side of the track. A queue is depicted in Figure 1.8 and is common for parking buses. Here, buses arrive at one side of the "track" or lane and leave via the other side. A queue is useful when driving backwards on the track is undesirable or forbidden, e.g. because of safety reasons. Notice the difference with a free track, where rolling stock can arrive and depart via both sides of the track.

When families of rolling stock differ only marginally, a departure might be served with rolling stock of an alternative type than prescribed. In general, this occurs for trams, while the differences between types of buses and train units are too large to allow type mismatches.

The main goals of shunting trams and buses are the same as the goals for shunting train units, namely to start up the operations as smoothly as possible and to create robust plans, requiring little changes in case of disruptions during the operations. Differences occur when e.g. type mismatches are allowed and when rolling stock is allowed to obstruct other rolling stock at a shunt track at a certain cost.

Parking of trains differs from parking of trams and buses since trams and buses typically consist of one unit. Furthermore, buses are less restricted by the infrastructure than trams and trains. In addition, we consider in this thesis other processes than parking, like the routing and cleaning of rolling stock. Therefore, we need to look at arrivals and departures `at the platforms`, since the arrivals and departures `at the shunt yard` are influenced by these related processes.

Tomii and Zhou [2000] report on similar problems regarding local shunting operations. However, in these problems, at most one train unit can be parked at a track. Boccalatte et al. [1994] discuss a decision support system for a similar problem at a Japanese shunt yard.

We conclude that the described literature provides a good starting point for studying the shunt planning problem at NSR. However, the problem at NSR has several distinctive characteristics, which require our attention, in order to ensure the practical relevance of our research.

1.8 RESEARCH QUESTIONS

The aim of this thesis is to develop quantitative models and algorithms, which can potentially support shunt planners in creating shunt plans. As mentioned before, it is a part of the research project RINTEL. The aim of this research project is to investigate which support for planners is desired in order to speed up the shunt planning process. The focus of the thesis is on algorithmic decision support for these planners.

The main research question studied in the thesis is the following:

What are appropriate quantitative models and algorithms for supporting shunt planners?

In order to answer this question, it is crucial to understand the shunting problem and the important aspects of shunt planning. In addition, we investigate important criteria for assessing the quality of a shunt plan. Based on the understanding of the shunt planning problem, we consider which quantitative models are appropriate for supporting shunt planners and how we can solve these models efficiently. The **main research question** breaks down into the following sub-questions:

1. *What are the important aspects of shunting and shunt planning?*

2. *How can the quality of a shunt plan be measured?*

3. *Which mathematical models can properly support the shunt planners?*

4. *How can we efficiently solve these mathematical models?*

By finding high-quality solutions fast to models, capturing the important characteristics, algorithmic decision support can be valuable for shunt planners.

1.9 RELEVANCE

The type of problems studied in the thesis has attracted little attention from the scientific community until now, especially in a railway setting. Because the subproblems of shunt planning are tightly intertwined with each other, decomposition is difficult. These complicated relations lead to scientifically challenging problems in the field of transportation science, which probably require decomposition in order to be able to solve the overall problem in a reasonable amount of computation time.

Currently, performance indicators for shunt plans are scarce. Moreover, performance indicators known in the scientific community might not be valid for the shunting problems described in this thesis. For instance, in many related problems, it is allowed but undesirable to serve certain trips with a different type of rolling stock than prescribed. However, these type mismatches are not allowed for the problems studied in the thesis.

From the research, new mathematical models and extensions of current models result for (sub)problems of shunt planning. Moreover, we propose extensions and adaptations of algorithms for solving these models.

The scientific relevance of our thesis is stressed by the scientific appreciation of the joint paper "Shunting of Passenger Train Units in a Railway Station" [Freling, Lentink, Kroon, and Huisman, 2005]. This paper is based on the research described in Chapters 3 and 4.

The models and algorithms developed for supporting shunt planners aim at reducing the throughput time of the shunt planning process. This enables managers to react faster and more accurately to market developments.

In the current planning process, shunt planning requires approximately 130 planners, which constitutes around 35 % of all planners at NSR. Because of this significant workload, creating shunt plans more efficiently is extremely relevant for management.

Moreover, since it is the last stage of the complete planning process, every modification in a previous stage requires adjustments to shunt plans at one or more stations. Therefore, shunt planning is the bottleneck of the rolling stock scheduling. A more efficient shunt planning process offers opportunities to improve the quality of the shunt plans, the quality of earlier planning processes, or a combination of both.

The potential reduction in the throughput time for creating such plans enables a later start of the planning process, which in turn means that input data will be more accurate. Accurate data result in better service to passengers, e.g. because they can rely more heavily on the published timetable. Moreover, it reduces the complexity in the planning process and the operations. In addition, the potential reduction in throughput time enables the possibility for scenario studies. Currently, such scenario studies are hardly feasible because the adaptation of previous shunt plans to a new timetable requires a throughput time of approximately 4 months, which is the approximately available time. However, such scenarios give a theoretical indication of the effect of changes in shunt plans, without actually implementing them and contain therefore valuable information. Moreover, they could be used in broader scenario studies.

Models and algorithms for supporting shunt planners also offer a possibility for improving the quality of the shunt plans. Moreover, the developed performance indicators provide a way to explicitly quantify trade-offs between various scenarios. This enables faster and better founded operational decision-making by managers.

As mentioned before, shunt planning is closely connected to other railway planning processes. Mostly, it serves as an evaluation tool for these processes since it is the last stage of the complete planning process. As such, improvements in the shunt planning process also impact these other processes.

In addition, the expected increased demand for mobility in the future will result in an increased size of the rolling stock fleet. In turn, this will increase the complexity of the shunt planning processes at different shunt yards. It might also result in expensive expansions of shunt yards. More efficient shunt plans can be a viable alternative for such infrastructure expansions.

From a societal perspective, an improved effectiveness of the shunt plans results in an increased quality of the railway system as a whole. For instance, more robust shunt plans result in less changes in the plans during the operations and allow passengers to rely more heavily on travel information gathered before their journey (similar to more accurate data, as mentioned before). In turn, an improved railway system might attract additional demand for railway transportation and might help to facilitate the expected mobility growth.

To conclude, the process of shunting passenger train units combines a challenging scientific problem with very relevant practical applications.

1.10 DEMARCATION

The characteristics of the operational processes described in this thesis pertain to the railway system in the Netherlands. Although railway shunting processes are to a large extent similar to shunting processes for other types of public transport, such as buses and trams, or in other countries, this is not considered in this thesis.

The research in the thesis mainly discusses the operational level of planning. This operational level looks at the problem per station and for a 24-hour period. Furthermore, we look at shunting trains, or more specifically train units. The arrivals and departures at a station are prescribed by a timetable, which also describes the configuration of each train service.

The railway infrastructure of a station is bounded by so-called entering and leaving points. In general, an entering point can also serve as a leaving point and vice versa. The railway infrastructure outside these points are irrelevant for the shunting processes considered in the thesis.

Finally, we focus on the planning of processes related to the rolling stock. Aspects of the crew planning problem will be described, but solution approaches for this problem are considered outside the scope of the thesis.

1.11 OUTLINE

Chapter 2 gives an in-depth introduction to the different shunt planning processes. After this introduction, several chapters discuss a subproblem of shunt planning in detail. Chapter 3 discusses the matching of arriving train units to departing ones. Chapter 4 gives insight into the parking of train units at shunt tracks. Chapter 5 describes how routes can be found for units over the local railway infrastructure. The subproblem of cleaning train units is described in Chapter 6. In Chapter 7, we investigate the potential benefit of combining the subproblems of Chapters 3 and 4 and solving these as an integrated problem. Each of these chapters contains a mathematical model with some theoretical considerations, a solution approach, computational results of this approach for real-life instances, and some conclusions. Finally, we present the conclusions and directions for further research in Chapter 8.

Chapter 2

Specification of the Shunt Planning Problems

As was mentioned in Chapter 1, the process of shunting passenger train units focuses on temporarily parking idle train units at shunt yards, which are located near larger railway stations. Typically, nearly all train units need to be parked at night, while some train units need to be parked during daytime between the morning and afternoon rush hours. During its stay at a shunt yard, rolling stock usually needs to undergo several additional processes, like cleaning or maintenance checks.

Plans are created for these local processes. During the operations, almost all plans are altered to some extent in order to react to disruptions. Nevertheless, the plans are considered valuable. Several reasons for the usefulness of these plans are:

- A decreased burden on the crews supervising the real-time operations.
- A better coordination between different stakeholders.
- An improved efficiency of the railway operations.
- An increased probability that sufficient capacity is available for the operations, since these plans also serve as capacity checks.

Of course, the plans have to comply with certain restrictions, such as environmental restrictions, and the capacities of the resources. The overall objective is to enable a smooth start-up of the railway services in the next morning.

Shunt planning is relevant at different levels of planning, where each level has its own characteristics. In this chapter, we introduce the main example that will be used throughout the remainder of the thesis. At the operational level of planning, several subproblems play a role in shunt planning. These subproblems are introduced and described from the point of view of NSR. However, this discussion is also relevant for other public transport operators with similar problems.

2.1 DIFFERENT LEVELS OF PLANNING

In the first chapter we briefly touched upon the different levels of planning, where shunt planning plays a role. In this paragraph we go into more detail.

At a strategic level, one is mainly interested whether or not the railway infrastructure will be sufficient for the expected future train services. The basis for this strategic planning is the expected passenger demand, the estimated growth of the total rolling stock fleet of all railway operators, future environmental restrictions, and the long term policy regarding mobility, infrastructure and transportation. On this basis, one can estimate the total need for shunting capacity in an entire network with several shunt yards. This need can be decomposed, such that analyses of specific locations become possible, thereby identifying potential bottlenecks, as reported by Cardol and Fledderus [2002] for example. These analyses might reveal bottlenecks for such shunting processes. In turn, these bottlenecks can be removed by expanding railway infrastructure. However, it is worthwhile to consider alternatives for such expansions, since these are very expensive. For example, Briginshaw [2004] reports on a cost of £ 2 million for upgrading and electrifying 3 shunt tracks for passenger railway operator South Central Trains in the UK. Furthermore, such expansions might have a large impact on the involved city, since stations and shunt yards are typically located in the city center. Therefore, such expansions should only be carried out when no viable alternatives exist.

The tactical level requires capacity checks at a specific station of e.g. the infrastructure, the capacity for the cleaning and maintenance processes, and the available crew. Three examples of such checks are:

- Checking whether the infrastructure connecting the platform tracks with the shunt tracks has sufficient capacity for routing the through train services and the rolling stock that needs to be parked at the shunt tracks. See for example Van den Broek [2002].

- Checking whether the capacity of the crews is sufficient to carry out the local shunting activities.

- Checking whether the capacity for cleaning rolling stock is sufficient.

The purpose of these checks is to identify potential bottlenecks as early as possible in the planning process. If such a check fails, one can act upon this, for example by making appropriate changes to the timetable or the rolling stock circulation.

At the operational level of planning, a planner tries to make a detailed plan for the near future. This will be discussed elaborately in Section 2.3, as it is the subject of the thesis.

Finally, real-time changes in the plan are probably necessary in the operations because of disruptions, which invalidate the operational plan. These disruptions result from minor deviations from the planning, e.g. a delay of one train service with a few minutes, or major deviations, e.g. a failure of the main railway infrastructure or the breakdown of a train leaving a shunt track. Both minor and major deviations may cause severe disruptions of the local shunting activities as well as of the network-wide operations. One would like the operations to stay as close as possible to the plans. In particular, these problems are quite complex and require much coordination, which invalidates a complete re-optimization of the planning. Therefore, robust plans are extremely valuable.

An extreme example of the impact of shunting operations on the railway system is the situation around Amsterdam on October 17, 2001. Railway traffic controllers were not able to handle large deviations from the shunt plans due to disruptions of the railway system caused by a collision. This resulted in a vicious circle, where deviations from the planned timetable implied disruptions of the shunt plans, and these disruptions in turn resulted in even more deviations from the timetable. In this particular case, some passenger train services were canceled because of these problems. Of course, the impact of such disruptions is to be minimized, among others by creating robust shunt plans. Although this is a rare situation, it demonstrates the impact of the shunting operations on the network-wide operations.

2.2 EXAMPLE OF A SHUNTING PROBLEM

In this paragraph, we introduce a small example in which several shunting activities are required. This example is used throughout the thesis to clarify certain aspects. It is based on a practical Dutch situation at station Zwolle.

Railway station	Abbreviation
Amersfoort	Amf
Enschede	Es
Groningen	Gn
Leeuwarden	Lw
Roosendaal	Rsd
Zwolle	Zl

Table 2.1: The railway stations of the main example and their abbreviations.

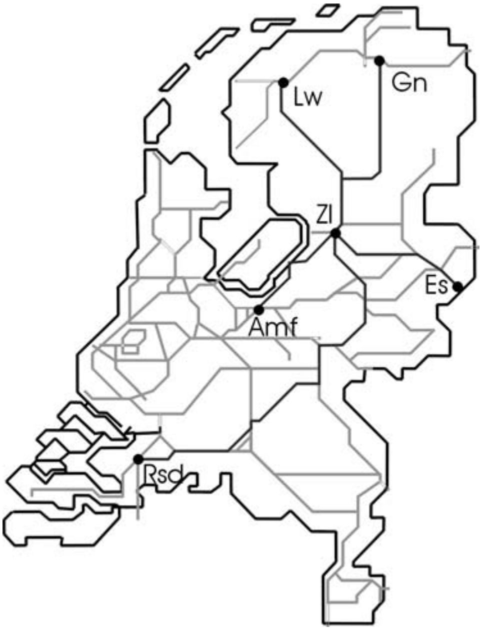


Figure 2.1: The Dutch railway network with the lines of the main example in bold.

The stations in the example and their abbreviations are shown in Table 2.1. Moreover, the relations between these stations in the Dutch railway network is depicted in Figure 2.1. Here, bold lines within the borders represent the railway lines in the example. The gray lines are other railway lines for passenger transportation, operated by NSR and other operators. The infrastructure of the station in the example is shown in Figure 2.2. This figure includes the names of all relevant tracks for the example. These names relate to the tracks below them, with the names "91" and "98" as exceptions, as these correspond to the tracks above the names. The tracks 1A, 1B, 3A, 3B, 5A, 5B, 6A, 6B, 7A, 7B are platform tracks. Tracks 19, 18C, 17, 100, 101, and 102 can be used for parking train units. Moreover, tracks 90 and 91 are located along a cleaning platform, and track 98 contains the train-wash for external cleaning of rolling stock. As mentioned before, a railway station is bounded by so-called entering and leaving points. These points are given in italics in the figure (*HA, VA, BA, AB, ZH, WO,* and *OW*). In practice, every switch is named as well, but for clarity's sake these names are omitted in this figure.

Each station has an A-side and a B-side. Given these sides, we define the A-side of a track as the side which is closest to the A-side of the station, and similarly the B-side of a track. A shunt track can be accessed from the A-side, the B-side, or both sides. This uniformly defines the side of a shunt track. Moreover, we introduce the A-side of a

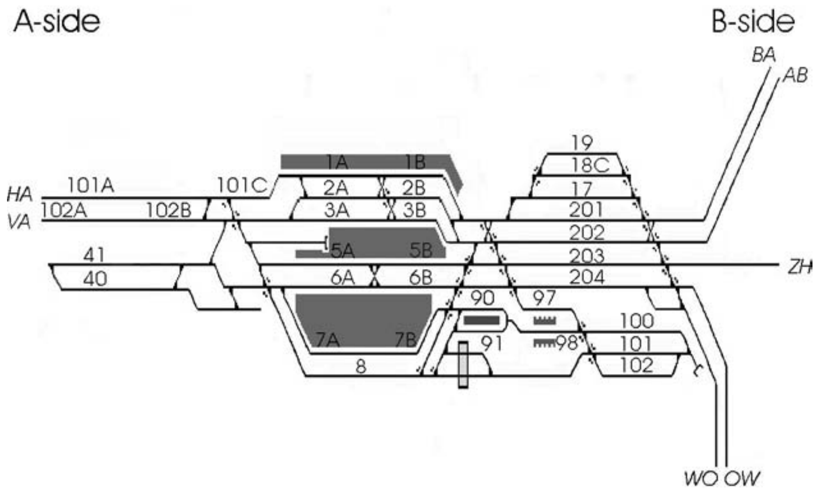


Figure 2.2: The railway infrastructure in of the main example [Zeegers, 2004].

train as the side of the train which is closest to the A-side of the station, whenever the train is within the boundaries of the station.

From Section 1.1, we know that rolling stock of a railway passenger operator typically consists of different families. In general, all train units are self-propelled and can move bi-directionally. This also holds for the types in this example. The different types of rolling stock in our example are introduced in Table 2.2 and depicted in Figure 2.3. Note that this example contains three families and a total of five types. The upper train unit (DH_2) is a diesel powered train unit and is mainly used for regional train services. The middle two IRM train units are double-deck units which are typically used for interregional services. Finally, the bottom two train units belong to the ICM family and are mainly used for intercity services. Both the IRM and ICM families consist of electrical train units only. In Section 1.5 we described some of the important characteristics of a

Family	Number of carriages	Abbreviation
Diesel Hydraulic	2	DH_2
InterRegional Rolling Stock	3	IRM_3
InterRegional Rolling Stock	4	IRM_4
InterCity Rolling Stock	3	ICM_3
InterCity Rolling Stock	4	ICM_4

Table 2.2: The types of rolling stock of the main example and their abbreviations.

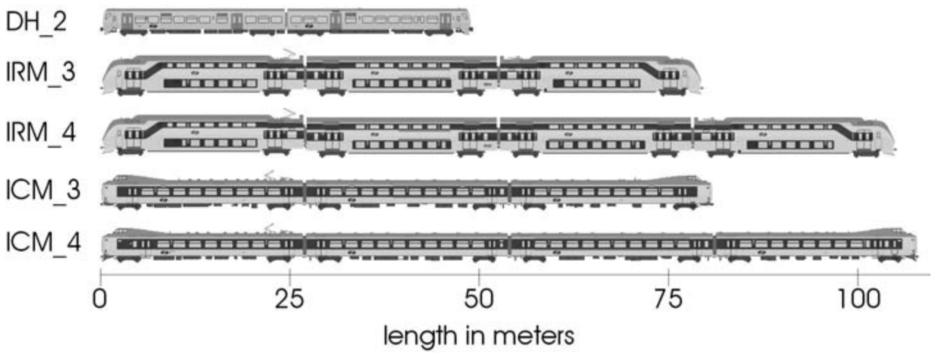


Figure 2.3: The different families (3) and types (5) of rolling stock from Table 2.3 [Pijpers, 2004].

family of train units. For shunting, the most important characteristics are the length of a type of train unit, and whether it is diesel-powered or electric-powered.

Table 2.3 describes the timetable that forms the basis for the example. In this table, the first column describes the ID of the train service, the second column describes the platform of arrival or departure, the third column gives the day and time of arrival or departure, the fourth column describes whether the train service arrives (A) or departs (D). Column 5 gives the configuration of the train operating the service, where the leftmost type is the unit closest to the A-side of the train, and similarly, the rightmost unit is closest to the B-side of the train. Column 6 gives the direction from which the train service arrived or to which it will depart, and between parentheses the entering or leaving point it uses. The last column indicates at which side of the station the entering or leaving point of the service is located.

2.3 OPERATIONAL TRAIN UNIT SHUNTING

In this paragraph, we describe the operational train shunting process in more detail. However, we start with some insights into operational railway practices in general. These insights provide a basis for the description of operational train unit shunting.

The railway tracks have several functions. The most important ones are: routing of train units, boarding and alighting of passengers, parking of train units, cleaning of rolling stock, and small maintenance of rolling stock. Note that one specific track can have multiple functions. For example, a platform track can be used for parking train units at night, when it is not used for passenger or freight services. Certain escape routes out of the shunt yard have to be kept free. Tracks also have several characteristics. For shunting, the most important characteristics of a track are:

Train ID	Platform	Time	Event	Configuration	Direction	Station side
771	3a/b	Tu 20:46	A	ICM_3 ICM_4 ICM_3 ICM_3	Amf (VA)	A
771	3a/b	Tu 20:49	D	ICM_3 ICM_3	Gn (AB)	B
10771	3a	Tu 20:52	D	ICM_4	Lw (AB)	B
3672	7a/b	Tu 22:09	A	IRM_4 IRM_3	Rsd (OW)	B
3687	7a/b	Tu 22:23	D	IRM_4	Rsd (WO)	B
7984	5b	Tu 23:12	A	DH_2	Es (ZH)	B
584	1a	Tu 23:18	A	ICM_3 ICM_3	Gn (BA)	B
3680	5b	We 0:09	A	IRM_4	Rsd (OW)	B
3623	5a	We 5:50	D	IRM_3 IRM_4	Rsd (WO)	B
516	1a	We 6:18	D	ICM_3 ICM_3	Amf (HA)	A
7917	5b	We 7:21	D	DH_2	Es (ZH)	B
721	3a	We 7:46	A	ICM_3	Amf (VA)	A
10721	3a	We 7:52	D	ICM_3 ICM_3	Lw (AB)	B

Table 2.3: The timetable for the main example.

- The length of a shunt track. This influences the amount of rolling stock that can be parked at a shunt track. With regard to a track available for routing, its length largely determines the duration of traversing the track.
- The sides from which rolling stock can approach a shunt track (see also Figure 1.1 on page 2). Tracks that can be approached from both sides provide additional possibilities for parking train units as compared to tracks with a dead-end side.
- The availability of catenary. In order to park train units with electric power at a track without catenary, a diesel locomotive is needed which should be avoided whenever possible.
- The availability of a railway safety system. Tracks which are not controlled by such a system, require the local traffic control organization to avoid collisions. In some exceptions the driver relies on his sight.
- The availability of several types of equipment along the track. Examples are a battery charger, which is needed for parking diesel powered train units, and equipment for filling the water tanks of toilets.

With these elaborations in mind, we start with a description of the matching of arriving and departing train units. This is followed by details on the parking of train units at shunt tracks and specifics on the routing over station infrastructure. Another important process is the cleaning of rolling stock, which is treated next. In addition, all these operational processes require local shunting crews. The shunting crew planning problem is discussed in Section 2.3.5.

2.3.1 Matching

In general, train units of the same type can be used interchangeably. This flexibility implies that part of a planner's job is to determine a matching of arriving units to departing units.

For a 24-hour period, the number of arriving train units of a specific type roughly equals the number of departing units. Otherwise, structural demand or supply differences would occur at a shunt yard, resulting in an unbalanced railway system. Exceptions are e.g. periods covering the transition to and from weekends. During weekends, less train units are required for operating the timetable, because demand for railway transportation is less than during weekdays. Therefore, from a typical Friday 8:00 in the morning until Saturday 8:00 in the morning, more units need to be shunted to the shunt tracks than units that need to be shunted from the shunt tracks. Vice versa, for a typical Sunday 8:00 in the morning until Monday 8:00 in the morning, more units depart from the shunt yard.

As input for this matching problem, **the planner** receives a timetable with planned arrivals and departures of all train services at the station under consideration. In addition, this timetable also prescribes the configuration of each train (see page 5 for the definition of the configuration of a train), which follows from the rolling stock circulation. More specifically, if the train configuration consists of different types, the order of the different types of train units in the train is given by the timetable. This timetable also prescribes the arrival and departure platforms for the train service. For a typical planning horizon, the first departure takes place before the last arrival and therefore the arrivals and departures overlap.

The configurations of the timetabled train services have to be respected by this matching problem. That is, it is not allowed to supply different types of train units for a train service, or to supply the right types of train units in an alternative order compared to the timetable and rolling stock schedule.

It is common that a large part of the matching has already been made, and therefore falls outside the scope of the shunt planning. The two most important reasons for such a prescribed part of the matching are the following:

1. The arriving and departing units are part of a through train service. In general, the timetable has sufficient time for alighting and boarding trains, and possibly for coupling or decoupling of train units. However, the timetable leaves insufficient time to replace an arriving unit before departure. In Table 2.3 for example, arriving train service 771 is split into departing services 771 and 10771, and one unit of

type ICM_3, which remains at the station. Another example is arriving service 721 and departing service 10721 with an ICM_3 coupled onto the train.

2. A train service arrives at a certain station, which is the end of a line. In such a case, units from the arriving train of this service might form a part of a departing train service. If the time difference between the departure and arrival is sufficiently small, this will result in a prescribed matching. Again in Table 2.3, the IRM_4 unit of train service 3672 returns to Rsd in train service 3687 and the resulting matching is fixed.

In order to match an arriving unit to a departing unit, the time difference between the corresponding arriving and departing train services needs to be sufficiently large. The minimum time difference is determined by the following aspects:

- the arriving and departing platform, which indicates the routing effort,
- whether or not parking is required, and
- the dwell time at the platform for boarding and alighting of passengers.

This matching problem results in a matching of arriving train units to departing train units. If, for a specific train unit, the time difference between the corresponding arriving and departing train services in this matching is below a certain threshold, the unit does not need to be parked. The units that do not need parking might need routing from the platform of the arriving service to the platform of the departing service. Of course, if these platforms are the same and this platform is not used in between, even routing is not necessary.

The main objective of the matching problem for the shunt planner is to keep units together as much as possible. An example is given in Table 2.4 where the subset of ICM train services from Table 2.3 are selected. Fixed parts of the matching of trains with ICM units are:

- the rightmost two ICM_3 train units of arriving service 771 to departing service 771,
- the ICM_4 unit of arriving service 771 to departing service 10771, and
- the unit from arriving service 721 to the rightmost unit in departing service 10721.

Therefore, the arriving units of the free part of the matching are the leftmost ICM_3 unit of arriving service 771 and the two ICM_3 units of service 584. The corresponding departing units are the units of service 516 and the leftmost unit in service 10721. A solution where the two units of arriving service 584 are matched with the units of departing service 516 is preferred over a solution where one of the two arriving units is matched with the leftmost unit of departing service 10721. Indeed, only this matching results in two entities of units that remain together, where alternatives require three entities.

Train ID	Time	Event	Configuration	Direction
771	Tu 20:46	A	ICM_3 ICM_4 ICM_3 ICM_3	Amf
771	Tu 20:49	D	ICM_3 ICM_3	Gn
10771	Tu 20:52	D	ICM_4	Lw
584	Tu 23:18	A	ICM_3 ICM_3	Gn
516	We 6:18	D	ICM_3 ICM_3	Amf
721	We 7:46	A	ICM_3	Amf
10721	We 7:52	D	ICM_3 ICM_3	Lw

Table 2.4: The ICM train units from the main example in Section A.2.

This objective results in a minimum required shunt effort, since each shunt movement requires expensive resources like energy, infrastructure and shunting crews. Because the capacity of the available crews is decided upon before the creation of the shunt plans and is fixed for a certain period of time, a reduction in the number of shunt movements increases the probability on an overall feasible shunt plan.

2.3.2 Parking

In order to achieve a good overall performance of the local shunting processes, it is crucial to park train units efficiently at shunt tracks. While planning this problem, physical characteristics of the tracks play an important role. For instance, the fact that certain tracks can be approached from both sides results in additional flexibility. This flexibility also complicates the problem, since a planner also needs to decide on the arrival and departure sides of a specific shunt track, for each train unit parked at it. Moreover, the robustness of the overall shunting plan is largely determined by the robustness of the plan for the parking subproblem. Finally, although the planned arrival and departure times are fixed, the arrival and departure times at the shunt track are flexible to some extent, see also Section 2.3.3. With minor changes to these arrival and departure times, better solutions might be obtained.

In **the physical process**, the family to which a train unit belongs might restrict the set of shunt tracks, where the unit may be parked. Indeed, train units with electric power can only be parked at shunt tracks with catenary and diesel train units can only be parked at tracks with a battery charger. Moreover, the length of the track can never be exceeded by the length of the rolling stock parked at it.

In the scientific literature, practical configurations of shunt yards are described where precedence restrictions with respect to the parking of rolling stock at certain tracks are present (see e.g. Winter [1999], Tomii and Zhou [2000], and Hamdouni et al. [2004]). These precedence restrictions reflect the fact that certain shunt tracks can only be approached via other shunt tracks. A small example is given in Figure 2.4, where track 2 is packed with two IRM_3 train units. In this situation, tracks 3 and 4 are inaccessible.

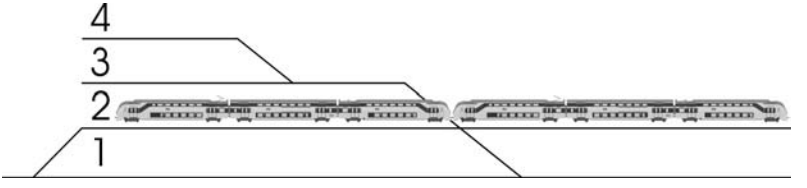


Figure 2.4: An example of precedence relationships for parking at tracks.

For this problem, **the planner** receives as input a set of train units, with their corresponding arriving or departing train services. For each arriving train unit, the planned arrival time as well as the arrival platform are known. The same holds for the departing trains. The second part of the input is described by the set of shunt tracks, with their characteristics, as described at the beginning of Section 2.3.

The arriving and departing services for the train units have been matched already in the matching subproblem. Moreover, estimates of the routing effort are taken into account in the preferences for certain shunt tracks over others.

Of course, a poor assignment of train units to shunt tracks might result in units blocking the arrival or departure of other units. In practical situations of NSR, such blocking units are rare, although they cannot be avoided at all times.

The output of this problem is an assignment of train units to tracks. In this assignment, no parked train unit is obstructing the departure or arrival of another unit.

If it is impossible to park all units at a yard, the excess units need to be parked at a different yard, which requires much additional resources, like crews, energy, and infrastructure. Therefore, it is important to park as many train units as possible at the shunt yard. Other relevant objectives include estimates of the quality of the resulting set of routes, and the number of different types of rolling stock parked at one shunt track. Indeed, if a shunt track contains only one type of rolling stock then the order of these units at the track is irrelevant. This is caused by the fact that these train units can be substituted for each other. In theory, a planner can park two types of rolling stock at a free track without any problem. All units of type “1” arrive and depart via the A-side of the track and all units of type “2” arrive and depart via the B-side. In practice, planners have preferences for a particular side of a free track, as will be described in Section 2.3.3. Moreover, if a track contains several types of train units, one would like these types to be grouped together. Finally, one would like to combine train units from different arriving trains but intended for the same departing train by parking them next to each other and in the right order at the same track. The train services in Table 2.5 illustrate such an opportunity. This table contains all the train services of IRM train units in our main example.

Train ID	Platform	Time	Event	Configuration
3672	7a/b	Tu 22:09	A	IRM_4 IRM_3
3687	7a/b	Tu 22:23	D	IRM_4
3680	5b	We 0:09	A	IRM_4
3623	5a	We 5:50	D	IRM_3 IRM_4

Table 2.5: An example with opportunity to combine train units of the same departing train at a shunt track.

Recall that the IRM_4 unit of service 3672 returns to Rsd in service 3687. It would be quite beneficial to combine the IRM_3 unit resulting from the train service 3672 and the IRM_4 unit from service 3680 at a shunt track. In this way, these train units can leave the track in one train destined for departing service 3623. In turn, this results in efficient resource usage, since the units require only one route to the departure platform and also only one driver. Note that units of different types need to be combined in the right order at a shunt track. Finally, planners can have preferences for certain tracks over other tracks. For example, certain tracks are also used at night for running freight trains through the station, such as tracks 2a and 2b in Figure 2.2 on page 25.

2.3.3 Routing

Shunt routing of train units takes place from a platform to the shunt yard and vice versa, and also from an arriving platform to a departing platform. The routes and corresponding railway infrastructure reservations of through trains have been decided upon in a previous planning process and are fixed. Additional routing could be necessary for local operations such as internal or external cleaning. The parking of train units as well as the local operations result in route requests over the station infrastructure. The routing subproblem aims at finding routes for such requests.

If shunt plans for the other subproblems are made without taking into account the routing of train units to and from the shunt yards, it is likely that the overall plans will be infeasible. This means that it is impossible to process all the route requests without conflicts. Therefore, the routing of the train units over the infrastructure of the railway station is important in shunt planning. In case of infeasibility, plans for one or more subproblems need to be altered or restrictions of the routing problem need to be relaxed.

Concerning the **physical process**, we know from Section 1.10 that the station infrastructure is bounded by entering and leaving points. Within these boundaries, the most prominent parts of the physical infrastructure are tracks and switches connecting tracks. Along these tracks, several signals are located. A signal is similar to a traffic light and indicates whether or not a train is allowed to enter a section of the infrastructure.

Of course, the shunt routes should neither conflict with each other nor with other infrastructure reservations for e.g. through trains or track maintenance. The procedure for reservation and release of railway infrastructure is described in Zwaneveld [1997, Section 3.1] and is the basis of this short description.

The station railway infrastructure is divided in a large number of `track sections`. A route over this infrastructure consists of a sequence of sections linking the route start point to its end point. For example, an inbound route of a timetabled train service links the entering point of the service with its platform by a number of succeeding sections. From a specific start point to a specific end point, a planner has a limited number of routes to choose from in the current Dutch practice.

Before a train starts its route over the station infrastructure, a specific route from the trains start point to its end point is claimed. A route cannot be claimed if it intends to use tracks or switches which have been claimed already by another train. After a train has passed a section of its claimed route, the section is released and becomes available for other trains.

During a route within a railway station, it might be required that a train unit needs to change directions. For example, a route from platform track 1a to track 8 in Figure 2.2 on page 25 requires at least one change in direction. In case of such a change, the driver locks the driver cabin on one side of the train, and walks to the other side of the train. Therefore, changing direction requires time.

The primary input for this subproblem of **the planner** is a set of route requests over the railway infrastructure of the station. Moreover, the timetable with planned arrivals and departures of train services also serves as input. The routes of the through train services and the corresponding reservations of the infrastructure have already been planned in a previous stage of the planning process and need to be considered fixed. Also, other infrastructure reservations need to be known, such as for example tracks going out of service because of maintenance. Finally, this subproblem requires a detailed formal description of the layout of the railway station infrastructure as input. This description includes the locations of all platforms, switches, signals, shunt tracks, and other tracks. Moreover, it also includes the set of prescribed routes between two specific start points and end points.

Two train movements over the infrastructure result in a route conflict if they use the same part of the infrastructure and have insufficient headway time between them. Thus, headway times are used to avoid conflicts. Exceptions to this rule are the start and end of a movement. Indeed, if a train unit is to be coupled onto a departing train, it is necessary to partly use the same infrastructure for both the train unit and the departing train.

An important characteristic of shunt routing of train units is the fact that the arrival times at and departure times from the platforms are flexible to some extent, as opposed to the planned arrival and departure times of the through train services. As an example, the train services 7984 and 3680 from the timetable in Table 2.3 offer opportunities to exploit this flexibility. The relevant characteristics of these services are repeated in Table 2.6.

Train ID	Platform	Time	Event	Configuration
7984	5b	Tu 23:12	A	DH_2
3680	5b	We 0:09	A	IRM_4

Table 2.6: An example of the flexible start times of routing.

Train service 7984 arrives on Tuesday evening at 23:12 at platform track 5b. After a dwell time of say 3 minutes at the platform, the resulting train unit of type DH_2 is ready to be routed from the platform track to a shunt track at 23:15. At 0:09 on Wednesday, the arrival of train 3680 is the next event at platform 5b, which requires a buffer time of say 4 minutes. Therefore, the DH_2 unit needs to be routed to a shunt track somewhere between 23:15 on Tuesday and 0:05 on Wednesday. This flexibility can be used to generate shunt plans of a higher quality.

A solution to the routing subproblem assigns a route to each route request. More specifically, each section in an assigned route is reserved for a certain time window. Since no conflicts occur in a solution, the time windows during which a section is reserved are separated by sufficient slack.

In general, there are many options that a planner can choose for the routing problem. Here, we only mention two of these options:

1. the duration of the dwell time at a platform for alighting and boarding a train.

2. the slack time between the infrastructure reservations of two subsequent routes.

While configuring these options, a planner has to balance between the robustness of the plans on one hand and a sufficiently large solution space for the routing problem on the other hand. For instance, more slack time implies a more robust plan, at the cost of a smaller solution space. Moreover, for simplicity and robustness, it can be beneficial to use certain parts of the infrastructure as little as possible. In general, planners have default norms for such options, with deviations from these norms only in rare exceptions.

A planner is looking for a solution with sufficient quality. The most important aspects of this quality are:

- the traveled distance,
- the number of changes in direction,
- deviations from the earliest possible start times, and the latest possible end times, and
- the number of routes operated simultaneously in time.

The number of simultaneous routes in time can be used as a proxy for the minimum number of drivers. Solutions with less simultaneous routes increase the chance of finding a good solution for the crew planning problem.

Note that for a free track, the route cost might differ for each approach side of the track. That is, in Figure 2.2, a route from platform track 3a to track 98 via the A-side of track 98 might have different cost than via the B-side. Because the chosen route for a certain request is also influenced by the traffic situation, this route can be different at different times. Indeed, if a route with already granted reservations of infrastructure is conflicting with the ideal route for a certain request, an alternative route for this request has to be found.

2.3.4 Cleaning

Like the punctuality of the railway system, the cleanliness of rolling stock and stations directly influences the perceived quality of the offered service to passengers. In fact, providing clean rolling stock is one of the 5 main objectives of Dutch Railways. Moreover, it has a considerable impact on the feelings of safety of passengers. Therefore, clean rolling stock and clean stations are quite important to a passenger railway operator. While rolling stock operates train services, it becomes dirty, and cleaning of rolling stock becomes necessary. Several factors which influence the extent to which rolling stock becomes dirty are:

- the number of passengers as well as the type of passengers,
- the season,
- the route the rolling stock has traveled,
- the type of rolling stock.

As described in Section 1.5, one of the main tasks of NedTrain is to clean the rolling stock of NSR. At night, over 600 cleaning crews work for NedTrain at 35 locations in the Netherlands in order to keep the rolling stock clean. In general, cleaning starts after the afternoon rush hours and ends before the start-up process in the next morning. However as an exception, cleaning can also be carried out during the day. This requires that train units remain idle at a station during part of the day and therefore cannot operate passenger services.

By analyzing performances of cleaning processes at different stations one can identify bottlenecks. One can also make similar analyses based on different types of rolling stock. Quarles van Ufford et al. [2002] report the results of such analyses.

In the Netherlands, the passenger perception of cleanliness is part of the valuation of NSRs services and is continuously monitored. Therefore, it is important that both aspects are sufficiently correlated with the perception of passengers.

This cleaning of rolling stock falls apart in cleaning of the exterior of the rolling stock and cleaning of the interior. First, we separately discuss the characteristics of both problems. This is followed by the physical aspects of both problems. Finally, we discuss both problems from the planners' perspective.

Different types of **internal cleaning** are:

1. Cleaning at the end of a railway passenger line. This consists of fast cleaning of the interior of the train and emptying trash cans.

2. Modular cleaning. All the standard internal cleaning activities are divided over several modules. Each module has a prescribed frequency.

3. Periodic thorough cleaning. Typically, this is scheduled once every few months together with large maintenance activities.

4. Urgent cleaning.

The standard modules for internal cleaning are given in Table 2.7.

Number	Name	Description
M0	Basic	Basic cleaning of the interior and sanitary facilities
M1	Seating	Cleaning of the benches and surroundings
M2	Floor	Cleaning of the floor
M3	Glass	Cleaning of all glass parts inside the unit
M4	Cabin	Cleaning of the cabin of the train crew
M5	Toilet	Cleaning of the sanitary facilities

Table 2.7: An overview of the different modules for internal cleaning of train units.

Recently, NedTrain changed its philosophy regarding internal cleaning. Typically, during a specific night the basic module M0 and one fixed additional module out of M1 - M5 are carried out at all NedTrain locations. Thus, on e.g. Monday nights the floors of all train units are cleaned. Previously, the exact additional modules to be carried out were determined for each train unit separately. The new approach makes the operations robust with respect to disturbances, such as delayed arrivals of train units and train units that end their duty in a different station than originally planned.

The **physical process of internal cleaning** takes place along dedicated platforms. If it is impossible to clean a train unit internally at a track along such a platform, one can consider cleaning a train unit at some other track. In such a case, strict rules concerning the safety of the cleaning crews apply. Train units can also be cleaned at a non-dedicated track when the station lacks dedicated tracks.

The **physical process of external cleaning** needs a train-wash, which is available at 15 stations in the Netherlands. Because it requires specialized equipment, external cleaning cannot be carried out at other tracks. Although external cleaning rules prescribe that train units should be washed once every 48 hours, it may be postponed for at most one day as an exception. Moreover, urgent cleaning is required for removing graffiti and after incidents, such as a suicide attempt.

In most cases, **the planner** of the cleaning process is employed at NedTrain. This planner has internal cleaning norms available for each type of rolling stock. These norms are given in man-minutes, which is the standard amount of work one person can do in one minute. In Table 2.8 we give the norms for the types of our example from Section 2.2. These norms represent the required time for the basic module M0 and the average required time over the additional modules. Similar norms are available for external cleaning, but represent the amount of time the train-wash needs to externally clean a type of rolling stock.

Type	Description	Norm (in man-minutes)
DH_2	Single-deck	52
IRM_3	Double-deck	233
IRM_4	Double-deck	317
ICM_3	Single-deck	127
ICM_4	Single-deck	151

Table 2.8: Internal cleaning norms for the types of rolling stock in our main example.

The internal cleaning is usually carried out by two shifts of cleaning crews. Typically, the first shift starts at 18:00 and works until 02:30, with a break scheduled in between, while the second shift starts at 22:00 and works until 06:30 in the next morning. This means that there is some overlap of the shifts.

The required internal cleaning time typically decreases when the number of crews in a shift increases. Caused by varying numbers of cleaning crews available at different times and the start and end times of these shifts, the processing time for internally cleaning a train unit is time dependent. For external cleaning, the norms are determined by the speed of the train-wash and are not influenced by the number of crews available.

Both internal and external cleaning receive the set of train units that need to be cleaned and the cleaning norms as input. Moreover, the structure of the cleaning shifts and the number of crews in each shift are input for the internal cleaning process. For each train unit, the planner knows from the matching the interval when the train unit is available for cleaning. We assume that the matching of train units is known. Therefore, a planner knows the time interval during which a train unit is available for cleaning. Since the cleaning typically takes place at dedicated tracks, routing to and from these tracks is also relevant for the planner.

The resulting internal cleaning plan and the external cleaning plan prescribe when each train unit should be cleaned internally and externally.

In general, one would like to schedule the cleaning as close as possible to the arrival time, because it requires less resources if these processes are combined. For each train unit that needs cleaning, three options are available:

1. It is cleaned shortly after its arrival at the station. This implies that it needs to be parked after it has been cleaned until it leaves the station.

2. It is first parked at a shunt track, then routed to a cleaning track, where it is cleaned. Thereafter it is parked at a possibly different track until it leaves the station.

3. It is first parked at a track, then routed to a cleaning track where it is cleaned, after which it directly leaves the station.

The last situation is undesirable, because it conflicts with the overall goal of the shunting process to start up the operations as smoothly as possible in the morning. As one can imagine, the first option is preferred over the second one, since train units will be parked once only, which results in less work for the shunting crews and reduces complexity of the shunting operations. Therefore, the objective of the cleaning process is to clean as many train units as possible "close" in time to their arrival time at the station, given an appropriate definition of "close". In real-life situations, it is clear that not all train units can be cleaned close to their arrival at the station.

If units are parked at a track before cleaning, the planner could couple several train units on a shunt track and send them as one train to the dedicated tracks for cleaning. Combining several train units into one train requires less routing capacity but the processing time of one train unit increases, since it has to wait until all units in the train are cleaned. In practice, train units are only combined for cleaning at a shunt track if the units also leave in the same departing train and are parked in the right order at the track. Moreover, if two train units arrive in the same train, but leave in different ones, it is likely that the units are cleaned together.

2.3.5 Crew Planning

Planning the tasks resulting from the shunting activities is another element of shunt planning worth discussing here. The local shunting activities are generally performed by local shunting crews. in principle, drivers operating the timetable can also perform these activities. These shunting activities fall apart in different types of tasks, requiring different types of crews. The types of crews for these processes are: shunting driver, shunting assistant, and cleaning crews. Shunting drivers have qualifications to route trains over railway infrastructure. Shunting assistants are responsible for activities resulting from coupling and decoupling train units, and preparation of trains before departure. Cleaning crews take care of cleaning the train units. Although each task has a preferred start time, many tasks have flexible start times, such as routing train units over local railway infrastructure, as described in Section 2.3.3. Shunting tasks are combined into duties, representing one day of work for a single crew.

The physical process distinguishes between different types of tasks. In Table 2.9, some specific types of tasks are given. The durations of the tasks for shunting assistants varies to some extent for different stations, in this table reported the values for station Zwolle. As can be seen, the tasks have typically a short duration. This means that many tasks can be combined in one duty. The duration of routing a train unit in Table 2.9 is highly dependent on the distance and the characteristics of a route, and the reservations of other trains, as was discussed in Section 2.3.3. In addition, the required time for cleaning a train depends on characteristics of the train, and the number of crews available as discussed in Section 2.3.4.

Coupling of train units typically occurs at a platform. However, if several units leaving in the same departing train are parked at the same track and in the right order then the units are combined at the shunt track, as discussed in Section 2.3.2. As described in Section 2.3.3, routing of train units is necessary from platform tracks to the shunt yard and vice versa, and possibly also for other local processes. This routing requires additional qualifications of crews as compared to the other tasks.

Finally, precedence relations between tasks for shunting crews might occur. For instance, before a train is routed to a shunt track, its doors have to be closed.

From **the planner's perspective**, the tasks for shunt crews can be derived from the other shunting plans. Table 2.9 reports the norms for several tasks in the column 'Duration'. In addition, many tasks only occur at a specific track and it is quite common for crews to walk between two tasks from the end track of the first task to the start track of the second task. Norms also exist for walking from one track to another. Both the norms for the tasks and for walking need to be respected by the planner in the resulting duties for the crews.

We discuss some typical tasks for local crews based on our main example in Section 2.2. We exclude tasks concerning the cleaning of the train units from this example.

Task	Description	Crew type	Duration (in min.)
Routing	Routing train units over local railway infrastructure	Shunting driver	Dependent on characteristics
Coupling	A train departing with more train units than arriving	Shunting assistant	3
Decoupling	A train departing with less train units than arriving	Shunting assistant	2
Preparing for departure	Perform some checks on e.g. brakes and open the train for boarding passengers	Shunting assistant	4
Closing	Closing the doors of a train prior to shunting	Shunting assistant	2
Internal cleaning	Internal cleaning of a train	Cleaning crew	Dependent on characteristics
External cleaning	External cleaning of a train	Cleaning crew	Dependent on characteristics

Table 2.9: Specific tasks resulting from shunting activities for crews.

In Table 2.10, we repeat the relevant characteristics of all train services, which take place either at platform 3a or at platform 5a. Although train services 10771, 721 and 10721 do not result in tasks for shunting crew, they do take place at platform 3a and are therefore incorporated. Note that arriving service 771 is split into departing services 771, 10771 and one ICM_3 arriving shunt unit. The tasks resulting from these train services are given in Table 2.11. In this table, most of the activities have little flexibility in their timing. However, no service arrives at or departs from platform 3a after service 771 until the next morning. Similarly, no service arrives at or departs from platform 5a

Train ID	Platform	Time	Event	Configuration
771	3a/b	Tu 20:46	A	ICM_3 ICM_4 ICM_3 ICM_3
771	3a/b	Tu 20:49	D	ICM_3 ICM_3
10771	3a	Tu 20:52	D	ICM_4
3623	5a	We 5:50	D	IRM_3 IRM_4
721	3a	We 7:46	A	ICM_3
10721	3a	We 7:52	D	ICM_3 ICM_3

Table 2.10: All train services of our main example at platforms 3a and 5a.

before service 3623. Therefore, the routing of the remaining ICM_3 unit of service 771 and the routing of the departing units of service 3623 have flexible start times. Of course, if the routing of the ICM_3 unit starts later, then this also results in some additional flexibility in the timing of the closing of this unit. This is similar for the routing of the departing units and the preparation of these units before departure.

Train ID	Time	Task
771	Tu 20:47 - 20:49	Decouple the ICM_4 unit
771	Tu 20:49 - 20:51	Decouple the rightmost two ICM_3 units
771	Tu 20:51 - 20:53	Close the remaining ICM_3 unit
771	Tu 20:53 - 20:55	Route the ICM_3 unit from its arrival platform to a shunt track
3623	We 05:43 - 05:45	Route the IRM_3 and IRM_4 units from a shunt track to their departure platform
3623	We 05:45 - 05:49	Prepare the train units for departure
3623	We 07:44 - 07:48	Route one ICM_3 unit from the shunt yard to platform 3a
10721	We 07:48 - 07:51	Couple the two ICM_3 units at platform 3a

Table 2.11: The tasks resulting from the services in Table 2.10.

In general, a duty contains some reporting time before the start of the first task and some sign off time after the last task of the duty. Moreover, certain restrictions concerning the timing of the meal break in the duty apply. Finally, the maximum length of a duty is typically determined by the start time of the duty. These restrictions are quite common in crew scheduling problems.

The short duration of the tasks combined with the flexible start times of some tasks, discerns this problem from known crew scheduling problems. More information on this problem can be found in Hoekert [2001].

The output of this subproblem is a set of duties that cover all the work resulting from the shunt plans. These duties have to comply with laws and union regulations and need to respect the norms for the tasks and the walking times.

The objective of this subproblem is to use the available crews as efficiently as possible.

2.4 RELATIONS BETWEEN SUBPROBLEMS

The subproblems described in Section 2.3 are interrelated parts of the overall planning problem that a shunt planner faces. In practice, planners are mostly unaware of such a decomposition because of these relations. The most important relations are:

- Matching and routing. The minimum time difference in a matching of an arriving unit to a departing unit is among others determined by the routing time from the arrival platform to the departing platform.

- Matching and parking. The result of the matching determines when train units are available for parking, and when these should leave the station again. Moreover, if the time difference between arrival and departure of a train unit is sufficiently small, parking is not required.

- Matching and cleaning. The matching determines for each train unit the duration of its stay at a station. In turn, this duration determines how much flexibility is available for the cleaning operations.

- Routing and parking. The routing effort influences preferences for certain tracks over other tracks for parking train units.

- Routing and cleaning. Since train units are normally routed to dedicated tracks for the cleaning operations, cleaning requires additional routing effort.

- Cleaning and parking. Cleaning can change the arrival time of a train unit at a shunt track and / or the departure time of a train unit from a shunt track. Moreover, the cleaning processes offer opportunities for rearranging train units at the shunt tracks with little additional effort.

- Crew planning and other subproblems. All the other subproblems typically result in tasks for local shunting crews. For example, the matching might prescribe coupling certain train units, which requires a crew.

As mentioned in Section 1.10, although crew planning for local crews is a part of the shunting problem, we consider it outside the scope of the thesis. In the remainder of the thesis, we will only briefly mention relations of other subproblems with crew planning whenever appropriate.

Chapter 3

Matching of Train Units

As mentioned before, a part of a shunt planner's job is to match arriving train units to later departing units. Although a large part of the solution to this matching problem has been made in previous planning processes, a significant number of arriving train units still needs to be matched to departing train units.

This matching problem has not been introduced or modeled before, which will be done in this chapter. It resembles certain problems in railway freight transportation to some extent. In railway freight transportation, carriages with different destinations are grouped and travel together to the same destination. Moreover, in railway freight transportation, the frequency and configuration of trains also need to be decided upon.

The general matching problem can be conveniently modeled as an integration of a set of shortest path problems and a classical matching problem. The decomposition and composition of each arriving respectively departing train results in one shortest path problem. The results of the shortest path problems of the arriving trains is simultaneously matched to the results of the shortest path problems of the departing trains. In turn, the general matching model is stated as an Integer Program.

In its most general form, the matching of arriving units to departing units belongs to the class of notoriously difficult mathematical problems: it is $\mathcal{NP}$-hard. However, some special cases can be solved in polynomial time and therefore belong to problem class $\mathcal{P}$.

Instances of the developed model for stations Zwolle and Enschede can be solved efficiently by commercially available solvers. The solution process requires no more than one second. Moreover, shunt planners have the opportunity to influence the solution in several ways. The solution process requires no more than one second. Moreover, shunt planners have the opportunity to influence the solution in several ways.

3.1 INTRODUCTION TO MATCHING

To properly define the problem of matching train units, we introduce the following terms:

- A shunt unit is a unit that needs to be parked at or supplied from the shunt yard.
- An arriving shunt unit is a shunt unit that has to be parked at a shunt yard. Such units come from complete ending train services or are decoupled from through train services. The set of arriving shunt units is also called the supply of shunt units.
- A departing shunt unit is a shunt unit that needs to be supplied from the shunt yard. These units form complete starting train services or are coupled onto through train services. The set of departing shunt units is also called the demand for shunt units.
- A part is an entity of one or more adjacent train units in one train. Similar to shunt units, we can distinguish between arriving and departing parts.
- A block is a matching of an arriving part to a later departing part. The train units in such a block are meant to remain together during their stay at the station under consideration.

In Section 1.10, we restricted our research to one station at a time, a 24-hour planning period, and a given timetable. In this chapter, we add the following two assumptions:

- For each type of rolling stock the arrivals and departures are balanced. This means that in a 24-hour period, the number of arriving train units of a specific type equals the number of departing train units of this type. Moreover, from the start of the planning period to each point in time, the number of arriving train units of a certain type is at least equal to the number of departing train units of the same type.
- The inventory of train units at the shunt yard under consideration is empty at the start and the end of the planning period.

Note that these assumptions can be relaxed easily by adding dummy arrivals and departures to the timetable, which will be done in Section 3.5.

In one block, the difference between the arrival and departure times should be sufficiently large. In addition, a shunt planner is not allowed to change the configuration of the timetabled train services, since these have been decided upon in previous planning processes. More specifically, type mismatches are not allowed and, in case a train consists of different types, the order of the train units in the train also needs to be respected. For instance, a train with configuration ICM_3 - ICM_4 - ICM_4 is considered different from a train with the same types but in a different order, say ICM_4 - ICM_3 - ICM_4.

As mentioned in Section 2.2, we use the following convention in the thesis: the leftmost train unit type in the order is the unit which is closest to the A-side of the train and similarly, the rightmost unit is closest to the B-side of the train. The A-side of a train is the side which is closest to the A-side of the station whenever the train is within the station boundaries.

The differentiation between these trains is due to work later on in the duties of the rolling stock. As an example, the relevant details of train services 3672 and 3687 from Table A.5 on page 207 are repeated in Table 3.1.

Train ID	Platform	Time	Event	Configuration	Direction
3672	7a/b	Tu 22:09	A	IRM_4 IRM_3	Rsd (OW)
3687	7a/b	Tu 22:23	D	IRM_4	Rsd (WO)

Table 3.1: The order of train units within one train is important.

Indeed, if the units in the train of arriving service 3672 would be in a different order, then the train of departing service 3687 would consist of a unit of type IRM_3 instead of IRM_4. However, interchanging these units results in imbalances in the inventories of rolling stock at stations, which is highly undesirable. Moreover, it is possible that the duty of the train unit in the through train 3687 contains tasks, which require 4 carriages due to expected demand.

In conclusion, a matching of an arriving part to a departing part is feasible whenever:

- the time difference between departure and arrival is sufficiently large,
- type mismatches do not occur, and
- the units in both parts have the same order.

Important objectives that shunt planners take into account in the shunt planning process include:

- Keeping the units resulting from the same train together as much as possible, since this results in a minimum of required resources. This has already been discussed in Section 2.3.1.
- Maximizing the number of blocks with a minimum difference between arrival and departure time. If the arrival platform differs from the departure platform, a small time difference represents the time for routing from the arrival platform to the departure platform.
- Preferring solutions with certain desirable characteristics. We consider matchings with a LIFO character to be better than matchings with a FIFO character.

The last aspect tries to leave sufficient solution space for the parking subproblem. Suppose we have two arriving shunt units a and c and two departing shunt units b and d. In addition, all units have the same type, and the sequence of arrival and departure is $a < b < c < d$. The two possible matchings for these units are depicted in Figure 3.1. We continue with a discussion of the differences between these solutions.

Figure 3.1: Two types of matchings.

Regarding the parking subproblem, two types of shunt tracks are available: LIFO tracks, and free tracks (see Figure 1.1 on page 2). At LIFO tracks, train units can only arrive at and depart from one side of the track. At free tracks, train units can arrive at and depart from both sides of the track. In Section 2.3.3, we already remarked that planners typically prefer one specific side of such a free track for arrivals and departures over the other side. This results in a preference for LIFO matchings, conform the left part of Figure 3.1. Indeed, when parking the results of the FIFO matching at a LIFO track, the departure of unit b is obstructed by the later departing unit d. This problem is avoided with a LIFO matching. At a free track, this obstruction can only be avoided by using the non-preferred side of the track for departing unit d.

All these aspects of the objective add to a solution with a minimum amount of required resources. Now, we are able to formally define the problem of matching arriving train units to departing train units:

Definition 3.1. Given a timetable of arriving and departing train services, the Train Matching Problem (TMP) is to find a feasible matching of arriving train units to departing train units of a minimum required amount of resources.

3.2 RAILWAY FREIGHT TRANSPORTATION AND MATCHING

Based on the definition of TMP, this section reviews several related problems in railway freight transportation. The goal of this review is to highlight the unique characteristics of TMP.

In railway freight transportation, trains typically consist of many carriages containing different commodities and with different destinations. These trains are hauled by one or more locomotives. At a shunt yard for freight trains, carriages are decoupled, regrouped and thereafter coupled to form new outbound trains. Blocks of carriages are formed by grouping carriages, which travel together to the destination of the block. This train formation process is called blocking. Note that the corresponding blocks differ from blocks that result from TMP. Subsequently, a solution to a routing problem determines the configurations and frequencies of trains. Finally, the makeup model aims at assigning blocks to trains. A survey of models and solution approaches for such problems can be found in Cordeau et al. [1998].

Dahlhaus et al. [2000] discuss the related problem of rearranging carriages in one freight train at a yard to group them by destination. Their goal is to use a minimum number of tracks for this rearrangement and they show that this problem is $\mathcal{NP}$-hard.

In addition, He et al. [2000] extend this problem by considering multiple trains. In their approach, the rearranging of carriages is a two-stage process where arriving carriages are firstly classified and secondly assembled before making up the departing train. Classification and assembly take place at different sets of parallel tracks. In their approach, the occupation of shunt tracks by carriages is not taken into account. He et al. [2003] propose an integrated model and solution approach for the problem described in He et al. [2000], where the occupation of the tracks by the trains is also taken into account.

The matching problem described in this chapter has some similarities with the above described makeup problem in railway freight transportation. Indeed, when the configurations and schedules of trains are known in the makeup problem, one has to decide how to match the carriages from arriving trains to departing trains. Di Stefano and Kŏci [2003] study the computational complexity of several problems combining aspects from the overall shunting problem and these freight transportation problems. Furthermore, they also present algorithms for solving some of these problems, including bounds on the objectives and on the complexity of the algorithms.

3.3 GENERAL MODEL FOR TMP

The matching model for TMP contains two components. The first component is a set of shortest path problems. The set contains one such problem for each train. The goal of such a shortest path problem is to determine the composition or decomposition of a train into sets of train units that remain together. The shortest path problems result in parts. The second component is the problem of matching the arriving parts to the departing parts, and resembles a classical matching problem. These two components are solved simultaneously. We start with the discussion of the shortest path problems, followed by the discussion of the matching problem. For these discussions, we introduce the following notation:

- $\mathcal{T}=\mathcal{T}^+ \cup \mathcal{T}^-$ is the set of trains, which can be decomposed in the set of arriving trains $\mathcal{T}^+$ and the set of departing trains $\mathcal{T}^-$.
- $\mathcal{K}$ represents the set of configurations of trains. A configuration $k \in \mathcal{K}$ consists of an ordered number of types of train units. The i-th type of a configuration k is denoted by $k[i]$. Moreover, $|k|$ represents the number of train units in configuration $k \in \mathcal{K}$.

For each train $t \in \mathcal{T}$, we define a network $\mathcal{G}_t = (\mathcal{N}_t, \mathcal{A}_t)$. For each train unit in train t, a node is introduced with an index. The unit closest to the A-side of the train receives index 1, and subsequent units are indexed increasingly. Together with a dummy node with index 0, we have $\mathcal{N}_t = \{0, \ldots, |k_t|\}$ for each train. The non-dummy nodes represent the places where the train can be divided into parts. The arcs of the network $\mathcal{G}_t$ represent the potential parts of the train: $\mathcal{A}_t = \{(i,j) \mid i < j, i \in \mathcal{N}_t, j \in \mathcal{N}_t\}$.

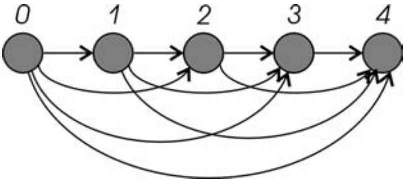


Figure 3.2: The network of arriving train 771 consisting of 4 train units.

We base our examples in this chapter on the main example. Details of this example can be found in Section A.2. Figure 3.2 depicts the network of arriving train 771 consisting of 4 units (see Table A.5 on page 207). Note that each arc corresponds to one potential part. Therefore, we have 10 potential parts in this example. Furthermore, each path from the first node to the last node in this network corresponds to a disassembly of the train into parts. For example, the path 0→1→2→4 in Figure 3.2 implies that train unit 1 makes up one part, just as train unit 2, and train units 3 and 4 form the last part.

The fixed matchings of our main example in Table A.5 are given in Table 3.2. Fixed matchings result from timetabled through trains. Moreover, we fix a matching if an arriving unit has precisely one option to be matched with a departing one. This is the reason why the IRM_3 unit of train 3672 is fixed. Finally, the ICM_3 unit of departing train 10721 in this table refers to the unit on the B-side of the train. That is, an additional ICM_3 unit needs to be coupled onto the A-side to complete its configuration of two ICM_3 units.

The second component of TMP combines the shortest path problems for the arriving and departing trains by matching the results of these problems. This matching takes into account the feasibility of a matching of an arriving part to a departing part.

Arriving train ID	Arriving units	Departing train ID	Departing units
771	ICM_4 ICM_3 ICM_3	771	ICM_3 ICM_3
		10771	ICM_4
3672	IRM_4 IRM_3	3687	IRM_4
		3623	IRM_3
7984	DH_2	7917	DH_2
3680	IRM_4	3623	IRM_4
721	ICM_3	10721	ICM_3

Table 3.2: The fixed part of the matching of Table A.5.

In Figure 3.3, we give a graphical representation of both components for our main example. In this figure, we intentionally left out the matching arcs between fixed matchings for clarity of exposition. The shortest path arcs conflicting with the fixed parts are also left out the figure. This leads to two isolated nodes, one in arriving train 771 and one in departing train 771.

Before stating the description of the overall model, we introduce some relevant notation.

- $\mathcal{N}_t' = \mathcal{N}_t \setminus \{0, |k_t|\}$. $\mathcal{N}_t'$ is the set of all intermediate nodes in $\mathcal{N}_t$.
- $\mathcal{A}_t^{i+} = \{j \mid (i,j) \in \mathcal{A}_t\}$, and $\mathcal{A}_t^{i-} = \{j \mid (j,i) \in \mathcal{A}_t\}$. For each node $i \in \mathcal{N}_t$, $\mathcal{A}_t^{i+}$ is the set of arcs emanating from this node, and similarly $\mathcal{A}_t^{i-}$ is the set of incoming arcs at this node.
- $\mathcal{A}^+ = \bigcup_{t \in \mathcal{T}^+} \mathcal{A}_t$ and $\mathcal{A}^- = \bigcup_{t \in \mathcal{T}^-} \mathcal{A}_t$. $\mathcal{A}^+$ represents all potential arriving parts, and $\mathcal{A}^-$ represents all potential departing parts.
- The set $\mathcal{T}_{t'}^-$ represents the set of trains that depart sufficiently later than arriving train $t' \in \mathcal{T}^+$. Similarly, the set $\mathcal{T}_{t'}^+$ represents the set of trains that arrive sufficiently earlier than departing train $t' \in \mathcal{T}^-$.
- For each arriving train $t' \in \mathcal{T}^+$ and for each arc $(i,j) \in \mathcal{A}_{t'}$, we introduce the set $\mathcal{A}_{(i,j)}^-$ representing the departing potential parts with the same configuration as the potential part denoted by (i,j):

$$\mathcal{A}_{(i,j)}^- = \bigcup_{t \in \mathcal{T}_{t'}^-} \bigcup_{(g,h) \in \mathcal{A}_t} \{(g,h) \mid k_t[g+1] = k_{t'}[i+1], \ldots, k_t[h] = k_{t'}[j]\}$$

- For each $t' \in \mathcal{T}^-$ and for each $(i,j) \in \mathcal{A}_{t'}$, we introduce the set $\mathcal{A}_{(i,j)}^+$ representing the arriving potential parts with the same configuration as the potential part

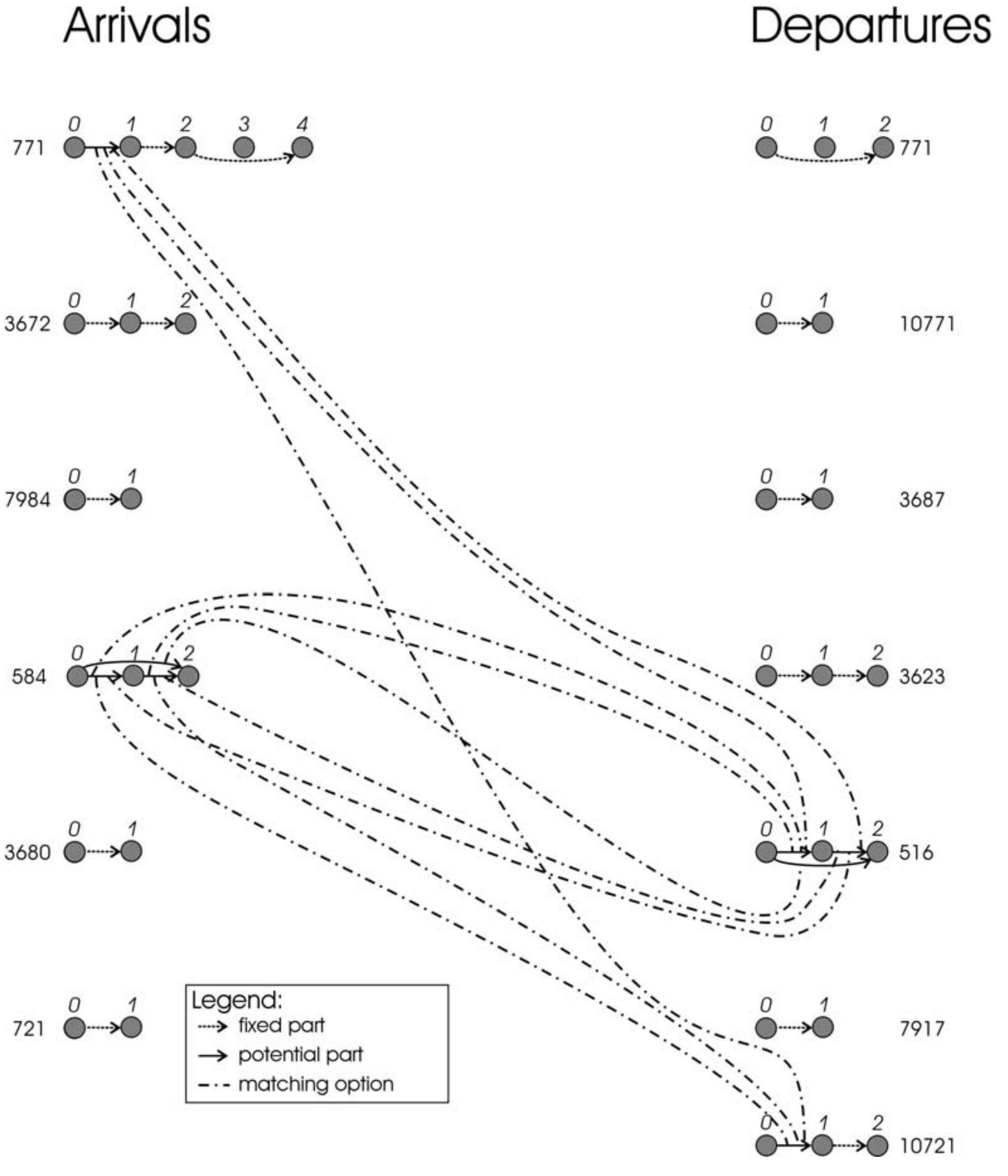


Figure 3.3: The matching network of our main example.

denoted by (i, j) resulting from departing train t':

$$\mathcal{A}^{+}_{(i,j)} = \bigcup_{t \in \mathcal{T}^{+}_{t'}} \bigcup_{(g,h) \in \mathcal{A}_t} \{(g,h) \mid k_t[g+1] = k_{t'}[i+1], \ldots, k_t[h] = k_{t'}[j]\}$$

The decision variables in the developed model for TMP are:

$$X_a = \begin{cases} 1 & \text{if arriving part } a \in \mathcal{A}^{+} \text{ is used;} \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_d = \begin{cases} 1 & \text{if departing part } d \in \mathcal{A}^{-} \text{ is used;} \\ 0 & \text{otherwise.} \end{cases}$$

$$Z_{a,d} = \begin{cases} 1 & \text{if arriving part } a \in \mathcal{A}^{+} \text{ is matched with departing part } d \in \mathcal{A}^{-}_{a}; \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, the parameter q models a penalty for each arriving part that is used. Note that imposing these penalties on the departing parts has the same effect. The parameters $m_{a,d}$ model the cost of matching arriving part $a \in \mathcal{A}^{+}$ with departing part $d \in \mathcal{A}^{-}_{a}$.

The model for TMP can be described as follows:

$$\text{minimize} \quad q \sum_{a \in \mathcal{A}^{+}} X_a + \sum_{a \in \mathcal{A}^{+}} \sum_{d \in \mathcal{A}^{-}_{a}} m_{a,d} Z_{a,d} \tag{3.1}$$

$$\text{subject to} \quad \sum_{a \in \mathcal{A}^{t+}_{0}} X_a = 1 \qquad \forall t \in \mathcal{T}^{+} \tag{3.2}$$

$$\sum_{a \in \mathcal{A}^{t+}_{n}} X_a - \sum_{a \in \mathcal{A}^{t-}_{n}} X_a = 0 \qquad \forall t \in \mathcal{T}^{+}, \forall n \in \mathcal{N}'_t \tag{3.3}$$

$$\sum_{d \in \mathcal{A}^{t+}_{0}} Y_d = 1 \qquad \forall t \in \mathcal{T}^{-} \tag{3.4}$$

$$\sum_{d \in \mathcal{A}^{t+}_{n}} Y_d - \sum_{d \in \mathcal{A}^{t-}_{n}} Y_d = 0 \qquad \forall t \in \mathcal{T}^{-}, \forall n \in \mathcal{N}'_t \tag{3.5}$$

$$\sum_{d \in \mathcal{A}^{-}_{a}} Z_{a,d} = X_a \qquad \forall a \in \mathcal{A}^{+} \tag{3.6}$$

$$\sum_{a \in \mathcal{A}^{+}_{d}} Z_{a,d} = Y_d \qquad \forall d \in \mathcal{A}^{-} \tag{3.7}$$

$$Z_{a,d}, X_a, Y_d \in \{0,1\} \qquad \forall a \in \mathcal{A}^{+}, \forall d \in \mathcal{A}^{-} \tag{3.8}$$

The objective (3.1) is to minimize the weighted sum of the number of parts and the matching cost. Flow conservation restrictions (3.2) and (3.3) assure the covering of each arriving unit by exactly one arriving part. Restrictions (3.4) and (3.5) assure the same for departing units. Assignment restrictions (3.6) guarantee that each arriving part is matched with a departing part if and only if the arriving part is a result of the train disassembly. Restrictions (3.7) model this in a similar way for the departing parts. Note that the definition of the sets $\mathcal{A}_a^-$ and $\mathcal{A}_d^+$ in (3.6) and (3.7) ensure that the matchings are feasible. We call (3.1)-(3.8) Model (3.a).

The objective to prefer matchings with a LIFO character can be represented by penalizing matchings with an 'average' time difference between the arriving and the departing part. This penalty can be incorporated in the parameters $m_{a,d}$. Fodor Birtalan [2003] discusses a more precise approach to prefer LIFO matchings. In this approach, additional decision variables are introduced to indicate FIFO matchings. These decision variables are then penalized in the objective function.

For a short discussion of the approach by Fodor Birtalan, we introduce a partial order $<$ on the parts $p \in \mathcal{A}^+ \cup \mathcal{A}^-$. The following holds: $p < p'$ if and only if the train service of part p arrives or departs before the train service of part p'. Note that this partial order disregards units from the same train service. Moreover, the set $\mathcal{F}$ and the binary decision variables $F_{a,a',d}$ are defined as follows:

$$\begin{aligned}\mathcal{F} &= \{(a,a',d) \mid a,a' \in \mathcal{A}^+, d \in \mathcal{A}_a^-, d \in \mathcal{A}_{a'}^-, \text{ and } a < a' < d\} \\ F_{a,a',d} &= \begin{cases} 1 & \text{if } (a,a',d) \in \mathcal{F}, a \text{ is matched to } d, \text{ and } a' \text{ to some } d' > d \\ 0 & \text{otherwise.}\end{cases}\end{aligned}$$

Given the set $\mathcal{F}$ and the decision variables $F_{a,a',d}$, the restrictions that ensure that the decision variables will be set to the appropriate values are:

$$Z_{a,d} + \sum_{d'>d} Z_{a',d'} \leq 1 + F_{a,a',d} \quad \forall (a,a',d) \in \mathcal{F}$$

Finally, the objective (3.1) is extended with the cost of FIFO matchings. The penalty for one FIFO matchings is $m_{aa'd}$. The new objective reads:

$$\text{minimize} \quad q \sum_{a \in \mathcal{A}^+} X_a + \sum_{a \in \mathcal{A}^+} \sum_{d \in \mathcal{A}_a^-} m_{a,d} Z_{a,d} + \sum_{(a,a',d) \in \mathcal{F}} m_{a,a',d} F_{a,a',d} \tag{3.9}$$

Extensions of these restrictions are required when arrivals are not matched with a departure in the planning period. The effectiveness of this extension of TMP is analyzed after solving the parking subproblem and reported for station Zwolle in Fodor Birtalan [2003]. For Tuesday / Wednesday instances, these restrictions seem useful. However, the characteristics of the Saturday / Sunday instances, with more arriving units than departing units, require additional modifications and tuning for good results.

3.4 COMPLEXITY OF MATCHING TRAIN UNITS

In this paragraph, we describe some results on the complexity of TMP. This paragraph is partly based on the work by Fodor Birtalan [2003]. First, we will show that TMP in its most general form is $\mathcal{NP}$-complete. This is followed by two special cases of TMP, which are solvable in polynomial time and space. Both cases are illustrated with an example. This paragraph requires the following definitions:

- $\mathcal{Y}$ is the set of different types of rolling stock.
- $\mathcal{B}_{k_t}$ is the set of potential disassemblies of a train t with configuration k_t.

For instance, if A and B represent different types, $k_t = AAB$, and ';' indicates a split in the train, then $\mathcal{B}_{AAB}$ contains 4 elements representing the different potential disassemblies of a train with configuration AAB, i.e. $\mathcal{B}_{AAB} = \{(AAB), (A; AB), (AA; B), (A; A; B)\}$.

3.4.1 TMP is $\mathcal{NP}$-hard

$\mathcal{NP}$-hardness of TMP is proven by a reduction from the 3 Partition Problem (3PP) of a special case of the decision version of TMP. 3PP is defined in Garey and Johnson [1979]. They also prove that 3PP is $\mathcal{NP}$-complete in the strong sense. For convenience, we repeat the definition of the decision version of 3PP here:

Definition 3.2. Given is a finite set $\mathcal{A}$ of $3m$ elements, a bound $s \in \mathbb{Z}^+$, and a "size" function $\sigma_a \in \mathbb{Z}^+$ for each $a \in \mathcal{A}$, such that each σ_a satisfies $s/4 < \sigma_a < s/2$ and such that $\sum_{a \in \mathcal{A}} \sigma_a = ms$. Then the question of 3PP is whether there exists a partitioning of $\mathcal{A}$ into m disjoint sets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$ such that, for $1 \le i \le m$, $\sum_{a \in \mathcal{S}_i} \sigma_a = s$.

Moreover, we define the decision problem TMP-0 as follows:

Definition 3.3. Given a timetable of arriving and departing train services, the question of TMP-0 is whether the arriving units can be matched to the departing units in a feasible manner, such that no arriving services are split.

Lemma 3.4. *TMP-0 is $\mathcal{NP}$-complete.*

Proof. It is not difficult to see that TMP-0 is in $\mathcal{NP}$. For a given matching, one can check in polynomial time whether sufficient time differences exist, whether type mismatches are lacking and whether the orders of the train units in the arriving and departing parts are the same.

Given an instance I of 3PP, we construct an instance I' of TMP-0 as follows. All train units have the same type, and the length of a train is defined as its number of train units. Moreover, the first departure in the timetable takes place after the last arrival and a sufficiently large buffer time have passed. The number of arriving trains is $3m$, each arriving train $a \in \{1, \ldots, 3m\}$ having length σ_a. Finally, I' contains m departing trains of length s.

Suppose I is a yes-instance of 3PP. Then, for each $\mathcal{S}_i$, all units of arriving trains $a \in \mathcal{S}_i$ leave in departing train i. This obviously is a feasible matching of arriving units to departing units in I'.

Conversely, suppose I' is a yes-instance of TMP-0. Then, each departing train i contains all train units of exactly 3 arriving trains $\mathcal{S}_i$. Thus, the sets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$ form a feasible solution to 3PP.

Concluding, I is a yes-instance of 3PP if and only if I' is a yes-instance of TMP-0. Since 3PP is $\mathcal{NP}$-complete and TMP-0 is in $\mathcal{NP}$, TMP-0 is $\mathcal{NP}$-complete as well. ■

Corollary 3.5. *TMP is $\mathcal{NP}$-hard.*

Proof. This proof follows directly from Lemma 3.4 and the fact that minimizing the number of split arriving services is a special case of the objective of TMP. ■

3.4.2 3TMP is Polynomially Solvable

Now, we consider a special case of TMP and show that it belongs to the class $\mathcal{P}$, since it can be formulated as an uncapacitated minimum cost flow problem. The definition of this problem is adapted from AHUJA ET AL. [1993, CHAPTER 9].

Definition 3.6. Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed network with cost $f_{i,j}$ associated with every arc $(i,j) \in \mathcal{A}$. We associate with each node $i \in \mathcal{N}$ a number b_i indicating its supply or demand, depending on whether $b_i > 0$ or $b_i < 0$. Moreover, decision variables $X_{i,j}$ denote the flow on arc $(i,j) \in \mathcal{A}$. The uncapacitated minimum cost flow problem (MCF) can be stated as follows:

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} f_{i,j} X_{i,j} \tag{3.10}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in\mathcal{A}} X_{i,j} - \sum_{j:(j,i)\in\mathcal{A}} X_{j,i} = b_i \qquad \forall i \in \mathcal{N} \tag{3.11}$$

$$0 \leq X_{i,j} \quad \forall (i,j) \in \mathcal{A} \tag{3.12}$$

At the time of writing, the fastest known algorithm for this problem is described by ORLIN [1993] and runs in $\mathcal{O}(|\mathcal{N}| \log |\mathcal{N}| (|\mathcal{A}| + |\mathcal{N}| \log |\mathcal{N}|))$ time. Subsequently, we define 3TMP and prove it to be polynomially solvable.

Definition 3.7. The 3 Train Matching Problem (3TMP) is TMP restricted to instances where trains consist of at most 3 units, where the objective is to minimize the number of blocks and the first departure takes place after the last arrival and a sufficiently large buffer time have passed.

Note that, by definition, the matching of an arriving part to a departing part has a sufficient time difference and therefore, the feasibility of such a matching only concerns the absence of type mismatches, and the same order of train units in the arriving and departing part.

Theorem 3.8. *3TMP can be solved in polynomial time and space.*

Proof. This theorem is proved by showing that the problem can be solved as an uncapacitated minimum cost flow problem on a network with numbers of nodes and arcs that are polynomial in the number of rolling stock types, $|\mathcal{Y}|$. The underlying network is based on three identical sets. The nodes in one set represent all possible configurations of trains with 2 or 3 train units and one node aggregating all train configurations with one train unit.

The first set, $\mathcal{S}$, represents the configurations of the arriving train units. The set $\mathcal{R}$ is an intermediate set. Finally, the set $\mathcal{D}$ represents the configurations of the departing train units. The complete set of nodes is defined as $\mathcal{N} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{D}$. We introduce the function $\rho : \mathcal{N} \rightarrow \mathcal{K}$, which gives the configuration of a node. For the three nodes representing all train configurations of 1 unit, ρ returns a configuration consisting of 1 unit of an arbitrary type. Moreover, $\rho(i)[n]$ indicates element n of configuration $\rho(i)$.

Arcs exist between the sets $\mathcal{S}$ and $\mathcal{R}$ and between the sets $\mathcal{R}$ and $\mathcal{D}$. More specifically, an arc (i, j) with $i \in \mathcal{S}$ and $j \in \mathcal{R}$ is present if and only if at least one of the following conditions hold:

- $|\rho(j)| = 1$. In this case, j is the node representing the train configurations of one unit.
- $\rho(i) = \rho(j)$. Here, nodes i and j represent the same configuration.
- $|\rho(i)| = 3$, $|\rho(j)| = 2$, and one of the following two conditions hold:
  1. $\rho(i)[1] = \rho(j)[1]$ and $\rho(i)[2] = \rho(j)[2]$
  2. $\rho(i)[2] = \rho(j)[1]$ and $\rho(i)[3] = \rho(j)[2]$

  In this case, the configuration of node j is a subset of the configuration of node i.

In addition, an arc (i, j) with $i \in \mathcal{R}$ and $j \in \mathcal{D}$ is present if and only if at least one of the following conditions hold:

- $|\rho(i)| = 1$. In this case, i is the node representing the train configurations of one unit.
- $\rho(i) = \rho(j)$. Here, nodes i and j represent the same configuration.
- $|\rho(i)| = 2$, $|\rho(j)| = 3$, and one of the following two conditions hold:
  1. $\rho(i)[1] = \rho(j)[1]$ and $\rho(i)[2] = \rho(j)[2]$
  2. $\rho(i)[1] = \rho(j)[2]$ and $\rho(i)[2] = \rho(j)[3]$

  In this case, the configuration of node i is a subset of the configuration of node j.

Arcs between nodes $\mathcal{S}$ and $\mathcal{R}$ represent the disassembly of arriving trains. Moreover, arcs between nodes $\mathcal{R}$ and $\mathcal{D}$ represent the assembly of departing trains. The arcs between $\mathcal{S}$ and $\mathcal{R}$ and between $\mathcal{R}$ and $\mathcal{D}$ form the set $\mathcal{A}$, which completes the construction of the network $\mathcal{G}$=($\mathcal{N}$, $\mathcal{A}$).

For arcs (i,j) with $i \in \mathcal{S}$ and $j \in \mathcal{R}$, we introduce costs $f_{i,j} = ||\rho(i)| - |\rho(j)|| + 1$. The costs represent the number of train configurations in which configuration $i \in \mathcal{S}$ is split. The arc cost for arcs between $\mathcal{R}$ and $\mathcal{D}$ are 0: $f_{i,j}$=0 if $i \in \mathcal{R}$ and $j \in \mathcal{D}$.

Let a_i be the number of arriving trains according to configuration $\rho(i) \in \mathcal{S}$, and define d_i similarly for departing trains. Then the supply or demand at a node $i \in \mathcal{N}$ is defined as:

$$b_i \quad = \quad \begin{cases} a_i & \text{if } i \in \mathcal{S} \text{ and } |\rho(i)| > 1; \\ \sum_{j:|\rho(j)|=1} a_j & \text{if } i \in \mathcal{S} \text{ and } |\rho(i)| = 1; \\ 0 & \text{if } i \in \mathcal{R}; \\ -d_i & \text{if } i \in \mathcal{D} \text{ and } |\rho(i)| > 1; \\ -\sum_{j:|\rho(j)|=1} d_j & \text{if } i \in \mathcal{D} \text{ and } |\rho(i)| = 1. \end{cases}$$

Decision variables $X_{i,j}$ represent the flow on arc $(i,j) \in \mathcal{A}$ and are non-negative. Finally, node $n_1^{\mathcal{R}}$ is the node in $\mathcal{R}$ representing train configurations of 1 unit, i.e. $n_1^{\mathcal{R}} = \{i \in \mathcal{R} \mid |\rho(i)| = 1\}$. Now, we are ready to state the model:

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} f_{i,j} X_{i,j} \tag{3.13}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in\mathcal{A}} X_{i,j} - \sum_{j:(j,i)\in\mathcal{A}} X_{j,i} = b_i \qquad \forall i \in \mathcal{N} \setminus n_1^{\mathcal{R}} \tag{3.14}$$

$$0 \leq X_{i,j} \quad \forall (i,j) \in \mathcal{A} \tag{3.15}$$

Note that the flow conservation restriction of node $n_1^{\mathcal{R}}$ is relaxed in restrictions (3.14). This relaxation resolves a complication which we will discuss after this proof.

Now, $|\mathcal{N}| = \mathcal{O}(|\mathcal{S}|) = \mathcal{O}(|\mathcal{Y}|^3)$, and $|\mathcal{A}| = \mathcal{O}(|\mathcal{S}|^2)$. Therefore, the numbers of variables $X_{i,j}$ and restrictions (3.14) in the formulation (3.13) - (3.15) are polynomial in the size of the input.

The cost on the arcs (i,j) with $i \in \mathcal{S}$ and $j \in \mathcal{R}$ together with the structure of the network $\mathcal{G}$=($\mathcal{N}$, $\mathcal{A}$) ensure that the cost of a minimum cost flow according to (3.13) - (3.15) corresponds to the minimum number of blocks in 3TMP. Therefore, we conclude that 3TMP can be solved in polynomial time and space by solving an uncapacitated minimum cost flow problem in an appropriate network. ■

Now, we discuss the complication mentioned in the proof. When solving a standard uncapacitated minimum cost flow problem (3.10) - (3.12) with $\mathcal{N}$, $\mathcal{A}$, b_i and $f_{i,j}$ as defined above, a complication arises. This complication is clarified in Figure 3.4 with an example instance. From the set $\mathcal{S}$, the figure only depicts the only node with strictly positive supply, i.e. $b_i > 0, i \in \mathcal{S}$. Similarly, from $\mathcal{D}$ only the nodes with strictly positive demand are depicted, i.e. $b_i < 0, i \in \mathcal{D}$. The nodes from $\mathcal{R}$, which are reachable from either the depicted node in $\mathcal{S}$ or a depicted node in $\mathcal{D}$ are also present in the figure. Arcs are present between $\mathcal{S}$ and $\mathcal{R}$ and between $\mathcal{R}$ and $\mathcal{D}$ when previously mentioned conditions are satisfied. In this instance, no feasible solution to the minimum cost flow formulation exists, although the supply and demand for types of rolling stock is balanced and a solution to 3TMP does exist. Notice that a similar instance with the supply and demand of the nodes in $\mathcal{S}$ and $\mathcal{D}$ interchanged is also infeasible.

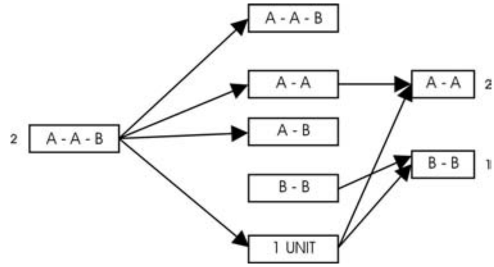


Figure 3.4: An instance of 3TMP resulting in an infeasible instance of the uncapacitated minimum cost flow problem.

This complication can be resolved by relaxing the flow conservation restriction on the node $n_1^{\mathcal{R}}$. The balances between the arrivals and departures of types of rolling stock ensure that this relaxation has the desired effect, and that the solution is still feasible.

As an example of the network $\mathcal{G}$ in the proof of Theorem 3.8, we show the network for the ICM family, which consists of the types ICM_3 and ICM_4, with at most 3 units in one train in Figure 3.5.

3.4.3 FSTMP is Polynomially Solvable

Besides the set of trains $\mathcal{T}$, introduced in the previous chapter, the second special case requires the following additional notation:

- $\overline{\kappa}$=$\max_{k\in\mathcal{K}}\{|k|\}$ is the maximum number of train units in one train.
- p is the maximum number of train units that is simultaneously parked at the shunt yard.
- $k(n,m)$ is a function returning the number of potential train configurations with n different types of rolling stock and at most m train units in a train. Note that $k(n, m) = \mathcal{O}(mn^m)$.

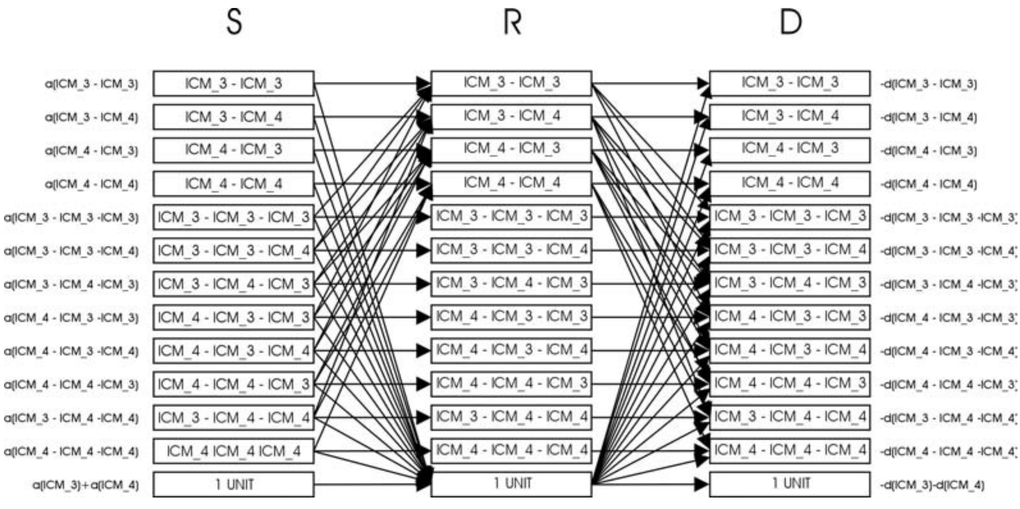


Figure 3.5: The network for an instance of 3TMP restricted to ICM_3 and ICM_4 units.

We show that a second special case of TMP can be solved in polynomial time and space by solving a shortest path problem in an appropriate network. The shortest path problem is well-known and reads:

Definition 3.9. Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a directed network with a nonnegative length $f_{i,j}$ associated with every arc $(i, j) \in \mathcal{A}$. The Shortest Path Problem from source $o \in \mathcal{N}$ to sink $d \in \mathcal{N}$ is to find a path from o to d with minimum length.

DIJKSTRA's algorithm [1959] for the shortest path problem with nonnegative arc lengths is the standard reference for solving the shortest path problem and runs in $\mathcal{O}(|\mathcal{N}|^2)$. Specialized data structures and other algorithms improved on this computation time, see AHUJA ET AL. [1993, CHAPTER 4] for an overview.

Definition 3.10. The Fixed Size Train Matching Problem is a special case of TMP, where the objective is to minimize the number of blocks, that is restricted to instances with at most p train units simultaneously at the yard, and at most $\overline{\kappa}$ train units in a train. We denote such a problem with FSTMP(p,$\overline{\kappa}$).

Theorem 3.11. *FSTMP(p,$\overline{\kappa}$) can be solved in polynomial time and space.*

Proof. We prove this theorem by showing that FSTMP(p,$\overline{\kappa}$) can be solved as a shortest path problem in an appropriate network. The numbers of nodes and arcs are polynomial in the number of types of rolling stock and the number of trains whenever the maximum number of train units in a train and the parking capacity in terms of the number of units at the shunt yard are fixed.

The network is based on the arriving trains $\mathcal{T}^+$, the departing trains $\mathcal{T}^-$, and an initial dummy train prior to all other trains. The nodes in the network are $k(|\mathcal{Y}|, \overline{\kappa})$ vectors, representing the potential inventories of potential configurations of train units at the shunt yard directly after an arrival or departure of a train, or after the dummy train. Note that the size of such a vector is polynomially bounded whenever $\overline{\kappa}$ is fixed. For each potential configuration $l \in \mathcal{K}$, i^l represents the inventory of potential configuration l and has domain $\{0, \ldots, p\}$.

For each train $t \in \mathcal{T}$, we define a layer $\mathcal{L}_t$ of nodes with all potential inventories of all potential train configurations directly after the arrival or departure of train t. In these potential inventories, we take into account the restriction on the maximum number of train units at the yard, that is $\sum_{l \in \mathcal{K}} i^l |l| \leq p$ for each node i in the network. Moreover, each layer $\mathcal{L}_t$ is identical to another layer $\mathcal{L}_{t'}$ for $t, t' \in \{1, \ldots, |\mathcal{T}| - 1\}$. Finally, layers $\mathcal{L}_0$ and $\mathcal{L}_{|\mathcal{T}|}$ represent the empty shunt yard at the start and end, respectively, of the planning period. These layers both contain precisely one node, which is the null vector.

Each potential disassembly $b \in \mathcal{B}_{k_t}$ of a train t with configuration k_t corresponds to a vector $(b[1], \ldots, b[k(|\mathcal{Y}|, \overline{\kappa})])$. Note that $b[l]$ is the l-th element of this vector. For an arriving train, this vector represents the number of occurrences of each configuration in potential disassembly b. For a departing train, this vector represents minus this number of occurrences. The size of the vector b is polynomially bounded whenever $\overline{\kappa}$ is fixed. Each arrival or departure results in a set of potential disassemblies of the corresponding train. In turn, each potential disassembly results in a set of arcs, representing inventory changes corresponding with this potential disassembly.

Arcs are only present between two consecutive layers $\mathcal{L}_t$ and $\mathcal{L}_{t+1}$, for some $t \in \{0, \ldots, |\mathcal{T}| - 1\}$. $\mathcal{B}_{k_{t+1}}$ is the set of potential disassemblies of train service $t+1$ and $b_h \in \mathcal{B}_{k_{t+1}}$ is one element of this set. Moreover, $(b_h[1], \ldots, b_h[k(|\mathcal{Y}|, \overline{\kappa})])$ is the vector corresponding with potential disassembly b_h. An arc from a node $i \in \mathcal{L}_t$ to a node $j \in \mathcal{L}_{t+1}$ is present if and only if the following holds:

$$\begin{aligned} &(i,j) \in \mathcal{A}, i \in \mathcal{L}_t, j \in \mathcal{L}_{t+1}, \text{for some } t \in \{0, \ldots, |\mathcal{T}| - 1\} \quad \Leftrightarrow \\ &\qquad \exists b_h \in \mathcal{B}_{k_{t+1}} : j^l = i^l + b_h[l], \forall l \in \{1, \ldots, k(|\mathcal{Y}|, \overline{\kappa})\} \end{aligned} \tag{3.16}$$

Exactly one potential disassembly b_{ij} exists in (3.16) for each arc $(i,j) \in \mathcal{A}$. Moreover, j^l represents the inventory of potential configuration l at node j in layer $\mathcal{L}_{t+1}$, and therefore directly after the arrival or departure of train $t+1$. In this manner, the numbers of configurations $b_h[l]$ in disassembly b_h, $l \in \{1, \ldots, k(|\mathcal{Y}|, \overline{\kappa})\}$, are accounted for in the inventories.

Now, the nodes of the network are defined as $\mathcal{N} = \bigcup_{t=0}^{|\mathcal{T}|} \mathcal{L}_t$. Moreover, the arcs are defined as $\mathcal{A} = \{(i,j) \mid (3.16) \text{ holds}\}$. This completes the definition of the network $\mathcal{G}$=($\mathcal{N}$, $\mathcal{A}$).

The number of nodes within a layer $\mathcal{L}_t$ is $\mathcal{O}(k(|\mathcal{Y}|, \overline{\kappa})^{p+1})$ for $t \in \{1, \ldots, |\mathcal{T}| - 1\}$, and is 1 for $t \in \{0, |\mathcal{T}|\}$. Since the number of layers is $|\mathcal{T}| + 1$, the number of nodes in the network $|\mathcal{N}|$ is polynomially bounded whenever $\overline{\kappa}$ and p are fixed. Moreover, since arcs only exist between two consecutive layers, the same holds for the number of arcs.

We introduce lengths $f_{i,j}$ on the arcs $(i, j) \in \mathcal{A}$ similar to the cost in the proof of Theorem 3.8. These lengths represent the number of parts that result from the disassembly of an arriving train. Formally, we define $f_{i,j}$ as follows:

$$f_{i,j} = \begin{cases} \sum_{l=0}^{k(|\mathcal{Y}|,\overline{\kappa})} b_{ij}[l] & \text{if } j \in \mathcal{L}_t \text{ for some } t \in \mathcal{T}^+; \\ 0 & \text{otherwise.} \end{cases}$$

A solution to the shortest path problem in $\mathcal{G}$ with cost $f_{i,j}, \forall (i, j) \in \mathcal{A}$, corresponds to a matching with a minimum number of blocks in FSTMP(p,$\overline{\kappa}$). Therefore, FSTMP(p,$\overline{\kappa}$) can be solved in polynomial time and space by solving a shortest path problem in an appropriate network. ■

Note that the result of Theorem 3.8 is mainly of theoretical interest. Its practical value is limited due to the huge size of the involved network. As an example, we discuss the trains consisting of ICM train units of our main example from Section A.2. The timetable of these trains is given in Table 2.4. However, we repeat it here for readability reasons.

Train ID	Time	Event	Configuration	Direction
771	Tu 20:46	A	ICM_3 ICM_4 ICM_3 ICM_3	Amf
771	Tu 20:49	D	ICM_3 ICM_3	Gn
10771	Tu 20:52	D	ICM_4	Lw
584	Tu 23:18	A	ICM_3 ICM_3	Gn
516	We 6:18	D	ICM_3 ICM_3	Amf
721	We 7:46	A	ICM_3	Amf
10721	We 7:52	D	ICM_3 ICM_3	Lw

Table 3.3: The ICM train units from our example in Section A.2.

As mentioned in the discussion around Table 2.4, most of the matching has already been fixed in previous planning processes. However, these fixed parts are also accounted for in the network.

The network is depicted in Figure 3.6, where ICM_3 units are denoted with “A” and ICM_4 units with “B”. Moreover, one layer of nodes is represented by vertically aligned nodes. We only depict the nodes that are reachable in this specific instance. Moreover, the figure only contains the subset of arcs between two depicted nodes. The vectors of the nodes only depict the train configurations with strictly positive inventories, with

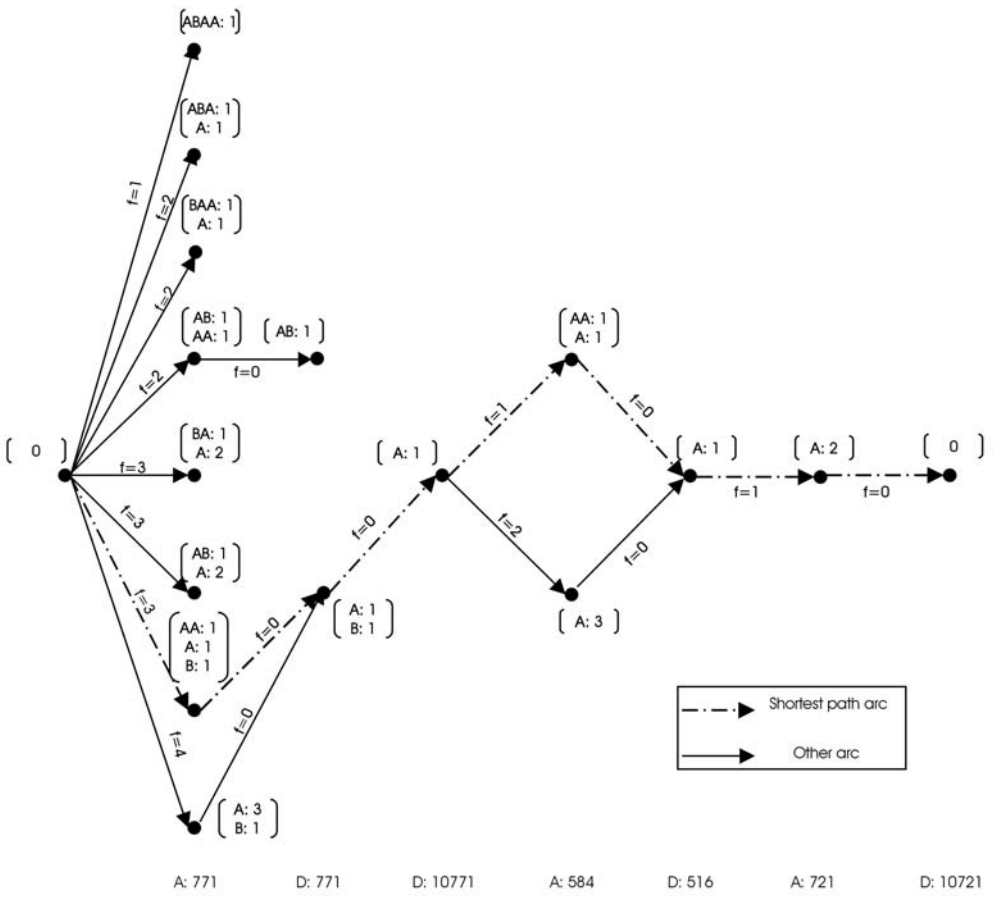


Figure 3.6: The network for the FSTMP for units of type 'A' and 'B' based on Table 3.3.

exceptions for the null vectors of the nodes in the first and last layer. Finally, with f we denote the cost of each arc.

The shortest path in this network can be found in a straightforward way. Its arcs have a different format in the figure. The cost of this path is 5, which corresponds to the minimum number of resulting blocks.

3.5 APPLICATIONS OF MATCHING

For solving the Integer Programs resulting from Model (3.a), we used the solver CPLEX 7.1 [ILOG CPLEX WEBSITE, 2004]. Unless stated otherwise, we used the default settings for all CPLEX parameters. All computational experiments reported in this thesis were performed on a computer with an Intel Pentium IV 2.8 GHz processor, 512MB of RAM, and operating under Windows XP Professional.

Our computational experiments represent real-life instances at stations Zwolle and Enschede, which were obtained from NSR. The instances are based on the Dutch railway system for 1999/2000, which is quite similar to the 2005/2006 railway system for these stations. Enschede is located in the eastern part of the Netherlands, see Figure A.1 on page 202.

For more detailed information on the instances, we refer to Appendix A. The sizes of Model (3.a) for these instances are reported in Table 3.4. The fact that many arriving train units have already been matched with departing train units, together with the strong preprocessing capabilities of CPLEX result in large reductions in numbers of variables and restrictions in the resulting Integer Programs.

Instance	ZT	ZS	ET	ES
Number of X variables	840	592	228	157
Number of Y variables	840	567	233	140
Number of Z variables	27,137	15,959	4,540	1,799
Total number of variables	28,817	17,118	5,001	2,096
Reduced number of variables	163	524	294	102
Number of restrictions (3.2)	306	276	81	72
Number of restrictions (3.3)	232	137	66	39
Number of restrictions (3.4)	304	270	79	66
Number of restrictions (3.5)	233	127	68	35
Number of restrictions (3.6)	840	592	228	157
Number of restrictions (3.7)	840	567	233	140
Total number of restrictions	2,756	1,970	756	510
Reduced number of restrictions	1,234	83	53	23

Table 3.4: Some characteristics of the TMP instances.

In the abbreviation of the instances in Table 3.4, the first character represents the station: 'Z' for Zwolle and 'E' for Enschede. The second character represents the planning period, with 'T' for an instance from Tuesday 08:00 until Wednesday 08:00, and 'S' for an instance from Saturday 08:00 until Sunday 08:00.

The instances that we used for our computational results differ from each other by the aspects in the objective function. We consider the following three objective functions:

A. Minimization of the number of blocks. Here, we only use a penalty q in the objective (3.1) and we set $m_{a,d}$ equal to 0 for all parts $a \in \mathcal{A}^+$ and $d \in \mathcal{A}_a^-$.

B. Minimization of the number of blocks and minimization of the deviation of the average length of stay. We model this by setting m_{ad} to a penalty times the square of the time difference between the arrival of part $a \in \mathcal{A}^+$ and the departure of part $d \in \mathcal{A}_a^-$. This penalty is added to the standard penalty of item A.

C. Minimization of the number of blocks and maximization of the deviation of the average length of stay. Maximizing the deviation of the length of stay prefers blocks that stay at the shunt yard either for a short period or for a long one. Therefore, we set m_{ad} to a penalty which is imposed if the difference in length of stay between parts $a \in \mathcal{A}^+$ and $d \in \mathcal{A}_a^-$ lies between 2 and 10 hours. This penalty is again added to the penalty of A.

In the remainder of this paragraph, the names from Table 3.4 are extended with a third character. This character represents the objective in this instance as mentioned above. We start with the discussion of the results for station Zwolle in Table 3.5. The first thing to note is that the computation times were very low: all instances are solved within one second. These computation times are reported for indicative purposes only. Moreover, the instances clearly demonstrate the very large fixed parts of the matching.

Instance	ZTA	ZTB	ZTC	ZSA	ZSB	ZSC
Number of blocks that need parking	54	57	54	34	34	33
Number of blocks that do not need parking	50	47	50	16	16	17
Number of fixed blocks	297			275		
Total number of blocks	401	401	401	325	325	325
Computation time for TMP (in sec.)	0.578	0.422	0.390	0.266	0.250	0.250

Table 3.5: Computational results for the matching instances of station Zwolle.

We see that the number of blocks is the same for the different instances, which is caused by the emphasis in the objective on minimizing the number of blocks. For station Zwolle, objective C consistently results in a minimum number of blocks that need parking. However, the differences with the other objectives are small. Although TMP in its most general form is $\mathcal{NP}$-hard, practical instances can be solved in extremely low computation times. The main reasons for these low computation times are the fact that many arriving units are already matched to departing units in previous planning processes and the efficient formulation combined with the power of CPLEX. Moreover, another important reason is that trains typically consist of no more than 3 units in practice, in which case the problem belongs to $\mathcal{P}$ (see Theorem 3.8). Computational tests showed that the instances can still be solved within 10 seconds when the fixed matchings are released. Note that the planner has the option to add or remove fixed matchings by himself. In this way, the planner has control over the algorithm.

Table 3.6 reports the results of the TMP instances for station Enschede for TMP. The table shows that these instances are significantly smaller than the instances for station Zwolle, which resulted in even smaller computation times. For station Enschede, we see that the different objectives do not result in different numbers of blocks that need parking. The matching for the Saturday / Sunday scenario does not differ for the reported objectives, and therefore these instances will result in the same input for subsequent problems.

Instance	ETA	ETB	ETC	ESA	ESB	ESC
Number of blocks that need parking	18	18	18	11	11	11
Number of blocks that do not need parking	18	18	18	3	3	3
Number of fixed blocks	75			65		
Total number of blocks	111	111	111	79	79	79
Computation time for TMP (in sec.)	0.094	0.093	0.078	0.047	0.047	0.047

Table 3.6: Computational results for the matching instances of station Enschede.

The precise effects of the different objectives for both stations Zwolle and Enschede will be discussed in subsequent chapters, where these blocks will serve as input for the other planning subproblems.

3.6 CONCLUSIONS

In this chapter, we have made both theoretical and practical contributions to the research field of shunting passenger train units.

Important restrictions for the Train Matching Problem (TMP) are the fact that type mismatches between prescribed train configurations and the provided ones are not allowed. Moreover, sufficient time between arrival and departure must be available. The quality of a matching of arriving units to departing units is measured by the number of blocks, the number of blocks that need parking because the time difference between arrival and departure is too large. Moreover, in Chapter 4 we will consider whether matchings with a LIFO or FIFO character are beneficial for the parking subproblem.

The theoretical contribution consists of the formulation of the model for TMP in Model 3.a, integrating shortest path problems and a matching problem. In addition, we studied the computational complexity of TMP. We showed that TMP in its most general form is $\mathcal{NP}$-hard by a reduction from the 3 Partition Problem. However, we considered two special cases of TMP for which polynomial time algorithms exist.

When each train consists of at most 3 units, the first departure takes place sufficiently later than the last arrival, and the objective is to minimize the number of blocks, the problem can be solved by an uncapacitated minimum cost flow in an appropriate network and with appropriate cost for each arc. This special case of TMP is called 3TMP.

The second special case occurs when the maximum number of train units in a train is fixed, the maximum number of train units at the yard is also fixed, and the objective is to minimize the number of blocks. The problem is denoted by FSTMP(n, m) with n the number of positions for parking train units at the shunt yard, and m the maximum number of train units in a train. This problem can be solved as a shortest path problem in a dedicated network by introducing certain arc costs.

We tested our model for TMP by running several instances for stations Zwolle and Enschede. The computation times for solving the mathematical models of these instances are all within one second. Moreover, we argued that by changing the objective function and adding fixed matchings, the planner can influence the solution. In subsequent chapters, we will describe mathematical models and algorithms for other subproblems. Then, we will be able to measure the effect of the different objectives used for TMP, as reported in this chapter.

Chapter 4

Parking of Train Units

Given a matching of arriving units to departing units, one needs to determine where to park the resulting blocks that stay sufficiently long at the shunt yard. This parking of rolling stock is subject to a number of restrictions. In general, the parking subproblem is the most difficult subproblem of the shunting problem to be solved. It is the subject of this chapter.

This chapter is structured as follows: after an in-depth introduction to the parking subproblem, related research topics like sorting permutations and container ship stowage are reviewed. Based on the acquired insights, a model for the parking problem is developed and we study the computational properties of it. The subproblem is modeled as a generalized Set Partitioning Problem (SPP). Because of the huge number of binary decision variables involved, a column generation heuristic has been developed for solving this model.

Based on the matchings computed in the previous chapter, we solve the parking subproblem for some instances at stations Zwolle and Enschede. It appears that the proposed solution approach is able to find high-quality solutions for these instances in a relatively fast way.

The chapter also discusses several extensions of the proposed algorithm for efficient re-optimization and generating 'good' initial sets of columns.

4.1 INTRODUCTION TO PARKING

In this chapter we use the terminology as defined in Section 3.1 and repeated in Appendix A.1. Examples of this terminology are shunt units, which are units that need to be parked at a shunt yard, and blocks, which are sets of train units from the same arriving train that will leave in the same departing train. Before describing the parking subproblem in more detail, we start with the introduction of the term crossing. A crossing occurs whenever a certain train unit obstructs another train unit during its departure or arrival. This term was first coined by GALLO AND DI MIELE [2001] in the context of parking buses. We will illustrate the concept of a crossing with an example. Table 4.1 repeats characteristics of selected train services resulting from the timetable of the main example in Table A.5 on page 207.

Train ID	Platform	Time	Event	Configuration	Direction
3672	7a/b	Tu 22:09	A	IRM_4 IRM_3	Rsd (OW)
3687	7a/b	Tu 22:23	D	IRM_4	Rsd (WO)
584	1a	Tu 23:18	A	ICM_3 ICM_3	Gn (BA)
3680	5b	We 0:09	A	IRM_4	Rsd (OW)
3623	5a	We 5:50	D	IRM_3 IRM_4	Rsd (WO)
516	1a	We 6:18	D	ICM_3 ICM_3	Amf (HA)

Table 4.1: Relevant train services for a potential crossing.

The arriving shunt units in this table are the IRM_3 unit of train 3672, the two ICM_3 units of train 584, and the IRM_4 unit of train 3680. The other arriving units are units of through train services, which do not require parking. Table 4.2 reports the set of blocks that need parking.

Arrival			Departure			Configuration
Train ID	Platform	Time	Train ID	Platform	Time	
3672	7a/b	Tu 22:09	3623	5a	We 5:50	IRM_3
584	1a	Tu 23:18	516	1a	We 6:18	ICM_3 ICM_3

Table 4.2: The blocks that need parking resulting from Table 4.1.

Suppose that the IRM_3 unit and the ICM_3 units are parked at the same LIFO track and the IRM_3 unit is parked there before the ICM_3 units. We know from Table 4.1 that the ICM_3 units will leave this track after the IRM_3 unit. Therefore, the situation on a track after parking the ICM_3 units is depicted in Figure 4.1. Since the IRM_3 unit needs to depart from the track before the ICM_3 units, this results in a crossing. Such crossings result in substantial additional resource consumption and are therefore infeasible. To resolve this crossing, one could park at least one block at a different track, or one could wait with routing the IRM_3 unit until the ICM_3 units have arrived at the track.

In addition to a crossing, we will also use the notion of a track assignment or assignment as an assignment of blocks to a specific shunt track using prescribed sides for arrival

Figure 4.1: A crossing at a LIFO track when the IRM_3 unit needs to depart first.

and departure of each block. Such an assignment is feasible whenever the following three criteria are met:

1. The assignment does not result in crossings.
2. The length of all the units parked at the shunt track exceeds the length of the shunt track at no point in time.
3. All the train units in the assignment are allowed to park at the shunt track.

In Section 1.10, we restricted ourselves to problems consisting of one station at a time with a 24-hour planning period and a given timetable. In this chapter, the following additional assumptions are made:

- No precedence relationships between shunt tracks exist. This implies that access to a shunt track is independent of the train units parked at a different shunt track, see Section 2.3.2 for an example of such a precedence relationship.
- The length of the train units parked at the track compared to the length of the track is sufficient for checking the capacity of a shunt track. This might be a problem for free tracks and will be discussed below.

We illustrate a problem with considering the length of the units at a free track with an example where the shunt units differ from our main example. The instance of the example consists of the blocks from Table 4.3 and one free track, namely shunt track 17 from Figure A.3 with length 275 meters (see Table A.4).

Arrival		Departure		Configuration	Length (in meters)
Platform	Time	Platform	Time		
3a/b	Tu 20:46	3a	We 7:52	ICM_3	81
7a/b	Tu 22:09	5a	We 5:50	IRM_3	82
7a/b	We 6:09	5a	We 7:50	IRM_3 IRM_3	164

Table 4.3: An example of a complication at a free track.

Suppose that the block with the IRM_3 unit is parked at the A-side of track 17 after the parking of the block with the ICM_3 unit. Moreover, the block with two IRM_3 units needs to be parked at the B-side of the track. Thus from 20:46 on Tuesday evening until 22:09 one ICM_3 unit is parked at track 17. From Tuesday 22:09 until Wednesday 5:50, an ICM_3 unit and an IRM_3 unit are parked. Between 5:50 and 6:09, only the ICM_3 unit is located at track 17. Between 6:09 and 7:50 the ICM_3 unit is joined by two IRM_3 units. In case the ICM_3 unit is not repositioned on track 17, the track needs to be at least 327 meters to park all blocks at it. Indeed, consider the situation at Wednesday 5:00 depicted in Figure 4.2. If the ICM_3 unit would not be repositioned, then the remaining length of the shunt track on the B-side of the ICM_3 unit at track 17 is 112 meters. This is insufficient for parking the two IRM_3 units, which have a total length of 164 meters. Although such repositionings can occur in theory, they are rare in practical situations.

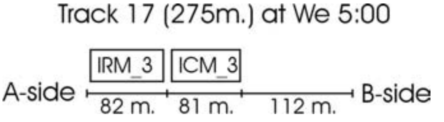


Figure 4.2: An example of a complication at a free track.

In order to provide maximum support of shunt planners, the most important objective is to park as many blocks as possible at the shunt tracks. Estimates of the route cost for each block to a certain shunt track are also incorporated. Note that the route cost might differ for different sides of a free track, as mentioned in Section 2.3.3. Preferences of planners for shunt tracks can also be taken into account. A planner has the additional option to prefer solutions where a shunt track has blocks with the same train configuration parked at it. This adds to the robustness of a solution. Suppose the operations are disrupted and only blocks with a specific configuration arrive at a specific shunt track. In this case, whenever a block with such a configuration arrives at the station, one does not need to consider the order of the blocks at this track. Finally, a planner may park blocks leaving in the same departing train in the right order at the same shunt track. This way, the blocks might be combined at the shunt track and require less resources, such as routing capacity and crews, in order for the blocks to leave the station.

The criteria for the feasibility of an assignment and the aspects of the objective function have also been discussed in Section 2.3.2 on page 30. Given the relevant terms and aspects of the objective, we are now ready to define the problem of parking train units at shunt tracks.

Definition 4.1. Given a set of blocks of shunt units, a set of shunt tracks, and estimates of the route cost for each block to and from each shunt track, the Track Assignment Problem (TAP) is to assign blocks to shunt tracks in a feasible manner, thereby minimizing the cost and maximizing the robustness of the solution.

Note that in this definition, we explicitly use the set of blocks as input, i.e. we assume that arriving shunt units have already been matched with departing shunt units.

4.2 PARKING AND RELATED PROBLEMS

In this paragraph, we discuss some literature on problems related to TAP. Topics include similar problems in a different railway setting, problems regarding the sorting of permutations, and a similar problem in maritime logistics.

An initial study for a tactical capacity check on parking space at shunt yards is discussed in DUINKERKEN [2003]. Based on the line plan and the corresponding frequencies of train services, DUINKERKEN derives the number of train units starting at each station for each type of rolling stock. For a specific station, these units need to be parked at the shunt yard of the station. The gross capacity of the shunt yard is typically measured as the number of meters of track available at the yard. However, several aspects need to be taken into account when determining the net capacity, such as the fraction of shunt tracks with catenary, the fraction of free tracks, the length of the shunt tracks and characteristics of the types of rolling stock to be parked. This net capacity is compared to the units that need to be parked at a station, indicating potential bottlenecks.

LÜBBECKE AND ZIMMERMANN [2005] discuss a problem related to TAP that arises at an in-plant private freight railroad. In this problem, one assigns transportation requests to certain regions of the in-plant railroad. Subsequently, one selects rail cars of specific types from a shunt track in this region for servicing a specific request. These rail cars have already been parked at shunt tracks and are selected in such a way that the number of crossings is minimized. Compared to the shunting problem studied in this thesis, the authors only discuss LIFO tracks and assume that there is no prescribed order of different types of rail cars in a train. In addition, it is assumed that there are no limitations for the temporary parking of rail cars, when these are not servicing a request.

KNUTH [1997] examined whether or not it is possible to obtain a permutation $p_1, \ldots, p_n$ from the trivial permutation $\{1, \ldots, n\}$ using a stack. He showed that this stack sorting problem has a feasible solution if and only if there are no indices $i < j < k$ such that $p_j < p_k < p_i$; a restriction independent of n. Consider an instance of TAP where each number represents a different type of rolling stock and the shunt yard consists of one LIFO track, all arrivals take place at one platform, and all departures at another one. Moreover, the last arrival of a block is before the first departure of a block, and the train units can be parked at the departure platform before the actual departure. Then, this restricted instance of TAP equals the described stack sorting problem. BÓNA [2003] gives an overview of related sorting problems for stacks and systems of stacks in parallel or in series. At a general deque, insertions and deletions can take place via both sides of the deque. PRATT [1973] considers necessary and sufficient conditions for permutations of any length to be sortable by such a deque. These conditions boil down to avoiding a set of subsequences in the requested permutations. The number of such subsequences grows as the size of the requested permutation grows. In case input or output is restricted to one side of the deque, the corresponding sets of subsequences to be avoided contain two elements each [KNUTH, 1997].

Finally, another stack sorting problem arises in maritime logistics: the Container Ship Stowage Problem. Here, a container ship visits several ports, and at each port a number of containers has to be loaded and unloaded. Every time a container has to be unloaded, all containers in the same stack on top of it have to be unloaded as well. This might result in unloading containers at a port different from its destination port, which is costly and therefore undesirable. Such unloading of containers at intermediate ports results in `shifts`. These shifts are related to the concept of crossings in train shunting. The problem formulation and a first solution approach can be found in AVRIEL AND PENN [1993], who consider container ships with one bay consisting of a fixed number of rows and columns and containers of the same standard size. AVRIEL ET AL. [2000] show that finding a stowage plan with a minimum number of shifts is $\mathcal{NP}$-hard. In addition, AVRIEL ET AL. [1998] propose a heuristic for solving this problem. This heuristic is able to solve real-life instances within 30 seconds. However, some practical restrictions are omitted, such as restrictions on the stability of the container ship. Such restrictions are incorporated in the genetic algorithm introduced by DUBROVSKY ET AL. [1999] at the cost of additional computation time and a slight increase in the number of shifts.

Some literature can be found on problems similar to the integration of TAP, studied in this chapter, and TMP, which was the subject of the previous chapter. In Section 7.3, we will discuss this literature in more detail.

4.3 PARKING MODEL FOR TAP

There are several ways to model TAP. In this paragraph, we formulate TAP as a Set Partitioning Problem (SPP) with side restrictions. The goal is to partition the blocks in feasible track assignments, minimizing some cost function. The cost estimate the required amount of resources and the robustness of such a set of track assignments. For completeness, we repeat the well known SPP here, see e.g. LENSTRA AND RINNOOY KAN [1979].

Definition 4.2. Given a finite set $\mathcal{A}$ of elements and a finite family $\mathcal{P}$ of subsets $\mathcal{P}_1, \ldots, \mathcal{P}_m$, SPP is defined as the problem of finding a subfamily $\mathcal{P}'$ of $\mathcal{P}$ consisting of pairwise disjoint sets such that $\cup_{p \in \mathcal{P}'} p = \mathcal{A}$.

The problem is also known as the exact cover problem and is $\mathcal{NP}$-complete if each family $\mathcal{P}$ contains at least 3 elements, see e.g. Garey and Johnson [1979]. The corresponding optimization problem, where one minimizes the cost of the subfamily $\mathcal{P}'$, is therefore $\mathcal{NP}$-hard. After introducing $a_{i,p}$ as an indicator whether or not subset p contains element i, f_p as the cost of subset p, and X_p as a binary decision variable indicating whether or not subset p is in a solution, the formulation of the optimization version of SPP reads:

$$\text{minimize} \quad \sum_{p \in \mathcal{P}} f_p X_p \tag{4.1}$$

$$\text{subject to} \quad \sum_{p \in \mathcal{P}} a_{i,p} X_p = 1 \qquad \forall i \in \mathcal{P} \tag{4.2}$$

$$X_p \in \{0,1\} \quad \forall p \in \mathcal{P} \tag{4.3}$$

The set covering problem is a relaxation of SPP, where each element needs to be covered at least once, i.e. with '=' replaced by '≥' in (4.2). Obviously, each feasible solution to SPP is also a feasible solution to the set covering problem.

Before we state the model for TAP, we introduce some relevant notation:

- $\mathcal{S}$ is the set of shunt tracks.
- $\mathcal{B}$ is the set of blocks that need to be parked.
- $\mathcal{V}_s$ is the set of potential assignments to track $s \in \mathcal{S}$.
- $\mathcal{V}_s^b$ is the set of potential assignments to track $s \in \mathcal{S}$ containing block $b \in \mathcal{B}$.

The decision variables in the parking model are:

$$X_a^s = \begin{cases} 1 & \text{if assignment } a \in \mathcal{V}_s \text{ is used for shunt track } s \in \mathcal{S}; \\ 0 & \text{otherwise.} \end{cases}$$

$$N_b = \begin{cases} 1 & \text{if block } b \in \mathcal{B} \text{ is not parked at any shunt track } s \in \mathcal{S}; \\ 0 & \text{otherwise.} \end{cases}$$

The parameters f_a^s model the cost of assignment a on track s. In addition, the parameter d models a penalty if a block is not assigned to any track. Typically, d is significantly larger than any f_a^s. Now, TAP can be formulated as follows:

$$\text{minimize} \quad \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{V}_s} f_a^s X_a^s + d \sum_{b \in \mathcal{B}} N_b \tag{4.4}$$

$$\text{subject to} \quad \sum_{s \in S} \sum_{a \in \mathcal{V}_s^b} X_a^s + N_b = 1 \qquad \forall b \in \mathcal{B} \tag{4.5}$$

$$\sum_{a \in \mathcal{V}_s} X_a^s \leq 1 \qquad \forall s \in \mathcal{S} \tag{4.6}$$

$$X_a^s \in \{0,1\} \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{V}_s \tag{4.7}$$

$$N_b \in \{0,1\} \quad \forall b \in \mathcal{B} \tag{4.8}$$

The goal is to minimize the total cost of parking shunt units, such that as many blocks as possible are assigned to the shunt tracks. Restrictions (4.5) state that each block is covered by exactly one assignment for one shunt track or it is not parked at all. Restrictions (4.6) describe that each shunt track can have at most one assignment. This model is called Model (4.a).

The structure of TAP allows for solving it as a set covering problem as well, since overcovers can be removed easily. However, our computational tests showed that computation times did not benefit from this relaxation.

An alternative model, integrating the matching and parking subproblems, has also been developed. This model is based on the order of arriving and departing train units at the station. It is capable of solving real-life instances in a reasonable amount of computation time. This alternative model is the subject of Chapter 7 of this thesis and has been developed by Schrijver [2003].

A major advantage of Model (4.a) is that difficult restrictions with respect to the feasibility of an assignment are taken into account implicitly. A major disadvantage is the fact that the number of potential track assignments may be exponential in the number of blocks. We return to this issue in Section 4.6 Moreover, this formulation assumes that assignments for different tracks are independent of each other, while in practice dependencies might occur.

4.4 COMPLEXITY OF PARKING

This paragraph investigates the computational complexity of several versions of TAP. We prove that TAP is $\mathcal{NP}$-hard and state that a special case of TAP can be solved in polynomial time and space.

We prove that TAP is $\mathcal{NP}$-hard by a reduction of a special case of the decision version of TAP from the Bin Packing Problem (BPP). BPP is $\mathcal{NP}$-complete in the strong sense [Garey and Johnson, 1979] and reads:

Definition 4.3. Given is a finite set $\mathcal{A}$ of items, a positive integer bin capacity s, a positive integer k, and a "size" $\sigma_a \in \mathbb{Z}^+$ for each $a \in \mathcal{A}$. Then the question of BPP is whether a partitioning of $\mathcal{A}$ into disjoint subsets $\mathcal{A}_1, \ldots, \mathcal{A}_k$ exists, such that the sum of the sizes of the items in each $\mathcal{A}_i$ does not exceed s.

In addition, we define TAP-0 as follows:

Definition 4.4. Given a set of blocks of shunt units, and a set of shunt tracks, the question of TAP-0 is whether all blocks can be assigned to shunt tracks in a feasible manner.

Theorem 4.5. *TAP-0 is $\mathcal{NP}$-complete.*

Proof. For a given assignment of blocks to shunt tracks, one can check in polynomial time whether the track assignments contain crossings, whether the total length of the blocks parked at a shunt track exceeds the length of this track at some point in time, and whether the blocks in an assignment are allowed to park at the corresponding track. Therefore, TAP-0 is in $\mathcal{NP}$.

Given an instance I of BPP, we construct an instance I' of TAP-0 in the following manner. The length of a block is defined as the total length of the train units in the configuration of the block. Moreover, all blocks are allowed to park at all shunt tracks. There are $|\mathcal{A}|$ blocks, each block $a \in \mathcal{A}$ having length σ_a, arrival time $t-a$ and departure time $t+a$, with t some point in time between the last arrival and the first departure. Finally, k shunt tracks are available for parking, each with length s.

Suppose I is a yes-instance of BPP. Then, for each $\mathcal{A}_i$, all blocks $a \in \mathcal{A}_i$ are parked at shunt track i. Since these blocks can be assigned at a LIFO track without crossings due to the chosen arrival and departure times, it is clear that this is a feasible assignment of blocks to shunt tracks in I'.

Conversely, suppose that I' is a yes-instance of TAP-0. Now, the blocks in $\mathcal{A}_i$ are assigned to shunt track i and the total length of the blocks in $\mathcal{A}_i$ does not exceed s. Therefore, the sets $\mathcal{A}_1, \ldots, \mathcal{A}_k$ form a feasible solution to BPP.

Concluding, I is a yes-instance of BPP if and only if I' is a yes-instance of TAP-0. Combined with the fact that BPP is $\mathcal{NP}$-complete and the earlier observation that TAP-0 is in $\mathcal{NP}$, this completes the proof. ■

Note that in the proof of Lemma 4.5 we assign arrival and departure times to the blocks in such a way that all blocks can be parked at a LIFO track without introducing one or more crossings.

Corollary 4.6. *TAP is $\mathcal{NP}$-hard.*

Proof. This proof follows directly from Lemma 4.5 and the fact that parking as many blocks as possible is a special case of the objective of TAP. ■

Now, we turn to a special case of TAP which is polynomially solvable. This special case is defined as:

Definition 4.7. The Fixed Size Track Assignment Problem (FSTAP) is a special case of the TAP, where the objective is to maximize the number of blocks parked at the shunt yard, and instances are restricted to at most $\bar{b}$ blocks simultaneously at the yard and the layout of the shunt yard is fixed. Such a problem is denoted with FSTAP($\bar{b}$).

Similar to the proof of Theorem 3.11 in this thesis and of Theorem 5 in Kroon et al. [1997], one can prove the following lemma by solving a shortest path problem in an appropriate network.

Theorem 4.8. *For each $\bar{b} \in \mathbb{N}$, FSTAP($\bar{b}$) can be solved in polynomial time and space.*

Just as Theorems 3.8 and 3.11, this is a theoretical result with very limited practical value because of the huge size of the involved network.

4.5 INTRODUCING COLUMN GENERATION

Since our solution approach will be based on column generation, we continue with a short introduction to column generation. Column generation has become a standard solution approach for solving mathematical models with a huge number of variables (or columns). The general idea is to start with solving a problem with a small set of variables. Iteratively, this reduced problem is solved and additional columns are added to the problem. The reduced problem is called the (restricted) master problem and the problem of generating additional columns is called the pricing problem. The concept has been introduced by Dantzig and Wolfe [1960]. Later, Desrosiers et al. [1984] and Crainic and Rousseau [1987] initiated a renewed interest in column generation including computational experiments for a vehicle routing problem respectively a crew scheduling problem. Without any doubt, the current popularity is largely influenced by improved speed and computing power of (personal) computers.

In the remainder of this paragraph, some well known elementary concepts from linear programming theory are used. Chvátal [1983] provides an excellent discussion of this theory. Consider the following model containing a huge number of columns n:

$$(P) \text{ minimize } \sum_{j=1}^{n} f_j X_j \tag{4.9}$$

$$\text{subject to } \sum_{j=1}^{n} a_{i,j} X_j \geq b_i \quad i = 1, \ldots, m \tag{4.10}$$

$$X_j \geq 0 \quad j = 1, \ldots, n \tag{4.11}$$

The dual formulation of (P) reads:

$$(D) \text{ maximize } \sum_{i=1}^{m} b_i Y_i \tag{4.12}$$

$$\text{subject to } \sum_{i=1}^{m} a_{i,j} Y_i \leq f_j \quad j = 1, \ldots, n \tag{4.13}$$

$$Y_i \geq 0 \quad i = 1, \ldots, m \tag{4.14}$$

Given a dual solution $Y_1, \ldots, Y_m$, the reduced cost of a primal variable X_k is defined as:

$$\overline{f_k} = f_k - \sum_{i=1}^{m} a_{i,k} Y_i \tag{4.15}$$

The complementary slackness theorem is well known and provides the characterization of an optimal primal solution. We repeat it here without proof.

Theorem 4.9 (e.g. in Chvátal [1983]). *A feasible solution $X_1^*, \ldots, X_n^*$ of (4.9) - (4.11) is optimal if and only if there are numbers $Y_1^*, \ldots, Y_m^*$ such that*

$$\begin{aligned} \sum_{i=1}^{m} a_{i,j} Y_i^* &= f_j \quad \text{whenever } X_j^* > 0 \\ Y_i^* &= 0 \quad \text{whenever } \sum_{j=1}^{n} a_{i,j} X_j^* > b_i \end{aligned}$$

and such that

$$\begin{aligned} \sum_{i=1}^{m} a_{i,j} Y_i^* &\leq f_j \quad j = 1, \ldots, n \\ Y_i^* &\geq 0 \quad i = 1, \ldots, m \end{aligned}$$

In other words, Theorem 4.9 states that a primal feasible solution is optimal if and only if all non-basic columns have non-negative reduced cost.

Suppose the problem (P) has a huge set $\mathcal{N}$ of variables or columns. In a column generation approach, one would start with an initial set of columns $\mathcal{N}_0 \subset \mathcal{N}$ containing a feasible solution to (P). After solving the restricted problem (P), the optimal dual variables of this restricted problem are used in the pricing problem to find columns with negative reduced cost. If no such column exists, the optimal solution to the initial problem (P) has been found. Note that it is not necessary to explicitly know all columns. Typically, the pricing problem is an optimization problem as well, since enumerative approaches require too much computation time.

A practical introduction to column generation can be found in Chvátal [1983, Chapter 26]. Recent overviews of applications of column generation and relevant topics in applying column generation can be found in Desaulniers et al. [2005], Lübbecke and Desrosiers [2004], Wilhelm [2001], and Van den Akker et al. [2005]. Column generation algorithms provide many possibilities for tuning the algorithm, as is also indicated in the above references.

In case some of the decision variables are required to be integer, an LP problem generalizes to a Mixed Integer Problem (MIP). These problems are typically solved by `branch-and-bound` type of algorithms. Here, column generation can be applied at different nodes of the branch-and-bound tree, resulting in a solution approach called `branch-and-price`. For a more elaborate introduction to this technique, we refer to Barnhart et al. [1998] and Vanderbeck and Wolsey [1996].

Although column generation is typically used in combination with solving the LP-relaxation of a MIP, the combination of column generation with Lagrangian relaxation is gaining popularity. In Lagrangian relaxation, the violation of "difficult" restrictions is penalized in the objective function, see e.g. Nemhauser and Wolsey [1988] for an introduction.

The link between column generation and Lagrangian relaxation has been established by Brooks and Geoffrion [1966] and recent applications can be found in Freling [1997], Huisman [2004], and Huisman et al. [2005a] for integrated vehicle and crew scheduling and others.

4.6 TAP AND COLUMN GENERATION

In order to handle the exponential number of potential assignments X_a^s in TAP, we propose a column generation approach, where columns are generated in the root node of the branch-and-bound tree.

Regarding TAP, the master problem consists of selecting a set of track assignments according to Model (4.a). In the pricing problem, assignments for individual shunt tracks are generated implicitly and independently. New columns are generated based on dual information obtained from the master problem. Therefore, we introduce dual variables λ_b, $b \in \mathcal{B}$, and μ_s, $s \in \mathcal{S}$, for restrictions (4.5) and (4.6), respectively. The details of the pricing problem may depend on the specific characteristics of TAP. However, the general structure of the pricing problem does not change. Other notation that will be used in this paragraph is:

- $\mathcal{F}_s$ is the set of different approach types to and from shunt track $s \in \mathcal{S}$.
- $\mathcal{B}_k^s$ is the set of blocks in assignment $k \in \mathcal{V}_s$ parked at track $s \in \mathcal{S}$.

For a LIFO track s, $\mathcal{F}_s$ contains only one element, namely arriving from and departing to the open side of the track. For a free track s, $\mathcal{F}_s$ contains the following four elements:

- Arriving from the left and departing to the left (LL).
- Arriving from the left and departing to the right (LR).
- Arriving from the right and departing to the left (RL).
- Arriving from the right and departing to the right (RR).

First, we discuss the network representation for the pricing problem per track, and thereafter we give a dynamic programming algorithm for solving a resource constrained shortest path problem in this network. This algorithm results in a set of potential assignments for a shunt track with negative reduced cost, if such assignments exist. Note that the structure of the involved network depends on the nature of the shunt tracks. An example of this nature is whether or not the shun track has catenary installed. However, for notational convenience, we describe the structure of the network as if it is the same for each shunt track. We conclude this paragraph with a description of our procedure to create an integer solution to TAP based on the LP-relaxation of Model 4.a, which is solved by column generation.

4.6.1 Network Representation

We assume that the set of blocks $\mathcal{B}$ that need to be parked is ordered by non-decreasing arrival time. In the network, each block $b \in \mathcal{B}$ is represented by a layer $\mathcal{L}_b$ consisting of nodes n_f^b with $f \in \mathcal{F}_s$ and a node n_{not}^b corresponding to *not* parking block b at the track currently under consideration. The network consists of these layers, a source n_1^0 and a sink $n_1^{|\mathcal{B}|+1}$. To simplify the notation later on, we define the layers $\mathcal{L}_0$ and $\mathcal{L}_{|\mathcal{B}|+1}$ as the layers containing the nodes n_1^0 and $n_1^{|\mathcal{B}|+1}$, respectively. The arcs in this network are directed from the source to every node in the first layer, from every node in the last layer to the sink. In addition, arcs between intermediate layers are present when at least one of the nodes is a '*not* node'. Between intermediate layers, arcs also exist when parking the blocks corresponding to the layers of the nodes at the same track, using the implied sides of the track, is feasible. Parking two blocks at the same shunt track is feasible when it does not result in a crossing and the types of units of both blocks are allowed to park at this track. Now, we can define our network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ for the pricing problem as follows:

$$\mathcal{N} = \bigcup_{b=0}^{|\mathcal{B}|+1} \mathcal{L}_b \tag{4.16}$$

$$\begin{aligned} \mathcal{A} = \{(n_i^b, n_j^{b+1}) | b = 0, \ldots, |\mathcal{B}|, \\ i \in \mathcal{L}_b, j \in \mathcal{L}_{b+1}, \text{ and } (n_i^b, n_j^{b+1}) \text{ is feasible}\} \end{aligned} \tag{4.17}$$

A path in this network represents a feasible assignment of blocks to the track if the three restrictions mentioned in Section 4.1 are satisfied. In the next paragraph, we describe how we check these restrictions. Note that one cannot entirely take into account these restrictions during the construction of the network. For example, because the length of the shunt track might be exceeded at some point in time.

The set $\mathcal{B}$ resulting from the matching instance of Table A.5 on page 207 is given in Table 4.4. Note that this is a matching with a minimum number of blocks.

We illustrate the network $\mathcal{G}$ in Figure 4.3 with the example of Table 4.4 for a single free shunt track. In this figure, we assume that all types of rolling stock are allowed

Index	Arrival		Departure		Configuration	Length
	Train ID	Time	Train ID	Time		(in meters)
1	771	Tu 20:46	10712	We 7:52	ICM_3	81
2	3672	Tu 22:09	3623	We 5:50	IRM_3	82
3	7984	Tu 23:12	7917	We 7:21	DH_2	44
4	584	Tu 23:18	516	We 6:18	ICM_3 ICM_3	162

Table 4.4: A matching for the example from Table 2.3.

to park at this track. Suppose that this track is track 17 in Figure A.3 on page 203, which has a length of 275 meters. It is immediately clear that it is impossible to park all blocks in Table 4.4 at this track because it has insufficient length. A feasible path with a maximum number of blocks parked at track 17 is depicted with special formatted arcs in the figure. In this path, blocks 1 and 3 arrive and depart via the A-side of the track, block 2 is not parked at the track, and block 4 arrives and departs via the B-side of the track. Note that there are no arcs from e.g. the node 'LL' in layer 1 to the nodes 'LR' and 'RL' in layer 2, because these arcs would be infeasible.

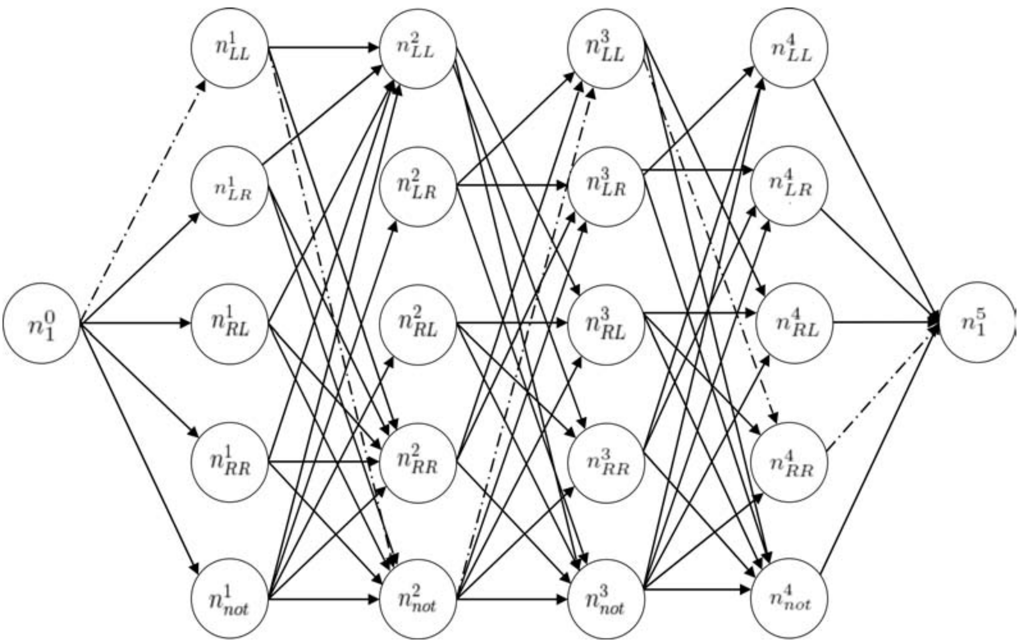


Figure 4.3: The network corresponding to the blocks in Table 4.4 for a free track.

Most costs are defined on the arcs of the network. The cost of an arc represents the cost of the node to which the arc is directed. In turn, the cost of a node is the cost of assigning a certain block to a certain shunt track, and arriving at and departing from this track via the specified sides. This cost mainly consists of estimated routing cost. Furthermore, the cost of the arcs directed to the sink node model the usage of the shunt track under consideration. For different arcs, these cost might differ to reflect preferences of planners for shunt tracks.

Moreover, a penalty for parking a block with a different train configuration than the previously parked block at this track, and a reward for parking two blocks leaving in the same departing train in the right order at the same shunt track. These penalties and rewards are dependent on the characteristics of a partial path, which might consist of several arcs.

The reduced cost of an arc (n_i^b, n_j^{b+1}), $0 \leq b < |\mathcal{B}|$, equals the cost of this arc minus the dual cost λ_{b+1} of block $b+1$ if $j \neq not$. If $j = not$ then the reduced cost of the arc equals the cost of the arc, which is 0. In addition, the reduced cost of an arc $(n_i^{|\mathcal{B}|}, n_1^{|\mathcal{B}|+1})$ is equal to its cost minus the dual cost μ_s of the corresponding shunt track $s \in \mathcal{S}$. Using the reduced cost of the arcs, the reduced cost of a path represents the reduced cost of the corresponding assignment a on shunt track s, which we denote by $\overline{f_a^s}$. Thus, the reduced cost of an assignment a to shunt track s in TAP is given by:

$$\overline{f_a^s} \; = \; f_a^s - \sum_{b \in \mathcal{B}_a^s} \lambda_b - \mu_s \quad \forall k \in \mathcal{V}_s, s \in \mathcal{S}. \tag{4.18}$$

Note that f_a^s contains the earlier mentioned penalties and rewards as well as the cost of the arcs in assignment a.

4.6.2 Dynamic Programming

In this paragraph, we propose a dynamic programming algorithm for solving the pricing problem, based on the network of the previous paragraph. To discuss the procedure, we need to introduce some notation first. The *time* of a node is defined as the arrival time of the block in the corresponding layer. Moreover, for a node u, $\mathcal{P}_u$ is defined as a set of feasible $(n_1^0 - u)$-paths in $\mathcal{G}$. Define f_p as the cost of path $p \in \mathcal{P}_u$ and f_p^v as the additional penalties for different train configurations and rewards for clustering of blocks for the same departing train incurred by extending path p with node v. Moreover, $\mathcal{B}_p^s$ is defined as the set of blocks at track s in path p. Next, l_p^s is defined as the remaining length of track s, i.e. the length of the track denoted by l^s minus the length of the blocks in $\mathcal{B}_p^s$, and r_p is defined as the earliest departure time of the blocks in path $p \in \mathcal{P}_i$, which did not leave yet at the time of node i. Furthermore, $f_{u,v}$ is defined as the cost of arc (u, v) (which equals 0 if v is a '*not* node'), and $\mathcal{R}^{uv}$ is defined as the set of blocks departing between the times of nodes u and v. Finally, we define for all nodes v, except for the '*not* nodes', l_v as the length and r^v as the departure time of the block corresponding to the layer of node v. If v is a '*not* node', then l_v equals 0.

For each node $u \in \mathcal{N}$, we try to extend each path $p \in \mathcal{P}_u$ with each potential arc $(u,v) \in \mathcal{A}$. If such an extension is feasible, then we obtain a new path p_v^{new} and we update the set of paths $\mathcal{P}_v$ with p_v^{new}.

The remaining question is thus whether such an extension is feasible. Therefore, we introduce several key values, called `resources` in Desrosiers et al. [1995], for checking the feasibility of the proposed extension.

In case of a LIFO track, these resources are the reduced cost of the path, the total length of the blocks currently at the track, and the earliest departure time of the blocks currently at the track. This earliest departure time is the departure time of the first block on the departure side of the track. Now, extending the path $p \in \mathcal{P}_u$ with the arc (u,v) is feasible if and only if v is a '*not* node' or

$$(l_v < l_p^s + \sum_{w \in \mathcal{R}^{uv}} l_w \text{ and } r^v < \min_{b \in \mathcal{P}_s^u \setminus \mathcal{R}^{uv}} r^b).$$

The first criterion is obvious: if v is a '*not* node', then nothing changes at the track. The second criterion states that the remaining track length should be nonnegative, i.e. the length of the blocks at the track should not exceed the track length, and a crossing is avoided. A crossing is avoided if the block corresponding to node v departs before the earliest departure time of the blocks which are already parked at the shunt track. Since the layers $\mathcal{L}_b$ are ordered increasingly on arrival time of blocks b, we know that the *time* of node v is not earlier than the time of node u.

If such an extension is feasible, then we update the resource variables for the path p_v^{new} as follows:

$$\begin{aligned}
f_{p_v^{new}} &= f_p + f_{u,v} + f_p^v, \\
l^s_{p_v^{new}} &= l_p^s - l_v + \textstyle\sum_{w \in \mathcal{R}^{uv}} l_w, \\
r^{p_v^{new}} &= \begin{cases} \min_{b \in \mathcal{B}^u \setminus \mathcal{R}^{uv}} r^b & \text{if } v \text{ is a '}\textit{not}\text{ node';} \\ r^v & \text{otherwise.} \end{cases}
\end{aligned}$$

In order to facilitate dominance checking, the paths $\mathcal{P}_u$ are kept in lexicographical order of the values of the resource variables. A path $p_i \in \mathcal{P}_u$ is dominated by a path $p_j \in \mathcal{P}_u$ if all the resource variables of p_i are dominated by those of p_j. For LIFO tracks, this implies:

$$f_{p_i} - \sum_{b \in \mathcal{B}^s_{p_i}} \lambda_b \geq f_{p_j} - \sum_{b \in \mathcal{B}^s_{p_j}} \lambda_b, \tag{4.19}$$

$$l^s_{p_i} \leq l^s_{p_j}, \tag{4.20}$$

$$r^{p_i} \leq r^{p_j}. \tag{4.21}$$

While updating the set of paths at a node, we purposely loop first over the paths and then over the arcs. Since a path is considered for several arcs, the removal of blocks that have departed and the corresponding updating of resource variables is done once for each path only, before extension of a path with an arc is considered. The list $\mathcal{P}_u$ can be updated in one loop: as long as the new path is lexicographically smaller than the paths in the list, we need to check if the new path is dominated. Otherwise, we can insert the new path in the list and loop through the remainder of the list, thereby removing paths that are dominated by the new path. The benefit of the lexicographic order is that we do not need to check the first resource variables, while looping through the remainder of the list.

In case of a free track, we register instead of the earliest departure time at one side, the earliest as well as the latest departure times at both sides of the track. Furthermore, we need to use in this case slightly more complicated feasibility checks and dominance rules.

Note that the penalties on blocks with different train configurations at the same track, and rewards for combining blocks leaving in the same departing train result in a heuristic feature of the pricing problem. This will be clarified with an example adapted from FIOOLE [2003] consisting of the blocks in Table 4.5.

Index	Arrival time	Departure time	Configuration	Length (in meters)
1	Tu 20:30	We 7:00	ICM_4	108
2	Tu 22:00	We 7:30	ICM_3	81
3	Tu 23:00	We 6:30	ICM_3	81
4	Tu 23:30	We 6:00	ICM_4	108

Table 4.5: Blocks illustrating a complication in the pricing problem.

The network of a LIFO track for these blocks is a simplified version of Figure 4.3 and is given in Figure 4.4, where n^b_{no} indicates that block b will not be parked at this track and n^b_{yes} is defined similarly.

Suppose a nonnegative penalty for parking different train configurations at one shunt track exists, the routing cost for block 1 are larger than the routing cost for block 2, and the dual variables for these blocks are equal. Moreover, suppose that all other costs are 0 and that the penalty for parking blocks with different configurations is higher than the difference in routing cost for blocks 1 and 2. Consider the two partial paths depicted in Figure 4.4, which are assumed to be feasible. With the information on the cost and the blocks, it follows from the dominance rules (4.19)-(4.21) that path 2 dominates path 1 at node n^3_{no}. Indeed, $f_1 \geq f_2$, $l^s_1 \leq l^s_2$, and $r^1 \leq r^2$. Therefore, path 1 is not considered anymore by the dynamic programming algorithm. However, the track assignment consisting of blocks 1 and 4 should be preferred over the track assignment consisting of blocks 2 and 4, assuming that both assignments are feasible. This is caused by the fact that the latter path results in a penalty for parking different configurations,

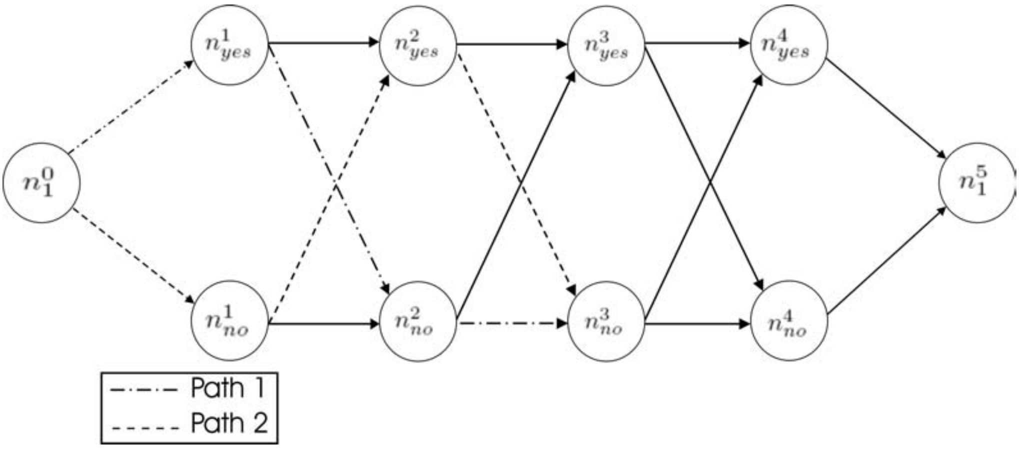


Figure 4.4: The network of a LIFO track for the blocks of Table 4.5.

while this is not the case for the former path. Unfortunately, this former path has been removed from the pricing problem at the node of block 3.

The concept of the '*not* node' can be used in many other applications using column generation, such as e.g. crew scheduling with fixed cost for each duty. The '*not* node' has two effects: it greatly reduces the number of arcs in the network, because arcs exist between consecutive layers only. Furthermore, it helps to concentrate path lists at a restricted number of nodes, because extending a feasible partial path with the '*not* node' of the next layer is always feasible. Because of the layered structure of the network and the usage of the '*not* nodes', the number of arcs $|\mathcal{A}|$ remains relatively small:

$$|\mathcal{A}| \;\leq\; |\mathcal{L}_b|^2(|\mathcal{B}|-1) + 2|\mathcal{L}_b| \tag{4.22}$$

Note that $|\mathcal{L}_b|$ is independent of b (5 for free tracks and 2 for LIFO tracks), and that $|\mathcal{A}|$ is linear in the number of nodes. Moreover, because of the concentration of paths on these '*not* nodes', dominance has more effect: more paths can be deleted than in a larger network without the '*not* nodes', where the 'smaller' lists with partial paths are divided over more nodes. A disadvantage might be that some additional computational effort is required. For example, in order to find the block previously assigned to the shunt track, one might need to traverse several '*not* nodes'.

4.6.3 Creating Integer Solutions

In case the optimal solution at the root node of the branch-and-bound tree is fractional, branching is required. In order to ensure optimality, one needs to generate columns at the different nodes of the branch-and-bound tree as well.

In our solution approach, we do not generate any columns at other nodes than this root node, thereby introducing a heuristic feature to our solution approach. The effect of this feature on the overall quality will be discussed in Section 4.7.

We create an integer solution, based on the LP-relaxation, by using CPLEX. We provide CPLEX with additional information to start branching down on the decision variables N_b, in order to find a solution with a maximum number of parked blocks as fast as possible. Moreover, whenever an integer solution is found during the column generation, it is preserved. Before starting CPLEX, we create an initial basic solution representing this integer solution.

4.7 APPLICATIONS OF PARKING

In this paragraph, we report some computational experiments for the algorithm described in the previous paragraph. At the start of Section 3.5, we introduced the hardware and software, which were used for these experiments. In this paragraph, we use the matchings of Section 3.5 as input for TAP. The instances ESA, ESB, and ESC resulted in exactly the same matching, see Section 3.5. Therefore, these three instances are summarized as ES. in the relevant tables of this paragraph.

Station Zwolle contains 19 shunt tracks, varying in length approximately between 110 to 400 meters. Out of these 19 tracks, 12 tracks are free tracks, and the remaining 7 tracks are LIFO tracks. Station Enschede contains 13 tracks, ranging in length between 55 and 650 meters. At station Enschede 11 tracks are free tracks, and 2 tracks are LIFO tracks.

We ensure that each assignment has an integer objective coefficient. Moreover, in the computational results, we report the objective of the integer solution and the objective of its LP-relaxation. This objective is especially useful for comparing different instances on the same day and the same station and the quality of the integer solution.

Table 4.6 gives the results based on the different matching instances at station Zwolle, where we focus on routing cost and parking as many blocks as possible at the yard. In these instance, the configurations of the shunt tracks correspond to the actual practical configurations, i.e. tracks that can be approached from both sides are modeled as free tracks. In order to reflect this, we append the character 'D' to the name of the instances.

We see that the instances can be solved in a satisfactory computation time, with sufficiently small gaps. Row 7 of the table shows that the algorithm is able to park all units at the shunt tracks. Moreover, we see that objective C of the matching consistently has the best results. This was already expected in the previous chapter because this objective prefers matchings "compatible" with LIFO tracks. Fodor Birtalan [2003] reaches a similar conclusion for a more precise approach to prefer such LIFO assignments. Finally, only a moderate number of columns is required in the restricted master problem.

Table 4.7 shows that the instances for station Enschede are much smaller, resulting in

Instance	ZTAD	ZTBD	ZTCD	ZSAD	ZSBD	ZSCD
Number of blocks to be parked	54	57	54	34	34	33
Comp. time for TAP (in sec.)	245.5	224.1	420.4	3.3	3.5	5.2
LP solution value	5198.40	5641.61	5053.00	5088.42	5085.67	4076.00
IP solution value	5313	5738	5155	5203	5088	4076
Gap	2.16%	1.59%	1.98%	2.15%	0.05%	0.00%
# Blocks not parked	0	0	0	0	0	0
# Columns generated	5397	5502	5157	4298	4733	4533
# Iterations column generation	38	36	35	22	30	25

Table 4.6: Computational results at station Zwolle with free tracks.

integer LP-relaxations for all but one instance. These instances can be solved easily by our algorithm. For the Tuesday / Wednesday instances, it is again clear that matching objective C performs best. Note that all Saturday / Sunday instances are based on the same matching, resulting in the same solutions to TAP.

For the next instances we consider different the track configurations: we restrict each free track to have arrivals and departures from one side only. The open side of such a track is the side mostly used in practice, which has been determined after consulting shunt planners. For these instances, we append 'R' to the names of the instances.

From Table 4.8, we conclude that the Tuesday / Wednesday instances are equally difficult to solve in terms of computation times. Moreover, we see that the objective function values are significantly higher than the ones reported in Table 4.6. This leads to the conclusion that restricting the free tracks to LIFO tracks considerably deteriorates the objective. Therefore, our additional effort to model free tracks pays off.

Instance	ETAD	ETBD	ETCD	ES.D
Number of blocks to be parked	18	18	18	11
Comp. time for TAP (in sec.)	1.2	1.1	1.2	0.2
LP solution value	5540.00	5573.00	5538.00	1982.00
IP solution value	5541	5573	5538	1982
Gap	0.02%	0.00%	0.00%	0.00%
# Blocks not parked	0	0	0	0
# Columns generated	1147	1043	1085	158
# Iterations column generation	16	11	14	6

Table 4.7: Computational results at station Enschede with free tracks.

Instance	ZTAR	ZTBR	ZTCR	ZSAR	ZSBR	ZSCR
Number of blocks to be parked	54	57	54	34	34	33
Comp. time for TAP (in sec.)	188.1	347.8	271.0	3.3	3.7	4.1
LP solution value	6567.00	7287.67	6765.96	5555.50	5564.00	5161.00
IP solution value	6673	7577	7011	5573	5564	5161
Gap	1.26%	2.80%	2.75%	0.31%	0.00%	0.00%
# Blocks not parked	0	0	0	0	0	0
# Columns generated	5595	5713	4695	4180	4826	4067
# Iterations column generation	35	41	31	29	35	22

Table 4.8: Computational results at station Zwolle with LIFO tracks.

For station Enschede, the differences between free tracks and LIFO tracks are smaller, see Table 4.9. Thus, although station Enschede has a larger fraction of free tracks compared to station Zwolle, the impact of restricting free tracks to LIFO tracks is less for the Enschede instances than for the Zwolle instances. This is caused by the fact that the solutions to the Enschede instances in Table 4.7 mostly use the preferred sides of the free tracks already. Again, the algorithm results in high quality solutions in very low computation times for these instances.

Tables 4.10 and 4.11 report computational experiments, where the objective function has been extended with penalties for parking blocks with different train configurations at the same track and with rewards for parking blocks leaving the station in the same departing train in the right order at the same shunt track. These instances are based on the instances reported in Tables 4.6 and 4.7 and the last character of their names is replaced with 'O'. Since the objective functions have changed, these cannot be compared

Instance	ETAR	ETBR	ETCR	ES.R
Number of blocks to be parked	18	18	18	11
Comp. time for TAP (in sec.)	1.1	1.1	1.1	0.2
LP solution value	5642.00	6465.00	5642.00	1982.00
IP solution value	5714	6465	5713	1982
Gap	1.26%	0.00%	1.24%	0.00%
# Blocks not parked	0	0	0	0
# Columns generated	854	840	893	151
# Iterations column generation	10	10	10	6

Table 4.9: Computational results at station Enschede with LIFO tracks.

Instance	ZTAO	ZTBO	ZTCO	ZSAO	ZSBO	ZSCO
Number of blocks to be parked	54	57	54	34	34	33
Comp. time for TAP (in sec.)	221.7	305.1	352.4	2.9	3.3	3.2
LP solution value	2842.00	2301.01	2477.00	5369.25	5327.83	4944.00
IP solution value	2842	2381	2477	5444	5470	4944
Gap	0.00%	2.99%	0.00%	1.37%	2.60%	0.00%
# Blocks not parked	0	0	0	0	0	0
# Columns generated	6214	6965	5749	4870	4755	3567
# Iterations column generation	36	40	32	25	29	17

Table 4.10: Computational results at station Zwolle with extended objective.

in absolute values. As mentioned before, no guarantee exists that the pricing problem is solved to optimality. Therefore, the reported LP-relaxations in Tables 4.10 and 4.11 are not necessarily lower bounds.

The computational results for these instances with extended objectives again indicate that the algorithm is able to find good solutions in short computation times. However, it is at least as interesting to investigate the effect of this extended objective on the number of train configuration changes at a track, and the number of combinations of blocks leaving in the same departing train. These results are depicted in Figure 4.5. The columns use the left vertical axis, while the lines use the right one.

For all instances, we see significant decreases in the number of configuration changes. Moreover, the Tuesday / Wednesday instances at station Zwolle, and to a lesser extent at station Enschede, benefit from the reward for parking blocks at the same track in the right order, whenever these blocks leave in the same departing train. This reward has

Instance	ETAO	ETBO	ETCO	ES.O
Number of blocks to be parked	18	18	18	11
Comp. time for TAP (in sec.)	1.2	1.2	1.2	0.2
LP solution value	4404.50	4609.67	4516.00	3769.00
IP solution value	4562	4756	4516	3769
Gap	3.20%	1.77%	0.00%	0.00%
# Blocks not parked	0	0	0	0
# Columns generated	1120	835	921	269
# Iterations column generation	12	11	11	7

Table 4.11: Computational results at station Enschede with extended objective.

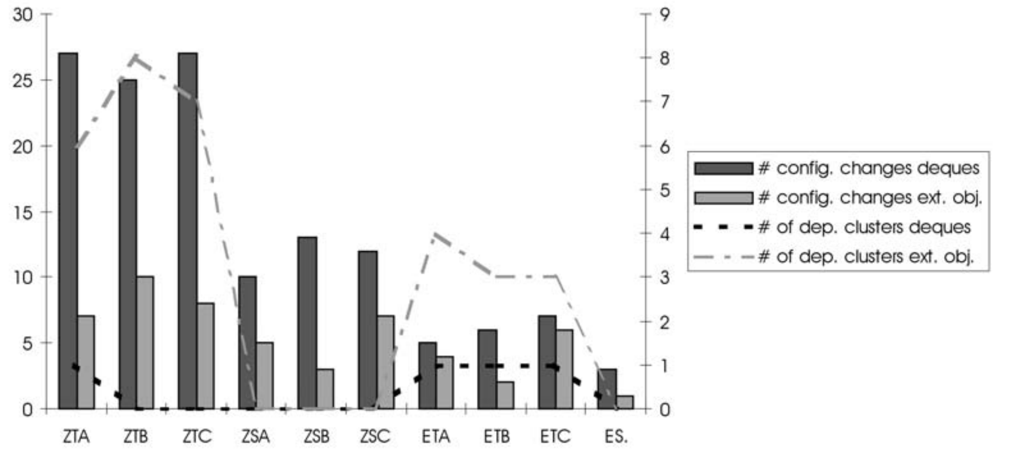


Figure 4.5: Effects of the extended parking objective for several instances.

no effect on the instances for Saturday / Sunday, since no such blocks exist.

The Tuesday / Wednesday instances for station Zwolle are all solved within 10 minutes, while the other instances require at most 10 seconds. Moreover, the algorithms are always able to park all units at shunt tracks. Finally, all gaps are within 4% and integer LP-relaxations are found quite frequently.

4.8 EXTENSIONS AND SPIN-OFFS

Fioole [2003] discusses several extensions of the parking algorithm described in Section 4.6. These extensions pay attention to the practical situation, in which a shunt planner typically solves multiple instances for the same scenario with different parameter settings in the objective. The planner searches a set of parameter settings that results in a solution corresponding with his wishes. In such similar instances, one could exploit the information of previously solved instances.

The first extension introduces a permanent inventory of columns, based on previous instances. Whenever a planner only changes parameters in the objective, the inventory still consists of feasible columns. Therefore, one only needs to determine the new objective coefficients for these columns. This inventory of columns typically results in good estimates of the overall optimal dual variables. In turn, such estimates result in a limited number of additionally generated columns. In a different configuration, one can generate a large inventory of initial columns based on likely parameter settings. Substantial savings in computation time for instances at station Zwolle are reported in Fioole [2003]. Moreover, the inventory can also serve as a start for initial explicit column generation. In this case, one starts the pricing problem with enumerating a subset of columns from this inventory. This subset is added to the restricted master problem. Then, the master problem is optimized, resulting in optimal dual variables for this restricted master problem. Given the values of these dual variables, one can iteratively add a new subset of columns from the inventory with negative reduced cost, and re-optimize the restricted master problem.

The second extension is the application of 2-OPT to TAP. The outline of 2-OPT for TAP is given in Algorithm 4.1. The goal of 2-OPT is to improve an initial solution by local search. In this heuristic, two blocks overlap if their periods at the shunt yard overlap. The best shunt track for a block i is determined as follows: for each shunt track, one determines whether or not the incumbent track assignment extended with block i is feasible. From the set of feasible shunt tracks for block i, the track resulting in minimum cost of its track assignment including block i is the best track.

This results in a significant reduction in computation time, especially when it is combined with the previously mentioned permanent inventory of columns. We will use 2-OPT in the computational results of Chapter 6. Moreover, this heuristic can also be used as an alternative solution approach as an alternative to the described column generation heuristic for TAP.

```
Name: 2-OPT applied to TAP.
Data: A set B of blocks, an initial solution to the parking problem, estimates of routing cost,
      and other objective elements.
Result: A high quality assignment of blocks to shunt tracks.

Determine the cost of the initial solution to the parking problem;
Set the current solution to the initial solution;
for i = 1, ..., |B| do
    for j = 1, ..., |B|, j ≠ i do
        if blocks i and j overlap then
            Remove blocks i and j from the current solution;
            Find the best shunt track for block j;
            Find the best shunt track for block i;
            Determine the cost of the new solution;
            if cost new solution < cost current solution then
                Set the current solution to the new solution
```

Algorithm 4.1: The outline of 2-OPT applied to TAP.

The third extension is an extension of the objective function. When a planner re-optimizes an instance, small changes in parameter settings might result in large changes in the solution. In order to minimize such changes, such large changes can be penalized, resulting in a solution similar to a reference solution. The parameters and penalties require careful tuning, since setting the penalty too high will absorb smaller changes in the parameter setting.

4.9 CONCLUSIONS

We started this chapter with an introduction of the parking problem, including a discussion of its restrictions and important elements of its objective. The restrictions ensure that:

- no block is obstructing the arrival or departure of another block,
- the sum of the length of the blocks parked at a specific track never exceeds the length of this track, and
- the blocks assigned to a shunt track are allowed to park at this track.

The most important element of the objective of the developed model is to park as many blocks as possible at the shunt tracks. Other elements include estimates of routing costs, penalties for undesired characteristics, and preferences of shunt planners for certain shunt tracks. This led to the formulation of TAP. We studied the relation of TAP with other research topics, such as sorting of permutations and container ship stowage.

We showed that TAP in its general form is $\mathcal{NP}$-complete via a reduction from the Bin Packing Problem. Moreover, we made plausible that a special case of TAP is polynomially solvable. This special case restricts the maximum number of blocks simultaneously parked at a shunt yard with fixed layout and can be solved by a shortest path problem. TAP has been formulated as a Set Partitioning Problem, resulting in a huge number of decision variables.

In order to solve this formulation, we propose a column generation heuristic. In this heuristic, we only generate columns while solving the LP-relaxation of the set partitioning formulation. A dynamic programming algorithm is used for solving the pricing problem. The underlying network structure is new and speeds up the pricing problem. If the LP-relaxation does not result in an integer solution, we resort to CPLEX for generating an integer solution.

Based on the matchings resulting from the previous chapter, we constructed a set of test instances for the stations Zwolle and Enschede. The heuristic performed well on these instances: they were solved quickly and resulted in high-quality solutions. Computation times lie within 10 seconds, except for Tuesday / Wednesday instances at station Zwolle, which require at most 10 minutes. Gaps between the LP-relaxation and the integer solution are within 4% with integer LP-relaxations for many instances. From a practical point of view, the heuristic is able to park all blocks in all instances.

We saw that modeling the free tracks, which can be approached from both sides, as LIFO tracks, which can be approached from one side, is quite restrictive, especially at station Zwolle. Moreover, the objective can be extended to penalize different train configurations of blocks at a track, and to reward combinations of blocks leaving in the same departing train from the same shunt track. The number of changes in block configurations simultaneously at a shunt track can be reduced significantly. Moreover, the combination of blocks leaving in the same departing train at a shunt track improves the Tuesday / Wednesday instances at station Enschede and especially at station Zwolle. The Saturday / Sunday instances did not benefit from this extension, which is caused by the fact that these instances do not have any possibility for such combinations.

In Chapter 3, we introduced 3 objectives for the Train Matching Problem (TMP): A, B, and C. Objective A focuses on a minimum number of blocks and a minimum number of blocks that need to be parked. Objective B extends this by preferring matchings with a FIFO character, which is convenient for tracks where train units arrive from one side of a shunt track and depart from the other. Finally, objective C prefers matchings with a LIFO character, which is convenient for LIFO tracks.

By studying the instances with different objectives for TMP, we see that objective B is outperformed by objectives A and C. In turn, objective C generally outperforms objective A.

Concluding, in this chapter we developed an adequate solution approach for TAP which is able to solve real-life instances within short computation times and resulting in high-quality solutions.

Chapter 5

Routing of Train Units

In this chapter, we study shunt routing of train units over the station infrastructure. Typically, shunt routing is required from platforms to shunt tracks and vice versa. Possibly, additional shunt routing is necessary for other shunting processes, such as cleaning. This shunt routing should not interfere with the routing of through train services over the station infrastructure. In addition, shunt routing plays a role in estimating routing cost for blocks. These estimates are used when solving the Track Assignment Problem (TAP) as discussed in the previous chapter. In a later phase, when the shunt tracks for the blocks have been decided upon, routes for each parked block to and from its shunt track have to be determined.

The structure of this chapter is as follows: we start with an introduction of the shunt routing problem. The introduction is followed by a review of related research topics. After some modeling issues and an introduction to state-space search, the proposed algorithm for shunt routing is outlined and we introduce some implementation details. Computational results are reported and we pay some attention to the practical support of planners in shunt route planning. Our research starts with the work of Van 't Woudt [2001] and Fioole [2003]. Lentink et al. [2003] discuss small adaptations and extensions are discussed and serves as the basis for this chapter.

5.1 INTRODUCTION TO ROUTING

In Section 1.10 we introduced the main assumptions of this thesis. For this chapter, the most important assumption is that we only consider the railway infrastructure within entering and leaving points of a station. In this chapter, we assume that the safety system, which avoids collisions between trains, works as described on page 33 in Section 2.3.3. This safety system claims an entire route over the station infrastructure. In case a route r_i intends to use tracks or switches which have been claimed by another route r_j, the claim for route r_i is denied. After a train has passed a section of its route, the section is released and can be claimed by other trains. Moreover, we also assume that through train services have been routed over the station infrastructure, resulting in a set of fixed reservations.

The most important terms we will use in this chapter are a route request, which is a request for a route over the station infrastructure, and a route conflict, which occurs when two train movements use the same infrastructure at nearly the same time (see Appendix A.1). In addition, we will use the terms free track, at which arrivals and departures can take place at both sides of the track, and shunt units, which are train units that require shunting, from this appendix. Finally, the term infrastructure is used throughout this chapter to denote the physical railway infrastructure, i.e. the tracks and the switches.

The routes for the requests are not allowed to conflict with other reservations of infrastructure. Three types of such infrastructure reservations are discerned:

1. Reservations for train movements. These movements consist of movements resulting from through train services and movements resulting from shunting processes.

2. Reservations for rolling stock standing still at tracks. For example, dwell times of timetabled trains at platforms.

3. Reservations for maintenance of station infrastructure. This results in station infrastructure being out of service for a period of time.

Typically, a reservation also has to be made for a certain minimum headway time between two subsequent reservations.

A planner needs to find routes for all requests, possibly making small modifications to one or more requests. Therefore, the main objective of algorithmic decision support is to find as many shunt routes as possible within the planning norms. Several other characteristics are also taken into account in the objective:

- the traveled distance. Minimizing the traveled distance results in a minimum resource usage. Examples of such resources are shunting crews, energy, and the infrastructure itself.

- the number of changes in direction. A change in direction requires the shunt driver to walk to the other end of the train, and therefore results in a longer duration of the shunt route.

- the number of simultaneous shunt routes. This number can be used as a proxy for the number of involved shunting crews.

- the deviations from preferred start times. Although the timing of shunt routes is flexible to some extent, planners typically have preferred start times for a route request.

Note that the cost of a shunt route to or from a free track might differ for the two sides of the track that can be used.

After this introduction, the problem of finding shunt routes for route requests is defined as follows:

Definition 5.1. Given the station railway infrastructure, a set $\mathcal{X}$ of assigned infrastructure reservations, and a set $\mathcal{R}$ of route requests, the Shunt Routing Problem (SRP) is to find a maximum number of shunt routes without conflicts, thereby minimizing the cost of the set of routes.

In case a solution to SRP is used for estimating the route cost as an input for TAP, each block in a solution to the Train Matching Problem (TMP), discussed in Chapter 3, results in one route request for each shunt track. A solution to TAP prescribes a shunt track for each block and possibly the sides to use for arrival and departure. Therefore, a solution to TAP results in an instance for SRP with the actual route requests.

5.2 RELATED RAILWAY ROUTING PROBLEMS

Initial studies on the capacity for routing trains to and from shunt tracks can be found in Egbers [2001], Van den Broek [2002], and Van den Broek and Kroon [2005]. These authors assume that capacity of parking, crew, and rolling stock is sufficiently available and can therefore be neglected. In addition, it is also assumed that the timetable and the corresponding rolling stock circulation have already been determined. The problem is to determine whether or not the capacity for routing train units between the platforms and the shunt tracks is sufficient. An important characteristic of the problem is that the timing of the route can be flexible to some extent, i.e. it is possible that the start time for a route is allowed to lie in a certain interval. This results in a flexible order in which trains use railway infrastructure between the platforms and the shunt tracks. Based on those characteristics and without knowing the exact shunt track at which shunt units will be parked, one looks at the capacity of the tracks that connect the platform tracks to the shunt tracks. Potential routing conflicts at the shunt tracks are disregarded.

Egbers [2001] aggregates the infrastructure in his model and checks whether or not the implied routes to and from zones of the shunt yard over the aggregated infrastructure are possible.

Van den Broek [2002] uses the notion of time dependency and route dependency between two train movements. Two train movements are time dependent whenever their time windows overlap. Two train movements are route dependent whenever they share a part of the infrastructure. Concluding, two train movements result in a route conflict whenever the movements are both time and route dependent. The resulting model determines the order of routes for train movements on different parts of the infrastructure, where each train movement uses its preferred route. In practice, if it is not possible to route a train along its preferred route, a planner can choose a route from a set of alternative routes. Van den Broek and Kroon [2005] introduce a mathematical model which incorporates such alternatives.

Zwaneveld et al. [1996] and Zwaneveld [1997] studied a similar routing problem. In this problem, one is looking for a set of routes and platforms for services in a one hour period, where arrival and departure times are fixed. The problem described in this chapter is on one hand more complicated by the flexible start and end times of routes, the longer planning horizon, and the fact that infrastructure might be partly reserved by e.g. through trains or maintenance. On the other hand, since we consider the platforms of the arrivals and departures as fixed, our problem is easier compared to the problem studied by these authors. Kroon et al. [1997] proved that the decision version of their problem is $\mathcal{NP}$-complete when considering the restrictions of the safety system and instances where each train has at least 3 routing possibilities. A corresponding optimization problem can be solved in polynomial time whenever the layout of the station is fixed. Very similar results hold for SRP but are not repeated in this thesis.

A similar problem as the previously described problem is discussed by Galaverna et al. [1994]. Influenced by their larger planning horizon of one day instead of one hour, the authors propose a train-by-train heuristic. If it is impossible to route a train t, a backtracking procedure changes the route of a previously planned train t' after which a new search is started for a route for train t.

De Luca Cardillo and Mione [1998] study the problem of assigning platforms to arriving or departing trains at a station. The assignments need to be without conflicts at the platforms. Important characteristics are the time window for arrival or departure as well as the arrival and departure sides of the platform used by a train. This problem is modeled as a graph coloring problem with additional restrictions. The authors propose a backtracking heuristic to solve the problem and report results on instances representing stations of different sizes. Billionnet [2003] reports a mathematical model for this problem with computational results for randomly generated instances as well as two real-life instances.

Carey and Carville [2003] investigated a problem similar to SRP. The authors also consider a planning period of one day and flexible arrival and departure times. It differs from the problem in this chapter because it also decides upon platforms for the trains. An important aim of the algorithm is to enable explanations of a solution to planners, railway operators and railway regulators. Therefore, a train-by-train heuristic has been developed, which closely resembles the approach of human planners. Computational experiments are reported for station Leeds, one of the largest stations in the United Kingdom.

Carey [1994] discusses the problem of integrating network-wide timetables in a railway network, including decisions on platform assignments for trains visiting stations. He considers several routes and platforms for a train to choose from. The problem is decomposed into a set of simpler problems, one for each train. Such a simpler problem is solved as an Integer Program. Computational results on a network with 10 trains show that the algorithm is able to quickly find acceptable solutions for these small instances.

5.3 MODELING STATION INFRASTRUCTURE

Figure 5.1 depicts an example of the standard network representation of a part of a station, as used by Dutch railway organizations. In this figure, tracks and switches are represented by nodes in the network. The tracks are given as open circles, while switches are represented by black dots. If two tracks or switches are physically connected then an undirected arc connects the nodes in the corresponding network. This example is part of the infrastructure of the main example of this thesis, see Figure A.3 on page 203.

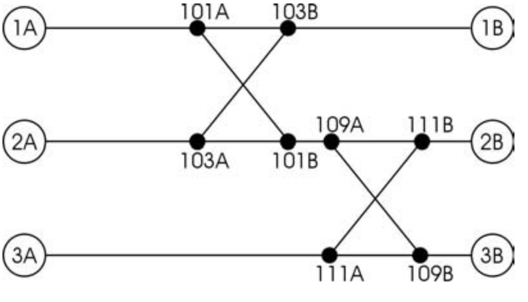


Figure 5.1: A straightforward way to model the station infrastructure.

Unfortunately, this representation is not suitable for solving SRP, since it is difficult to detect all potential conflicts. Consider for example the routes of the trains in Figure 5.2: one from track 1A to 2B and another one from track 2A to 1B, both at the same time. From a practical point of view, it is obvious that the routes in this example conflict because they simultaneously use crossing switches. However, since the routes do not share any node or arc in the graph, detection of the conflict is not straightforward.

In order to detect the conflict of Figure 5.2 more easily, we need to use a more sophisticated model, which is depicted in Figure 5.3. In this model, a route from track 1A to 2B passes three nodes when using switch 101 (101A, 103A and 101B). This model enables us to easily detect all conflicts on the infrastructure. Note that this model

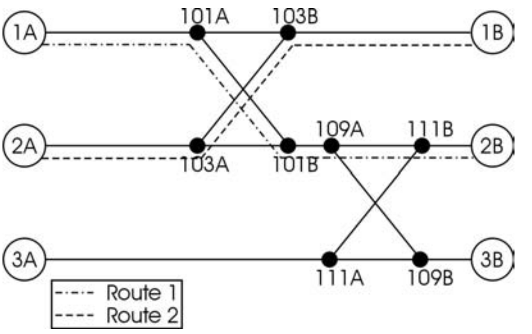


Figure 5.2: A problem with the straightforward model of the station infrastructure.

requires some additional information on the nodes of the network. This information indicates whether or not it is possible to change directions at a node. For example, without this information it is impossible to prohibit the path 1Ar→101A→103A→2Ar, which is not possible on the physical infrastructure, since it implies changing direction at switch 101A.

Note that in Figure 5.3 the nodes representing tracks are split. Indeed, for each track that can be approached from both sides, we explicitly model these sides. This enables the differentiation of routing cost for different sides of a track.

The actual network that will be used in the proposed routing algorithm for SRP is an extended version of the network of Figure 5.3. Each node in this network is duplicated in the extended network and is connected to other nodes by directed arcs. The first copy of the node represents arriving from the "left" and departing to the "right" side of the node. Vice versa, the second copy represents arriving from the "right" and departing to the "left" side of the node. For instance, a part of the network of Figure 5.3 in its extended

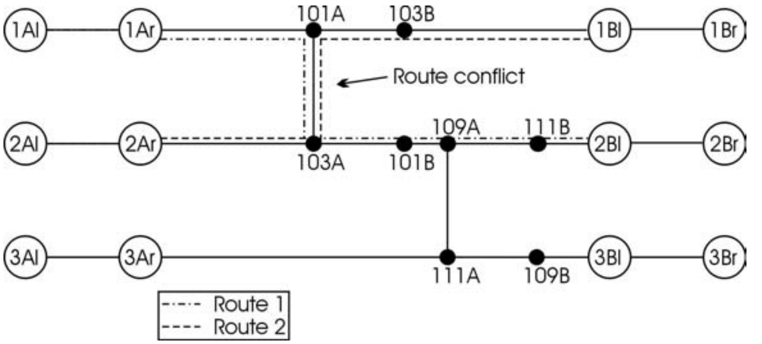


Figure 5.3: An updated model of the station infrastructure.

form is depicted in Figure 5.4. For reasons of clarity, the part of the original network representing nodes 109A, 111B, 2B, 3A, 111A, 109B, and 3B has been left out. Both copies of a node are connected with two directed arcs whenever changing direction at the original node is possible. In this extended network, a path 1Ar→101A→103A→2Ar is impossible: both copies of the node representing the switch 101A are not connected by arcs, since changing direction is not possible there.

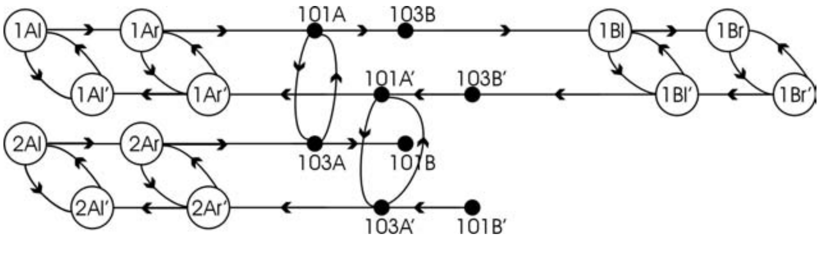


Figure 5.4: The extended version of a part of the network from Figure 5.3.

Nodes representing parts of the station infrastructure that are unavailable for certain requests are left out the network for these requests. Examples are tracks without catenary, which cannot be used by electrical train units, in case the request involves such units.

Another relevant dimension of SRP is time: in the extended network nodes might be occupied for certain periods of time. In order to model this, the extended network is extended itself. For each node, one copy for each minute in the planning period should be created, assuming that infrastructure can be reserved on a minute-by-minute basis. This leads to many layers in the network, in which infrastructure reservations are modeled by deleting corresponding nodes for certain minutes.

5.4 INTRODUCTION TO STATE-SPACE SEARCH

In this paragraph, we briefly introduce some of the main concepts in state-space search. Based on this discussion, we outline our algorithm for solving SRP. Most of the concepts in this paragraph can be found in Pearl [1984] and Luger and Stubblefield [1998].

In general, state-space search is a solution procedure that systematically explores the state-space of a problem, which has been represented by a network. Formally, Luger and Stubblefield [1998] define state-space search as follows:

Definition 5.2. Given is a network $\mathcal{G}$=($\mathcal{N}$, $\mathcal{A}$), with $\mathcal{N}$ the set of nodes and $\mathcal{A}$ the set of arcs connecting nodes, a set $\mathcal{O}$ with origin nodes, and a set $\mathcal{D}$ with destination nodes. Then, state-space search is defined as looking for a solution path through the network $\mathcal{G}$ from an origin node $o \in \mathcal{O}$ to a destination node $d \in \mathcal{D}$.

Note that the nodes $\mathcal{N}$ represent the state-space, while the arcs $\mathcal{A}$ correspond to steps in the solution process.

Potentially, state-space search algorithms use some kind of prior knowledge of the network $\mathcal{G}$. Consider for example the network in Figure A.1 on page 202, and suppose one is interested in the shortest path from Roosendaal (Rsd) to Groningen (Gn). In case a partial route from Roosendaal to Zwolle (Zl) has been found, the prior knowledge is the `estimated` length of the remaining shortest path from Zl to Gn, which might be the geometric distance between Zl and Gn. The set of algorithms that use such prior knowledge is called the set of informed search algorithms, and is a subset of all state space search algorithms.

Additional notation is introduced before the discussion of search algorithms. With $k(n)$ we denote a heuristic evaluation function, which estimates the lowest cost of reaching a destination node $d \in \mathcal{D}$ from a node $n \in \mathcal{N}$, while $f_{n,n'}$ denotes the cost of arc $(n, n') \in \mathcal{A}$. Moreover, $\mathcal{U}$ is the set of unexpanded nodes of $\mathcal{G}$ and $\mathcal{E}$ is the set of expanded nodes. Finally, k_n denotes the evaluation of the incumbent $o{\to}n$ path.

We assume that the objective is to find a minimum cost path. Moreover, we assume that the next node to be expanded is the node with the lowest evaluation $k(.)$. Finally, in case two paths lead to the same node, the path with the higher evaluation $k(.)$ can be discarded.

The value $k(n)$ might reflect characteristics of the path $o{\to}n$, characteristics of the node n, and problem specific knowledge to be used in estimating the cost of a path $n{\to}d$ from n to a destination node $d \in \mathcal{D}$. Since we only consider one path to each node, the $o{\to}n$ path is uniquely defined for each node. Given these assumptions and notation, the framework of Best-First Search (BFS) algorithms is given in Algorithm 5.1. After some header information, the algorithm is given in pseudo code, with comment lines starting with '//'.

The algorithm starts with initializing the set of unexpanded nodes $\mathcal{U}$ with the set of origins $\mathcal{O}$ on line 5.1.1. Then, while $\mathcal{U}$ is non-empty, the node n^* with the lowest evaluation value is selected in line 5.1.3. The next line moves the node from $\mathcal{U}$ to $\mathcal{E}$. Line 5.1.5 states that if the selected node is a destination node, the algorithm returns the induced path. Otherwise, all emanating arcs (n^*, n') are evaluated. If node n' has not been considered by the algorithm before (line 5.1.9), then it is added to $\mathcal{U}$, and its evaluation is saved in $k_{n'}$. Otherwise, line 5.1.13 states that if we can reach this node with a lower evaluation than the incumbent path, we update the incumbent path and the evaluation of the node. In this case, BFS makes sure that n' is an element of $\mathcal{U}$, see lines 5.1.15 - 5.1.17. In BFS algorithms, nodes can be added to $\mathcal{U}$ several times. Indeed, every time a better path from a source $o \in \mathcal{O}$ to an expanded node $n \in \mathcal{E}$ is found, n is removed from $\mathcal{E}$ and added to U in lines 5.1.15 - 5.1.17, and might therefore be re-examined.

Name : Best-First Search.
Data: A network $\mathcal{G}$=($\mathcal{N}$, $\mathcal{A}$), a set of origins $\mathcal{O}$, a set of destinations $\mathcal{D}$, and a heuristic evaluation function $k(n)$ for $n \in \mathcal{N}$.
Result: A path from an origin $o \in \mathcal{O}$ to a destination $d \in \mathcal{D}$ or 0 if none exists.

5.1.1 $\mathcal{U} = \mathcal{O}$;
5.1.2 **while** $\mathcal{U} \neq \emptyset$ **do**
5.1.3 $n^* = \arg\min_{n \in \mathcal{U}} k(n)$;
    `// break ties arbitrarily in the above minimum`
5.1.4 $\mathcal{U} = \mathcal{U} \setminus \{n^*\}, \mathcal{E} = \mathcal{E} \cup \{n^*\}$;
5.1.5   **if** $n^* \in \mathcal{D}$ **then**
5.1.6   return the induced $o \to n^*$ path;
5.1.7   **forall** $(n^*, n') \in \mathcal{A}$ **do**
5.1.8   Calculate $k(n')$;
5.1.9   **if** $n' \notin \mathcal{U}$ *and* $n' \notin \mathcal{E}$ **then**
            `// the first` $o \to n'$ `path found`
5.1.10 $\mathcal{U} = \mathcal{U} \cup \{n'\}$;
5.1.11 $k_{n'} = k(n')$;
5.1.12   **else**
5.1.13   **if** $k(n') < k_{n'}$ **then**
                `// a better` $o \to n'$ `path found`
5.1.14 $k_{n'} = k(n')$;
5.1.15   **if** $n' \in \mathcal{E}$ **then**
5.1.16 $\mathcal{U} = \mathcal{U} \cup \{n'\}$;
5.1.17 $\mathcal{E} = \mathcal{E} \setminus \{n'\}$;

5.1.18 return 0; `// since no solution exists`

Algorithm 5.1: The outline of Best-First Search algorithms.

Suppose $k(n)$ is restricted to the form $k(n) = l(n) + m(n)$, with $l(n)$ the cost of the induced $o{\to}n$ path and $m(n)$ the estimated cost of reaching a destination node $d \in \mathcal{D}$ from node n. Moreover, $l(n)$ has the form $l(n') = l(n) + f_{n,n'}, \forall (n, n') \in \mathcal{A}$. In case $m(n)$ always is an optimistic estimate, i.e. the estimated cost of reaching any destination is never higher than the real cost, this gives rise to the well known A* search algorithm [Hart et al., 1968], which is outlined in Algorithm 5.2. Indeed, the evaluation function $k(.)$ of Algorithm 5.2 is a special case of the one in Algorithm 5.1. This can be seen easily by comparing line 5.1.8 in Algorithm 5.1 with lines 5.2.8 and 5.2.9 in Algorithm 5.2. This difference is the only difference between both algorithms.

Name: A* Search.
Data: A network $\mathcal{G}$=($\mathcal{N}$, $\mathcal{A}$), a set of origins $\mathcal{O}$, a set of destinations $\mathcal{D}$, and a heuristic evaluation function $k(n) = l(n) + m(n)$ for $n \in \mathcal{N}$.
Result: A path from an origin $o \in \mathcal{O}$ to a destination $d \in \mathcal{D}$ or 0 if none exists.

$\mathcal{U} = \mathcal{O}$;
while $\mathcal{U} \neq \emptyset$ **do**
  $n^* = \arg\min_{n \in \mathcal{U}} k(n)$;
  `// break ties in the above minimum arbitrarily`
  if $n^* \in \mathcal{D}$ **then**
    return the induced $o \rightarrow n^*$ path;
  $\mathcal{U} = \mathcal{U} \setminus \{n^*\}, \mathcal{E} = \mathcal{E} \cup \{n^*\}$;
  forall $(n^*, n') \in \mathcal{A}$ **do**
    $l(n') = l(n^*) + f_{n^*,n'}$;
    $k(n') = l(n') + m(n')$;
    if $n' \notin \mathcal{U}$ *and* $n' \notin \mathcal{E}$ **then**
      `// the first` $o \rightarrow n'$ `path found`
      $\mathcal{U} = \mathcal{U} \cup \{n'\}$;
      $k_{n'} = k(n')$;
    else
      if $k(n') < k_{n'}$ **then**
        `// a better` $o \rightarrow n'$ `path found`
        $k_{n'} = k(n')$;
        if $n' \in \mathcal{E}$ **then**
          $\mathcal{U} = \mathcal{U} \cup \{n'\}$;
          $\mathcal{E} = \mathcal{E} \setminus \{n'\}$;

return 0; `// since no solution exists`

Algorithm 5.2: The outline of A* Search algorithms.

Different choices for $k(.)$ lead to different algorithms:

- $k(n') = k(n) + f_{n,n'}, \forall (n, n') \in \mathcal{A}$ and $k(o) = 0, \forall o \in \mathcal{O}$ results in an extension of the algorithm of DIJKSTRA [1959] for the Shortest Path Problem (see Definition 3.9 on page 58) with multiple origins and destinations.
- $k(n') = k(n) + 1, \forall (n, n') \in \mathcal{A}$ and $k(o) = 0, \forall o \in \mathcal{O}$ results in Breadth-First Search.
- $k(n') = k(n) - 1, \forall (n, n') \in \mathcal{A}$ and $k(o) = 0, \forall o \in \mathcal{O}$ results in Depth-First Search.

Due to the well-behaved nature of the functions $k(.)$ and $l(.)$, A* Search has several nice properties. The three most important ones are:

- Completeness. If a solution exists, A* Search will find it.
- Optimality. If a solution exists, A* Search will find the optimal one.
- Efficiency. Given the same information $m(.)$, algorithm A is efficient over a class of algorithms if every expanded node in A is also expanded by each algorithm in the class.

Of course, completeness follows directly from optimality. These properties are stated here without proof. The interested reader is directed to Pearl [1984] and Dechter and Pearl [1983] for formal proofs of these properties as well as extensions of A* Search, additional properties and their proofs. Note that in these texts optimality is denoted with admissibility and efficiency is called optimality.

In the remainder of this chapter, we only consider a network with nodes out of which at most one emanating arc can be in a solution, i.e. the network consists of so-called OR-nodes. In a more general setting, AND-nodes are also possible. In this case, one needs to select all or none of the emanating arcs from such an AND-node. This generalization is treated in Pearl [1984].

The success of A* Search algorithms is largely influenced by the quality of the estimate for the remaining cost $m(.)$. SRP contains high-quality information available for this estimation. This high-quality information consists of the distance from the last node in the path to a destination node in a network without reservations. In case no other infrastructure reservations are present, this minimum distance represents the best way to complete a route. Already planned reservations might require detours or deviations from the desired start times of a route.

In principle, the estimate for the remaining cost $m(.)$ can be improved upon by taking into account infrastructure reservations of through trains. Since these trains have been planned before, the corresponding reservations have to be considered fixed. The nodes representing these routes are unavailable for shunt routes. A disadvantage of this approach is that it requires a substantial computation time to determine these minimum distances given a set of already planned reservations. Moreover, these minimum distances need to be recomputed whenever the set of through trains changes. In Section 5.7, we address the effect of the estimate for the remaining cost $m(.)$ on the computational properties of our solution approach.

5.5 ROUTING ALGORITHM

In this paragraph, we will discuss how A* Search serves as the basis for solving SRP. Based on the introduction to A* Search and the introduction of SRP in Section 5.1, this paragraph introduces the developed routing algorithm for SRP. In general, algorithms for searching routes can be partitioned into two classes:

1. Algorithms that search routes simultaneously.

2. Algorithms that search routes sequentially.

An obvious advantage of algorithms belonging to the first class over algorithms of the second class, is the possibility to compute an overall optimal solution for routing all shunt units over the infrastructure, while taking into account all restrictions and interdependencies. However, a disadvantage of such an approach is typically the amount of computation time that is needed for finding an optimal solution.

The methodology described in Zwaneveld [1997] belongs to the first class of algorithms and therefore searches routes for a set of requests simultaneously. However, a sequential approach, where the algorithm searches routes on a request-by-request basis, better resembles the current practice of planners, which is stressed by Van Wezel and Barten [2002] and Van Wezel and Jorna [2004] among others. Moreover, Carey and Carville [2003] argue that the results of a sequential approach can be explained better to planners, railway operators and other parties. Finally, a sequential approach typically finds routes fast. This motivates our choice for a sequential approach.

As mentioned at the end of Section 5.3, a node in the network represents a track or a switch at a specific minute in the planning period. Since a route request might have a flexible start time, the set of origins $\mathcal{O}$ might contain several nodes, representing the same physical location at different points in time. Similarly, $\mathcal{D}$ might contain multiple nodes.

5.5.1 Occupied Network A* Search

This paragraph describes the Occupied Network A* (ONA*) Search algorithm that has been developed for SRP. Extensions of A* Search are required since costs are not only present at the arcs but also at the nodes. The costs at a node include for example the preferences of planners. Moreover, several additional stop-criteria are introduced.

Cost are defined on the nodes and arcs instead of solely on the arcs. Examples of the cost of a node are cost for traveled distance, and preferences of planners for using certain parts of the infrastructure. An example of the cost of an arc is the cost of changing direction.

Before we discuss the outline of ONA* Search, we introduce some additional notation. Parameter e denotes the maximum number of expansions of nodes for one route request, v represents the upper bound on the cost of a route. Moreover, w represents the maximum number of changes in direction in a route, and the function $w(.)$ returns

Name : ONA* Search.

Data: A network $\mathcal{G}$, a set of origins $\mathcal{O}$, a set of destinations $\mathcal{D}$, a maximum number of iterations e, an upper bound v on the cost of a shunt route, a maximum number of changes in direction w.

Result: A path from an origin $o \in \mathcal{O}$ to a destination $d \in \mathcal{D}$ in $\mathcal{G}$ or 0 if none exists.

$\mathcal{U} = \mathcal{O}, i = 0$;
while $\mathcal{U} \neq \emptyset$ *and* $i \leq e$ **do**
    $n^* = \arg\min_{n \in \mathcal{N}} k(n)$;
    `// break ties in the above minimum arbitrarily`
    if $n^* \in \mathcal{D}$ **then**
        return the induced $o - n^*$ path p^*;
    $\mathcal{U} = \mathcal{U} \setminus \{n^*\}, \mathcal{E} = \mathcal{E} \cup \{n^*\}$;
    if $k(p^*) > v$ **then** return 0;
    `// the cost of a shunt route for this request exceeds` v
    if $w(p^*) > w$ **then** $i = i+1$;
    `// more than` w `changes in direction for this request`
    else
        $i = i+1$;
        forall $(n^*, n') \in \mathcal{A}$ **do**
            $l(n') = l(n^*) + f_{n^*,n'}$;
            $k(n') = l(n') + m(n')$;
            if $n' \notin \mathcal{U}$ *and* $n' \notin \mathcal{E}$ **then**
                `// the first` $o - n'$ `path found`
                $\mathcal{U} = \mathcal{U} \cup \{n'\}$;
                $k_{n'} = k(n')$;
            else
                if $k(n') < k_{n'}$ **then**
                    `// a better` $o - n'$ `path found`
                    $k_{n'} = k(n')$;
                    if $n' \in \mathcal{E}$ **then**
                        $\mathcal{U} = \mathcal{U} \cup \{n'\}$;
                        $\mathcal{E} = \mathcal{E} \setminus \{n'\}$;
return 0; `// since no solution exists`

Algorithm 5.3: The structure of the ONA* Search algorithm.

the number of changes in direction of a path in $\mathcal{G}$. Finally, we denote with p^* the induced $o \to n^*$ path at node n^*. Based on these definitions and previously discussed characteristics, we outline ONA* Search in Algorithm 5.3.

The main differences with A* Search in Algorithm 5.2 are the additional stop-criteria for the algorithm:

- a maximum number of node expansions for a request in line 5.3.2,
- a maximum cost for a route in line 5.3.7,
- a maximum number of changes in direction in line 5.3.8.

These stop-criteria ensure that no routes without practical value are found. Note that by implementing these stop criteria, ONA* Search is no longer complete, and therefore also no longer optimal. The paths that are neglected have no practical value since these are usually too complex for the operations. In such cases, alternative routes need to be found for one or more previous route requests. If no such alternatives are available, an alternative solution to a previous process needs to be found, which avoids this route request, for instance by parking a block at a different shunt track.

Note the subtle difference between the maximum number of changes in direction in a request and the other two stop-criteria. In case the algorithm finds a partial route with a maximum number of changes in direction, it still continues to consider other partial routes. However, when the other two criteria are met, the algorithm terminates the search for a route. This is logical, since the number of expansions and the cost of the partial route so far are non-decreasing, which does not hold for the number of changes in direction.

In principle, the ONA* algorithm is executed sequentially for each route request, with a random order of the requests. Before searching a shunt route for a new request, the infrastructure reservations of the shunt route for the current request are added to the set $\mathcal{X}$. In this manner, the reservations of the current shunt route are taken into account when searching for future requests.

5.5.2 2-OPT Applied to SRP

An obvious disadvantage of the train-by-train heuristic introduced in the previous paragraph, is its sequential nature. Indeed, the overall quality of the set of shunt routes is largely influenced by the order of handling the route requests. In an attempt to reduce the impact of the order of handling requests, Fioole [2003] proposes to improve the solution by applying a 2-OPT interchange heuristic for handling the requests. This paragraph describes the heuristic.

We start with an example where the interchange heuristic improves the solution found based on Fioole [2003]. Consider the infrastructure of Figure A.3 on page 203.

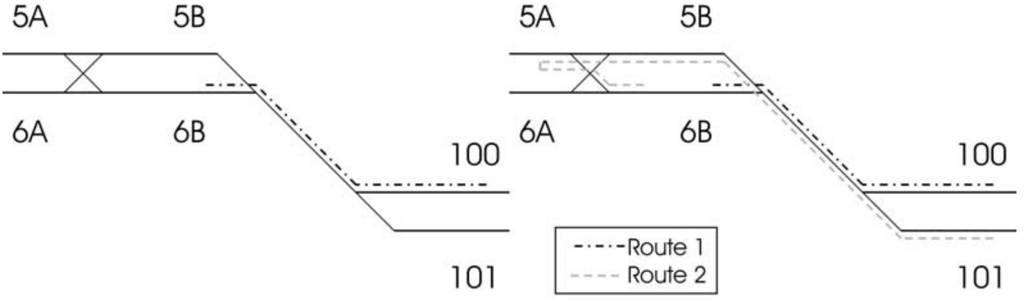


Figure 5.5: A solution for two requests without interchanging the order.

Suppose we have two route requests: the first from track 100 to the right side of track 6B, and the second from track 101 to the left side of track 6B. Moreover, the routes have sufficient timing flexibility to plan them in both orders. In addition, the train of request 1 leaves the station after the train of request 2 has arrived at platform 6B. Finally, suppose that without the interchange heuristic, request 1 is routed first, resulting in route 1, depicted in the left side of Figure 5.5. Since the train of route 1 remains at platform 6B, it obstructs a route of request 2 via the right side of platform 6B. Therefore, the route for request 2 contains a detour with changing direction at track 5A as shown in the right side of Figure 5.5.

However, if the order of routing the requests would be reversed, a better overall solution can be found. In this case, route 2 can use the right side of platform 6B before the train of request 1 is parked there. In Figure 5.6, the situation with a route for request 2 is shown in the left part. The right part of the figure shows the routes for both requests. Worse instances than the above example can be found. Indeed, planning a route for request i firstly prohibits finding a route for a request j, while planning a route for request j firstly leaves sufficient options for planning request i. In this case, the order i,j results in 1 planned route, while the order j,i finds both routes.

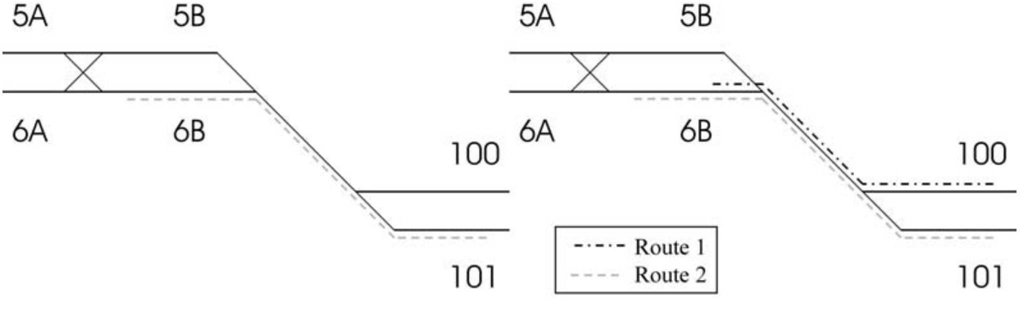


Figure 5.6: A solution for two requests with interchanging the order.

Name : 2-OPT applied to SRP.

Data: A set of routes $\mathcal{R}$, a network $\mathcal{G}$, an evaluation function $u(.)$ for a set of routes, and a number of iterations for improvement h, and parameters e,v, and w for ONA* Search (see Algorithm 5.3).

Result: A set of routes $\mathcal{R}'$ hopefully improved by exchanging the order of handling requests.

5.4.1 $\mathcal{R}' = \mathcal{R}$, $\mathcal{G}' = \mathcal{G}$;
5.4.2 **for** $g = 1, \ldots, h$ **do**
5.4.3   **for** $i = 1, \ldots, |\mathcal{R}'| - 1$ **do**
5.4.4   **for** $j = i + 1, \ldots, |\mathcal{R}'|$ **do**
5.4.5 $u = k(\mathcal{R}')$;
5.4.6 $\mathcal{R}' = \mathcal{R}' \setminus \{r_i, r_j\}$, $\mathcal{G}' = \mathcal{G}' \setminus \{r_i, r_j\}$;
5.4.7 r'_j = ONA* Search $(\mathcal{G}', \mathcal{O}_{r_j}, \mathcal{D}_{r_j}, e, v, w)$;
5.4.8 $\mathcal{R}' = \mathcal{R}' \cup r'_j$, $\mathcal{G}' = \mathcal{G}' \cup r'_j$;
5.4.9 r'_i = ONA* Search $(\mathcal{G}', \mathcal{O}_{r_i}, \mathcal{D}_{r_i}, e, v, w)$;
5.4.10 $\mathcal{R}' = \mathcal{R}' \cup r'_i$, $\mathcal{G}' = \mathcal{G}' \cup r'_i$;
5.4.11   **if** $u(\mathcal{R}') \geq u$ **then**
5.4.12 $\mathcal{R}' = \mathcal{R}' \setminus \{r'_i, r'_j\}$, $\mathcal{G}' = \mathcal{G}' \setminus \{r'_i, r'_j\}$;
5.4.13 $\mathcal{R}' = \mathcal{R}' \cup \{r_i, r_j\}$, $\mathcal{G}' = \mathcal{G}' \cup \{r_i, r_j\}$;
                    `// no improvement ⇒ undo the interchange`
5.4.14 return $\mathcal{R}'$;

Algorithm 5.4: The outline of the 2-OPT for SRP.

This example clarifies an important weakness in the train-by-train heuristic, which justifies 2-OPT outlined in Algorithm 5.4. This outline uses the following additional notation: h represents the number of improvement rounds, and the function $u(.)$ returns the cost of a set of routes $\mathcal{R}$. Routes are denoted with variables $r \in \mathcal{R}$. Moreover, we introduce the operator '$\setminus$' for a network $\mathcal{G}$ and a set or routes $\mathcal{R}$: $\mathcal{G} \setminus \mathcal{R}$ results in releasing the infrastructure reservations of the routes in $\mathcal{R}$ and thereby making nodes in $\mathcal{G}$ related to the routes in $\mathcal{R}$ available for other requests. Similarly, $\mathcal{G} \cup \mathcal{R}$ reserves the infrastructure of routes $\mathcal{R}$ for certain time periods, which removes nodes related to routes in $\mathcal{R}$ from $\mathcal{G}$. Finally, $\mathcal{O}_r$ represents the set of origins for the request corresponding to route r, and $\mathcal{D}_r$ represents the set of destinations for this request.

The algorithm computes potential improvements on the overall solution by pairwise interchanging the order of handling request. After some initialization, the cost of the incumbent overall plan u is determined in line 5.4.5. Hereafter, the reservations for requests r_i and r_j are released. Then, we start with finding a route for request r_j

by calling the ONA* algorithm in line 5.4.7. The resulting infrastructure reservations are added to the network $\mathcal{G}$ and subsequently, we call ONA* Search for route r_i (see line 5.4.9). If these interchanges do not improve upon the evaluation function, we undo the changes in lines 5.4.12 and 5.4.13.

5.6 SOME IMPLEMENTATION DETAILS

In the implementation of the algorithm, and the model of the station infrastructure, we did not actually copy each node for each point in time as suggested in Section 5.3. This would result in an enormous number of nodes, typically well over 100,000. Instead, we keep at each node a list of time intervals during which the node is occupied by another reservation of the infrastructure. This requires some minor modifications to the algorithm as it is presented in Algorithm 5.3, since expanding a node with an edge might not be feasible, for example because the node to which the arc is directed has already been reserved for relevant time intervals. Moreover, the dominance criterium $k(n') < k_{n'}$ needs to be extended. Indeed, at a node we need to keep several paths representing different timing of partial routes, and therefore several $o{\rightarrow}n$ paths might be present at node n.

Moreover, we assumed that certain reservations of the infrastructure, denoted with $\mathcal{X}$, are already known when requests $\mathcal{R}$ are routed. However, taking the already reserved routes as a part of the input requires an enormous amount of additional input data, which might be difficult to obtain. Therefore, we simulate the reservations $\mathcal{X}$, by finding routes for the through train services. After the routes for the through train services have been found, resulting in the reservations $\mathcal{X}$, the algorithm starts with routing the requests from $\mathcal{R}$.

5.7 APPLICATIONS OF ROUTING

This paragraph reports on the results of computational experiments with the routing algorithm of Section 5.5, consisting of ONA* Search and 2-OPT. The hardware that was used for these experiments was described in Section 3.5. The instances consist of the routing requests resulting from the instances of TAP in the previous chapter. The instances related to Tuesday / Wednesday scenarios at stations Zwolle and Enschede are treated in-depth. Moreover, we give an overview of the instances concerning Saturday / Sunday at station Zwolle. The reader is directed to Appendix A for the computational results for Saturday / Sunday instances at stations Zwolle and Enschede.

For each instance treated in-depth, we report the number of route requests (including routes of the through train services) and the minimum routing cost. The minimum routing cost is determined as the sum of the estimates of the routing cost in a network without any infrastructure reservation. This means that infrastructure reservations of the through trains and other shunt routes are not taken into account. Note that the routes of the through trains could have been taken into account, resulting in better estimates of the shunt routes. The number of route requests depends on the number of through train services, the number of blocks that need parking, the number of blocks that only need routing, and the number of combinations of blocks leaving in the same departing train (see Section 4.1). The number of blocks that need parking consists of blocks that need routing to *and* from the shunt yard, and blocks that need routing to *or* from the shunt yard. Blocks that require only one route are either blocks that arrive at the shunt track but do not leave before the end of the planning period, or blocks that are already parked at a shunt track at the start of the planning period and only need to be routed to their departure platform.

For each instance, we ran the algorithm with 0, 1, and 2 applications of 2-OPT. Besides the previously described characteristics, we report the following statistics:

- the number of requests that could not be routed,
- the number of changes in direction,
- the number of deviations from the preferred start times,
- the total routing cost,
- the gap with the minimal routing cost,
- the computation time.

To start with, we report the computational results for the Tuesday / Wednesday instances at station Zwolle in Table 5.1. The number of route requests for instance ZTAD is determined as follows:

- the number of through trains is 610,
- 18 blocks need routing from an arrival platform to a departure platform only,
- 6 blocks need routing to or from the shunt yard,
- 48 blocks need routing to and from the shunt yard
- 2 blocks are combined into 1 at a shunt track.

This leads to 610 + 18 + 6 + 2*48 - 1 = 729 requests. The number of blocks that need parking is 6 + 48 = 54. From Table 3.5 on page 63, we know that 50 blocks do not need parking for this instance. Out of the 50 blocks that do not need parking, 18 need routing from the arrival platform to the departure platform (see the first column of Table 3.5 on

Instance	ZTAD	ZTBD	ZTCD	ZTAR	ZTBR	ZTCR	ZTAO	ZTBO	ZTCO
Number of route requests	729	737	731	728	737	730	724	729	724
Minimum routing cost	34308	34745	34685	35121	35851	35558	34590	34570	34718
No 2-OPT									
Number of routes not found	30	29	30	26	29	26	31	30	30
Number of changes in direction	60	62	61	47	56	55	56	67	58
Number of start time deviations	51	53	53	67	67	66	53	56	46
Routing cost	62575	62250	65246	59612	64154	60222	64071	63812	64183
Gap with minimal routing cost	45.17%	44.18%	46.84%	41.08%	44.12%	40.96%	46.01%	45.83%	45.91%
Computation time (in sec.)	5.2	4.3	6.1	4.2	5.7	4.6	6.5	4.8	5.3
Apply 2-OPT once									
Number of routes not found	13	16	13	12	15	15	16	16	16
Number of changes in direction	99	89	89	86	99	96	88	103	86
Number of start time deviations	66	69	71	78	81	73	67	71	59
Routing cost	47640	50492	47852	47696	51363	50590	50522	50390	50467
Gap with minimal routing cost	27.98%	31.19%	27.52%	26.36%	30.20%	29.71%	31.53%	31.40%	31.21%
Computation time (in sec.)	108.9	98.0	137.1	55.0	154.7	112.4	109.7	82.0	127.7
Apply 2-OPT twice									
Number of routes not found	13	15	12	12	14	13	15	15	15
Number of changes in direction	94	81	79	76	93	99	80	102	80
Number of start time deviations	66	70	71	78	83	74	67	71	59
Routing cost	47387	49225	46535	47312	50110	48873	49279	50390	49384
Gap with minimal routing cost	27.60%	29.42%	25.46%	25.77%	28.46%	27.2%	29.81%	29.98%	29.70%
Computation time (in sec.)	176.4	131.6	201.7	83.3	246.6	199.6	208.7	127.6	264.0

Table 5.1: Computational results for SRP at station Zwolle for Tuesday / Wednesday instances.

page 63). The numbers of combinations of blocks leaving in the same departing train for these instances can be found in Figure 4.5 on page 89. The differences between the numbers of requests of the instances ZT.D and ZT.O correspond to the differences in the number of combinations of blocks in the same departing train. For instance, the difference between the number of requests of instance ZTAD (729) and ZTAO (724) in Table 5.1 is caused by the fact that instance ZTAD results in 1 departure cluster, while instance ZTAO contains 6 of these clusters.

The penalty for changing direction is 5, the penalty for deviating from the preferred start time is 1, and the penalty for not finding a route for a request varies between 850 and 1000, depending on the minimum distance between the start and end point of the request. These parameters give some feeling for the reported routing cost.

In general, for the Tuesday / Wednesday instances at station Zwolle, the gaps without 2-OPT are around 45%, applying the heuristic once decreases the gaps to approximately 30%, while a second application only adds marginal value. Note that gaps around 0% are unlikely to occur, since this would imply that the routes of the blocks are marginally influenced by the routes of the through trains and the routes of the other blocks.

ONA* Search is able to find routes quickly for these instances, however the order of routing the requests has a large influence on the quality of the solution. This requires the application of 2-OPT, which is computationally expensive compared to ONA* Search. The first application of the interchange heuristic typically improves upon the number of routed requests, while a second application saves some changes in direction and most of the times a marginal number of unplanned requests.

The flexible start times provide added value, since many shunt routes have small deviations from their preferred start times. Moreover, on average nearly each shunt route contains a change in directions.

Considering the Tuesday / Wednesday instances in Table 5.2 at station Enschede, it is immediately clear that these instances can be solved more easily by ONA* Search, compared with the instances at station Zwolle. In nearly half of the instances, the algorithm is able to route all requests, while the other instances have one unplanned request. The first application of 2-OPT saves some changes in direction, at the cost of 2 to 3 additional deviations from the preferred start times. A second application of 2-OPT does not result in improvement in any of these instances.

From Tables 5.1 and 5.2, we conclude that especially the first application of 2-OPT to the instances at station Zwolle is very effective. The second application of 2-OPT at station Zwolle as well as the first application of it at station Enschede provide some small improvements. Finally, the second application of 2-OPT at station Enschede has no added value.

The interchange heuristic is more effective for the instances at station Zwolle than for the instances at station Enschede, since station Zwolle is used more heavily. This

Instance	ETAD	ETBD	ETCD	ETAR	ETBR	ETCR	ETAO	ETBO	ETCO
Number of route requests	196	195	195	193	193	193	193	193	193
Minimum routing cost	23889	23921	23898	23137	23221	23129	23633	23113	23613
No 2-OPT									
Number of routes not found	0	0	0	1	0	1	1	1	1
Number of changes in direction	26	27	27	22	20	22	19	22	21
Number of start time deviations	11	11	11	10	10	8	10	9	9
Routing cost	25073	25099	25106	25251	24290	25440	25341	25006	25518
Gap with minimal routing cost	4.72%	4.69%	4.81%	9.09%	4.40%	9.08%	6.74%	7.57%	7.47%
Computation time (in sec.)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Apply 2-OPT once									
Number of routes not found	0	0	0	1	0	1	1	1	1
Number of changes in direction	23	24	24	20	19	20	17	20	19
Number of start time deviations	13	13	13	13	12	11	12	11	11
Routing cost	24687	24713	24914	24725	23846	24714	25181	24844	25156
Gap with minimal routing cost	3.23%	3.20%	4.08%	6.42%	2.62%	6.41%	6.15%	6.97%	6.13%
Computation time (in sec.)	0.3	0.2	0.2	1.0	2.9	1.0	1.2	0.2	0.1
Apply 2-OPT twice									
Number of routes not found	0	0	0	1	0	1	1	1	1
Number of changes in direction	23	24	24	20	19	20	17	20	19
Number of start time deviations	13	13	13	13	12	11	12	11	11
Routing cost	24687	24713	24914	24725	23846	24714	25181	24844	25156
Gap with minimal routing cost	3.23%	3.20%	4.08%	6.42%	2.62%	6.41%	6.15%	6.97%	6.13%
Computation time (in sec.)	0.4	0.3	0.3	2.1	5.8	2.1	2.3	0.4	0.2

Table 5.2: Computational results for SRP at station Enschede for Tuesday / Wednesday instances.

implies that the set of reservations $\mathcal{X}$ has a larger effect on the available capacity of the infrastructure. Moreover, the time intervals of the route requests at station Zwolle overlap to a larger extent, compared to the intervals of the route requests at station Enschede. Indeed, if two subsequent route requests do not overlap in time, interchanging them has no effect. Therefore, we expect the 2-OPT heuristic to perform particularly well for instances at busy stations, with many overlaps between the time intervals of the route requests. In a similar train-by-train heuristic for a similar problem, Carey and Carville [2003] found that interchanging the order marginally improved the results, which is in line with our findings at station Enschede, and conflicting with our findings at station Zwolle. This might be explained by the fact that the railway infrastructure in the Netherlands is used more heavily than in the United Kingdom (see for example Vromans [2005]).

Additionally, we experimented with the value of the estimates $m(.)$ of a remaining path for several instances. For the instances ZTBR, ZSAD, ETAO, and ESAR, we reran the algorithm with $m(n) \equiv 0, \forall n \in \mathcal{N}$, resulting in Dijkstra's algorithm [1959] with multiple origins and destinations. Since the routing algorithm has no longer any incentive to direct the search to a destination node, the maximum number of iterations needs to be augmented in order to find similar results as the instances with high-quality estimates $m(.)$. For the instances ZTBR and ZSAD, we set the maximum number of iterations for one route from 20000 to 60000, while this maximum is set from 5000 to 50000 for the instances ETAO and ESAR. The instance ZTBR could not be solved satisfactorily within 20 minutes of computation time. This approach resulted in some solutions for the instance ZSAD. If 2-OPT is not applied, computation time increased from 0.1 second to 15.6 seconds for a similar solution. Applying the heuristic once for this instance costs 726.2 seconds compared to 27.2 for the original instance. Applying the heuristic twice for this instance did not result in a solution within 20 minutes of computation time.

Concerning the instances at station Enschede, the original instances ETAO and ESAR are all solved within 3 seconds. For 0, 1, and 2 applications of 2-OPT the computation times for the ETAO instances with $m(n) \equiv 0$ are 1.3, 14.5 and 27.0 seconds, respectively. Similarly, the computation times for the ESAR instances rose in this case to 0.8, 2.8 and 4.8 seconds.

Note that the instances ZSAD and ESAR are not reported in Tables 5.1 or 5.2. The interested reader is referred to Appendix A.1 for the details concerning these instances. These computational results show that good estimates $m(.)$ of future routing cost enormously improve the speed of ONA* Search.

Figure 5.7 shows the gaps for the Saturday / Sunday instances at station Zwolle for 0, 1, and 2 applications of 2-OPT. Note that these gaps are smaller than the gaps reported in Tables 5.1 and 5.2. The gaps show a similar structure as the gaps in Tables 5.1

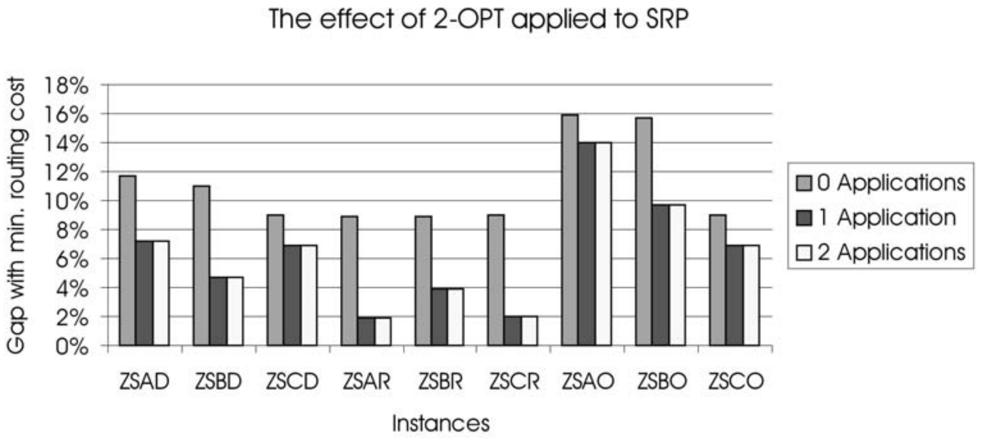


Figure 5.7: The effect of 2-OPT on the gaps for the ZS.. instances.

and 5.2: the first application of 2-OPT is very useful, while the second application only adds marginal value, if any at all.

5.8 SUPPORTING THE PLANNER

In order to be able to discuss results of algorithms supporting the planning of shunting processes and to enable planners to use the algorithms, a prototype support system was developed. In this paragraph, we briefly discuss some of the functionalities of the prototype regarding the routing process. After a discussion of a visualization of the routes, we describe the control of the planner over the parameters for the routing algorithm as introduced in Section 5.5.

When a solution to SRP has been found, a planner typically inspects this solution to see if it matches his expectations. One way to do this is by graphs like the one in Figure 5.8. Given a route r, the graph shows for each track or switch, the earliest successor of r and latest predecessor at a track or switch. Note that these successors and predecessors can be different for different tracks or switches. With r_s we denote the route with minimum time difference between r and its successor at any shared node. Moreover, r_p is the route with the minimum time difference between r and its predecessor at any shared node. The route itself is depicted on the horizontal axis, representing the different parts of the infrastructure. The vertical axis represents time.

The panel above the graph shows textual information of a selected route r, containing start and end times and locations, the involved train service, and the distance, among others. The upper panel on the right gives information on the successor route r_s of

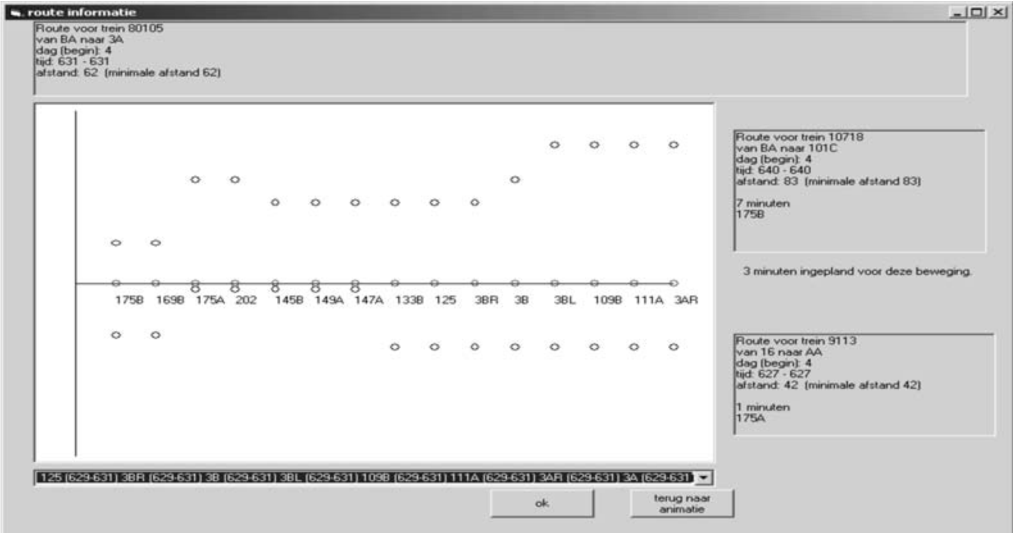


Figure 5.8: A graph depicting the amount of flexibility for a route.

route r. This contains the same information as the information of route r extended with information on the point where r_s is closest in time to r. The lower panel on the right shows similar information for the predecessor route r_p of the route r.

The graph can be used for changing the timing of a route request. Moreover, such graphs can also be used to determine the robustness of a set of routes.

As mentioned in Section 2.3.3, a shunt planner has many parameters for influencing the solution process of SRP. These parameters include the duration of dwell times along platforms, the buffer time between two routes, the minimum time for changing direction for different trains, a penalty for simultaneous routes, and more. Although most of the planning norms are changed rarely, these norms might be too restrictive to route all requests, as can be concluded from Tables 5.1 and 5.2. In such cases, planning in every detail might not be very useful. This is caused by the fact that during the operations changes are inevitable, and therefore a completely detailed plan needs real-time alterations. Alternatively, a planner might change the planning norms, or make small modifications to the requests in order to find routes for requests.

5.9 CONCLUSIONS

This chapter introduced the Shunt Routing Problem (SRP), which is to find shunt routes not conflicting with already planned infrastructure reservations for e.g. through trains and other shunt routes. An important characteristic of SRP is that the start and end times of the shunt routes are flexible to some extent. When searching for such routes, important objectives are the traveled distance, the number of changes in direction and the deviations from the preferred start times. Obviously, a planner needs to find routes for all requests. Therefore, this is the most important objective for an algorithm to support him. After the introduction of SRP, we studied several related railway routing problems in order to determine distinctive characteristics of SRP and to present some proposed solution methodologies.

The routing problem SRP requires a specific model of the station infrastructure in order to detect all possible conflicts between routes. This model has been introduced in Section 5.3. After a discussion of state-space search algorithms and A* Search in particular, we proposed Occupied Network A* (ONA*) Search as an extension of A* Search for finding a route for each route request. ONA* Search is applied sequentially and therefore the quality of the set of routes is largely dependent on the order of finding routes. In order to decrease the influence of this order, a 2-OPT procedure has been applied to improve the quality of the overall set of routes.

The computational results show that the sequential application of ONA* Search is fast. Interaction with shunt planners is explicitly required for the Tuesday / Wednesday instances at station Zwolle, since the results leave several requests unplanned. However, most of the requests of the Tuesday / Wednesday instances at station Enschede can be routed. One application of 2-OPT results in a significant improvement in the quality of the set of routes for the instances at station Zwolle. A second application of the heuristic at station Zwolle and a first application of it at station Enschede result in marginal improvements at most. This might be caused by the fact that 2-OPT is unable to leave a local optimum. More advanced interchange heuristics, e.g. based on Lin and Kernighan [1973] might provide better results at the cost of additional computation time. However, our main point is that the combination of ONA* Search with an interchange heuristic forms a solid basis for a tool, supporting shunt planners in solving SRP. We expect that such heuristics are especially effective for instances at large stations, with many route requests, which overlap in time.

Note that in many cases not all route requests can be assigned a route. In such cases, planners are required to make small modifications to e.g. the start times of the routes to assign more routes to requests. Alternatively, planners can change the norms of route planning in order to increase the solution space. We briefly touched upon decision support of the planner in these cases. This support contains graphical visualization of the routes and the time difference between a route and its successor route. Moreover, a variety of parameters is available to shunt planners to tune the algorithm to their needs.

In conclusion, this chapter introduced an algorithm for supporting planners in finding routes for route requests. In the planning process of finding shunt routes, planners are required to use their knowledge of the infrastructure and the shunting process and to interact with the algorithm to provide the best results.

Chapter 6

Cleaning of Train Units

Cleaning of train units is important since it has a large influence on the quality of the services offered to the passengers, as well as the perceived quality. Therefore, providing clean rolling stock and stations is one of the main objectives of Dutch Railways. The cleaning of rolling stock falls apart in the cleaning of the interior of rolling stock, and the cleaning of its exterior. In this chapter we study the internal cleaning of train units.

Internal cleaning of rolling stock is often combined with external cleaning and maintenance checks, which are also carried out locally at a station. Maintenance checks of rolling stock typically need to be performed at dedicated tracks. At NSR, preventive maintenance checks are due every 48 hours. Moreover, external cleaning takes place at a train-wash and is also due every 48 hours in general. Note that the planning period of shunting is typically 24 hours. This planning period results in a discrepancy with the intervals of maintenance checks and external cleaning. Therefore, approximately half of the rolling stock that lays over at a shunt yard requires external cleaning, and also approximately half requires a maintenance check. Thus, planning these processes requires additional input characteristics of physical train units.

Internal cleaning is carried out every night at most shunt yards, typically in a 12-hour planning period starting at 18:00. In this period, each train unit is cleaned internally. Internal cleaning typically takes place along dedicated platforms, which are available at a restricted number of stations.

As described in Section 2.3.4, different types of internal cleaning of rolling stock are:

- Cleaning at the end of a passenger line.
- Modular cleaning.
- Urgent cleaning.
- Periodic thorough cleaning.

Time for cleaning rolling stock at the end of a passenger line has been incorporated in the timetable. Time for urgent cleaning is not explicitly accounted for in the operational plans. Moreover, periodic thorough cleaning is also disregarded in these operational plans. However, modular cleaning is considered in the operational plans.

In modular cleaning, different cleaning activities are divided over several modules, each module with its own prescribed frequency and duration. Typically, in one modular cleaning session a basic cleaning module and one additional module are performed. Internal cleaning norms for different types of rolling stock are available. These internal cleaning norms represent the average number of man-minutes required for a basic cleaning module and one additional module.

A typical characteristic of internal cleaning is that the corresponding activities are performed by shifts of cleaning crews with different working hours. Since these shifts generally overlap in time, and the numbers of crews in each shift differ, the required time for cleaning train units varies during the planning period.

After an in-depth introduction of the most important characteristics, we define the Shunt Unit Cleaning Problem, and we introduce an example based on the real-life situation at station Zwolle. Moreover, an overview of fields related to the cleaning problem is presented. The main purpose of this overview is to discuss some backgrounds on this cleaning problem. Subsequently, a model is developed and discussed, and we study some computational properties of the problem. Computational experiments on real-life instances are reported. The fact that the cleaning process takes place at dedicated tracks implies that additional routing is required. In addition, it results in increased flexibility since the process allows for changes in the order of the train units parked at the shunt tracks. By considering the cleaning subproblem, we capture some of the dynamics at a shunt yard at night. The effect of the increased flexibility of the parking subproblem will be discussed in Section 6.6. Finally, the chapter ends with some conclusions.

6.1 INTRODUCTION TO CLEANING

In addition to the main assumptions of this thesis, as described in Section 1.10, the following assumptions are made in this chapter to treat the cleaning problem in an adequate way:

- Although maintenance checks and the external cleaning activities are also relevant in practice, we decided after a discussion with shunt planners that supporting the scheduling of internal cleaning activities has the first priority.

- The only internal cleaning activities that are relevant for the operational shunt planners result from the modular cleaning.

- In the Netherlands, NSR outsources the cleaning of train units to NedTrain. However, NSR typically plans a large part of the process because it is tightly related with the operations of NSR.

- Logistics regarding the cleaning crews are disregarded by NSR. Cleaning crews are planned by NedTrain, and the results serve as an input for NSR. These results typically consist of the structure of the shifts and the number of cleaning crews in each shift.

- The matching of arriving to departing train units is known. Therefore, we know the arrival and departures times of blocks (see Appendix A.1), which provide release times and deadlines for internal cleaning activities of each block. Moreover, we consider the blocks fixed. In theory, the configurations of the blocks can change but shunt planners rarely make use of this option in practice.

- We consider the blocks as the entities to be planned. This might result in a discrepancy with current practice. In practice, train units arriving in the same train, but assigned to different departing trains, are likely to be cleaned together.

- Typically, a station has one cleaning platform with two cleaning tracks along it. We assume that at most one cleaning track at a time is used for cleaning rolling stock. The other track is used for routing blocks to and from the tracks along the platform. When a block is clean, the cleaning crews move to the other side of the platform and start cleaning the train parked at that track. Shunting drivers subsequently move the cleaned train away from the cleaning platform, and park a new train there.

At NedTrain, the cleaning activities are carried out by two shifts of cleaning crews. Typically, the late shift starts at 18:00 and works until 02:30, with a break scheduled in between, while the night shift starts at 22:00 and works until 06:30. This means that there is some overlap in the shifts. An example of the availability of cleaning crews over time is depicted in Figure 6.1. Here, 6 crews start working at 18:00 in the late shift, and their break is scheduled from 21:30 to 22:00. The night shift consists of 6 crews and starts at 22:00. Therefore, from 22:00 12 crews are available. At 1:30, the 30-minute break of the night shift is scheduled, leaving 6 crews of the late shift working. At 2:00, the crews in the late shift finish, while the crews in the night shift start working again until 6:00.

As mentioned before, the processing time for cleaning a train unit is time dependent, caused by the different numbers of crews available for cleaning in different times of the planning period and the start and end times of the shifts. For example, suppose an ICM_4 unit needs to be cleaned, and the crew availability is modeled by Figure 6.1. Then, scheduling the job in one of the intervals [18:00, 21:30] or [1:30, 6:00] will require

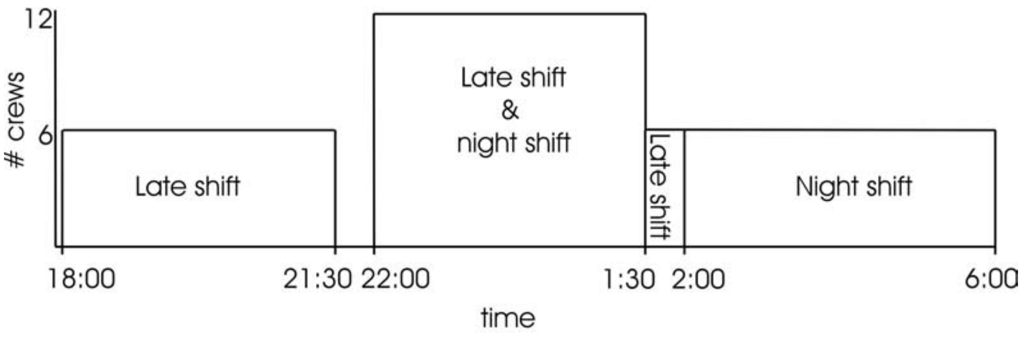


Figure 6.1: An example of cleaning crew availability.

in general twice as much time as scheduling the train unit in the interval [22:00, 1:30], since twice as many crews are available in this latter interval. In Section 6.4, we return to this aspect and treat it in detail.

For each block that needs cleaning, three options are available:

1. It is cleaned shortly after arrival at the station. This implies that the block needs to be parked after it has been cleaned until it leaves the station in the next morning.

2. It is first parked at a shunt track, then routed to a cleaning track where it will be cleaned. Thereafter it is rerouted to a (possibly different) shunt track, and finally it is routed to its departure platform just before departure.

3. It is first parked at a shunt track, then routed to a cleaning track, and after it has been cleaned it is routed to the departure platform.

The last situation is undesirable, because it conflicts with the overall goal of shunting, which is to start up the railway operations in the morning as smoothly as possible. As one can imagine, the first option is preferred over the second one, since train units will be parked only once, which results in less work for the crews that carry out the tasks resulting from the shunting process. Moreover, the first option results in reduced complexity of the shunting operations. Finally, the second option results in more produced noise during the night, which is undesirable since this disturbs people living around the shunt yard. Therefore, our objective is to clean as many blocks as possible “close” in time to their arrival at the station, for an appropriate definition of “close”.

Before defining the problem in detail, we introduce some notation. $\mathcal{B}$ is used to denote the set of blocks that need cleaning. With a_b we denote the release time of job (or block) $b \in \mathcal{B}$, q_b is the deadline of job (or block) b, and p_b indicates the required amount of work on job b in man-minutes. Moreover, the parameter z is the number of minutes after the arrival of a block, which is still considered “close” in time to its release time. $\mathcal{M}$ is the set of relevant points in time. Typically, cleaning takes place during a period of 12 hours (see Figure 6.1) and is discretized on a one-minute basis. Therefore $|\mathcal{M}| \approx 720$. Finally, $c(m)$ gives the number of cleaning crews available at time $m \in \mathcal{M}$.

Definition 6.1. Given is a set of blocks $\mathcal{B}$, where each block $b \in \mathcal{B}$ is assigned a release time a_b, a deadline q_b and an amount of work p_b, a parameter z, and a function $c(m)$ representing the number of man-minutes available at each point in time $m \in \mathcal{M}$. The Shunt Unit Cleaning Problem (SUCP) is to assign p_b consecutive man-minutes to each block $b \in \mathcal{B}$ within $[a_b, q_b]$, while maximizing the number of jobs that start cleaning in $[a_b, a_b + z]$, and $c(m)$ man-minutes are available at each point $m \in \mathcal{M}$.

SUCP belongs to the class of machine scheduling problems. In this class, one or more machines need to operate a set of jobs, and the problem is to find a schedule of the jobs optimizing some objective. Lawler et al. [1993] distinguish the following important characteristics of a machine scheduling problem:

- The number of available machines and their relations.
- Whether or not preemption of jobs is allowed.
- The existence of certain types of precedence relationships between jobs.
- The existence of release dates and due dates of jobs.
- The processing times of the jobs.
- The optimization criterium.

In general, important objectives are finding the earliest completion time of a schedule, a minimum total lateness, and the minimum total time in the system. This last criterium is also called the minimum total flow time.

In machine scheduling terms, the blocks are equivalent with jobs and the set of cleaning crews represents the machine. The processing times differ over time since the number of cleaning crews fluctuates over time. Moreover, preemption of jobs is not allowed. If a job is not started within z minutes after its release time, we set an indicator to 1, otherwise it equals 0. Thus, the indicator is set when the flow time of a job exceeds a certain threshold. The objective is to minimize the sum of these indicators. In Section 6.3.3, we pay some more attention to this field of research.

6.2 EXAMPLE OF A CLEANING SCHEDULE

In this paragraph, we discuss an example of a cleaning schedule. This schedule is based on a real-life situation at station Zwolle. An overview of the station infrastructure at Zwolle is depicted in Figure A.2 on page 203, while Figure A.3 on page 203 is more detailed. At station Zwolle, the cleaning tracks are tracks 90 and 91. Moreover, we consider the availability of the cleaning crews as given in Figure 6.1.

The different types of rolling stock in the example are enumerated in Table 6.1. This table also contains the prescribed time in man-minutes for cleaning a certain type of rolling stock. We also indicate whether the type is single-deck or double-deck rolling stock, since double-deck units require more time to clean than single-deck units.

Type	Description	Norm (in man-minutes)
DH_2	Single-deck	52
DM90_2	Single-deck	55
SM90_2	Single-deck	55
MAT64_2	Single-deck	60
MAT64_4	Single-deck	118
ICM_3	Single-deck	127
ICM_4	Single-deck	151
DD-AR_3	Double-deck	199
MDDM_4	Double-deck	229
IRM_3	Double-deck	233
DD-AR_4	Double-deck	262
IRM_4	Double-deck	317

Table 6.1: Internal cleaning norms for the types of rolling stock in our main example.

After the arrival of a block, it requires some pre-processing time before the block can be cleaned. Similarly, some post-processing time is required after the cleaning of the block. After consultation of the shunt planners, we set the pre-processing time for each block to 13 minutes and the post-processing time to 30 minutes. Finally, we consider 5 minutes still close in time to the release time of a block, i.e. z is set to 5 minutes.

Table 6.2 gives an example of a cleaning schedule. In this table, the first column gives the track at which the cleaning activities take place. Columns 2 and 3 describe the start and end times of the cleaning respectively. If the cleaning of a block starts within 5 minutes of its release time, then the indication (*) is added after the start time in the second column. The fourth column shows the number of cleaning crews scheduled to clean a block. Columns 5 and 6 give some information on the release times and deadlines of the blocks. In these columns the arrival and departure times of the blocks are given within parentheses, while the actual release times and deadlines for cleaning include the pre-processing respectively post-processing time. The last column gives the configuration of the block. Note that the ICM_4 and ICM_3 units that are cleaned from 3:50 to 4:37 do not leave the shunt yard at the end of the planning period.

Track	Start time	End time	Crews	Block release	Block deadline	Block configuration
	18:00	START LATE SHIFT: 6 crews				
	18:00	START CLEANING				
90	18:00 (*)	18:09	6	17:57 (17:44)	5:14 (5:44)	DH_2
91	18:26 (*)	19:19	6	18:26 (18:13)	5:50 (6:20)	IRM_4
90	19:41	20:01	6	18:46 (18:33)	6:19 (6:49)	MAT64_4
91	20:03 (*)	20:13	6	19:59 (19:46)	5:44 (6:14)	MAT64_2
90	20:23	21:02	6	18:22 (18:09)	6:21 (6:51)	MDDM_4
91	21:03 (*)	21:25	6	20:59 (20:46)	5:22 (5:52)	ICM_3
	21:30	START BREAK LATE SHIFT: 30 minutes				
	22:00	END BREAK LATE SHIFT: 30 minutes				
	22:00	START NIGHT SHIFT: 6 crews				
90	22:00	22:22	12	20:22 (20:09)	4:51 (5:21)	DD-AR_4
91	22:25 (*)	22:42	12	22:22 (22:09)	5:21 (5:51)	DD-AR_3
90	22:45	22:56	12	19:28 (19:15)	7:22 (7:52)	ICM_3
91	22:57 (*)	23:02	12	22:55 (22:42)	6:21 (6:51)	DM90_2
90	23:03 (*)	23:14	12	22:59 (22:46)	5:48 (6:18)	ICM_3
91	23:22 (*)	23:41	12	23:22 (23:09)	5:51 (6:21)	MDDM_4
90	23:42	23:52	12	13:07 (12:54)	5:44 (6:14)	MAT64_2 MAT64_2
91	23:53 (*)	23:58	12	23:53 (23:40)	6:56 (7:26)	MAT64_2
90	23:59 (*)	0:22	12	23:59 (23:46)	5:18 (5:48)	ICM_3 ICM_4
91	0:23 (*)	0:28	12	0:22 (0:09)	5:57 (6:27)	MAT64_2
90	0:29 (*)	0:34	12	0:25 (0:12)	6:21 (6:51)	DM90_2
91	0:35 (*)	0:39	12	0:35 (0:22)	5:14 (5:44)	DH_2
90	0:41 (*)	0:52	12	0:37 (0:24)	5:48 (6:18)	ICM_3
91	0:58 (*)	1:03	12	0:57 (0:44)	5:26 (5:56)	MAT64_2
90	1:04 (*)	1:15	12	1:00 (0:47)	5:48 (6:18)	ICM_3
91	1:16	1:26	12	23:22 (23:09)	4:50 (5:20)	MAT64_4
90	1:30	1:52	6	0:28 (0:15)	5:47 (6:17)	ICM_3
	1:30	START BREAK NIGHT SHIFT: 30 minutes				
	2:00	END BREAK NIGHT SHIFT: 30 minutes				
	2:00	END LATE SHIFT: 6 crews				
91	2:00	2:53	6	0:22 (0:09)	5:50 (6:20)	IRM_4
90	2:55	3:05	6	23:07 (22:54)	6:22 (6:52)	SM90_2
91	3:10	3:49	6	18:53 (18:40)	6:21 (6:51)	MDDM_4
90	3:50	4:37	6	23:58 (23:45)	—	ICM_4 ICM_3
91	4:38	5:04	6	0:28 (0:15)	6:45 (7:15)	ICM_4

90	5:05	5:15	6	0:57 (0:44)	6:16 (6:46)	MAT64_2
91	5:17	6:00	6	0:01 (23:48)	6:49 (7:19)	ICM_3 ICM_3
	6:00	END NIGHT SHIFT: 6 crews				

Table 6.2: An example of a cleaning schedule.

In this example, 16 out of 30 jobs are cleaned "close" to their release times. General internal cleaning schedules are similar in structure to this specific cleaning schedule: early in the evening the cleaning is started, and some blocks that lay over are already at the shunt yard and can be cleaned. After the afternoon rush hours, blocks that lay over at the yard arrive faster than the cleaning crews can clean the blocks. This results in a backlog of blocks that need cleaning and this backlog is scheduled during the night, since no blocks arrive after 2:00.

6.3 CLEANING AND RELATED PROBLEMS

In this paragraph, several developments related to cleaning of rolling stock are discussed to introduce relevant practical aspects and to position SUCP from a scientific point of view. These developments fall apart in practical developments, research on maintenance planning of rolling stock, and research on machine scheduling problems.

6.3.1 Practical Aspects

Efficient planning of the resources available for cleaning and maintenance of rolling stock is important. Indeed, inefficient resource usage might lead to additional required infrastructure, such as specialized tracks and platforms for cleaning and maintenance. Masse [2005] reports a cost of €220 million for rebuilding a cleaning and maintenance facility near Paris dedicated to TGV trains. This indicates that such tracks need to be used as efficiently as possible, since expansions are expensive.

In practice, an important development is the transfer of rolling stock maintenance from passenger operators to rolling stock manufacturers. Currently, the passenger operators are typically responsible for the maintenance of their rolling stock. These operators might outsource this maintenance to others. For instance, NSR outsources the rolling stock maintenance to NedTrain. Two of the largest suppliers of rolling stock in the world, Siemens and Bombardier, include optional maintenance of rolling stock in their products [Knutton, 2003; International Railway Journal, 2003]. According to Veenstra et al. [2005], similar developments on a larger scale might occur at the port of Rotterdam for providing additional services to maritime freight transport operators. Therefore, manufacturers providing additional services seem to have the future.

Moreover, the current state of the art rolling stock contains vehicle information systems, which are able to perform certain maintenance checks and communicate the results in real-time [Railway Gazette International, 2005]. Similar information systems are becoming available in many industries [Van Nunen and Zuidwijk, 2004; Lee, 2003]. This availability leads to better information which is faster available. Therefore, maintenance activities can be improved resulting in increased availability of the rolling stock.

Also improvements in the cleaning products ensure faster and easier cleaning of rolling stock. Finally, due to technical innovations in the design of rolling stock, modern rolling stock needs less maintenance than old rolling stock.

6.3.2 Related Maintenance Planning Problems

One strategy that is applied for the maintenance of rolling stock at NSR is condition based maintenance (see e.g. Waeyenbergh and Pintelon [2002] for an overview of strategies). In this strategy, preventive maintenance checks reveal the condition of the rolling stock. Based on the results of these checks, preventive or even corrective maintenance can be performed.

At NSR, preventive maintenance checks for railway rolling stock are due every 48 hours. In addition, capacity for maintenance activities is reserved in the rolling stock circulation and therefore maintenance tasks end up in the duties of the rolling stock. Maintenance rolling stock planners interchange rolling stock duties in such a way that the units that need maintenance in the next couple of days arrive in time in a maintenance facility. However, such interchanges might require adjustments in the shunt plans at stations. Therefore, local shunt planners need to assess whether or not such adjustments are possible. Since the communication between maintenance rolling stock planners and shunt planners takes time, only a limited number of such interchanges can be evaluated.

The urgency of maintenance checks differs from the urgency of cleaning rolling stock. Regarding maintenance checks, regulations prohibit rolling stock to be used for passenger service whenever their maintenance checks are tardy. However, NSR prefers dirty rolling stock in punctual service over clean rolling stock resulting in delays. Therefore cleaning of rolling stock can be delayed to some extent.

Maróti and Kroon [2004] introduce the Interchange Model for maintenance routing. They consider several "changing scenarios". Such a changing scenario consists of a set of interchanges of duties, for which the resulting shunting processes can be carried out. The goal is to select a set of pairwise independent interchanges, such that the urgent units receive their maintenance in time. The authors developed a heuristic as well as an exact formulation for this problem. Real-life instances with randomly generated urgent units are solved satisfactorily by both algorithms. Typically, the heuristic requires at most 5 seconds for a solution within approximately 2.5% of the optimum. The Integer Program is solved to optimality by CPLEX within 90 seconds.

The drawback of the Interchange Model is that a lot of detailed input data are required for the model, mainly caused by the sets of feasible interchanges. In a subsequent paper, Maróti and Kroon [2005] introduce the Transition Model, which requires significantly less input. In this model, the shunting effort of interchanging two duties is taken into account by requiring sufficient time for the shunting operations to be performed. Note that this is a relaxation, since it is not known whether or not a set of interchanges results in a feasible shunt plan. For solving the Transition Model, an Integer Program has been developed. Typically, less than 10 seconds are required for solving real-life instances to optimality with randomly generated urgent train units. An extensive discussion on these topics can be found in Maróti [2006].

Haghani and Shafahi [2002] discuss the scheduling of preventive bus maintenance inspections at a bus depot. Given a daily operating timetable and maintenance resources, the objective is to schedule all buses due for inspection, while minimizing the disruption of the operational schedule and maximizing the utilization of the maintenance facilities. A real-life instance is provided by the Mass Transit Authority in Baltimore, United States. This instance consists of 181 buses, while at most 12 buses can be inspected simultaneously. In addition, 5 types of inspections exist, which are due with different mileage intervals. Although this problem is difficult to solve to optimality, heuristics perform well. In a simulation of 182 days of the operations, the proposed heuristics improve the current manual planning, while requiring an average of 40 seconds of computation time per day.

6.3.3 Related Machine Scheduling Problems

An overview of algorithms and complexity of deterministic machine scheduling problems is presented in Lawler et al. [1993]. The popularity of this field of research is illustrated by the fact that this survey contains over 375 citations. Many complexity results in this survey result from Lenstra et al. [1977]. It treats among others Single Machine Scheduling Problems (SMSPs), including topics like minmax objectives (e.g. minimizing the maximum lateness) and weighted objectives (e.g. minimizing total weighted completion time). Moreover, the authors also discuss parallel machine scheduling problems for different configurations (including identical machines and unrelated machines) and multi-operation models, where each job requires execution on more than one machine.

More specifically, deterministic SMSPs belong to the classical problems of Operations Research, finding their roots in the 1950's. A review and classification of research on SMSPs can be found in Gupta and Kyparisis [1987]. By no means, we intend to give an exhaustive overview of the literature on machine scheduling problems here. However, we discuss some literature on machine scheduling problems which resemble

SUCP. The overview contains scientific work on SMSPs with varying machine capacity, which resembles the varying availability of cleaning crews.

An interesting stream of research focuses on SMSPs, where a trade-off between overtime of the machine and detailed scheduling costs is considered. By scheduling overtime at a certain cost, additional capacity becomes available for the machine. In this case, the capacity of the machine varies, but explicit decisions on this capacity are made. Gelders and Kleindorfer [1973] develop several lower bounds on the total costs for a variant of this problem. Some of these lower bounds are useful in a branch-and-bound algorithm. In a companion paper, Gelders and Kleindorfer [1975] introduce a detailed branch-and-bound algorithm using the previously mentioned lower bounds. Their proposed algorithm is able to handle non-identical release dates of jobs. Computational results are reported for instances with 10 and 15 jobs, which required up to several minutes on a computer of that time.

Moreover, Vickson [1980] introduces rules for expediting jobs in order to minimize the total weighted flow cost plus job-processing cost on a single machine. A heuristic based on an advanced priority rule is shown to be able to find optimal solutions for most tested instances.

For a similar SMSP with controllable processing times, Daniels and Sarin [1989] study characteristics of non-dominated solutions both in the number of tardy jobs and the amount of allocated resources. A solution is called non-dominated if it is impossible to improve on one of the two objectives without deteriorating the other.

Hirayama and Kijima [1992] show that, under some assumptions, several optimal priority rules in deterministic SMSPs are also optimal in stochastic SMSPs, where the machine capacity varies stochastically. The authors consider a general stochastic machine capacity as well as machine breakdown models where the capacity at a point in time is either 0 or 1.

To our knowledge, Baker and Nuttle [1980] are the first to consider a given varying capacity of a single machine. Consider the objective of minimizing a function of the job completion times. In this case, certain results for problems consisting of a machine with a fixed capacity can be extended to problems with a machine with a given variable capacity.

In a setting where jobs require operations on different machines and machines with varying capacities, Adiri and Hamberg [1998] provide an extensive characterization of the computational complexity of many SMSPs in this class.

Amaddeo et al. [1997] introduce a problem similar to SUCP occurring at KLM Baggage Handling. Here, luggage from flights to Amsterdam needs to be handled by shifts of crews. In different shifts, different numbers of crews are available and breaks within a shift need to be respected. The objective in this problem is to minimize the sum of weighted completion times. The authors prove that their problem is $\mathcal{NP}$-complete.

Dyer and Wolsey [1990] propose a time-indexed formulation for SMSPs as a result of a comparison of different integer programming formulations of SMSPs, where the objective is to minimize the sum of weighted start times of jobs. Subsequently, Van den Akker et al. [2000] generalize the objective and introduce a column generation approach for this time-indexed formulation, which is the strongest formulation in Dyer and Wolsey [1990]. Since this formulation requires many variables, column generation is applied. In this approach, the restricted master problem ensures that each job is started exactly once. The pricing problem generates so-called pseudo-schedules where the capacity restrictions are met, but the number of times a job is started is not restricted. The authors report computational results of solving the LP-relaxation of the formulation with instances consisting of 20 and 30 jobs. These results indicate that the column generation approach is faster than CPLEX in case of relatively long processing times. When the processing times are relatively short, CPLEX outperforms the column generation technique in terms of computation time.

Van den Akker et al. [1999] study the polyhedron of the time-indexed formulation of Van den Akker et al. [2000], resulting in sets of valid inequalities. Such inequalities are subsequently embedded in a branch-and-cut algorithm, where these inequalities are generated at different nodes in the branch-and-bound tree. Although the original LP-relaxation of this formulation is already quite strong, the authors are able to improve significantly upon the quality of this relaxation by adding valid inequalities, resulting in reduced computation times. This enables them to solve many instances without branching. However, for instances with many jobs and / or large processing times, solving the LP-relaxation still requires a prohibitive amount of computation time.

An interesting direction of research is the combination of the column generation approach of Van den Akker et al. [2000] with cut generation based on the results in Van den Akker et al. [1999]. However, so far this has led to disappointing computational results [Van den Akker et al., 2000].

6.4 MODEL AND THEORETICAL ASPECTS

Before introducing the developed model and the computational complexity, we make the following assumptions, which are valid for the remainder of this chapter:

- Time is discretized by dividing the planning horizon in periods of 1 minute. The cleaning of a block can only start at the beginning of a 1 minute period.
- At one point in time, the cleaning crews can only clean one block. Suppose a block requires one man-minute of processing time and one shift of two cleaning crews is available. Cleaning the block requires half of the man-minutes available in one minute. However, the other man-minute in this point of time is lost.

- More crews assigned to the cleaning of a block typically results in lower processing times. However, for example the filling of the water tanks is in principle not influenced by the number of crews. Therefore, cleaning a block might require a minimum amount of time. Nevertheless, we do not consider such fixed minimum processing time for cleaning rolling stock.

As mentioned in Section 6.1, the cleaning process can be formulated as a SMSP. In this formulation, the blocks are the jobs to be scheduled and the cleaning crews represent the machine. The speeds of the machine are given because they can be derived from the number of crews in each shift and the start and end times of these shifts, which are input data. Before describing the model, we introduce $p_{b,m}$ as the processing time of job $b \in \mathcal{B}$ in order to be ready precisely at time $m \in \mathcal{M}$. We have

$$p_{b,m} = m - \max\{\tilde{m} \in \mathcal{M} \mid \sum_{\hat{m}=\tilde{m}}^{m} c(\hat{m}) \geq p_b\}$$

Moreover, the set of decision variables is defined by:

$$R_{b,m} = \begin{cases} 1 & \text{if job } b \in \mathcal{B} \text{ starts cleaning at time } m \in \mathcal{M}; \\ 0 & \text{otherwise.} \end{cases}$$

Since preemption of jobs is not allowed, the processing of a job will not be interrupted by the processing of another job, nor by a break of a shift of cleaning crews. In case a shift of crews stop working for their break but a different shift continues working, this is not seen as an interruption of the processing of a job. However, in case the crews in a shift take their break and no other shift is working, it is not allowed to leave a partially processed block at the cleaning track. This leads to some decision variables, which can be set to 0 in advance:

$$R_{b,m-p_{b,m}} = 0 \quad \forall b \in \mathcal{B}, m \in \mathcal{M} : \min\{c(\tilde{m}) \mid \tilde{m} \in \{m - p_{b,m}, \ldots, m\}\} = 0$$

Now, the model is stated as

$$\text{maximize} \quad \sum_{b=1}^{|\mathcal{B}|} \sum_{m=a_b}^{a_b+z} R_{b,m} \tag{6.1}$$

$$\text{subject to} \quad \sum_{m=a_b}^{q_b-p_{bq_b}} R_{b,m} = 1 \qquad b = 1, ..., |\mathcal{B}| \tag{6.2}$$

$$\sum_{b=1}^{|\mathcal{B}|} \sum_{m=\tilde{m}-p_{b,\tilde{m}}}^{\tilde{m}} R_{b,m} \leq 1 \qquad \tilde{m} = 1, ..., |\mathcal{M}| \tag{6.3}$$

$$R_{b,m} \in \{0,1\} \quad b = 1, ..., |\mathcal{B}|, m = 1, ..., |\mathcal{M}| \tag{6.4}$$

Our objective (6.1) is to clean as many blocks as possible close to their arrival times. Restrictions (6.2) state that each block needs to start cleaning at exactly one point in time. Together with restrictions (6.2), the inner sum of restrictions (6.3), $\sum_{m=\tilde{m}-p_{b,\tilde{m}}}^{\tilde{m}} R_{b,m}$, indicates whether or not the cleaning crews are working on job b at time $\tilde{m}$. Therefore, restrictions (6.3) prohibit working on more than one block at the same time. Note that the different operating speeds of the machine are represented by the parameters $p_{b,m}$. We call (6.1) - (6.4) Model (6.a).

Before discussing computational results of this model, we give two results on the computational complexity of SUCP. We start with a proof that SUCP is $\mathcal{NP}$-complete in the strong sense. This is followed by a proof that a special case of SUCP can be solved in pseudo-polynomial time, and therefore this special case is weakly $\mathcal{NP}$-complete. Based on SUCP, we define the decision problem SUCP-∞ as follows:

Definition 6.2. Given is a set of blocks $\mathcal{B}$, where each block $b \in \mathcal{B}$ is assigned a release time a_b, a deadline q_b and an amount of work p_b, and $c(m)$ represents the capacity of the cleaning crews at each point $m \in \mathcal{M}$ in time. The SUCP-∞ is to decide whether it is possible to assign p_b consecutive man-minutes to each block $b \in \mathcal{B}$ within $[a_b, q_b]$.

The difference between SUCP and SUCP-∞ lies in the objectives. While in SUCP the objective is to schedule a maximum number of jobs close to their release times, in SUCP-∞ one is only concerned with finding a feasible solution. We will use the Sequencing within Intervals Problem (SIP) to prove that SUCP-∞ is $\mathcal{NP}$-complete in the strong sense. SIP is defined in Garey and Johnson [1979] as:

Definition 6.3. Given a set $\mathcal{B}$ of tasks, each task $b \in \mathcal{B}$ having a length p_b, and a time interval $[a_b, q_b]$ within which b is to be executed, the Sequencing within Intervals Problem (SIP) is to determine whether the tasks can be sequenced to obey these restrictions, with at most one task ever being executed at a time.

Lenstra et al. [1977] proved that a special case of SIP without due dates is $\mathcal{NP}$-complete in the strong sense. In order to prove that SUCP-∞ is $\mathcal{NP}$-complete in the strong sense, we need a lemma from Garey and Johnson [1979]:

Lemma 6.4. *(Lemma 4.1 in* Garey and Johnson [1979]) If decision problem Π is $\mathcal{NP}$-complete in the strong sense, decision problem Π' belongs to $\mathcal{NP}$, and there exists a pseudo-polynomial transformation from Π to Π', then Π' is $\mathcal{NP}$-complete in the strong sense as well.

Proof. See Garey and Johnson [1979]. ■

Note that Lemma 6.4 uses the term "pseudo-polynomial transformation", which is not properly defined here. The interested reader is referred to Garey and Johnson

[1979] for a precise definition. Loosely speaking, however, a pseudo-polynomial transformation from Π to Π' is a transformation of an instance I of Π to an instance I' of Π', such that I is a yes-instance of Π if and only if I' is a yes-instance of Π'. Moreover, the transformation can be done in amounts of time and space that are polynomial in the length and the largest number of the instance I.

Now, we are ready to prove $\mathcal{NP}$-completeness of SUCP-∞, which is done via a reduction of a special case of SUCP-∞ from SIP.

Lemma 6.5. *SUCP-∞ is $\mathcal{NP}$-complete in the strong sense.*

Proof. For a given schedule, it is trivial to decide whether or not all jobs $b \in \mathcal{B}$ are scheduled within $[a_b, q_b]$ and therefore SUCP-$\infty \in \mathcal{NP}$.

Given an instance I of SIP, an instance I' of SUCP-∞ is constructed by adding the resource capacity function $c_m \equiv 1$ for all $m \in \mathcal{M}$. It is obvious that I' is a yes-instance of SUCP-∞ if and only if I is a yes-instance of SIP.

It is trivial to see that the transformation from I to I' can be done in pseudo-polynomial time and space. Therefore SUCP-∞ is $\mathcal{NP}$-complete in the strong sense, since SIP is $\mathcal{NP}$-complete in the strong sense as well. ■

Corollary 6.6. *SUCP is $\mathcal{NP}$-hard.*

Proof. This corollary follows trivially from Lemma 6.5. ■

A slightly less negative result is the fact that a special case of SUCP is weakly $\mathcal{NP}$-complete. The proof is analogous to the proof of Theorem 14 in Amaddeo et al. [1997] and is based on a transformation from the well-known Partition Problem (PP) [Karp, 1972]:

Definition 6.7. Given a finite set $\mathcal{A}$ and a "size" $\sigma_a \in \mathbb{Z}^+$ for each $a \in \mathcal{A}$, the question of the Partitioning Problem (PP) is whether there exists a subset $\mathcal{A}' \subseteq \mathcal{A}$ such that $\sum_{a\in\mathcal{A}'} \sigma_a = \sum_{a\in\mathcal{A}\setminus\mathcal{A}'} \sigma_a$.

PP is proven to be $\mathcal{NP}$-complete by Karp [1972]. Subsequently, Garey and Johnson [1979] showed that this problem can be solved in pseudo-polynomial time by dynamic programming. Therefore, PP is weakly $\mathcal{NP}$-complete. The special case of SUCP considered here relaxes release times and deadlines, and assumes a special structure of the resource capacity function. It is defined as follows:

Definition 6.8. The Shunt Cleaning Resource Availability Problem (SCRAP) is a special case of SUCP, where $a_b \equiv 0, q_b \equiv \infty$ for all $b \in \mathcal{B}$, and the capacity of the cleaning crews models a resource with constant capacity and a break of one period, i.e.

$$c(m) = \begin{cases} 1 & \text{for } 0 \le m \le A \text{ and } A+1 \le m \le 2A+1 \\ 0 & \text{for } A < m < A+1. \end{cases}$$

Lemma 6.9. *SCRAP is weakly $\mathcal{NP}$-complete.*

Proof. Given an instance I of PP with $\sum_{a \in \mathcal{A}} \sigma_a$ even, we construct an instance I' of SCRAP by choosing $p_b = \sigma_b$ for all $b \in \mathcal{A}$ and $A = \frac{1}{2} \sum_{a \in \mathcal{A}} \sigma_a$. It follows that instance I' of SCRAP has a solution if and only if instance I of PP has a solution, therefore SCRAP is $\mathcal{NP}$-complete.

However, SCRAP can be solved in $\mathcal{O}(|\mathcal{B}| \sum_{b \in \mathcal{B}} p_b)$ by dynamic programming techniques analogous to the technique presented in Garey and Johnson [1979] for PP. ■

Note that in the instance I' in the proof of Lemma 6.9 half of the total processing time is scheduled before the break, and the other half is scheduled after the break. The dynamic programming technique mentioned in the proof of Lemma 6.9 builds a table with truth values whether or not subsets of $\mathcal{B}$ exist for which the sum of the processing times equals $1, \ldots, A$. In case $c(m)$ is constant, an instance of SCRAP is a yes-instance if and only if $\sum_{a \in \mathcal{A}} \sigma_a \leq |\mathcal{M}| c(m)$.

6.5 APPLICATIONS OF CLEANING

In order to test the practical applicability of Model (6.a) developed in the previous paragraph, we apply the model to the instances related to station Zwolle. The general structure of internal cleaning schedules has been discussed in Section 6.2. We are interested in both the Tuesday / Wednesday and the Saturday / Sunday scenarios. For both scenarios, we vary the definition of "close", denoted with z, between 1, 2, 5, 10, and 1440 minutes and we derive three cleaning instances for each parameter z. These three instances correspond to the different objectives in TMP, as described in Chapter 3. Recall that these matching objectives are denoted with A, B, and C:

- Matching objective A: minimization of the number of blocks.
- Matching objective B: minimization of the number of blocks and minimization of the deviation of the average length of stay.
- Matching objective C: minimization of the number of blocks and maximization of the deviation of the average length of stay.

Concerning the parameters for the cleaning process, we use the settings conform Section 6.2. In casu, the preprocessing of each job requires 13 minutes, while the post-processing requires 30 minutes. The availability of the cleaning crews is described by Figure 6.1, and the norms for cleaning rolling stock in man-minutes are given in Table 6.1. Cleaning takes place at the tracks 90 and 91 in Figure A.3 (at page 203). This figure shows part of the railway infrastructure of station Zwolle. Specifications of the hardware and software that were used can be found in Section 3.5 on page 61. The Integer Programs (IPs) are solved by CPLEX using standard settings.

To start with, we report the sizes of the instances before and after the standard reduction techniques of CPLEX in Table 6.3. Note that the sizes of the IPs are fixed for a given matching and varying parameter z. Moreover, many variables can be set to 0 because they result in an infeasible period of cleaning a block; e.g. because of starting before its release time, ending after its deadline or processing during a period when no cleaning crew is available.

	ZTA	ZTB	ZTC	ZSA	ZSB	ZSC
Number of jobs	31	32	33	24	23	24
Number of variables	22320	23040	23760	17280	16560	17280
Number of restrictions	751	752	753	744	743	744
Reduced number of variables	11400	11337	12781	8847	8460	8847
Reduced number of restrictions	677	666	684	647	646	647

Table 6.3: Some characteristics of the SUCP instances.

Note that the number of jobs varies for different matching objectives. This is caused by a different number of blocks that lay over at the shunt yard. Moreover, we see that the preprocessing phase of CPLEX is especially effective in reducing the number of variables: it saves nearly 50% of the variables.

The results of the computational tests of Model (6.a) can be found in Table 6.4. In this table, the first column gives the name of the instance. The structure of these names is similar to the structure of the names of the TMP instances (see page 61), extended with a number indicating the value of z. The third character in this name is always a dot. It represents the fact that the matching objective varies in the table. For example, the ZT.5 instances contain ZTA5, ZTB5, and ZTC5. Columns 2, 3 and 4 present the objective value of the instances, while columns 5, 6, and 7 report the value of the LP-relaxation. Finally, the last three columns represent the computation times for solving the instances.

All instances are solved well within 10 seconds except for the instances ZTB2, ZTB5, and ZTB10, which require at most 35 seconds. Moreover, the difference between the LP-relaxation and the integer solution is at most 1. The only instance with a difference of exactly one is instance ZTC2, which has an integer objective by chance. Rounding down the LP-relaxation results in all but one case in the optimal integer objective. This indicates that the model formulation is quite strong indeed.

Of course, the numbers of crews available in each shift largely affect the difficulty of the resulting cleaning problem. For the instances at station Zwolle on a Tuesday / Wednesday, the 6 crews in both the late shift and the night shift are just sufficient.

Instance	# jobs close			LP solution value			Comp. time (in sec.)		
	A	B	C	A	B	C	A	B	C
ZT.1	13	13	14	13.00	13.09	14.49	2.3	4.4	4.5
ZT.2	13	13	14	13.00	13.62	15.00	1.9	34.1	6.8
ZT.5	17	17	19	17.00	17.82	19.00	2.2	10.7	2.2
ZT.10	22	21	22	22.33	21.50	22.33	1.6	20.7	3.0
ZT.1440	31	32	33	31.00	32.00	33.00	2.6	2.7	3.2
ZS.1	10	10	10	10.00	10.00	10.00	1.2	1.2	1.2
ZS.2	11	11	11	11.00	11.00	11.00	1.2	1.6	1.2
ZS.5	13	12	13	13.00	12.00	13.00	1.3	1.6	1.3
ZS.10	16	15	16	16.00	15.00	16.00	1.1	1.6	1.1
ZS.1440	24	23	24	24.00	23.00	24.00	1.2	1.1	1.2

Table 6.4: Statistics on solving SUCP at station Zwolle.

However, for the Saturday / Sunday instances one might be able to clean all blocks using less cleaning crews. In Table 6.5, we present the results of the Saturday / Sunday instances at station Zwolle with 4 crews in the late shift and in the night shift. We see that this reduced capacity of the cleaning crews results in a significant reduction in the number of blocks that is cleaned close to the release times. This holds especially for the small values of the parameter z. Compared to the shifts with 6 crews, the cleaning crews obviously have less idle time between their tasks, while costs are lower since less cleaning crews are employed. This information is relevant on the tactical level of planning for the management of NedTrain. Concerning the computation times, we see that these are within 15 seconds, with instance ZSA1 as a negative outlier. Since the computation times of the Saturday / Sunday instances in Table 6.4 are within 2 seconds, we conclude that the computational burden also significantly increases with these reduced numbers of available crews.

Instance	# jobs close			LP solution value			Comp. time (in sec.)		
	A	B	C	A	B	C	A	B	C
ZS.1	3	4	3	4.07	5.51	4.07	85.7	13.1	4.8
ZS.2	6	7	6	6.27	7.64	6.27	5.8	3.7	7.6
ZS.5	9	9	9	9.00	9.77	9.00	4.0	3.3	4.6
ZS.10	12	12	12	12.00	12.10	12.00	2.7	1.9	3.1
ZS.1440	24	23	24	24.00	23.00	24.0	4.2	2.2	3.7

Table 6.5: SUCP instances with reduced numbers of cleaning crew.

6.6 EFFECT OF CLEANING ON PARKING

The results for solving SUCP are not only of interest in itself, but the relation with other shunting processes is also worth further examination. VELDHORST [2005] introduces the integrated problem of matching, parking and cleaning of train units. This problem is formulated as an Integer Program with a huge number of variables and restrictions. With some assumptions, instances with up to 5 tracks and 10 trains can be solved within a reasonable computation time. Although this model is currently not able to solve instances of practical importance, it might serve as a basis for future research.

In this paragraph, we investigate the effect of a cleaning schedule on TAP, as described in Chapter 4. Recall from Chapter 4 that each solution to TMP resulted in 3 instances for TAP: one instance with the track configurations shunt planners at NSR use, a second instance where track access is restricted to the mostly used side of each track in practice, and a third instance with the track configurations from practice and an extended objective to take robustness of a solution into account. All these instances take into account preferences of planners for shunt tracks as well as estimates of routing costs.

Since the blocks that need cleaning are split into two blocks for TAP, the number of blocks for TAP is significantly larger than the number of blocks in the original instances, resulting in larger computation times. These larger instances give us the opportunity to consider the effect of the 2-OPT heuristic that can be used for generating an initial set of columns, as described in Section 4.8 and by FIOOLE [2003].

We start with the results of TAP for the different values of z and consider the instances ZTAD and ZTAR. These instances are Tuesday / Wednesday instances at station Zwolle with matching objective A and practical track configurations (ZTAD) or restricted track configurations (ZTAR). For these instances, we report the computation time, some characteristics of the quality of the solution (LP solution value, IP solution value, and the gap between these two), the number of blocks not parked, and some characteristics of the column generation process (the number of columns generated by 2-OPT, the total number of columns generated, and the number of iterations of column generation). We solve each instance without and with the 2-OPT heuristic for an initial column set. Note that we left out the ZTAO instances for purposes of exposition. The interested reader is referred to Table A.8 in Appendix A.3 on page 207 for these results.

Analyzing Table 6.6, we see that for each of the 20 TAP instances, all blocks can be parked at the shunt yard. When 2-OPT is not applied, the average computation time is 757 seconds for the ZTAD instances. The upper part of the table shows computation times ranging between 555 and 1079 seconds without the application of 2-OPT (i.e. in the even columns) for ZTAD instances. The computation time for the corresponding ZTAD instance without cleaning is 246 seconds, as can be seen in Table 4.6 on page 86. Similarly, the lower part of Table 6.6 results in an average computation time for instances

Use 2-OPT Start?	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Number of blocks	68		66		63		63		68	
Instance **ZTAD**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	1079	226	715	299	555	256	567	283	870	488
LP solution value	10441.99	10407.86	10041.63	10052.27	9722.38	9722.24	9669.00	9642.00	10443.82	10442.23
IP solution value	10679	10560	10130	10129	9888	9823	9944	9913	10748	10714
Gap	2.22%	1.44%	0.87%	0.76%	1.67%	1.02%	2.77%	2.73%	2.83%	2.54%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	316	0	266	0	262	0	282	0	330
# Columns generated	7164	3454	7336	4312	7129	3501	6526	2616	7446	4087
# Iterations of col. gen.	42	36	47	38	47	37	41	38	46	41
Instance **ZTAR**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	811	736	1667	531	417	184	713	187	1637	690
LP solution value	11656.60	11630.47	11465.36	11413.95	10914.40	10907.76	10883.14	10840.69	11813.53	11805.83
IP solution value	12119	11950	11817	11792	11040	11056	11225	10937	12341	12129
Gap	3.82%	2.67%	2.98%	3.21%	1.14%	1.34%	3.05%	0.88%	4.27%	2.66%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	247	0	259	0	260	0	239	0	268
# Columns generated	7310	2669	7475	2837	6651	2360	5809	2378	7793	2807
# Iterations of col. gen.	46	29	49	28	41	28	36	25	47	30

Table 6.6: The effect of cleaning on TAP for station Zwolle on Tuesday / Wednesday for matching objective A in TMP.

based on instance ZTAR of 1049 seconds. In this case, the computation time for ZTAR without cleaning is 188 seconds (see Table 4.8 on page 87). The additional computation time can be saved by applying 2-OPT, which typically requires approximately 10 seconds to find an initial set of columns. This computation time is not included in the computation times reported in this table. This reduction in computation time is mostly caused by a reduction in the number of iterations of column generation. Moreover, we also observe a smaller number of columns in the final LP-relaxation, which results in a smaller branch-and-bound tree. The differences in LP solution values between applying 2-OPT or not can be attributed to tailing off effects: if the LP solution value improves little over a number of iterations, column generation is terminated. In all instances except ZTAR5, the application of 2-OPT results in a better solution.

In addition, we present the effects of the cleaning schedule on the instances of TAP with extended matching objective C in TMP, preferring LIFO matchings. These results are reported in Table 6.7, where we left out the ZTCR instances intentionally. The interested reader will find these results in Table A.8. As mentioned in Chapter 4, the LP-relaxation is not necessarily a lower bound of TAP for the ZTCO instances. This is caused by the additional penalties and rewards that are considered.

The instances with matching objective C appear to be more difficult to solve in terms of computation times. When 2-OPT is not applied, the average computation time is nearly 30 minutes. In contrast, the corresponding instances without cleaning are solved within 7 minutes (see Tables 4.6 and 4.10). These computation times are quite high for interactive decision support. However, in all instances except instance ZTCD1440 the computation time with the application of 2-OPT is at most 8 minutes, which is reasonable. Concerning the quality of the solutions, the application of 2-OPT results in comparable solutions of the instances. Therefore, the added value of 2-OPT for these instances lies mainly in the reduction of the computation times. The results for instances based on matching objective B can be found in Tables A.9 and A.10 in Appendix A.3.

In general, we conclude that cleaning schedules can be computed within short computation times. Moreover, these schedules have a large effect on the instances for TAP. These instances require a heuristic to generate initial columns in order to find good solutions within reasonable computation times. Therefore, the model developed in this chapter provides a solid basis for supporting shunt planners in the generation of internal cleaning schedules.

6.7 CONCLUSIONS

This chapter started with an in-depth introduction to the planning of the internal cleaning process at railway stations in the Netherlands. The introduction included its restrictions and objectives. In short, sufficient capacity of the cleaning crews needs to

Use 2-OPT Start?	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Number of blocks	65		67		65		65		70	
Instance **ZTCD**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	1230	446	1563	366	1174	365	1793	447	5406	2444
LP solution value	9828.39	9963.17	10265.30	10243.36	10043.68	10044.31	10001.74	9976.30	10683.97	10763.63
IP solution value	10101	10350	10552	10527	10309	10087	10261	10421	11137	10852
Gap	2.70%	3.74%	2.72%	2.69%	2.57%	0.42%	2.53%	4.27%	4.07%	0.82%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	287	0	250	0	269	0	299	0	266
# Columns generated	7414	3960	7143	3166	6644	2315	6247	2995	7648	3679
# Iterations of col. gen.	46	30	42	28	46	25	38	26	48	30
Instance **ZTCO**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	793	335	1594	270	993	315	1136	344	1374	380
LP solution value	7721.50	7794.50	8401.44	8281.79	7523.00	7310.98	7940.32	7965.03	9136.54	9071.71
IP solution value	7885	7991	8703	8577	7533	7578	8047	8275	9426	9394
Gap	2.07%	2.46%	3.47%	3.44%	0.13%	0.91%	3.52%	3.75%	3.07%	3.43%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	400	0	286	0	335	0	459	0	420
# Columns generated	7349	3846	7508	3573	7311	3234	8594	3242	8202	2813
# Iterations of col. gen.	39	30	41	24	40	26	44	23	44	31

Table 6.7: The effect of cleaning on TAP for station Zwolle on Tuesday / Wednesday for matching objective C in TMP.

be assigned to each block that lays over at a station, where the cleaning crews clean one block at a time. Typically, cleaning crews work in a late shift and a night shift. The objective is to schedule as many blocks as possible “close” in time to their release times, since this might save additional routing and parking, and results in less disturbances for people living in the surroundings of the shunt yard. Because the shifts of the cleaning crews and the number of crews in a shift are given on beforehand, the capacity of the “machine” cleaning the blocks varies over time. This varying machine capacity is found rarely in scientific literature. Moreover, the objective of scheduling as many jobs as close as possible to their release times is new to our knowledge. This led to the formulation of the Shunt Unit Cleaning Problem (SUCP), which can be seen as a single machine scheduling problem. After an example, we presented a short overview of literature on practical developments, on related maintenance planning problems, and on machine scheduling.

A time-indexed Integer Program for SUCP has been developed in this chapter. Moreover, we showed that SUCP is $\mathcal{NP}$-complete in the strong sense by a reduction from the problem Sequencing within Intervals. However, a special case of SUCP is weakly $\mathcal{NP}$-complete. In this special case, release times and deadlines are relaxed and the machine capacity has a certain form. This special case is called the Shunt Cleaning Resource Availability Problem (SCRAP). SCRAP can be solved in pseudo-polynomial time and space by dynamic programming.

The resulting Integer Program (IP) for the general problem is solved by CPLEX. Based on the matchings computed in Section 3.5 for station Zwolle, we tested the formulation with different definitions of starting close in time and 6 crews in both the late shift and the night shift. Most instances were solved to optimality within 10 seconds computation time, with 3 instances requiring at most 35 seconds to solve. Since the capacity of the cleaning crews is rather large for the matchings in the Saturday / Sunday instances, we reduced the numbers of crews in the late shift and the night shift to 4 persons and reran the instances. This obviously resulted in less blocks that could be scheduled close to their release times. In addition, computation times typically increased from approximately 2 seconds to 15 seconds. In one outlier, the computation time increased from 1.17 seconds to 85.67 seconds. For all tested instances, the formulation of the IP model is quite strong, since the value of the LP-relaxation is typically close to the IP solution value.

The short computation times enable planners to explore different parameter settings. Therefore, several alternatives for an instance can be developed and the best one can be selected. Similar to previous chapters, interaction between the algorithm and shunt planners is likely to produce the best results.

Of course, the cleaning schedule is tightly related with the parking of the blocks. Therefore, we looked at the effect of the cleaning schedules for the Tuesday / Wednesday instances at station Zwolle on the instances for the Track Assignment Problem (TAP). TAP was the subject of Chapter 4. Here, we saw that the column generation heuristic of Chapter 4 is still able to park all blocks. However, the computations required significantly more time. In order to overcome this additional computational burden, we applied the 2-OPT heuristic of Fioole [2003] for generating an initial set of columns, as described in Section 4.8. This resulted in a decrease of the computation times to the levels of the instances in Chapter 4.

Summarizing, in this chapter we developed a model for SUCP. The developed model can be used for supporting shunt planners in the creation of internal cleaning schedules. Although SUCP is a challenging problem from a scientific point of view, the developed model is able to solve real-life instances to optimality within seconds. Moreover, with an additional 2-OPT heuristic for generating an initial set of columns, the resulting instances of TAP can be solved satisfactorily.

Chapter 7

Integrated Matching and Parking of Train Units

The focus of this chapter is on the problem of integrating matching and parking of train units. The matching and parking problems have been discussed separately in Chapters 3 and 4, respectively.

Although both matching and parking have been treated in depth in previous chapters, the most important characteristics are repeated in this chapter. After the integrated problem has been stated, we discuss the computational complexity of several variants of this problem and we relate it to similar problems in other modes of transport, including buses and trams.

In this chapter, the problem is shortly introduced and some computational properties are discussed. Hereafter, it is related to several other problems in the scientific literature. Moreover, we apply several variants of a model, which have been developed by Schrijver [2003]. We have implemented parts of these models and we have carried out extensive computational experiments. The results are presented for Tuesday / Wednesday instances at stations Zwolle and Enschede. These results are compared with the results presented in Chapters 3 and 4.

7.1 INTRODUCTION

In this paragraph, assumptions and definitions are introduced in order to define the integrated problem. Additional assumptions to those in Section 1.10 that are relevant in this chapter are:

- The arrivals and departures of each train unit type are balanced in the planning period. This means that the number of arriving units of a type of train unit equals the number of departing units of this type. Moreover, at each point in time, the number of arriving units of each type from the start of the planning period is at least equal to the number of departing units in this period (see also Section 3.1).

- No train units are parked at the shunt yard at the start or end of the planning period (see also Section 3.1).
- No precedence relationships between different shunt tracks are present. That is, units parked at one shunt track do not influence the possibilities to park other units at a different track. These precedence relationships are clarified in Section 2.3.2 on page 30, while the assumption is taken from Section 4.1.
- The length of the units parked at a shunt track and the length of a track are sufficient to check the capacity of a shunt track. This might be a problem for free tracks, as has been discussed in Section 4.1 on page 67. In short, since arrivals and departures might take place using both sides of a free shunt track, repositioning of rolling stock might be required. This assumption states that such repositionings can be neglected (taken from Section 4.1).

In addition to these assumptions, we repeat relevant definitions which have been introduced in previous chapters. These definitions include:

- A shunt unit, which is a train unit that requires shunting and can either be an arriving or a departing shunt unit.
- A LIFO track, which can be approached from one side, and a free track, which can be approached from both sides.
- A part to denote one or more adjacent train units in a train, and a block, which is a matching of an arriving part to a departing part.
- A crossing, which occurs when a train unit is blocking the arrival or departure of another train unit at a shunt track.

Note that these definitions can be found in Appendix A.1.

Each station has an A-side and a B-side. Given these sides, we define the A-side of a track as the side which is closest to the A-side of the station, and similarly the B-side of a track. A shunt track can be open at the A-side, the B-side, or both sides. This uniformly defines the side of a shunt track. Moreover, we introduce the A-side of a train as the side of the train which is closest to the A-side of the station, whenever the train is within the boundaries of the station.

Note that in the instances resulting from practice, arrivals and departures are mixed in time. This implies that, within the planning horizon, the first departure typically takes place before the last arrival.

From Chapter 3, we reuse the restrictions on type mismatches and the order of train units in a train. The timetabled trains have fixed configurations, since these have been decided upon in previous planning processes. Therefore, it is not allowed to supply a unit in a train of a different type than prescribed in the timetable. Moreover, in case a train consists of different types of units, the order of the types in the train has to be adhered to as well.

Regarding the parking problem, the main elements that determine whether an assignment of train units to a shunt track is feasible are repeated from Chapter 4. Crossings are not allowed in such assignments, while also the length of the train units parked at a shunt track can never exceed the length of the track. The last relevant criterium for parking is that the train units must be allowed to park at the shunt track. For example, electrical train units require shunt tracks with catenary installed.

Elements of the costs of a solution to the integrated problem include cost for each block, cost for shunt tracks with multiple types of train units, and penalties for train units leaving in the same departing train but parked at different shunt tracks. Note that preferences for LIFO matchings, as described in Chapter 3, are not taken into account explicitly, but are left as a degree of freedom in the problem.

Given the description of the most important aspects of the integrated problem, we formally define the Train Unit Shunting Problem (TUSP) as follows:

Definition 7.1. Given

- a railway station,
- a shunt yard, usually geographically separate from the station,
- a timetable, with for each train service the arrival and / or departure time and platform at the involved station, and its configuration,

the Train Unit Shunting Problem (TUSP) consists of (*i*) matching the arriving and departing shunt units, and (*ii*) parking these shunt units at the shunt tracks, such that the total shunting costs are minimal and neither crossings nor type mismatches occur, while the capacity of each shunt track is never exceeded by the train units parked at it.

A solution to TUSP assigns arriving shunt units to departing ones and to a shunt track. Moreover, if such a track can be approached from both sides, the solution also describes the arrival and departure sides for each train unit parked at the track.

7.2 COMPLEXITY RESULTS FOR TUSP

In this paragraph we state that TUSP is $\mathcal{NP}$-hard. Note that this result is not surprising since both TMP and TAP are already $\mathcal{NP}$-hard (see Corollaries 3.5 and 4.6 respectively).

However, it is stated here for completeness. An important related problem is the 0-Tram Dispatching Problem (0-TDP), studied by Winter [1999] and defined as follows:

Definition 7.2 (Definition 3.2.5 in Winter [1999]). Given

- an ordered set of arriving trams, where each tram is of a certain type,
- an ordered set of departing trams, where the first departure takes place after the last arrival, and
- a set of depot positions located in a number of stacks of certain sizes,

the 0-Tram Dispatching Problem (0-TDP) is to find an assignment of arriving trams to positions and of positions to departing trams without type mismatches and without crossings.

Winter proves that 0-TDP is $\mathcal{NP}$-complete via a reduction of 0-TDP from 3-Dimensional Matching [Garey and Johnson, 1979]. It remains $\mathcal{NP}$-complete if one considers a variable number of stacks with 4 positions in each stack. We start with the complexity of the decision version of TUSP, where the question is whether there exists a feasible solution without type mismatches and without crossings.

Theorem 7.3. *The decision version of TUSP is $\mathcal{NP}$-complete.*

Proof. This follows directly from the $\mathcal{NP}$-completeness of TMP or TAP. These results can be found in Theorems 3.4 respectively 4.5. ∎

Note that it is also easy to see that 0-TDP is a special case of TUSP.

Corollary 7.4. *TUSP is $\mathcal{NP}$-hard.*

Proof. This follows trivially from Theorem 7.3. ∎

We proceed by showing that a special case of TUSP can be solved in time and space polynomial in the number of trains and the number of types of train units parked simultaneously at a shunt yard. First, we define the special case:

Definition 7.5. The Fixed Size Train Unit Shunting Problem (FSTUSP) is a special case of TUSP, where each train consists of exactly one train unit and at most p units can be simultaneously parked at the shunt yard. Moreover, all tracks are only accessible from the A-side only and each train unit type has the same size. In addition, the objective is to minimize the total estimated routing costs while parking all units. Such a problem is denoted with FSTUSP(p).

In the proof of the following theorem, we require some additional notation. This notation is introduced first:

- The set $\mathcal{S}$ contains all the shunt tracks at which units can be parked.
- $\mathcal{T}$ is the set of trains, consisting of arriving trains in $\mathcal{T}_+$ and departing trains in $\mathcal{T}_-$.
- $\mathcal{Y}$ is the set of types of train units.
- p is the maximum number of train units simultaneously at the shunt yard.
- p_s is the number of positions at track $s \in \mathcal{S}$, with $\sum_{s\in\mathcal{S}} p_s = p$. The positions at a track are numbered increasingly on the distance from the A-side of the track. Thus, higher indices are further away from the open side of the track.
- $\psi_t \in \mathcal{Y}$ denotes the type of the train unit in train $t \in \mathcal{T}$.
- l_t denotes the length of the type of train unit corresponding to train $t \in \mathcal{T}$.
- $r_{t,s}$ is the estimated cost of routing train $t \in \mathcal{T}$ to / from track $s \in \mathcal{S}$.

Similar to the proof of Theorem 5 in Kroon et al. [1997] and Theorem 3.11 in this thesis, we show that FSTUSP(p) belongs to $\mathcal{P}$ by constructing a dynamic programming algorithm in a certain network. This network has a size, which is polynomial in the number of trains $|\mathcal{T}|$ and the number of types of train units $|\mathcal{Y}|$. At a node i in this network, we register for each track s an index p_i^s and a vector o_i^s. The index represents the farthest position from the A-side of track s, which is available for parking a train unit. The vector represents the types of train units parked at the occupied positions of the track. When a train unit is parked at a shunt track, it closes up and leaves no empty positions between its position and the positions of the previously parked train units. The network contains an initial layer of nodes and one layer of nodes for arriving or departing train $t \in \mathcal{T}$. For each node in the initial layer, $p_i^s = p_s$ and o_i^s is empty.

Theorem 7.6. *FSTUSP(p) can be solved in time and space which are polynomially bounded by the number of trains $|\mathcal{T}|$ and the number of types of train units $|\mathcal{Y}|$.*

Proof. We start with the construction of a network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, based on the arrivals and departures $\mathcal{T}$. For each train $t \in \{1, \ldots, \mathcal{T} - 1\}$, we construct a layer $\mathcal{L}_t$ of nodes representing all possible states of types of train units parked at positions of the shunt yard. In this layer $\mathcal{L}_t$, the changes related to the arrival or departure of train t are already taken into account. Layers $\mathcal{L}_0$ and $\mathcal{L}_{|\mathcal{T}|}$ represent the empty shunt yard at the start respectively end of the planning period. Arcs are only present between two consecutive layers $\mathcal{L}_t$ and $\mathcal{L}_{t+1}$ for some $t \in \{0, \ldots, |\mathcal{T}| - 1\}$. More precisely, if train $t + 1$ is an arriving train then

$$(i,j) \in \mathcal{A}, i \in \mathcal{L}_t, j \in \mathcal{L}_{t+1}, \text{for some } t \in \{0,\ldots,|\mathcal{T}|-1\} \quad \Leftrightarrow$$

$$\exists s \in \mathcal{S} : p_j^s = p_i^s - 1, p_j^s \geq 0, o_j^s[p_i^s] = \psi_{t+1}, \text{ and} \tag{7.1}$$

$$o_i^{s'} = o_j^{s'}, p_i^{s'} = p_j^{s'} \text{ for } s' \in \mathcal{S} \setminus s \tag{7.2}$$

If train $t+1$ is a departing train then

$$(i,j) \in \mathcal{A}, i \in \mathcal{L}_t, j \in \mathcal{L}_{t+1}, \text{for some } t \in \{0,\ldots,|\mathcal{T}|-1\} \quad \Leftrightarrow$$

$$\exists s \in \mathcal{S} : p_j^s = p_i^s + 1, p_j^s \leq p_s, o_j^s[p_i^s] = \psi_{t+1}, \text{ and} \tag{7.3}$$

$$o_i^{s'} = o_j^{s'}, p_i^{s'} = p_j^{s'} \text{ for } s' \in \mathcal{S} \setminus s \tag{7.4}$$

In (7.1), the arriving train unit closes up at one shunt track, respecting the length of the track. Equation (7.2) ensures that exactly one vector and one index change. The relations for a departing train unit are similar.

Now, we can define the nodes of our network as $\mathcal{N} = \bigcup_{t=0}^{|\mathcal{T}|} \mathcal{L}_t$. In addition, the arcs are defined by (7.5):

$$\begin{aligned} \mathcal{A} = \{&(i,j) \mid (7.1) \text{ and } (7.2) \text{ hold and } j \in \mathcal{L}_t \text{ for some } t \in \mathcal{T}_+\} \\ \cup\{&(i,j) \mid (7.3) \text{ and } (7.4) \text{ hold and } j \in \mathcal{L}_t \text{ for some } t \in \mathcal{T}_-\} \end{aligned} \tag{7.5}$$

The number of nodes in one layer is bounded by $(|\mathcal{Y}|+1)^p$. Since p is considered fixed, $|\mathcal{N}|$ is polynomially bounded by $|\mathcal{Y}|$ and $|\mathcal{T}|$. In addition, since arcs only exist between consecutive layers, $|\mathcal{A}|$ is also bounded by these numbers.

For an arc $(i,j) \in \mathcal{A}$, we set its length $f_{i,j}$ to

$$f_{i,j} = \begin{cases} r_{t,s} & \text{if } j \in \mathcal{L}_t \text{ for some } t \in \{1,\ldots.|\mathcal{T}|-1\} \text{ with } o_i^s \neq o_j^s; \\ 0 & \text{otherwise.} \end{cases}$$

Now, a shortest path in $\mathcal{G}$ with lengths $f_{i,j}$, $(i,j) \in \mathcal{A}$, corresponds to a solution to FSTUSP(p) with minimum estimated routing costs. This proves the theorem. ■

At the cost of cumbersome additional notation, Theorem 7.6 can be extended to differentiate between using the A-side and the B-side of a free track and to incorporate tracks with different lengths. Moreover, by introducing even more notation, trains consisting of multiple train units and train unit types with different lengths can be handled. Due to the enormous size of the involved network, Theorem 7.6 has little practical significance.

Blasum et al. [2000] study amongst others the computational complexity of a related tram scheduling problem. In their case, arriving trams have been assigned already to shunt tracks. In general, this problem is $\mathcal{NP}$-complete, even when the number of types is fixed and at least 2. In Chapter 3 of his thesis, WINTER [1999] proves complexity results for a number of related problems.

7.3 RELATED LITERATURE

The integrated problem as described in Section 7.1 has also been studied by HAIJEMA ET AL. [2005]. These authors propose a heuristic based on dynamic programming for solving the problem. Based on the timetable, a set of periods are identified, where one period consists of a number of arrivals followed by a number of departures. Since the departures in the morning are most critical, a blueprint of the shunt yard is generated. This blueprint contains the ideal location of the departing train units at the shunt yard in the last period. Subsequently, one tries to assign the arriving train units at minimum cost to the blueprint. The objective consists of minimizing the costs for shunting the parts. Typically, some of the departing train units in the blueprint cannot be served by the arriving train units in the last period. In this case, the remaining part of the blueprint serves as a starting point for the next period. Iteratively, the proposed algorithm works towards the start of the planning period. Since it is computationally infeasible to use all possible blueprints as starting points for the next period, only the best blueprint is used. This introduces a heuristic feature of the algorithm. Computational results for a real-life instance at station Zwolle are reported for Friday / Saturday. The instance consists of 55 arriving trains and 45 departing ones, divided over 23 periods. The algorithm finds a solution within one second on a standard PC, which is promising.

VERMEULEN [2005] develops a first step in the direction of integrating matching, parking and routing into one problem. He is able to find high-quality solutions for restricted instances with an Integer Program.

Several problems similar to TUSP have been dealt with by WINTER AND ZIMMERMANN [2000] and BLASUM ET AL. [2000] for dispatching trams in a depot. Much initial work on dispatching trams has been done by WINTER [1999]. This includes online versions of several variants, where information on the arriving and departing trams is revealed gradually during the planning period. Moreover, he theoretically extends his approach with length restrictions and mixed arrivals and departures. Finally, he also discusses an application to a bus depot, including computational results.

Furthermore, GALLO AND DI MIELE [2001] discuss an application for dispatching buses in a depot. Their basic model does not include mixed arrivals and departures, but an extension to include this aspect is also introduced. Another application of bus dispatching is described in HAMDOUNI ET AL. [2004]. Here, robust solutions are emphasized by having as little different types of buses as possible in one lane, and within one lane by grouping together the buses of the same type as much as possible. In case buses are mispositioned in a lane, these are repositioned during the night. In subsequent work, Hamdouni et al. [2005] develop an alternative formulation, where type-mismatches between requested and supplied buses are allowed at some cost.

Tomii et al. [1999] and Tomii and Zhou [2000] propose a genetic algorithm that takes into account some related processes of TUSP. However, their parking problem is of a less complex nature, since in their context at most one train unit can be parked at a shunt track at the same time.

7.4 BASIC MODEL FOR LIFO TRACKS

In this paragraph, we discuss the most basic model for TUSP described by Schrijver [2003]. This model considers the special case that all tracks can only be accessed from the A-side. Before discussing this model, some notation is introduced. As mentioned in Section 7.2, the timetable consists of a set of arriving trains $\mathcal{T}^+$ and a set of departing trains $\mathcal{T}^-$. Moreover, we define $\mathcal{U} = \{1, \ldots, |\mathcal{U}|\}$ as the set of train units that arrive or depart. For each train unit $u \in \mathcal{U}$, we know whether it is arriving or departing since we know to which train a unit belongs. Therefore, we can partition the sequence $\mathcal{U}$ in a set of arriving train units, denoted by $\mathcal{U}_+$, and a set of departing train units, denoted by $\mathcal{U}_-$, with $\mathcal{U} = \mathcal{U}_+ \cup \mathcal{U}_-$ and $\mathcal{U}_+ \cap \mathcal{U}_- = \oslash$. Moreover, we introduce the mapping ψ_u, which maps a train unit $u \in \mathcal{U}$ to its type $\psi \in \mathcal{Y}$. Also, the mapping l_u maps a unit $u \in \mathcal{U}$ to its length. For each $u \in \mathcal{U}$, we know the train service i_u in which it arrives at or departs from the station.

For each train service, we know the planned time of arrival or departure as well as the exact configuration of the train operating this service. The set $\mathcal{U}$ is sorted according to the partial ordering $<_A$ on the train units. By definition $u_i <_A u_j$ if and only if one of the following conditions is satisfied:

1. Unit u_i arrives or departs in a train with an earlier planned time than the train to which unit u_j belongs.
2. Arriving units u_i and u_j arrive in the same train and u_j is closer to the A-side of the train than u_i.
3. Departing units u_i and u_j depart in the same train and u_i is closer to the A-side of the train than u_j.

Consider an arriving train configuration $u_1 \cdots u_k$, where u_1 is closest to the A-side of the train and u_k is farthest from it. Case 2 states that it is ordered as $u_k <_A \ldots <_A u_1$ in $\mathcal{U}$. Note that this is the order in which the units arrive at a shunt track open at the A-side. In Figure 7.1, the arriving train at 08:00 has configuration IRM_3 IRM_4 and is ordered as IRM_4 $<_A$ IRM_3. Since the units of a departing service $u_1 \cdots u_k$ leave the shunt track via the A-side in the order $u_1 <_A \ldots <_A u_k$, Case 3 states that a departing service is ordered in this manner in $\mathcal{U}$. In this example, we numbered the train units ①,②,③,④ resulting in $\mathcal{U} = \{①, ②, ③, ④\}$ with order ①$<_A$②$<_A$③$<_A$④. Moreover, the relation $u_i \leq_A u_j$ holds if and only if $u_i <_A u_j$ or $u_i = u_j$.

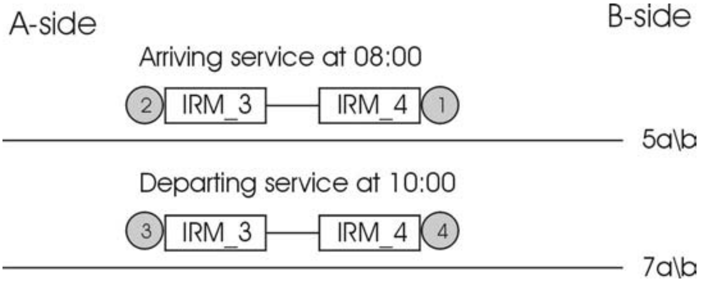


Figure 7.1: An example of the ordering of train units.

The set $\mathcal{I}$ is defined as the set of pairs of train units $u, u+1 \in \mathcal{U}$ that arrive or depart in the same train service, i.e. $i_u = i_{u+1}$. Finally, we introduce $\mathcal{L}$ as the set of pairs of train units (t, u) that can be matched:

$$\mathcal{L} := \{(t, u) \mid t \in \mathcal{U}_+, u \in \mathcal{U}_-, t <_A u, \psi_t = \psi_u\}. \tag{7.6}$$

Note that in this approach, train units are considered individually and the concept of parts as used in Chapters 3 and 4 is only present implicitly. Moreover, it is assumed that the conditions in the definition of the set $\mathcal{L}$ are sufficient for a feasible matching of two train units, although this is not necessarily the case. Indeed, even if the previous conditions are met, a train unit t cannot be matched to a train unit $u >_A t$ if no train unit of type ψ_t is present at some point in time between the arrival of unit t and the departure of unit u. The set $\mathcal{X}$ of potential crossings at a LIFO track is defined as:

$$\mathcal{X} := \{\{(t, u), (v, w)\} \mid \{(t, u), (v, w)\} \in \mathcal{L}^2, t <_A v <_A u <_A w\}$$

Given these definitions, Schrijver [2003] introduces the following decision variables:

$$P_{t,s} = \begin{cases} 1 & \text{if train unit } t \text{ is parked at or retrieved from track } s; \\ 0 & \text{otherwise.} \end{cases}$$

$$M_{t,u,s} = \begin{cases} 1 & \text{if arriving train unit } t \text{ is matched to departing unit } u \\ & \text{and parked at track } s; \\ 0 & \text{otherwise.} \end{cases}$$

$$L_{t,s} = \begin{cases} \text{the length of the train units at track } s \text{ after the arrival} \\ \text{or departure of unit } t. \end{cases}$$

$$K_t = \begin{cases} 1 & \text{if units } t \text{ and } t+1 \text{ are related to the same train} \\ & \text{and are parked at or retrieved from different tracks;} \\ 0 & \text{otherwise.} \end{cases}$$

$$O_{\psi,s} = \begin{cases} 1 & \text{if at least one unit of type } \psi \text{ is parked at track } s; \\ 0 & \text{otherwise.} \end{cases}$$

$$E_s = \text{the number of types } \psi \text{ in excess of 1 parked at track } s.$$

For each shunt track $s \in \mathcal{S}$, l^s gives the length of track s. The penalty g on the variables E_s models a preference for solutions with less different types parked at a track. After introduction of the weights a for splitting of trains, the model reads:

$$\begin{aligned}
\text{minimize} \quad & a \sum_{t \in \mathcal{U}} K_t + g \sum_{s \in \mathcal{S}} E_s && && (7.7)\\
\text{subject to} \quad & \sum_{s \in \mathcal{S}} P_{t,s} = 1 && \forall t \in \mathcal{U} && (7.8)\\
& \sum_{u:(t,u) \in \mathcal{L}} M_{t,u,s} = P_{t,s} && \forall t \in \mathcal{U}_+, s \in \mathcal{S} && (7.9)\\
& \sum_{u:(u,t) \in \mathcal{L}} M_{t,u,s} = P_{t,s} && \forall t \in \mathcal{U}_-, s \in \mathcal{S} && (7.10)\\
& M_{t,u,s} + M_{v,w,s} \leq 1 && \forall s \in \mathcal{S}, \{(t,u),(v,w)\} \in \mathcal{X} && (7.11)\\
& L_{t,s} = L_{t-1,s} + l_t P_{t,s} && \forall t \in \mathcal{U}_+, s \in \mathcal{S} && (7.12)\\
& L_{t,s} = L_{t-1,s} - l_t P_{t,s} && \forall t \in \mathcal{U}_-, s \in \mathcal{S} && (7.13)\\
& L_{t,s} \leq l^s && \forall t \in \mathcal{U}_+, s \in \mathcal{S} && (7.14)\\
& K_t \geq P_{t,s} - P_{t+1,s} && \forall s \in \mathcal{S}, (t,t+1) \in \mathcal{I} && (7.15)\\
& P_{t,s} \leq O_{\psi_t,s} && \forall s \in \mathcal{S}, t \in \mathcal{U} && (7.16)\\
& \sum_{\psi \in Y} O_{\psi,s} \leq E_s + 1 && \forall s \in \mathcal{S} && (7.17)\\
& M_{t,u,s} \in \{0,1\} && \forall (t,u) \in \mathcal{L}, s \in \mathcal{S} && (7.18)
\end{aligned}$$

In this model, restrictions (7.8) state that each train unit needs to be parked at a track. Restrictions (7.9) state that if arriving unit t is parked at track s, then exactly one departing unit u exists for which $M_{t,u,s} = 1$ and $(t,u) \in \mathcal{L}$. This also holds for departing train units in restrictions (7.10). In addition, restrictions (7.11) prohibit crossings. Restrictions (7.12) are used for administrating the length of the units parked at a track at arrival of a train unit. Again, restrictions (7.13) are the same, but for departing units. Restrictions (7.14) ensure that the total length of the units parked at a track never exceeds the capacity of the track. Note that these restrictions are only checked after arrivals of trains. Moreover, restrictions (7.15), (7.16), and (7.17) are used for determining the right values of the decision variables in the objective function (7.7). Finally, restrictions (7.18) are integrality restrictions on the M variables. We call this model Model (7.a). The bottleneck of this model is the large number of restrictions (7.11), which equals $|\mathcal{S}| \times |\mathcal{X}|$. In order to reduce this number of crossing restrictions, we aggregate them. This aggregation also strengthens the formulation.

For a convenient discussion of the aggregation of crossing restrictions, we introduce the set $\mathcal{Z}$ as the set of pairs $(v,u) \in \mathcal{U}^2$ such that there exist $(t,u),(v,w) \in \mathcal{L}$ with $t <_A v <_A u <_A w$. Note that this requires $u \in \mathcal{U}_-$ and $v \in \mathcal{U}_+$. An element of $\mathcal{Z}$ describes an arrival, followed by a departure, which might be involved in a crossing. Given $\mathcal{Z}$, restrictions (7.11) can be replaced with:

$$\sum_{t<_A v:(t,u)\in\mathcal{L}} M_{t,u,s} + \sum_{w>_A u:(v,w)\in\mathcal{L}} M_{v,w,s} \;\leq\; 1 \quad \forall (v,u) \in \mathcal{Z}, s \in \mathcal{S} \tag{7.19}$$

These restrictions sharpen the restrictions (7.11) and are far less in number. Given this replacement, we define Model (7.b) by (7.7)-(7.10),(7.19), (7.12)-(7.18). In the remainder of this chapter, we try to reduce the number of restrictions even further.

7.5 RESTRICTING THE NUMBER OF MIXED TRACKS

Suppose, one would know on beforehand that only units of one type are parked at a certain shunt track. For such a track, the aggregated crossing restrictions (7.19) are superfluous. Indeed, since train units of the same type can be used interchangeably, the order of the train units at such a track is irrelevant. This would add to the robustness of a solution, it being better able to handle disruptions in the operations, as mentioned in Section 7.1.

However, one does not want to choose on beforehand which tracks are mixed tracks, containing several types, and which ones are not. SCHRIJVER [2003] achieves this flexible selection of mixed tracks by introducing virtual tracks and by assigning these virtual tracks to physical tracks. Let $\mathcal{S}$ represent the set of virtual tracks. Moreover, $\mathcal{P}$ is the set of physical tracks, with $|\mathcal{S}| = |\mathcal{P}|$. A matching from $\mathcal{S}$ to $\mathcal{P}$ assigns the virtual tracks to the physical ones. Let $\mathcal{S}'$ be the set of mixed virtual tracks and let $\mathcal{S}'' = \mathcal{S} \setminus \mathcal{S}'$ be the set of non-mixed virtual tracks. Then it suffices to activate the crossing restrictions (7.19) only for the mixed tracks in $\mathcal{S}'$ instead of for all virtual tracks in $\mathcal{S}$:

$$\sum_{t<_A v:(t,u)\in\mathcal{L}} M_{t,u,s} + \sum_{w>_A u:(v,w)\in\mathcal{L}} M_{v,w,s} \;\leq\; 1 \quad \forall (v,u) \in \mathcal{Z}, s \in \mathcal{S}' \tag{7.20}$$

Of course, only units of one type can be parked at a track in $\mathcal{S}''$, which results in this additional set of restrictions:

$$E_s \;=\; 0 \quad \forall s \in \mathcal{S}'' \tag{7.21}$$

In fact, this reduces the number of decision variables, since these E variables can be removed from the model. The matching is described by the following decision variables:

$$A_{s,p} = \begin{cases} 1 & \text{if virtual track } s \in \mathcal{S} \text{ is assigned to physical track } p \in \mathcal{P}; \\ 0 & \text{otherwise.} \end{cases}$$

The parameters l^s describing the length of a track $s \in \mathcal{S}$ are replaced with l^p representing the length of the physical track $p \in \mathcal{P}$. For the new decision variables, the restrictions that result in a matching are:

$$\sum_{s \in \mathcal{S}} A_{s,p} = 1 \qquad \forall p \in \mathcal{P} \tag{7.22}$$

$$\sum_{p \in \mathcal{P}} A_{s,p} = 1 \qquad \forall s \in \mathcal{S} \tag{7.23}$$

$$A_{s,p} \in \{0,1\} \quad \forall s \in \mathcal{S}, p \in \mathcal{P} \tag{7.24}$$

Given this matching, SCHRIJVER [2003] rewrites the restrictions (7.14) on the track capacity as:

$$L_{t,s} \leq \sum_{p \in \mathcal{P}} l^p A_{s,p} \quad \forall t \in \mathcal{U}_+, p \in \mathcal{P} \tag{7.25}$$

This results in Model (7.c) consisting of (7.7)-(7.10),(7.12),(7.13),(7.15)-(7.18), (7.20)-(7.25). Note that (7.7)-(7.10),(7.12),(7.13), (7.15)-(7.18) remain unchanged, and are defined in terms of virtual tracks now.

Further reductions in the resulting models are possible by choosing a type of train unit to be assigned to a non-mixed shunt track [SCHRIJVER, 2003]. For types of train units, which occur frequently at a station it seems logical to assume that at least one track exists with only this type of train units parked at it. This holds especially for types without other types in the same family and long types. For a non-mixed virtual track $s \in \mathcal{S}''$ with a pre-assigned type ψ of train unit, restrictions (7.17) can be omitted while restrictions (7.16) simplify to $P_{t,s} = 0$ if $\psi_t \neq \psi$. The latter implies that restrictions (7.9) and (7.10) are only relevant for virtual track s if $\psi_t = \psi$. Note that if no mixed virtual tracks are allowed, we know that each type of train unit has at least one track which only consists of units of this type.

7.6 MODELS EXTENDED TO FREE TRACKS

Until now, we have assumed that the shunt tracks can only be approached from the A-side of the track. In this paragraph, we describe SCHRIJVER's extension of Model (7.c) to incorporate shunt tracks which can be approached from the B-side, and even from both sides of the track.

Before describing the extension of the model, we need to take a closer look at the ordering $<_A$ on the train units. In Figure 7.2, both the IRM_3 and IRM_4 as well as the ICM_3 and ICM_4 units arrive respectively depart in one train. Via the A-side, the partial ordering $<_A$ results in the order ①,②,③,④. For a track open at the B-side, this order should be: ②,①,④,③. Indeed, the IRM_3 unit arrives at a B-side open track before the IRM_4 unit. Therefore, the ordering $<_A$ is wrong for configurations with

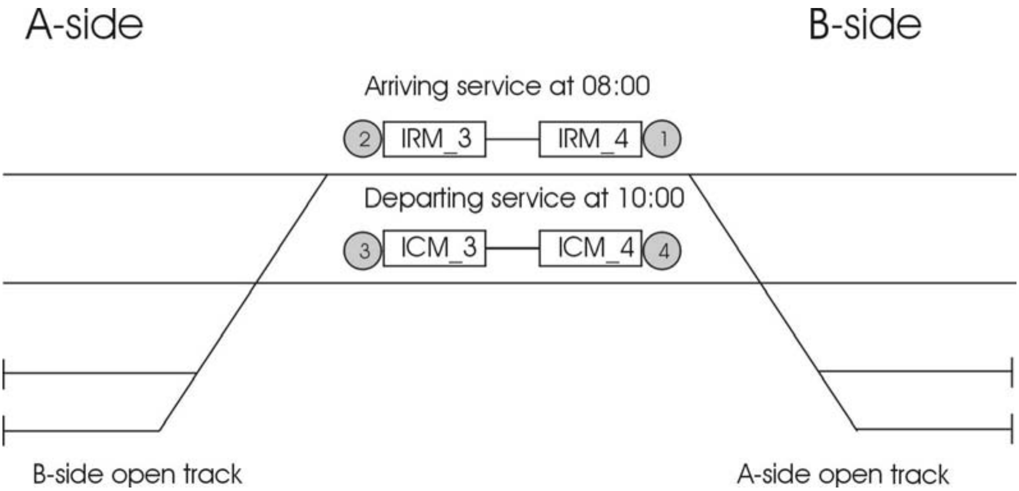


Figure 7.2: Tracks open at different sides may require a different ordering of train units within a train.

multiple train units and needs to be reversed in this case. This results in the partial ordering $<_B$ of the train units. The relation $u_i <_B u_j$ holds if and only if one of the following conditions is satisfied:

1. Unit u_i arrives or departs in a train service with an earlier planned time than the service to which unit u_j belongs.
2. Arriving units u_i and u_j arrive in the same train and unit u_i is closer to the A-side of the train than arriving unit u_j.
3. Departing units u_j and u_i depart in the same train and unit u_j is closer to the A-side of the train than departing unit u_i.

Note that the orderings $<_A$ and $<_B$ are the same for train units in different train configurations and only differ for train units in the same train configuration.

The set $\mathcal{P}$ is partitioned into the set $\mathcal{P}_A$ with tracks open at the A-side only, $\mathcal{P}_B$ with tracks open at the B-side only, and $\mathcal{P} \setminus \{\mathcal{P}_A \cup \mathcal{P}_B\}$ with tracks that can be approached from both sides.

As a straightforward extension, one could add two indices to the decision variables $M_{t,u,s}$, resulting in $M_{t,u,s,d,e}$ variables, where d describes the arrival side at track s used for unit t, and e describes the departure side at track s used for unit u. Subsequently, crossing restrictions, similar to restrictions (7.11) or (7.19), can be determined. Moreover, some restrictions for the sides of the tracks which can be used are required. However, since the number of decision variables roughly increases by a factor 4, this approach will not result in a model that is able to solve real-life instances. Therefore, Schrijver [2003] continues with an alternative approach. In this approach, one decision variable for each train unit is introduced, which indicates the side of a track via which the unit arrives or departs:

$$S_t = \begin{cases} 0 & \text{if train unit } t \text{ arrives or departs via the A-side of a track;} \\ 1 & \text{if train unit } t \text{ arrives or departs via the B-side of a track.} \end{cases}$$

Note that, compared to the straightforward extension, this results in a huge reduction in the number of decision variables, since the decision variables S_t make the additional indices d and e in the $M_{t,u,s,d,e}$ decision variables superfluous. One variable S_t is used for each train unit to denote the side of the shunt track at which the unit is parked, instead of explicitly taking into account the sides of all shunt tracks.

In order to extend the Model (7.b) to include free tracks, we need restrictions like $P_{t,s} - S_t \leq 0$ if track s is not accessible from the A-side, and $P_{t,s} + S_t \leq 1$ if track s is not accessible from the B-side. However, the model with virtual tracks, Model (7.c), requires the following restrictions:

$$P_{t,s} - S_t + \sum_{p \in \mathcal{P}_B} A_{s,p} \leq 1 \quad \forall s \in \mathcal{S}', t \in \mathcal{U} \tag{7.26}$$

$$P_{t,s} + S_t + \sum_{p \in \mathcal{P}_A} A_{s,p} \leq 2 \quad \forall s \in \mathcal{S}', t \in \mathcal{U} \tag{7.27}$$

Indeed, suppose that train unit t is parked at track s, resulting in $P_{t,s} = 1$ and virtual track s is assigned to a physical track $p \in \mathcal{P}_B$, which can only be approached from the B-side. Then, restrictions (7.26) imply $S_t = 1$, ensuring that unit t approaches track s from the B-side. Restrictions (7.27) can be explained similarly.

7.6.1 Trains with One Unit

For ease of discussion, Schrijver starts with the crossing restrictions in the special case that each train consists of exactly one train unit. Note, that in this case both orders $<_A$ and $<_B$ are the same. The resulting restrictions are generalized later on. For each $s \in \mathcal{S}$, and $((t,u),(v,w)) \in \mathcal{L}^2$, the conditions on S_t are:

$$\text{if } M_{t,u,s} = 1 \text{ and } M_{v,w,s} = 1 \text{ and } t <_A v <_A u <_A w, \text{ then } S_v \neq S_u \tag{7.28}$$

$$\text{if } M_{t,u,s} = 1 \text{ and } M_{v,w,s} = 1 \text{ and } v <_A t <_A u <_A w, \text{ then } S_t = S_u \tag{7.29}$$

Theorem 7.7. *If each train consists of exactly one unit, restrictions (7.28) and (7.29) are necessary and sufficient for describing the crossing restrictions for $s \in \mathcal{S}$, and $((t,u),(v,w)) \in \mathcal{L}^2$.*

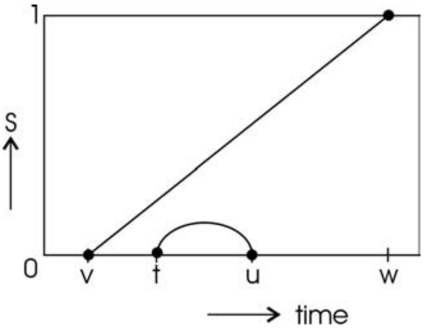


Figure 7.3: A case where (t, u) and (v, w) can be assigned to the same track.

Proof. The theorem is equivalent to showing that for each (t, u) and (v, w) in $\mathcal{L}$ with $\{t, u\} \cap \{v, w\} = \emptyset$, there exist disjunct curves in $\mathbb{R} \times [0, 1]$ from (t, S_t) to (u, S_u) and from (v, S_v) to (w, S_w) if and only if restrictions (7.28) and (7.29) hold. A case where this is possible is represented in Figure 7.3. Figure 7.4 gives the special case with $t <_A v <_A u <_A w$ and $S_v = S_u = 0$, where a crossing occurs indeed. Without loss of generality, one can assume $u <_A w$, and therefore $t <_A u <_A w$. Now, there are three possibilities for the position of v compared to t, u, and w. If $u <_A v <_A w$ then such curves can be drawn, because the train units do not have a common time interval at track s. Moreover, if $t <_A v <_A u$, such curves exist only if $S_v \neq S_u$, conform Figure 7.4. In addition, if $v <_A t$, these curves exist only if $S_v = S_u$. Exactly these restrictions on the S variables are described by restrictions (7.28) and (7.29). Note that if $v >_A w$ then v arrives later than w, which contradicts $(v, w) \in \mathcal{L}$. ∎

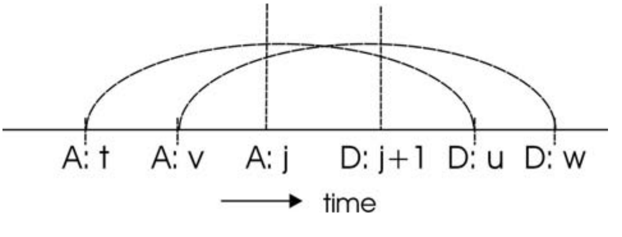


Figure 7.4: An example of a crossing at a specific LIFO track.

Restrictions (7.28) and (7.29) are rewritten in linear form for each $s \in \mathcal{S}$, and $((t, u), (v, w)) \in \mathcal{L}^2$, resulting in:

$$M_{t,u,s} + M_{v,w,s} \leq 3 - S_u - S_v \quad \text{if } t <_A v <_A u <_A w \tag{7.30}$$

$$M_{t,u,s} + M_{v,w,s} \leq 1 + S_u + S_v \quad \text{if } t <_A v <_A u <_A w \tag{7.31}$$

$$M_{t,u,s} + M_{v,w,s} \leq 2 - S_t + S_u \quad \text{if } v <_A t <_A u <_A w \tag{7.32}$$
$$M_{t,u,s} + M_{v,w,s} \leq 2 + S_t - S_u \quad \text{if } v <_A t <_A u <_A w \tag{7.33}$$

For example, restrictions (7.33) prohibit the situation depicted in Figure 7.5, where there is no possibility to park (v, w) at the same track as (t, u) without introducing a crossing.

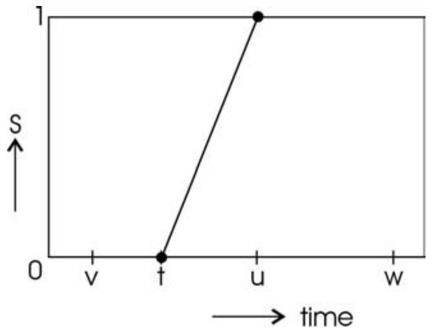


Figure 7.5: A graphical representation of restrictions (7.33).

7.6.2 Trains with Multiple Units

When trains possibly consist of several train units, the restrictions (7.30)-(7.33) generalize to the following, again for each $s \in \mathcal{S}$, and $((t, u), (v, w)) \in \mathcal{L}^2$ [SCHRIJVER, 2003]:

$$M_{t,u,s} + M_{v,w,s} \leq 1 + S_u + S_v \quad \text{if } t <_A v <_A u <_A w \tag{7.34}$$
$$M_{t,u,s} + M_{v,w,s} \leq 3 - S_u - S_v \quad \text{if } t <_B v <_B u <_B w \tag{7.35}$$
$$M_{t,u,s} + M_{v,w,s} \leq 2 - S_t + S_u \quad \text{if } v <_B t, u <_A w \tag{7.36}$$
$$M_{t,u,s} + M_{v,w,s} \leq 2 + S_t - S_u \quad \text{if } v <_A t, u <_B w \tag{7.37}$$
$$M_{t,u,s} + M_{v,w,s} \leq 3 + S_t - S_u - S_v \quad \text{if } t <_A v <_B t, u <_B w \tag{7.38}$$
$$M_{t,u,s} + M_{v,w,s} \leq 3 + S_w - S_u - S_v \quad \text{if } t <_B v, u <_A w <_B u \tag{7.39}$$
$$M_{t,u,s} + M_{v,w,s} \leq 2 - S_t + S_u + S_v \quad \text{if } v <_A t <_B v, u <_A w \tag{7.40}$$
$$M_{t,u,s} + M_{v,w,s} \leq 2 - S_w + S_u + S_v \quad \text{if } t <_A v, w <_A u <_B w \tag{7.41}$$
$$M_{t,u,s} + M_{v,w,s} \leq 3 + S_t - S_u - S_v + S_w \quad \text{if } t <_A v <_B t, u <_A w <_B u \tag{7.42}$$

We define Model (7.d) as the model consisting of (7.7)-(7.10),(7.12),(7.13),(7.15)-(7.18), (7.21)-(7.25),(7.26),(7.27),(7.34)-(7.42). Note that in restrictions (7.34) - (7.42) a relation like $u <_A w <_B u$ implies that the train units u and w belong to the same train. These restrictions are explained by the following discussion on restrictions (7.39).

A restriction (7.39) is only restrictive if $S_u = 1, S_v = 1$, and $S_w = 0$ since otherwise the righthand side is at least two, and the restriction is trivially fulfilled because $M_{t,u,s}$ and $M_{v,w,s}$ are binary. We know that units u and w will leave in the same train. In addition, we know that $v <_B u$ because otherwise $w <_B u <_B v$, which contradicts $(v, w) \in \mathcal{L}$. The situation at track s after both arrivals t and v is given in Figure 7.6, which is independent of the value of S_t. In this figure, the arrows from (t, u), respectively (v, w), represent the fact that these units need to depart from the B-side, respectively A-side, of shunt track s.

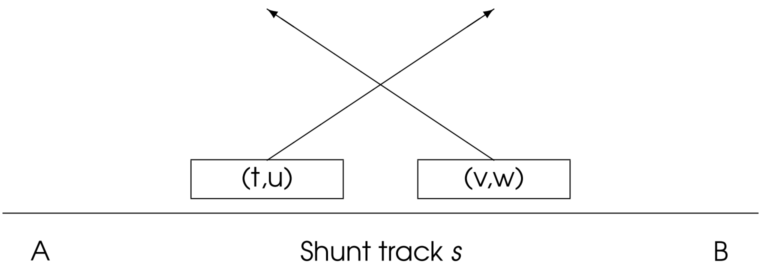


Figure 7.6: The situation at track s after both arrivals.

This results in a crossing because w needs to leave from the A-side of s, while u needs to leave from the B-side and the units are parked simultaneously at s.

Lentink [2004] showed that these restrictions are necessary by simple enumeration of the possible relations between two train units regarding the orders $<_A$ and $<_B$. Subsequently, Veldhorst [2004] showed these restrictions to be sufficient as well. Aggregations similar to the ones in Model (7.b) can be applied to reduce the numbers of these restrictions.

7.7 APPLICATIONS OF TUSP

In this paragraph, we present computational results of Models (7.a) - (7.d). The models are applied to the Tuesday / Wednesday instances at stations Zwolle and Enschede. The computational results of the matching problem TMP in Section 3.5 for these instances showed that approximately 50% of the resulting blocks do not need parking because the time difference between departure and arrival is sufficiently small. Since the models developed in this chapter require each train unit to be parked at a shunt track, we reduced the original instances by removing the train units in the blocks that do not need parking. Most of these blocks result from fixed assignments and can be determined unambiguously. However, some of the blocks result from the solution of the TMP and depend on the matching objective used. We choose to remove the blocks resulting from matching objective A with a sufficiently small time difference. The resulting reduced timetable for station Zwolle for a Tuesday / Wednesday instance contains 126 train units divided over 8 families and 12 types. Similarly, the reduced timetable at station Enschede contains 44 train units divided over 2 families and 3 types. Originally, station Zwolle contains 306 train units for the same 12 types. Station Enschede originally contains 81 train units from 3 families and 4 types.

We start the discussion of the computational results with the number of restrictions required for prohibiting crossings presented in Table 7.1. The relevant models are Models (7.a), (7.b), and (7.d). Note that the number of crossing restrictions for one track in Model (7.c) is equal to the similar number of Model (7.b). The table shows that aggregation of the crossing restrictions is extremely successful, reducing the number of restrictions with a factor 20. By looking at the instance at station Zwolle, it is also clear that Model (7.a) grows too large to solve real-life instances. Finally, we see that enabling the use of both sides of a free shunt track results in a significantly larger number of crossing restrictions.

	Zl: 19 tracks		Es: 13 tracks	
	Per track	In total	Per track	In total
Model (7.a)	21.150	401.850	2.294	29.822
Model (7.b)	1.023	19.437	129	1.677
Model (7.d)	18.833	357.827	2.452	31.876

Table 7.1: The number of crossing restrictions in TUSP.

For the computational results, we set the cost of a broken arriving train as well as a broken departing train, denoted with a, to 3. Moreover, the cost for each type of train unit in access of 1 parked at a track, denoted with g, is set to 1. The resulting models are solved using CPLEX. For these models, we changed some of the settings based on ideas by Schrijver [2003]. CPLEX is directed to branching on decision variables in the following order: first K_t variables, then $O_{\psi,s}$ variables, and finally $P_{t,s}$ variables. In the branching process, the first branch tries to set the variable to 1. The emphasis is put on finding a feasible solution within 3 hours of computation time. Moreover, the problem is probed aggressively. This means that substantial preprocessing time is devoted to considering the effects of fixing one variable on the bounds of the remaining problem. Finally, the objective function is perturbed. The computer that was used for these experiments has been introduced in Section 3.5.

In Table 7.2, we report computational results for different models at stations Zwolle and Enschede. The table contains the numbers of broken arrivals and departures, as well as the sum of the E_s variables in the column 'Type'. Moreover, the computation time is reported in seconds. In case the optimal solution has not been found in 3 hours, the indication (*) is added here. For the Models (7.c) and (7.d), we assumed 0 tracks with multiple types of train units parked at it. As mentioned before, this means that each train unit type has at least one virtual track, where only units of this type are parked. The instances at station Zwolle contains 12 types, while at station Enschede only 3 types.

	Station Zwolle				Station Enschede			
	A	D	Type	Time	A	D	Type	Time
Model (7.a)	N/A	N/A	N/A	N/A	0	0	2	434.02
Model (7.b)	2	3	14	10800 (*)	0	0	2	286.34
Model (7.c)	3	1	0	23.81	4	4	0	1.20
Model (7.d)	3	1	0	9.14	4	4	0	0.39

Table 7.2: Computational results for different models.

From this table, we can conclude that restricting solutions to tracks with only one type of train unit parked at it is very strong. Computation times decrease significantly at the cost of several broken trains which cannot be avoided. At station Zwolle, the first model cannot be solved within three hours of computation time. In this case, CPLEX requires all three hours for preprocessing the problem, and therefore not even a feasible solution has been found after three hours. This justifies the extensions described in this chapter. The results presented in this table represent two ends of a spectrum of possible models, where only Model (7.d) considers free tracks. On one hand, there are instances with much flexibility, but requiring large computation times. On the other hand, there are restrictive instances with low computation times. In order to support shunt planners in a practical way, a combination of these two extremes is required.

Now, we continue with some of these combinations of the extremes. Based on the number of pre-assigned shunt tracks, different cases are considered. For station Zwolle, we used the following ordered list of pre-assigned types:

1. ICM_3.
2. ICM_3.
3. SM90_2.
4. MDDM_4.
5. MAT64_4.

This list should be interpreted as follows: whenever the case consists of n pre-assigned types of train units, the types $1, \ldots, n$ are used in the instance to restrict virtual tracks

$1, \ldots, n$ to these types. The order and types of train units were selected based on the timetable, as well as results for solutions to instances with different numbers of mixed tracks and no tracks with pre-assigned types. For station Enschede, we selected 3 such pre-assigned types, and use the following order:

1. ICM_4.

2. ICM_3.

3. DM90_2.

To start with, we present computational results for Models (7.c) and (7.d), with at most one track with multiple types of train units. These results can be found in Table 7.3, where the first two columns describe the model and the number of non-mixed virtual tracks that were used. Columns 3 to 10 are the same as columns 2 to 9 of Table 7.2. Note that the instances at station Enschede contain at most 3 pre-assigned virtual tracks.

	# pre-assigned	Station Zwolle				Station Enschede			
		A	D	Type	Time	A	D	Type	Time
Model (7.c)	0	2	0	1	10800(*)	1	1	1	2.34
	1	2	0	1	101.48	1	1	1	1.39
	2	2	0	1	102.75	1	1	1	2.44
	3	2	0	1	91.33	1	1	1	1.30
	4	2	0	1	102.73	—	—	—	—
	5	2	0	1	152.13	—	—	—	—
Model (7.d)	0	2	0	1	10800(*)	1	1	1	12.19
	1	1	1	1	10800(*)	1	1	1	17.72
	2	2	0	1	10800(*)	2	0	1	13.19
	3	2	0	1	10800(*)	2	0	1	9.00
	4	2	0	1	10800(*)	—	—	—	—
	5	2	0	1	10800(*)	—	—	—	—

Table 7.3: Computational results with one track with multiple types of train units.

By analyzing Table 7.3, we see that the instances at station Zwolle of Model (7.d) are not solved to optimality within three hours of computation time. This is caused by the fact that the LP-relaxation is weaker for these instances than for the instances with LIFO tracks. At station Enschede, we also see that free track instances are more difficult to solve. However, since these instances are significantly smaller, they can be solved to optimality within 20 seconds of computation time. Moreover, using one virtual shunt track with multiple types of train units is insufficient to prevent broken trains.

Since broken trains are considered more important than tracks with multiple types of train units, we continue the computational experiments by increasing the number of mixed virtual tracks from 1 to 2. The results of these experiments are reported in Table 7.4.

	# pre-	Station Zwolle				Station Enschede			
	assigned	A	D	Type	Time	A	D	Type	Time
Model (7.c)	0	0	1	2	10800(*)	0	0	2	2.19
	1	0	1	2	10800(*)	0	0	2	1.59
	2	2	0	1	10800(*)	0	0	2	1.31
	3	0	1	2	10800(*)	0	0	2	0.84
	4	0	1	2	10800(*)	—	—	—	—
	5	0	1	2	10800(*)	—	—	—	—
Model (7.d)	0	0	0	2	1033.00	0	0	2	10.16
	1	0	0	2	598.03	0	0	2	19.18
	2	0	0	2	938.36	0	0	2	26.66
	3	0	0	2	213.14	0	0	2	49.66
	4	0	0	2	338.48	—	—	—	—
	5	0	0	2	504.49	—	—	—	—

Table 7.4: Computational results with two tracks with multiple types of train units.

For the instances at station Zwolle, we see that the instances with free tracks can be solved to optimality within somewhat more than 15 minutes. However, the instances with LIFO tracks cannot be solved to optimality within 3 hours of computation time. These results are opposite to the results reported in Table 7.3. Similar to the latter table, the negative results are caused by a weak LP-relaxation for the instances with LIFO tracks. We conclude that for the Zwolle instances the additional effort of modeling tracks as free tracks pays off. Observe that the instances with free tracks are able to avoid broken trains. At station Enschede, the instances with LIFO tracks already provide sufficient flexibility for avoiding broken trains. Therefore, computational results for the instances with LIFO tracks are better since a solution with the same quality is found faster.

When allowing up to 4 mixed virtual tracks, the found solutions have the same quality as the ones presented in Table 7.4. Moreover, the relations between the instances are similar to the results presented in this table. The tables for these instances can be found in Appendix A.3.

At station Zwolle, restricting tracks to LIFO tracks deteriorates the LP-relaxation and therefore a solution with satisfactory quality cannot be found within 3 hours of computation time. The model with free tracks finds an optimal solution. However, the model with free tracks requires significantly more computation time with additional mixed virtual tracks. For example, the model with 4 mixed virtual tracks and 4 preassigned types requires a little more than one hour to solve. This additional computation time is caused by the increased sizes of the instances. From Table 7.1, we see that allowing one more mixed virtual track requires 18.833 additional restrictions for the instance at station Zwolle.

At station Enschede, the instances with LIFO tracks and 4 tracks with multiple types of train units already provide sufficient flexibility and require on average 2.61 seconds of computation time. The instances with free tracks require on average 66.23 seconds of computation time.

Finally, we compare the results of this integrated approach with the sequential approach of matching followed by parking of train units. Recall that we decided to remove matchings which do not need parking based on the results of TMP with matching objective A. Therefore, we compare the results of the integrated approach to TUSP with the results of sequentially solving TMP and TAP for the instances based on matching objective A. In Section 4.7, we studied the effect of extending the objective in TAP to include robustness measures. This measure consisted of a penalty for parking two blocks with different configurations next to each other at the same shunt track and a reward for parking blocks which leave in the same departing train at the same track and in the right order.

Here, we apply slightly different objective elements. The first element is the number of configuration changes of train units parked at a shunt track. Compared to the sequential approach, the entities accounted for are the train units and not the blocks. Therefore, parking one block consisting of an ICM_3 and an ICM_4 train unit results in one configuration change. In Section 4.7, this would result in 0 configuration changes. Moreover, the models for TUSP described in this chapter count the number of types of train units to be parked at a track. For example, if a track is used from 10:00 to 16:00 for parking MAT64_2 units and during the night for parking ICM_3 units, this is penalized while these types of train units are not parked simultaneously at the track. In this case, the objective in TUSP is too restrictive. In addition, in case two types of train units are parked at a track simultaneously, the models for TUSP do not consider clustering of types at the track. For example, suppose 4 train units are parked at a track in the order ICM_3 ICM_4 ICM_3 ICM_4. This order will be penalized to the same extent as the order ICM_3 ICM_3 ICM_4 ICM_4. However, from a robustness point of view, the second order is preferred over the first. Similar observations are made by Hamdouni et al. [2004]. The second objective element is the number of broken arriving and departing trains. In Section 4.7, we only considered combining broken departing trains in the objective. Therefore, we need to put more emphasis on the number of broken departing trains compared to the number of broken arriving trains.

We choose to report the computational results for the integrated approach with 2 mixed virtual tracks and 1 type of train unit pre-assigned to a shunt track. These results are compared with the results for instances of TAP with a standard as well as an extended objective. The results can be found in Table 7.5. In this table, the computation times for the sequential approach equals the sum of the computation times for solving TMP and TAP.

	Station Zwolle				Station Enschede			
	A	D	Type	Time	A	D	Type	Time
.TAD	2	7	29	247	1	7	7	1
.TAO	2	1	10	222	1	5	5	1
Integrated	0	0	3	598	0	0	3	19

Table 7.5: A comparison of the sequential approach and the integrated approach.

The first thing that stands out in this table is that the integrated approach does not require any broken train. The sequential approach does require broken trains, especially at station Enschede. Moreover, we already noted in Section 4.7 that the extended objective is effective in combining blocks for the same departing train, and reducing the number of neighboring train units with different types. This is confirmed in this table. In the integrated approach, 3 type changes occur at two shunt tracks for both stations, while the sequential approach requires more. Finally, the higher quality solutions produced by the integrated approach come at a cost of additional computation time.

7.8 CONCLUSIONS

In this chapter, we considered the integrated problem of matching and parking train units, which is formulated as the Train Unit Shunting Problem (TUSP). These matching and parking problems have been described extensively in Chapters 3 and 4, and therefore only the main characteristics were repeated here. This introduction is followed by some results on the computational complexity of TUSP. It is clear that TUSP is $\mathcal{NP}$-complete, since this already holds for the matching subproblem as well as the parking subproblem. However, we show that a special case, called the Fixed Size Train Unit Shunting Problem, can be solved in polynomial time. In this special case, each train consists of one unit, a maximum number of train units are simultaneously parked at the shunt yard, all tracks are LIFO tracks, and each train unit has the same length. In addition, we present some related literature with regard to other applications for buses and trams aimed at discussing similarities and differences with TUSP.

Several models are discussed in this chapter, each with an increasing complexity. These models have been proposed by Schrijver [2003]. In the first model, tracks are restricted to LIFO tracks and each crossing explicitly results in one restriction in an integer program. As a first improvement, one can aggregate many of these restrictions, resulting in an integer program with a significantly lower number of restrictions. A second improvement is found by the introduction of virtual tracks, which are matched to the physical tracks. By selecting a restricted number of virtual tracks at which train units with different types can be parked, the number of restrictions can be reduced even further. This is caused by the fact that train units of the same type can be used interchangeably, and therefore no crossing restrictions are required for tracks with only one type of train unit parked at it. Since the restrictions are placed on the virtual tracks, one does not restrict physical tracks to have only one type of train unit parked at it. Therefore, the selection of the physical tracks with one type of train unit is left as a degree of freedom to the model.

Moreover, it is likely that planners are able to determine on beforehand that certain types of train units will have tracks with only this type parked at it. By taking this into account, the problem can be reduced even further. Finally, the restrictions on the free tracks are released, and we consider a model by SCHRIJVER, which considers free tracks as well. Free tracks require many additional crossing restrictions in the Integer Programs since more possibilities for crossings at free tracks exist.

Computational results at station Zwolle show that only the models with a restricted number of mixed virtual tracks are able to find high-quality solutions within minutes of computation times. When allowing one such a virtual track, the Tuesday / Wednesday instance requires 2 trains with units parked at different tracks. Such trains are called broken trains. In this case, the model with LIFO track configurations finds the optimal solution much faster than the model with the free track configurations. By adding more mixed virtual tracks, broken trains are avoided, with as an exception the model with LIFO tracks and 2 virtual tracks. For these instances, the models with free tracks are able to find the optimal solution faster than the models with LIFO tracks. Typically, these models require at most 10 minutes of computation time.

Regarding station Enschede, the first two basic models are able to find optimal solutions in at most 8 minutes of computation time. Moreover, two virtual tracks with multiple types of train units are necessary to prevent broken trains. For these instances, the LIFO tracks already provide sufficient flexibility and perform better since their computation times are lower than the instances based on the model with free tracks. The instances of the model with LIFO tracks are solved within 3 seconds.

Finally, we compared the integrated approach with the sequential approach. We showed that the integrated approach significantly improves the number of broken trains and the number of neighboring train units with different types. However, this result comes at the cost of additional computation time.

Regarding the support of shunt planners by the integrated model, we conclude that the insights of shunt planners are useful in selecting the right configuration of the shunt tracks and the number of tracks with multiple types of train units. Accurate insights are able to reduce the Integer Program significantly and therefore reduce the required computation time, without deteriorating the quality of the solution found.

The integrated models discussed in this chapter provide considerable improvements compared to the sequential approach. However, further research is required to extend the scope of these models and to reduce the computation times. Examples of such extensions are incorporating the internal cleaning process, as described in Chapter 6, and taking into account estimates of routing cost and preferences of shunt planners to park train units at certain shunt tracks.

Chapter 8

Conclusions and Further Research

An increase in demand for passenger railway transportation, new legislation restricting the environmental impact of railway operations at stations, and the introduction of competition in European railways make shunt planning a challenging problem to study. As discussed in Chapter 1, algorithmic decision support for shunt planners has significant potential value to increase the efficiency of the shunt planning process. This can be achieved mainly by reducing its throughput time. The research presented in this thesis proposes several mathematical models and algorithms, which can serve as a basis for an advanced shunt planning system.

This last chapter starts with an overview of the main results of the thesis. Subsequently, these results are related to the research questions posed in Section 1.8. The chapter is concluded with some limitations of the thesis and several interesting directions for further research and development steps in support for shunt planners from both a scientific as well as a practical point of view.

8.1 MAIN RESULTS

After introducing shunt planning in Chapter 2, we described models and algorithms for several subproblems at the operational level of planning in subsequent chapters. Although different subproblems are tightly intertwined with each other, shunt planning is decomposed into subproblems because of the operational complexity of the integrated problem. Moreover, without decomposition, the resulting models would be too large and complex to provide high-quality solutions in reasonable computation times. The models and algorithms have been applied to real-life instances at stations Zwolle and Enschede.

In Chapter 3, the problem of matching arriving train units to departing train units is described. Given a timetable with arriving and departing trains, the Train Matching Problem (TMP) is the problem of finding a matching of arriving units to departing units. In this problem, it is essential that the types of matched units correspond to each other and that the order of the types of units in a train is adhered to. Important characteristics of such a matching are the number of blocks and the number of blocks that need to be parked. With different weights in the objective, one is able to express a preference for solutions with desirable properties, such as matchings with a Last-In-First-Out (LIFO) character. In addition, a planner can add and remove fixed matchings of arriving train units to departing train units, and thereby influence the solution.

Although TMP in its most general form is $\mathcal{NP}$-hard, two special cases are shown to be solvable in polynomial time and space. The first special case restricts the number of train units in a train, the order of arrivals and departures and the objective. This case is solved by an uncapacitated minimum cost flow problem in a specialized network. The second special case restricts the maximum number of train units in a train, the maximum number of train units simultaneously parked at the shunt yard and the objective, and can be solved as a shortest path problem in an appropriate network. The developed model for the general problem integrates shortest path problems and a matching problem. Real-life instances of this model at stations Zwolle and Enschede can be solved by a general purpose solver within one second of computation time. These instances are based on the Dutch railway system in 1999/2000. The planning period starts at 08:00 and lasts 24 hours. We studied instances from Tuesday 08:00 to Wednesday 08:00 and from Saturday 08:00 until Sunday 08:00.

The parking of train units at shunt yards is the subject of Chapter 4, resulting in the definition of the Track Assignment Problem (TAP). When parking train units at shunt tracks, one should ensure that no train unit is blocking the arrival or departure of another train unit and the length of a shunt track is never exceeded by the length of the train units parked at it. By parking as little different types of train units as possible at one track, one is able to increase the robustness of a solution. In addition, opportunities for combining train units destined for the same departing train, but resulting from different arriving trains, might occur. Since these units can be routed to their departure platform as one entity, such a combination results in reduced usage of resources, such as crews, physical infrastructure capacity, and energy. Moreover, these opportunities also enable a reduction in the operational complexity of the shunt plans.

TAP in its general form is $\mathcal{NP}$-hard. In the developed model for TAP as a Set Partitioning Problem, a huge number of variables might occur. In order to be able to solve the practical instances of interest with this model, we applied a column generation heuristic, using dynamic programming in a network with a specialized structure. An instance of TAP is based on a matching, which is a solution to TMP. Most instances are solved within one minute. However, the Tuesday / Wednesday instances at station Zwolle require at most 10 minutes computation time. Changing the objective of TAP to take into account robustness and efficiency of a solution mostly affects the Tuesday / Wednesday instances and especially at station Zwolle. Also, computational results show that preferring LIFO matchings in TMP typically results in better solutions to TAP.

Chapter 5 is devoted to the Shunt Routing Problem (SRP) of routing train units over the station railway infrastructure. This routing is required to and from shunt tracks. Moreover, while other shunt processes such as e.g. cleaning, might require additional routing. A resulting shunt route should not conflict with the routes of through trains or with other shunt routes. An important characteristic of this problem is that start times and end times are flexible to some extent. For example, the last arriving train at a platform track can be routed to a shunt track in a certain time interval starting some time after the arrival. Typically, planners try to minimize the traveled distance, the number of changes in direction, and the deviations from the desired start times. The first two objectives add to the overall objective of efficiently using available resources, in particular the railway infrastructure and the number of shunt drivers. The last objective element of a minimum number of deviations from desired start times adds to the robustness of a shunt plan.

In order to solve SRP, a heuristic extension of the A* Search algorithm has been developed, resulting in Occupied Network A* (ONA*) Search. In ONA* Search, nodes might be unavailable during certain time intervals. Resembling the current practice of shunt planners, the developed heuristic finds shunt routes on a train-by-train basis. Obviously, the quality of the resulting set of routes is influenced by the order in which the shunt routes are planned. A simple 2-OPT interchange heuristic on the order of the shunt routes has been implemented to reduce this influence. Computational tests showed that solutions for the Tuesday / Wednesday instances at station Zwolle require up to 5 minutes, including 2 applications of the interchange heuristic, with some routes left unplanned by the algorithm. Especially for these instances, the creativity of shunt planners is required to find good solutions. Similar instances at station Enschede indicate that all shunt routes are found within seconds of computation time by applying the 2-OPT interchange heuristic only once. We concluded that the combination of ONA* Search with the interchange heuristic provides a solid basis for a tool that supports shunt planners in finding shunt routes over local railway infrastructure.

Cleaning of rolling stock at a station has been studied in Chapter 6, resulting in the formulation of the Shunt Unit Cleaning Problem (SUCP). All train units that lay over at a shunt yard need to be cleaned internally. This cleaning takes place along a dedicated cleaning platform and is performed by shifts of cleaning crews. Since the number of crews working in a shift varies and these shifts typically overlap in time, the required time for cleaning a train unit varies over time. While planning the order of cleaning of train units, one tries to maximize the number of trains that is cleaned at a moment close in time to their arrival time at the station. This adds to the overall objective to start up the railway operations as smoothly as possible in the morning. Moreover, train units are not required to be parked at a shunt track when cleaning is due close to its arrival time. In turn, this results in reduced required resources, reduced complexity of the shunt plans, and reduced noise levels at the shunt yard.

Before introducing a model for the planning of the cleaning process, we reported two results on the computational complexity of SUCP. In general, the decision version of this problem is $\mathcal{NP}$-complete in the strong sense. In addition, a special case with relaxed release times and deadlines and a certain structure of the availability of cleaning crews remains weakly $\mathcal{NP}$-complete. We propose a time-indexed Integer Program (IP) for solving SUCP. The combination of a varying capacity of the cleaning crews with the new objective to schedule as many jobs as possible close to their release times, in such an IP results in a scientifically challenging model.

The model is tested with several instances at station Zwolle, resulting in optimal solutions computed in at most 35 seconds in all but one instance. In order to capture the dynamics of the overall shunt planning, we studied the effect of the cleaning schedule on TAP. Although the algorithm presented in Chapter 4 is still able to park all train units at the yard, computation times rose significantly, since many blocks are split into two blocks. In order to overcome this additional computation time, a 2-OPT interchange heuristic for generating an initial set of columns is applied. This heuristic is similar to the one used in Chapter 5. Computation times decreased to previous levels, while the quality of the solutions remained comparable.

Finally, Chapter 7 describes an approach for integrating the matching and parking subproblems of Chapters 3 and 4 respectively. It is clear that TUSP is $\mathcal{NP}$-hard, since this is already the case for the subproblems TMP and TAP. We showed that a special case can be solved in polynomial time and space. The special case is restricted to instances with LIFO tracks, one unit per train, a maximum number of units to be parked at the shunt yard, and all types of train units having the same size. Moreover, the objective is to minimize the estimated routing costs, while parking all train units. The special case can be solved as a shortest path problem in a network with specific arc costs.

Several models in this approach are developed by Schrijver [2003]. These models result in IPs. Computational results show that knowledge of shunt planners provides ample opportunities for reduction of computation times. For station Zwolle, this knowledge is crucial for finding high-quality solutions within e.g. 15 minutes of computation time. The potential of the integrated approach is shown by comparing its solutions with the solutions of sequentially solving TMP and TAP. By solving TUSP in an integrated manner, better solutions are available compared to the sequential approach, at the cost of additional computation time.

The models and algorithms in this thesis provide a first step towards algorithmic decision support of shunt planners. In such a setting, most of the "routine work" in shunt planning can be performed by the algorithms and a reduction in the throughput time of shunt planning lies within reach. In turn, algorithmic decision support facilitates a speed up of this bottleneck in the current planning process at NS Reizigers.

However, caused by different peculiarities at different stations and the complexity of shunt planning, it is very unlikely that the entire shunt planning can be fully automated. Therefore, although the computation times for the presented algorithms are lower than for manual planning, some modifications need to be made by shunt planners. Such modifications can be made manually, or by the application of these algorithms on modified problems.

8.2 ANSWERS TO THE RESEARCH QUESTIONS

In Section 1.8, we introduced the central research question of this thesis as well as a breakdown of this question into several sub-questions. In this paragraph, we first relate the results described in the previous paragraph to the sub-questions. Thereafter, we formulate an answer to the central research question, based on the answers to the sub-questions.

1. *What are the important aspects of shunting and shunt planning?*

Operational shunt planning is a critical part of the planning process of a railway passenger operator, because it is highly sensitive to changes in previous planning processes, such as timetabling and the planning of the rolling stock circulation. Therefore, it is typically the bottleneck of the planning process. The overall planning process will benefit especially from improvements in the throughput time of the shunt planning.

The most important shunt processes are routing, parking, cleaning and maintenance of train units, and scheduling of shunting crews. Typically, these processes are tightly intertwined with each other. However, they each are already so complex that it is not wise to consider them as a whole. In the thesis, we focused on the shunt planning processes regarding rolling stock. This implies that the scheduling of shunting crews is only considered implicitly.

The main goal of operational shunt planning is to enable a smooth start-up of the railway operations in the morning, while certain restrictions with respect to the shunt processes have to be met. This goal is related to the fact that shortly after the start-up of the railway operations in the morning, the demand for railway transportation is at its maximum for the day. Therefore, disturbances in this start-up can easily spread out during the day and throughout the railway network. Another important goal is the robustness of operational shunt plans, since railway operations are likely to endure disturbances. By increasing the robustness of operational shunt plans, these plans become less vulnerable to such disturbances. Finally, shunt planners try to create efficient shunt plans, which require a minimum amount of resources, such as crews, railway infrastructure, and energy.

When generating shunt plans, several restrictions have to be taken into account. Here, we briefly mention the most important ones. When parking train units at a shunt track, one has to ensure that no train unit blocks the arrival or departure of another unit. Moreover, the prescribed composition of departing trains has to be respected: type mismatches or a wrong order of the right types of train units in a train are not allowed. Finally, the maximum railway noise level for shunting processes is more and more restricted. This influences the flexibility of the shunting operations.

2. *How can the quality of a shunt plan be measured?*

The answer to the previous question mentioned robustness and efficient resource usage as the most important objectives in shunt planning. By answering this research question, we provide means to quantify these characteristics.

When considering the parking subproblem of shunt planning, robustness of shunt plans is taken into account by parking units with the same type at the same track. In case the operations are disturbed and all train units parked at a track have the same type, the order of train units at a track is irrelevant. In addition, the smooth start-up of the railway operations in the morning is supported by combining train units from different arriving trains for the same departing train at a shunt track, and by minimizing the number of train units that finish cleaning just before these have to depart in the timetable. Cleaning rolling stock just before it departs the station increases the dependencies between the cleaning process at a shunt yard and the network-wide railway services. Indeed, these dependencies reduce the robustness of the shunt plans and should be minimized. In addition, changes in direction in shunt routes increase the complexity of the routing activities. An increased complexity of the operations has a negative impact on the robustness of the shunt plans and therefore should be avoided.

Efficient usage of resources is also taken into account in all presented models for subproblems of shunt planning. When matching train units, the matching model aims at a minimum number of resulting entities. In addition, a maximum number of entities with little time difference between arrival and departure is also aimed at. The reason is that entities with little time difference between arrival and departure do not need parking. In turn, this results in less work for the shunting crews. Moreover, by applying a Last-In-First-Out (LIFO) structure to the resulting matching, less resources are required for routing the train units, since these can typically use the preferred sides of the shunt tracks. As mentioned before, the combination of train units from different arriving trains for the same departing train increases robustness. In addition, it requires less shunting crews and railway infrastructure capacity. Indeed, such combined train units can be considered one entity after the arrivals at the shunt track, and therefore the train units can be routed as one entity to the departure platform. In addition, the proposed algorithm for routing train units penalizes traveled distance as well as changes in direction. These penalties result in solutions where railway infrastructure capacity and shunting drivers are used efficiently.

Finally, cleaning train units close in time to their arrivals ensures that these units are only parked after their cleaning. In case a train unit needs to wait too long before it is cleaned, it needs to be parked at the shunt yard, resulting in additional routing and additional capacity of infrastructure and shunting drivers.

3. *Which mathematical models can properly support the shunt planners?*

The mathematical models developed in this thesis, can properly support the shunt planners. Note that the different models differ in resemblance with current practice of shunt planners as well as in mathematical complexity. For example, the algorithm for routing train units, described in Chapter 5, is close to the current practice of shunt planners, while the integrated model of Chapter 7 is further away.

It is important that planners have some control over the algorithms and models. By changing parameters, objectives and restrictions, shunt planners are able to influence the solution process. Examples of such control mechanisms are fixing parts of the matching of train units, and requiring train units to be parked at a specific track.

In order to support the shunt planners, the mathematical models should be able to plan as many entities as possible, besides considering the objectives of the shunt plans as described in the answer of the previous question. In this manner, shunt planners can focus on bottlenecks in a shunt plan, and on creatively developing several scenarios and choosing the best one for these bottlenecks, while algorithms perform the routine work.

In addition, a trade-off between computation times and quality of the resulting solution is relevant. Since it is important to find acceptable solutions fast, computation times can be reduced at the cost of solution quality.

4. *How can we efficiently solve these mathematical models?*

The models for matching, cleaning and integrated matching and parking resulted in Integer Programs (IPs). Typically, these IPs can be solved satisfactorily by general-purpose solvers. The model for parking also resulted in an IP. However, the size of this IP is too large to be handled by such general-purpose solvers. Therefore, a special-purpose solver has been developed for this problem, which uses a general-purpose solver for subproblems. In addition, for the routing problem, a special-purpose solver has been developed, taking advantage of the specific characteristics of this problem.

Based on the practical aspects of the operational shunt planning process as described in Chapter 2 and the above answers to the sub-questions, we answer the central research question:

What are appropriate quantitative models and algorithms for supporting shunt planners?

Keeping in mind the practice of operational shunt planning, and having answered the sub-questions of the thesis, we conclude that the models and algorithms described in the thesis provide a first step in algorithmic decision support for shunt planners. Although the presented models differ in the similarities with current practice, all models have the potential to provide support for relevant aspects of shunt planning and give opportunities for shunt planners to influence the solution process. Moreover, since computation times are typically at most several minutes, the models and algorithms can facilitate a reduction in throughput time of shunt planning. This is relevant since approximately 130 planners are involved in shunt planning and it is the bottleneck of the current planning process at NS Reizigers.

Practical applicability of these algorithms require that they are embedded in a user-friendly decision support system. This decision support systems should be able to provide user-friendly opportunities for planners to tune the algorithms and to modify shunt plans. Moreover, communication with other systems for retrieving input and communicating resulting shunt plans need to be addressed. Finally, since each shunt planner develops his own routines for solving shunt plans, human machine interaction needs to be taken into account and training of planners is required.

However, it is unlikely that all aspects of shunting can be modeled and the resulting models can be solved fast. Therefore, shunt planners will also be actively involved in the generation of future shunt plans. As mentioned before, the developed models and algorithms in this thesis provide a basis for future algorithmic decision support of shunt planners. Work on extending the scope of these models remains to be done before measurable practical results can be achieved. Therefore, we conclude that the quantitative models and algorithms described in this thesis provide a first step for advanced planning support for shunt planners.

8.3 SUGGESTIONS FOR FURTHER RESEARCH

Regarding the application of the models and algorithms described in this thesis, several directions are worth further exploration. Here, we describe the most important ones. Moreover, we also point at some general areas for future attention in railways.

Developed models and algorithms in the thesis are tested based on real-life instances at stations Zwolle and Enschede. Although station Zwolle is one of the most complex stations in the Netherlands regarding shunting, each station has its own peculiarities. Therefore, it would be worthwhile to consider other stations and to investigate whether refinements or adjustments in the presented models and algorithms are required.

An important limitation of the thesis is the fact that shunting crews are only taken into account implicitly. It is obvious that the shunting crews are an important resource in shunting and its planning. Therefore, future research on the precise relations between shunting crew planning and other shunt processes, described in the thesis, will provide valuable insights, from a scientific perspective as well as from a practical one.

In this thesis, we focused on the operational shunt planning problem. It would be interesting to see to what extent the developed models and algorithms could be used during the operations, where decisions have to be taken within minutes. Therefore, we see further investigation of fast heuristics as an important direction for future research. During the operations, even more emphasis is put on generating acceptable shunt plans quickly instead of finding optimal shunt plans. In case suitable algorithmic support of the operations has been developed, one might choose to create less detailed operational shunt plans and rely more heavily on the operations for finding shunt plans with acceptable quality. In particular since stochastic elements play a predominant role in the operations.

In relation with the previous subject, actual realizations of arrivals and departures provide good means to compare the robustness of different shunt plans. This comparison could result in refinements of the current robustness measures, and / or the introduction of additional ones.

The focus of the models and algorithms in the thesis is on applications for shunting of train units. It might be not that hard to extend this focus to tackle instances of shunting of e.g. buses and trams. Advantages of the models and algorithms in this thesis over models and algorithms known in the scientific literature are possibilities to take into account local processes at a shunt yard, and the generalization of the shunt tracks, where both sides of a shunt track might be used for arrivals and departures.

A different, but very relevant, further step is the application of the models and algorithms in the actual planning process at NSR as described earlier. This requires a convenient graphical user-interface, interfaces with other planning systems for retrieving input data and communicating the resulting shunt plans. Moreover, shunt planners need to be trained to work with the models and algorithms, while the user-friendliness of the developed algorithms needs to be improved.

From a more general perspective, innovations in information and communication technology will improve the services to passengers. Such innovations include better information to passengers, for example in case of delays, but also innovations in the operations will improve available real-time data, which will improve algorithmic decision support of the operations, thereby reducing the complexity of the operations.

Bibliography

I. Adiri and O. Hamberg (1998). *Openshop scheduling under linear resources constraints*. Naval Research Logistics, 45(1):51–66.

Adviesdienst Verkeer en Vervoer website (2005). http://www.rws-avv.nl/vv2020. In Dutch.

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin (1993). *Network flows: theory, algorithms and applications*. Prentice Hall, New Jersey, United States.

H.F. Amaddeo, W.M. Nawijn, and A. van Harten (1997). *One-machine job-scheduling with non-constant capacity - Minimizing weighted completion times*. European Journal of Operational Research, 102(3):502–512.

M. Avriel and M. Penn (1993). *Exact and approximate solutions of the container ship stowage problem*. Computers and Industrial Engineering, 25(1–4):271–274.

M. Avriel, M. Penn, and N. Shpirer (2000). *Container ship stowage problem: Complexity and applications to coloring of circle graphs*. Discrete Applied Mathematics, 103(1–3):271–279.

M. Avriel, M. Penn, N. Shpirer, and S. Witteboon (1998). *Stowage planning for container ships to reduce the number of shifts*. Annals of Operations Research, 76:55–71.

K.R. Baker and H.L.W. Nuttle (1980). *Sequencing independent jobs with a single resource*. Naval Research Logistics Quarterly, 27:499–510.

C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance (1998). *Branch-and-price: Column generation for solving huge integer programs*. Operations Research, 46(3):316–329.

A. Billionnet (2003). *Using integer programming to solve the train-platforming problem*. Transportation Science, 37(2):213–222.

U. Blasum, M.R. Bussieck, W. Hochstättler, C. Moll, H.H. Scheel, and T. Winter (2000). *Scheduling trams in the morning*. Mathematical Methods of Operations Research, 49(1):137–148.

A. Boccalatte, R. Minciardi, M. Paolucci, and R. Pesenti (1994). *Optimal train scheduling in presence of track circuits*. In T.K.S. Murthy, B. Mellitt, C.A. Brebbia, G. Sciutto, and S. Sone (editors), *Proceedings of COMPRAIL 1994*, volume 2, pages 73–80. WIT Press, Southampton, United Kingdom.

M. Bóna (2003). *A survey of stack-sorting disciplines*. Electronic Journal of Combinatorics, 9(2):#A1. `http://www.combinatorics.org/Volume_9/PDF/v9i2a1.pdf`.

D. Briginshaw (2004). *Govia's strategy to improve performance at South Central*. International Railway Journal, 2:10–11.

R. Brooks and A.M. Geoffrion (1966). *Finding Everett's Lagrange multipliers by linear programming*. Operations Research, 14(6):1149–1153.

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

A. Cardol and M. Fledderus (2002). *Opstellen: prognose op basis van Referentie 2010 (Parking: forecast on the basis of Referentie 2010)*. Technical report, Railned. Internal document in Dutch.

M. Carey (1994). *A model and strategy for train pathing with choice of lines, platforms and routes*. Transportation Research Part B, 28(5):333–353.

M. Carey and S. Carville (2003). *Scheduling and platforming trains at busy complex stations*. Transportation Research Part A, 37(3):195–224.

V. Chvátal (1983). *Linear Programming*. Freeman, New York, United States.

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

T.G. Crainic and J.M. Rousseau (1987). *The column generation principle and the airline crew scheduling problem*. INFORS, 25(2):136–151.

A. Crockett and J. Carlisle (2003). *West Rail sets new standards in lineside noise control*. Railway Gazette International, 9:579–580.

E. Dahlhaus, P. Horak, M. Miller, and J.F. Ryan (2000). *The train marshalling problem*. Discrete Applied Mathematics, 103(1–3):41–54.

R.L. Daniels and R.K. Sarin (1989). *Single machine scheduling with controllable processing times and number of jobs tardy*. Operations Research, 37(6):981–984.

G.B. Dantzig and P. Wolfe (1960). *Decomposition principle for linear programs*. Operations Research, 8(1):101–111.

D. De Luca Cardillo and N. Mione (1998). k L-*list* τ *colouring of graphs*. European Journal of Operational Research, 106(1):160–164.

R. Dechter and J. Pearl (1983). *The optimality of A* revisited*. In M.R. Genesereth (editor), *Proceedings of the National Conference on Artificial Intelligence*, pages 95–99. AAAI Press, Menlo Park, United States.

G. Desaulniers, J. Desrosiers, and M.M. Solomon (2005). *Column Generation*, volume 5 of *GÉRAD 25th Anniversary Series*. Springer-Verlag, Berlin, Germany.

J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis (1995). *Time constrained routing and scheduling*. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (editors), *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35–139. North-Holland, Amsterdam, the Netherlands.

J. Desrosiers, F. Soumis, and M. Desrochers (1984). *Routing with time windows by column generation*. Networks, 14:545–565.

G. Di Stefano and M.L. Kŏci (2003). *A graph theoretical approach to the shunting problem*. Electronic Notes in Theoretical Computer Science, 92(1):16–33.

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

O. Dubrovsky, G. Levitin, and M. Penn (1999). *A genetic algorithm with a compact solution encoding for the container ship stowage problem*. Journal of Heuristics, 8(6):585–599.

E. Duinkerken (2003). *Toets op opstelcapaciteit (Check on parking capacity)*. Master's thesis, University of Amsterdam, the Netherlands. In Dutch.

M.E. Dyer and L.A. Wolsey (1990). *Formulating the single machine sequencing problem with release dates as a mixed integer program*. Discrete Applied Mathematics, 26(2–3):255–270.

J.M. Egbers (2001). *Knelpunten op knooppunten (Bottlenecks at junctions)*. Technical report, Technical University, Eindhoven, the Netherlands. In Dutch.

European Commission (1991). *Directive 91/440/EEC*. Official Journal of the European Communities.

——— (2001). *European transport policy for 2010: time to decide*.

P.J. Fioole (2003). *Het automatisch genereren en verbeteren van een rangeerplanning*. Master's thesis, Erasmus University, Rotterdam, the Netherlands. In Dutch.

I. Fodor Birtalan (2003). *Matching arrivals to departures in the train unit shunting problem*. Master's thesis, Erasmus University, Rotterdam, the Netherlands.

R. Freling (1997). *Models and techniques for integrating vehicle and crew scheduling*. Ph.D. thesis, Tinbergen Institute, Erasmus University, Rotterdam, the Netherlands.

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

R. Freling, R.M. Lentink, and A.P.M. Wagelmans (2004). *A decision support system for crew planning in passenger transportation using a flexible branch-and-price approach*. Annals of Operations Research, 127:205–224.

M. Galaverna, A. Imovilli, and G. Sciutto (1994). *An object-oriented approach to the train platforming problem*. In T. Murthy, B. Mellitt, C.A. Brebbia, G. Sciutto, and S. Sone (editors), *Proceedings of COMPRAIL 1994*, volume 1, pages 19–25. WIT Press, Southampton, United Kingdom.

G. Gallo and F. Di Miele (2001). *Dispatching buses in parking depots*. Transportation Science, 35(3):322–330.

M.R. Garey and D.S. Johnson (1979). *Computers and intractability, a guide to the theory of $\mathcal{NP}$-completeness*. Freeman, New York, United States.

L. Gelders and P.R. Kleindorfer (1973). *Coordinating aggregate and detailed scheduling in the one-machine job shop: Part I. Theory*. Operations Research, 22(1):46–60.

——— (1975). *Coordinating aggregate and detailed scheduling in the one-machine job shop: Part II. Computation and structure*. Operations Research, 23(2):312–324.

J.W.H.M. Goossens (2004). *Models and algorithms for railway line planning models*. Ph.D. thesis, University of Maastricht, the Netherlands.

S.K. Gupta and J. Kyparisis (1987). *Single machine scheduling research*. Omega, 15(3):207–227.

A. Haghani and Y. Shafahi (2002). *Bus maintenance systems and maintenance scheduling: model formulations and solutions*. Transportation Research Part A, 36(5):453–482.

R. Haijema, C.W. Duin, and N.M. van Dijk (2005). *Train shunting: A practical heuristic inspired by dynamic programming*. Technical report, University of Amsterdam, the Netherlands.

M. Hamdouni, G. Desaulniers, O. Marcotte, F. Soumis, and M. van Putten (2004). *Dispatching buses in a depot using block patterns*. Technical Report G-2004-051, accepted for publication in Transportation Science, GÉRAD, Montréal, Canada.

M. Hamdouni, G. Desaulniers, and F. Soumis (2005). *Parking buses in a depot using block patterns: a Benders decomposition approach for minimizing type mismatches*. Technical Report G-2005-70, accepted for publication in Computers and Operations Research, GÉRAD, Montréal, Canada.

P.E. Hart, N.J. Nilsson, and B. Raphael (1968). *A formal basis for the heuristic determination of minimum cost paths*. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107.

S. He, R. Song, and S.S. Chaudhry (2000). *Fuzzy dispatching model and genetic algorithms for railyards operations*. European Journal of Operational Research, 124(2):307–331.

——— (2003). *An integrated dispatching model for railyard operations*. Computers and Operations Research, 30(7):939–966.

E.H.G. Heimerl (1997). *De spoorwegen in het grensoverschrijdende verkeer (Railways in cross-border traffic)*. Railforum. Dutch summary of the German report *Die Eisenbahn im Grenz überschreitenden Verkehr*.

T. Hirayama and M. Kijima (1992). *Single machine scheduling problem when the machine capacity varies stochastically*. Operations Research, 40(2):376–383.

W. Hoekert (2001). *Het maken van diensten voor rangeerpersoneel (Creating schedules for shunting crew)*. Master's thesis, Erasmus University, Rotterdam, the Netherlands. In Dutch.

D. Huisman (2004). *Integrated and dynamic vehicle and crew scheduling*. Ph.D. thesis, Tinbergen Institute, Erasmus University, Rotterdam, the Netherlands.

D. Huisman, R. Jans, M. Peeters, and A.P. Wagelmans (2005a). *Combining column generation and Lagrangian relaxation*. In G. Desaulniers, J. Desrosiers, AND M.M. SOLOMON (editors), *Column Generation*, volume 5 of *GÉRAD 25th Anniversary Series*, pages 247–270. Springer-Verlag, Berlin, Germany.

D. HUISMAN, L.G. KROON, R.M. LENTINK, AND M.J.C.M. VROMANS (2005b). *Operations Research in passenger railway transportation*. Statistica Neerlandica, 59(4):467–498.

ILOG CPLEX WEBSITE (2004). `http://www.ilog.com/products/cplex`.

INTERNATIONAL RAILWAY JOURNAL (2003). *Full fleet maintenance the Bombardier way*. International Railway Journal, 10:32–33.

R.M. KARP (1972). *Reducibility among combinatorial problems*. In R.E. MILLER AND J.W. TATCHER (editors), *Complexity of computer computations*, pages 85–103. Plenum Press, New York, United States.

D.E. KNUTH (1997). *Fundamental algorithms*, volume 1 of *The art of computer programming*. Addison Wesley, Reading, United States, third edition.

M. KNUTTON (2003). *Siemens adopts pit-stop style maintenance*. International Railway Journal, 10:34–35.

L.G. KROON (2001). *Opsporen van sneller naar beter; modelling through (Trekking from faster to better; modelling through)*. Erasmus University, Rotterdam, the Netherlands. Inaugural address. In Dutch.

L.G. KROON, H.E. ROMEIJN, AND P.J. ZWANEVELD (1997). *Routing trains through railway stations: complexity issues*. European Journal of Operational Research, 98(3):485–498.

B. KUENEN (2003). *Vervoer gevaarlijke stoffen versus stedelijke verdichting (Transport of hazardous substances versus urban condensation)*. Trens, 5:12–15. In Dutch.

E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN, AND D.B. SHMOYS (1993). *Sequencing and scheduling: algorithms and complexity*. In S.C. GRAVES, A.H.G. RINNOOY KAN, AND P.H. ZIPKIN (editors), *Logisitics of Production and Inventory*, volume 4 of *Handbooks in Operations Research and Management Science*, pages 445–522. North-Holland, Amsterdam, the Netherlands.

J. LEE (2003). *Smart products and service systems for e-business transformation*. International Journal of Technology Management, 26(1):45–62.

J.K. LENSTRA AND A.H.G. RINNOOY KAN (1979). *Computational complexity of discrete optimization problems*. In P.L. HAMMER, E.L. JOHNSON, AND B.H. KORTE (editors), *Discrete Optimization 1*, volume 4 of *Annals of Discrete Mathematics*, pages 121–140. North-Holland, Amsterdam, the Netherlands.

J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker (1977). *Complexity of machine scheduling problems*. In P.L. Hammer, E.L. Johnson, B.H. Korte, and G.L. Nemhauser (editors), *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pages 343–362. North-Holland, Amsterdam, the Netherlands.

R.M. Lentink (2004). *An analysis of the restrictions (26) from the paper "Planning van opstelsporen" by A. Schrijver*. Unpublished note.

R.M. Lentink, P.J. Fioole, L.G. Kroon, and C. van 't Woudt (2003). *Applying Operations Research techniques to planning of train shunting*. Technical Report ERS-2003-094-LIS, Erasmus University, Rotterdam, the Netherlands.

T. Li (2005). *Smart card impact on transportation yield*. Technical report, Erasmus University, Rotterdam, the Netherlands.

S. Lin and B.W. Kernighan (1973). *An effective heuristic algorithm for the Traveling Salesman Problem*. Operations Research, 21(2):498–516.

M.E. Lübbecke and J. Desrosiers (2004). *Selected topics in column generation*. Technical Report 2004/08, accepted for publication in Operations Research, Technical University, Berlin, Germany.

M.E. Lübbecke and U.T. Zimmermann (2005). *Shunting minimal rail car allocation*. Computational Optimization and Applications, 31(3):295–308.

G.F. Luger and W.A. Stubblefield (1998). *Artificial Intelligence*. Addison-Wesley, Reading, United States.

G. Maróti (2006). *Operations Research models for railway rolling stock planning*. Ph.D. thesis, Technical University, Eindhoven, the Netherlands.

G. Maróti and L.G. Kroon (2004). *Maintenance routing for train units: the scenario model*. Technical Report PNA-E0414, accepted for publication in Computers and Operations Research, CWI, Amsterdam, the Netherlands.

——— (2005). *Maintenance routing for train units: the transition model*. Transportation Science, 39(4):518–525.

J.P. Masse (2005). *Paris depot reconstructed to maintain TGV Est Européen fleet*. Railway Gazette International, 3:149–150.

G.L. Nemhauser and L.A. Wolsey (1988). *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, United States.

NS (Nederlandse Spoorwegen) (2000). *De Koppeling, nr. 1619, 12-16-2000*. Utrecht, The Netherlands.

J.B. Orlin (1993). *A faster strongly polynomial minimum cost flow algorithm*. Operations Research, 41(2):338–350.

J. Pearl (1984). *Heuristics: intelligent search strategies for computer problem solving*. Addison Wesley, Reading, United States.

L.W.P. Peeters (2003). *Cyclic railway timetable optimization*. Ph.D. thesis, Erasmus Research Institute of Management, Erasmus University, Rotterdam, the Netherlands.

A. Pijpers (2004). `http://www.arthurstreinenpagina.nl/`. In Dutch.

J.P. Poort (2002). *Grenzen aan benutting (Limits on utilization)*. NYFER, Breukelen, the Netherlands. In Dutch.

V.R. Pratt (1973). *Computing permutations with double-ended queues, parallel stacks and parallel queues*. In A.V. Aho, A.B. Borodin, R.L. Constable, R.W. Floyd, M.A. Harrison, R.M. Karp, and H.R. Strong (editors), *Proceedings the fifth annual ACM symposium on Theory of Computing*, pages 268–277. ACM Press, New York, United States.

M. Prins (1998). *Het vervoersproces van planning tot uitvoering (The transportation process from planning to operation)*. Technical report, NS Reizigers. Internal document in Dutch.

D. Quarles van Ufford, J. IJzerman, M. Groenewoud, M. Bijvank, N. Marengo, O. Ibrahim, R. Visser, and W. Radder (2002). *OctoVU Solutions Eindrapport (Final report OctoVU Solutions)*. Technical report, Free University, Amsterdam, the Netherlands. In Dutch.

Railway Gazette International (2005). *Intelligent trains provide real-time feedback*. Railway Gazette International, 3:156–157.

C.J.M. Reinartz and P.M. Fassaert (1999). *Schets toekomstig planproces van NS Reizigers (Blueprint of the future planning process of NS Reizigers)*. Technical report, NS Reizigers. Internal document in Dutch.

A. Schrijver (2003). *Planning van opstelsporen (Planning of shunt tracks)*. Technical report, Centrum voor Wiskunde en Informatica. In Dutch.

Statistical Office of the European Communities (2004). `http://europa.eu.int/comm/eurostat/`.

N. Tomii and L.J. Zhou (2000). *Depot shunting scheduling using combined genetic algorithm and PERT*. In J. Allan, R. Hill, C.A. Brebbia, G. Sciutto, and S. Sone (editors), *Proceedings of COMPRAIL 2000*, pages 437–446. WIT Press, Southampton, United Kingdom.

N. Tomii, L.J. Zhou, and N. Fukumara (1999). *An algorithm for station shunting scheduling problems combining probabilistic local search and PERT*. In I. Imam, Y. Kodratoff, A. El-Dessouki, and M. Ali (editors), *Multiple approaches to intelligent systems: 12th international conference on industrial and engineering applications of artificial intelligence and expert systems*, volume 1611 of *Lecture Notes in Artificial Intelligence*, pages 788–797. Springer-Verlag, Berlin, Germany.

J.M. van den Akker, H. Hoogeveen, and S.L. van de Velde (2005). *Applying column generation to machine scheduling*. In G. Desaulniers, J. Desrosiers, and M.M. Solomon (editors), *Column Generation*, volume 5 of *GÉRAD 25th Anniversary Series*, pages 303–330. Springer-Verlag, Berlin, Germany.

J.M. van den Akker, C.A.J. Hurkens, and M.W.P. Savelsbergh (2000). *Time-indexed formulations for machine scheduling problems: column generation*. INFORMS Journal on Computing, 12(2):111–124.

J.M. van den Akker, C.P.M. van Hoesel, and M.W.P. Savelsbergh (1999). *A polyhedral approach to single-machine scheduling*. Mathematical Programming, 85(3):541–572.

J.J.J. van den Broek (2002). *Toets op inplanbaarheid van rangeerbewegingen (Check on planability of shunt movements)*. Master's thesis, Technical University, Eindhoven, the Netherlands. In Dutch.

J.J.J. van den Broek and L.G. Kroon (2005). *Capacity test for shunting movements*. Technical report, Technical University, Eindhoven, the Netherlands.

J.A.E.E. van Nunen and R.A. Zuidwijk (2004). *E-enabled closed-loop supply chains*. California Management Review, 46(2):40–54.

C. van 't Woudt (2001). *Zoeken naar rangeerroutes op een station (Searching shunt routes through a station)*. Master's thesis, Erasmus University, Rotterdam, the Netherlands. In Dutch.

W.M.C. van Wezel and B. Barten (2002). *Hierarchical mixed initiative planning support*. Paper presented at the Plansig 2002 workshop in Delft, the Netherlands.

W.M.C. van Wezel and R.J.J.M. Jorna (2004). *Tracing cognition, tasks and support: Shunting support in the Netherlands Railways (NS)*. Technical Report accepted for publication in the Journal of Cognition, Technology and Work, University of Groningen, the Netherlands.

W.M.C. van Wezel, R.J.J.M. Jorna, and D.J. Kiewiet (2003). *Does task follow structure? Domain representations and task execution of human planners at the Dutch Railroads*. Technical report, University of Groningen, the Netherlands.

F. Vanderbeck and L.A. Wolsey (1996). *An exact algorithm for IP column generation*. Operations Research Letters, 19:151–159.

A.W. Veenstra, R.A. Zuidwijk, and S. de Zwaan (2005). *Developing a maritime service centre in the port of Rotterdam*. Paper presented at the International Conference on Port-Maritime Development and Innovation in Rotterdam, the Netherlands.

M. Veldhorst (2004). *Ordering of train units on free shunt tracks and constraints in ILP-models*. Technical report, NSR. Internal document.

——— (2005). *Cleaning, shunting and matching*. Technical report, NSR. Internal document.

S. Vermeulen (2005). *Scheduling shunting movements at Alkmaar station*. Master's thesis, Maastricht University, Maastricht, the Netherlands.

R.G. Vickson (1980). *Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine*. Operations Research, 28(5):1155–1167.

M.J.C.M. Vromans (2005). *Reliability of railway systems*. Ph.D. thesis, Erasmus Research Institute of Management, Erasmus University, Rotterdam, the Netherlands.

G. Waeyenbergh and L. Pintelon (2002). *A framework for maintenance concept development*. International Journal of Production Economics, 77(1):299–313.

R. Watson and M. Sohail (2003). *Reducing railway noise levels - the impact of forthcoming European legislation*. Technical report, Loughborough University, United Kingdom.

W.E. Wilhelm (2001). *A technical review of column generation in integer programming*. Optimization and Engineering, 2:159–200.

T. Winter (1999). *Online and real-time dispatching problems*. Ph.D. thesis, Technical University, Braunschweig, Germany.

T. Winter and U.T. Zimmermann (2000). *Real-time dispatch of trams in storage yards*. Annals of Operations Research, 96:287–315.

S. Zeegers (2004). `http://www.sporenplan.nl/`. In Dutch.

P.J. Zwaneveld (1997). *Railway planning - routing of trains and allocation of passenger lines*. Ph.D. thesis, Erasmus University, Rotterdam, the Netherlands.

P.J. Zwaneveld, S. Dauzère-Peres, C.P.M. van Hoesel, L.G. Kroon, H.E. Romeijn, M. Salomon, and H.W. Ambergen (1996). *Routing trains through railway stations: model formulation and algorithms*. Transportation Science, 30(3):181–194.

Appendix A

Definitions

A.1 RELEVANT TERMS

A* Search	a special case of Best-First Search with a restricted structure of the evaluation function $k(.)$, see page 101.
arriving shunt unit	a train unit that has to be parked at a shunt yard, see page 44. Such units are decoupled from through train services, which continue after a short dwell time, or come from complete ending train services. The set of arriving shunt units is also called the supply of shunt units.
Best-First Search	a search algorithm where the most promising node is selected for expansion, see page 100.
block	a matching of an arriving part to a departing part, see page 44. The train units in such a block are meant to remain together during their stay at the station under consideration.
configuration	an ordered number of train unit types in a train or a train service, see page 5.
crossing	a train unit obstructing another train unit during its departure or arrival at a shunt track, see page 68. This term was first coined by Gallo and Di Miele [2001] in the context of buses.
demand for shunt units	see 'departing shunt unit'.

departing shunt unit	a train unit that needs to be supplied from the shunt yard, see page 44. These units are coupled onto through train services or form complete starting train services. The set of departing shunt units is also called the demand for shunt units.
deque	a linear list for which all insertions and deletions are made at the ends of the list. Deques are used to model shunt tracks which can be approached from both sides, see page 3. `Output-restricted` and `input-restricted` deques are deques in which deletions or insertions, respectively, are allowed to take place at only one end. If no such restrictions apply, one also speaks of a `general` deque [Knuth, 1997].
feasible assignment	a track assignment without crossings, where the units in the assignment are allowed to park at the track, and where the length of the units is never exceeded by the length of the units parked at it, see page 69.
feasible matching	a matching of arriving shunt units to departing shunt units with sufficient time difference between arrival and departure, no type mismatches, and the units in both parts in the same order, see page 45.
free track	a deque shunt track, see page 3.
informed search algorithms	a search algorithm that uses prior knowledge about the network being searched, see page 100.
LIFO track	a stack shunt track, see page 3.
mixed track	a shunt track at which different types of rolling stock are parked. See page 153.
modular cleaning	the operational internal cleaning process of rolling stock at NedTrain, see page 120.
Occupied Network A* Search	A specialization of A* Search for solving SRP.
part	an entity of one or more adjacent train units in one train, see page 44. Thus we can distinguish arriving and departing parts, which result from arriving and departing train services, respectively.
queue	a shunt track where rolling stock arrives at one side and departs from the other side of the track. See page 15.

pricing problem	the problem of generating additional columns, which price out, see pages 76,130.
restricted master problem	a mathematical program with a restricted number of columns, see pages 76,130.
route conflict	two train movements over the infrastructure that use the same part of the infrastructure at nearly the same time, see page 33. Exceptions are the start and end of a movement. Indeed, if a train unit is to be coupled onto a departing train, it is necessary to partly use the same infrastructure for both the train unit and the departing train.
route request	a request for a route over the station infrastructure, see page 32.
shunt unit	a train unit that needs to be parked at or to be supplied from the shunt yard, see page 44.
stack	a linear list for which all insertions and deletions are made at the end of the list [KNUTH, 1997]. Stacks are used to model shunt tracks which can be approached from one side, see page 3.
state-space search	a solution procedure that systematically explores the state-space of a problem which has been represented by a network, see page 99.
supply of shunt units	see 'arriving shunt unit'.
through train service	a service that continues passenger service after a short dwell time, see page 13.
track assignment	an assignment of blocks to a specific shunt track using prescribed sides for arrival and departure of each block, see page 68.
virtual tracks	tracks enabling a restricted number of mixed tracks in TUSP without deciding on beforehand which tracks are mixed, see page 153.

A.2 PROBLEMS

0-TDP	0-Tram Dispatching Problem, see page 146.
3PP	3 Partition Problem, see page 53.
3TMP	3 Train Matching Problem, see page 54.

BPP	Bin Packing Problem, see page 74.
FSTAP	Fixed Size Track Assignment Problem, see page 75.
FSTMP	Fixed Size Train Matching Problem, see page 58.
FSTUSP	Fixed Size Train Unit Shunting Problem, see page 146.
MCF	Uncapacitated Minimum Cost Flow, see page 54.
PP	Partition Problem, see page 133.
SCRAP	Shunt Cleaning Resource Availability Problem, see page 133.
SHPP	Shortest Path Problem, see page 58.
SIP	Sequencing within Intervals Problem (SIP), see page 132.
SMSP	Single Machine Scheduling Problem, see page 128.
SRP	Shunt Routing Problem, see page 95.
SPP	Set Partitioning Problem, see page 72.
SUCP	Shunt Unit Cleaning Problem, see page 123.
SUCP-∞	Shunt Unit Cleaning Problem where the objective is to schedule the blocks within their available time windows, see page 132.
TAP	Track Assignment Problem, see page 70.
TAP-0	Track Assignment Problem where all blocks need to be parked, see page 74.
TMP	Train Matching Problem, see page 46.
TMP-0	Train Matching Problem where no arriving trains can be split, see page 53.
TUSP	Train Unit Shunting Problem, see page 145.

A.3 PARAMETERS

Name	Description
a	the penalty for splitting of trains in TUSP, see page 152.
a_i	the number of arriving trains with configuration $\rho(i), i \in \mathcal{S}$, see page 56.
a_b	the release time of job (or block) $b \in \mathcal{B}$ in SUCP, see page 122.
$a_{i,p}$	an indicator whether subset p contains element i in SPP, see page 73.
$\overline{b}$	the maximum number of blocks simultaneously at the shunt yard, see page 75.
b_i	the supply or demand at a node $i \in \mathcal{N}$ in MCF, see pages 54, 56.
$c(m)$	the number of cleaning crews available at time $m \in \mathcal{M}$, see page 123.
d	• the destination in SHPP, see page 58.

	• the penalty for not parking a block, see page 73.
d_i	the number of departing trains with configuration $\rho(i), i \in \mathcal{S}$, see page 56.
e	the maximum number of node expansions for one route request in ONA* Search, see page 104.
f_p	• the cost of subset p in SPP, see page 73. • the cost of path p in the pricing problem of TAP, see page 81.
f_p^v	the additional penalties for different train configurations and rewards for clustering of blocks for the same departing train incurred by extending path p with node v in the pricing problem of TAP, see page 81.
$f_{i,j}$	the cost of an arc $(i, j) \in \mathcal{A}$ used at several places, see pages 54,58,60,81,100,148.
f_a^s	the cost of assignment a on track s in TAP, see page 73.
g	an objective element in TUSP modeling a preference for solutions with less different types parked at a track, see page 152.
h	the number of 2-OPT improvement rounds in SRP, see page 108.
i_u	the train service in which train unit u arrives at or departs from the station in TUSP, see page 150.
k	the number of bins in BPP, see page 74.
k_n	the evaluation of the incumbent $o{\rightarrow}n$ path in informed search algorithms, see page 100.
$k(n)$	a heuristic evaluation function, which estimates the lowest cost of reaching a destination node $d \in \mathcal{D}$ from a node $n \in \mathcal{N}$ in informed search algorithms, see page 100.
$k(n,m)$	a function returning the number of potential train configurations with n different types of rolling stock and at most m train units in a train, see page 57.
$\overline{\kappa}$	the maximum number of train units in a train in the proof that FSTMP can be solved in polynomial time and space, see page 57.
l_u	the length of train unit u in TSUP, see page 150.
l^s	the length of track s in TUSP, see page 152.
l^p	the length of physical track p in Models (7.c) and (7.d), see page 154.
l_t	the length of the type of train unit corresponding to train $t \in \mathcal{T}$ in FSTUSP, see page 147.
l_v	the length of the block corresponding with node v in the pricing problem of TAP, see page 81.

l_p^s	the remaining length of track s when the blocks $\mathcal{B}_p^s$ are parked at it, see page 81.
$l(n)$	the cost of the induced $o \to n$ path in Best-First Search algorithms, see page 101.
m	the number of subsets in 3PP, see page 53.
$m(n)$	the estimated cost of reaching a destination node $d \in \mathcal{D}$ from node n in Best-First Search algorithms, see page 101.
$m_{a,d}$	the cost of matching arriving part a to departing part d in 3TMP, see page 51.
$n_1^{\mathcal{R}}$	the node in layer $\mathcal{R}$ representing the configurations consisting of 1 unit, see page 56.
n_f^b	a node in layer $\mathcal{L}_b$ with $f \in \mathcal{F}_s$ in the pricing problem of TAP, see page 79.
n_{not}^b	a node in layer $\mathcal{L}_b$ representing *not* parking block b at shunt track s in the pricing problem of TAP, see page 79.
n_1^0	the source in the pricing problem of TAP, see page 79.
$n_1^{\lvert\mathcal{B}\rvert+1}$	the sink in the pricing problem of TAP, see page 79.
o	the origin in SHPP, see page 58.
o_i^s	the types of train units parked at the occupied positions of track s at node i in SUCP, see page 147.
p	the maximum number of train units parked simultaneously at a shunt yard, see pages 57,146.
p_s	the number of positions at track $s \in \mathcal{S}$ in FSTUSP, see page 147.
p_b	the required amount of man-minutes work for job (or block) $b \in \mathcal{B}$ in SUCP, see page 122.
$p_{b,m}$	the processing time of job $b \in \mathcal{B}$ in order to be ready precisely at time $m \in \mathcal{M}$, see page 131.
p_i^s	the farthest position from the A-side of track s at node i, which is available for parking a train unit in FSTUSP, see page 147.
q	a penalty for each arriving part that is used in 3TMP, see page 51.
q_b	the deadline of job (or block) $b \in \mathcal{B}$ in SUCP, see page 122.
r_p	the earliest departure time of the blocks in path $p \in \mathcal{P}_i$, which did not leave yet at the time of node i, see page 81.
r^v	the departure time of the block corresponding to the layer of node v in the pricing problem of TAP, see page 81.
$r_{t,s}$	the estimated cost of routing train $t \in \mathcal{T}$ to / from track $s \in \mathcal{S}$ in FSTUSP, see page 147.

$\rho(n)$	a function returning the configuration corresponding with node n in a network $\mathcal{G}$. The specific network is created in the proof that 3TMP can be solved in polynomial time and space, see page 55.
s	the size of a subset in 3PP, see page 53, and of a bin in BPP, see page 74.
σ_a	the size of an item $a \in \mathcal{A}$ at several places, see pages 53,74,133.
u	the cost of the incumbent overall set of routes in SRP, see page 108.
$u(\mathcal{R})$	a function that returns the cost of a set of routes $\mathcal{R}$ in SRP, see page 108.
v	an upper bound on the cost of a route for a request in ONA* Search, see page 104.
ψ_t	the type of the train unit in train $t \in \mathcal{T}$ in FSTUSP, see page 147.
ψ_u	the type of train unit u in TUSP, see page 150.
w	an upper bound on the number of changes in direction of a route in ONA* Search, see page 104.
$w(p)$	a function returning the number of changes in direction of a partial path p in ONA* Search, see page 104.
z	the number of minutes after the arrival of a block which is still considered “close” in time to its release time in SUCP, see page 122.

A.4 DECISION VARIABLES

Name	Description
$A_{s,p}$	decision variable indicating whether virtual track $s \in \mathcal{S}$ is assigned to physical track $p \in \mathcal{P}$ in Models (7.c) and (7.d), see page 154.
E_s	decision variable indicating the number of types ψ in excess of 1 parked at track s in TUSP, see page 152.
$\overline{f_a^s}$	the reduced cost of assignment a to shunt track s in TAP, see page 81.
K_t	decision variable indicating whether units t and $t+1$ are related to the same train and are parked at or retrieved from different tracks in TUSP, see page 151.
$L_{t,s}$	decision variable registering the length of the train units at track s after the arrival or departure of unit t in TUSP, see page 151.
λ_b	dual variable corresponding with the restriction (4.5) concerned with block b, see page 78.

$M_{t,u,s}$	decision variable indicating whether train unit t is matched to unit u and parked at or retrieved from track s in TUSP, see page 151.
μ_s	dual variable corresponding with the restriction (4.6) concerned with shunt track $s \in \mathcal{S}$, see page 78.
N_b	decision variable indicating whether block $b \in \mathcal{B}$ is not parked at any shunt track $s \in \mathcal{S}$, see page 73.
$O_{\psi,s}$	decision variable indicating whether at least one unit of type ψ is parked at track s in TUSP, see page 152.
$P_{t,s}$	decision variable indicating whether train unit t is parked at or retrieved from track s in TUSP, see page 151.
$R_{b,m}$	decision variable indicating whether job $b \in \mathcal{B}$ starts cleaning at time $m \in \mathcal{M}$, see page 131.
S_t	decision variable indicating whether train unit t arrives or departs via the A-side or the B-side of a track, see page 156.
X_a	decision variable indicating whether arriving part $a \in \mathcal{A}^+$ is used in 3TMP, see page 51.
X_p	decision variable indicating whether subset p is in a solution to SPP, see page 73.
$X_{i,j}$	the flow on arc $(i,j) \in \mathcal{A}$, see pages 54,56.
X_a^s	decision variable indicating whether assignment $a \in \mathcal{V}_s$ is used for shunt track $s \in \mathcal{S}$, see page 73.
Y_d	decision variable indicating whether departing part $d \in \mathcal{A}^-$ is used in 3TMP, see page 51.
$Z_{a,d}$	decision variable indicating whether arriving part a is matched to departing part d in 3TMP, see page 51.

A.5 SETS

Name	Description
$\mathcal{A}$	• the set of items at several places, see pages 53,72,74,133. • the arcs in a network used at several places, see '$\mathcal{G}$'.
$\mathcal{A}'$	a subset of $\mathcal{A}$ in PP, see page 133.
$\mathcal{A}_t$	the arcs in the network $\mathcal{G}_t$, see '$\mathcal{G}_t$'.
$\mathcal{A}_i$	the set of items assigned to bin $i, i = 1, \ldots, k$ in BPP, see page 74.
$\mathcal{A}_t^{i+}$	the arcs emanating from node $i \in \mathcal{N}_t$, see page 49.
$\mathcal{A}_t^{i-}$	the arcs directed at node $i \in \mathcal{N}_t$, see page 49.
$\mathcal{A}^+$	the set of arriving potential parts, see page 49.

$\mathcal{A}^-$	the set of departing potential parts, see page 49.
$\mathcal{A}^-_{(i,j)}$	the departing potential parts with the same configuration as the arriving potential part (i, j), see page 49.
$\mathcal{A}^+_{(i,j)}$	the arriving potential parts with the same configuration as the departing potential part (i, j), see page 49.
$\mathcal{B}$	the set of blocks that need to be parked and is used at several places, see page 73,122.
$\mathcal{B}_{k_t}$	the set of potential disassemblies of a train $t \in \mathcal{T}$ with configuration k_t, see page 53.
$\mathcal{B}^s_k$	the set of blocks in assignment $k \in \mathcal{V}_s$ parked at track $s \in \mathcal{S}$, see page 78.
$\mathcal{B}^s_p$	the set of blocks at track s in path p in the pricing problem of TAP, see page 81.
$\mathcal{D}$	• the configurations of departing trains in 3TMP, see page 55. • the set of destination nodes in SRP, see page 99.
$\mathcal{D}_r$	the set of destination nodes for route request r in SRP, see page 108.
$\mathcal{E}$	the set of expanded nodes in informed search algorithms, see page 100.
$\mathcal{F}_s$	the set of different approach types to and from shunt track $s \in \mathcal{S}$, see page 78.
$\mathcal{G}$	a network used at several places, see pages 54,56,58,79,99,147.
$\mathcal{G}_t$	the matching network for disassembly of train $t \in \mathcal{T}$, see page 48.
$\mathcal{I}$	the set of pairs of train units $u, u+1 \in \mathcal{U}$ that arrive or depart in the same train service, i.e. $i_u = i_{u+1}$ in TUSP, see page 151.
$\mathcal{K}$	the set of possible configurations of trains, see page 48.
$\mathcal{L}$	the set of pairs of train units (t, u) that can be matched in TUSP, see page 151.
$\mathcal{L}_b$	the layer of nodes representing block $b \in \mathcal{B}$ in the pricing problem of TAP, see page 79.
$\mathcal{L}_0$	the initial layer containing the source n^0_1 in the pricing problem of TAP, see page 79.
$\mathcal{L}_{\lvert\mathcal{B}\rvert+1}$	the last layer containing the sink $n^{\lvert\mathcal{B}\rvert+1}_1$ in the pricing problem of TAP, see page 79.
$\mathcal{L}_t$	a layer of nodes of nodes representing all possible states of types of train units parked at positions of the shunt yard in FSTUSP after the arrival of departure of train t, see page 147.
$\mathcal{M}$	the set of relevant points in time in SUCP, see page 123.

$\mathcal{N}$	the nodes in a network $\mathcal{G}$ used at several places, see '$\mathcal{G}$'.
$\mathcal{N}_t$	the nodes in the network $\mathcal{G}_t$, see '$\mathcal{G}_t$'.
$\mathcal{N}'_t$	the set of all intermediate nodes in $\mathcal{N}_t$, see page 49.
$\mathcal{O}$	the set of origin nodes in SRP, see page 99.
$\mathcal{O}_r$	the set of origin nodes for route request r in SRP, see page 108.
$\mathcal{P}$	• the set of subsets of $\mathcal{A}$ in SPP, see page 72. • the set of physical shunt tracks in Models (7.c) and (7.d), see page 153.
$\mathcal{P}_u$	the set of feasible $(n_1^0 - u)$-paths in the pricing problem of TAP, see page 81.
$\mathcal{P}_A$	the set of physical tracks in TUSP open at the A-side only, see page 155.
$\mathcal{P}_B$	the set of physical tracks in TUSP open at the B-side only, see page 155.
$\mathcal{R}$	• an "intermediate set" in the proof that 3TMP can be solved in polynomial time and space, see page 55. • the set of route requests in SRP, see page 95.
$\mathcal{R}^{uv}$	the set of blocks departing between the times of nodes u and v in the pricing problem of TAP, see page 81.
$\mathcal{S}$	• the configurations of arriving trains in 3TMP, see page 55. • the set of shunt tracks and is used at several places, see pages 73,147. • the set of virtual shunt tracks in Models (7.c) and (7.d), see page 153.
$\mathcal{S}'$	the set of mixed virtual tracks in Models (7.c) and (7.d), see page 153.
$\mathcal{S}''$	the set of non-mixed virtual tracks in Models (7.c) and (7.d), see page 153.
$\mathcal{S}_i$	the subsets $i = 1, \ldots, m$ in 3PP, see page 55.
$\mathcal{T}$	the set of trains, which can be decomposed in the set of arriving trains $\mathcal{T}^+$ and the set of departing trains $\mathcal{T}^-$, see pages 48,147.
$\mathcal{T}^-_{t'}$	the set of trains that leave sufficiently later than arriving train $t' \in \mathcal{T}^+$ in order to match train units of these train, see page 49.
$\mathcal{T}^+_{t'}$	the set of trains that arrive sufficiently earlier than departing train $t' \in \mathcal{T}^-$ in order to match train units of these train, see page 49.
$\mathcal{U}$	• the set of unexpanded nodes in informed search algorithms, see page 100. • the set of train units, which can be decomposed in the set of

	arriving train units $\mathcal{U}^+$ and the set of departing train units $\mathcal{U}^-$, see page 150.
$\mathcal{V}_s$	the set of potential assignments to track $s \in \mathcal{S}$, see page 73.
$\mathcal{V}_s^b$	the set of potential assignments to track $s \in \mathcal{S}$ containing block $b \in \mathcal{B}$, see page 73.
$\mathcal{X}$	• the set of assigned infrastructure reservations in SRP, see page 95. • the set of potential crossings at a LIFO track in TUSP, see page 151.
$\mathcal{Y}$	the set of different types of rolling stock, see pages 53,147.
$\mathcal{Z}$	the set of pairs $(v, u) \in \mathcal{U}^2$ such that there exist $(t, u), (v, w) \in \mathcal{L}$ with $t <_A v <_A u <_A w$, see page 153.

Appendix B

Test Instances

The real-life instances in the thesis are based on the stations Zwolle and Enschede in the 1999 / 2000 railway system, which is quite similar to the 2005 / 2006 system for these stations. This appendix starts with some information on the lines that occur at these stations. Typically, one hundred consecutive numbers are reserved for one line. One train is indicated with the last two digits in the line number, where even numbers mostly are trains directed to Amsterdam. Moreover, the numbers of the trains increase throughout the day.

Table B.1 introduces the railway lines that were operated in the 1999 / 2000 railway system of NS Reizigers, which visit Zwolle, Enschede or both. The table reports the line-number, the type of line, which can be InterCity (IC), InterRegional (IR), or Regional (R). The frequency is the number of trains per hour for the line. The last column of the table shows some of the important stations located of the line. In this table, abbreviations of the stations are used. These stations are: Amersfoort (Amf), Apeldoorn (Apd), Arnhem (Ah), Den Bosch (Ht), Den Haag Centraal (Gvc), Deventer (Dv), Emmen (Emn), Enschede (Es), Groningen (Gn), Kampen (Kpn), Leeuwarden (Lw), Roosendaal (Rsd), Schiphol (Shl), Utrecht (Ut), Zwolle (Zl).

The odd-numbered trains in the 10500 line are split at station Zwolle from the trains from the 500 line, while the odd-numbered trains of the 10700 line are split at the same station from the 700 line. Similar, the even-numbered trains in these series are combined with the even-numbered trains of the 500 respectively 700 line at this station. Similar processes take place for the 500, 700, 1600, and 1700 lines at stations Utrecht and Amersfoort.

In the following paragraph, we discuss some details on the stations of Zwolle and Enschede. We will discuss some aspects of the relevant railway infrastructure at these stations as well as some aspects of the 1999 / 2000 railway system, passing stations Zwolle and / or Enschede.

Line-number	Type	Frequency	Main stations
500	IC	1	Gvc - Ut - Amf - Zl - Gn
10500	IC	1	Zl - Lw
700	IC	1	Shl - Amf - Zl - Gn
10700	IC	1	Zl - Lw
1600	IC	1	Shl - Amf - Apd - Dv - Es
1700	IC	1	Gvc - Ut - Amf - Apd - Dv - Es
3600	IR	2	Zl - Dv - Ah - Ht - Rsd
3800	IR	1	Zl - Emn
5600	R	2	Ut - Amf - Zl
7900	R	2	Zl - Es
8000	R	1	Zl - Emn
8500	R	2	Zl - Kpn
9100	R	1	Zl - Gn

Table B.1: The lines passing stations Zwolle and / or Enschede in 1999 / 2000.

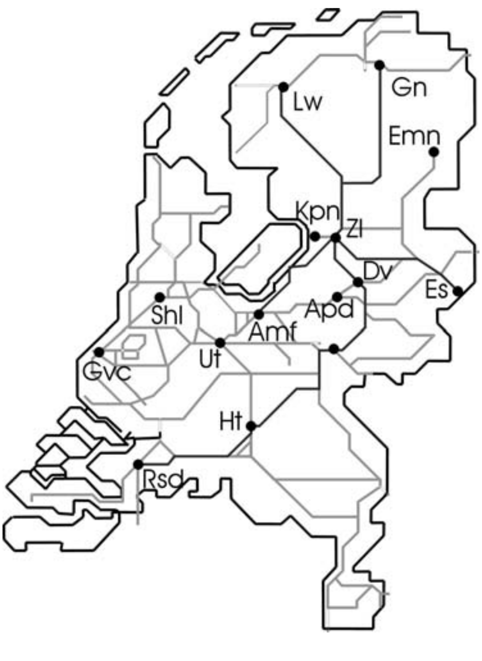


Figure B.1: The Dutch railway network with the lines of the main example from Section 2.2 in bold.

B.1 REAL-LIFE INSTANCES

Figure B.2 depicts an overview of the railway infrastructure of station Zwolle. A more detailed figure of the layout of station Zwolle is given in Figure B.3, where the cleaning tracks are the tracks 90 and 91. Here, the platform tracks are located in the center of the figure, such as tracks 1A, 1B, 3a, and 3B. Around these platform tracks, several shunt tracks are located, including tracks 17, 18C, 19, 100, 101, and 102. Station Zwolle contains platform tracks, 19 shunt tracks and 2 cleaning tracks. The shunt tracks range in length from 114 meter to 415 meter. Moreover, 7 IU-tracks are available for entering or leaving station Zwolle: HA and VA in the direction of Amersfoort, BA and AB in the direction of Leeuwarden and Groningen, WO and OW in the direction of Deventer and ZH in the direction of Enschede.

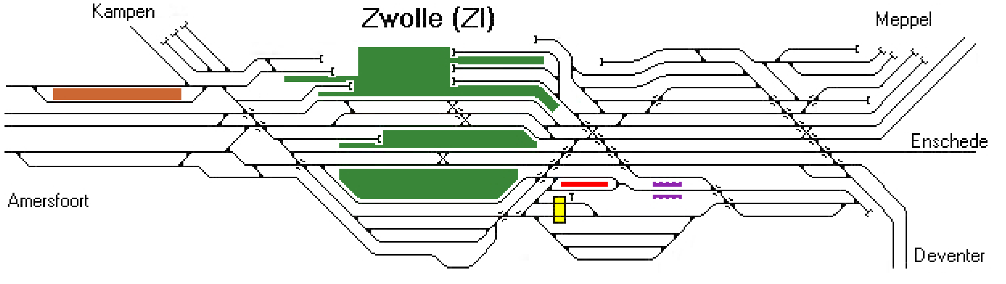


Figure B.2: The layout of the railway station Zwolle, based on [Zeegers, 2004].

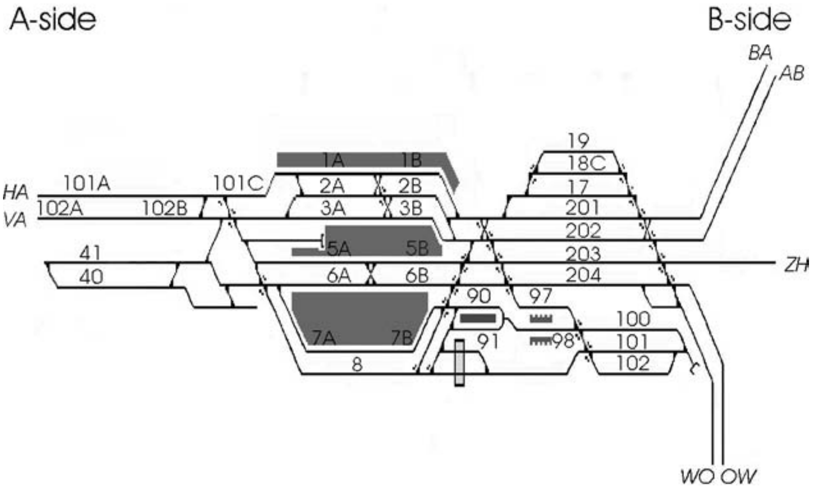


Figure B.3: The railway infrastructure in our example [Zeegers, 2004].

The types of train units that passed station Zwolle in the 1999 / 2000 railway systems are given in Table B.2. For each type, we report the number of carriages, the length, and whether it is a single-deck or a double-deck train unit. Several pairs of train unit types belong to the same family. These pairs are: DD-AR_3 and DD-AR_4, ICM_3 and ICM_4, IRM_3 and IRM_4, and MAT'64_2 and MAT'64_4.

Abbreviation	Number of carriages	Length (in meters)	Deck
DD-AR_3	3	98	Double-deck
DD-AR_4	4	124	Double-deck
DH_2	2	44	Single-deck
DM90_2	2	53	Single-deck
ICM_3	3	81	Single-deck
ICM_4	4	108	Single-deck
IRM_3	3	82	Double-deck
IRM_4	4	108	Double-deck
MAT'64_2	2	53	Single-deck
MAT'64_4	4	102	Single-deck
MDDM_4	4	102	Double-deck
SM90_2	2	53	Single-deck

Table B.2: The types of rolling stock and some characteristics.

The layout of station Enschede is given in Figure B.4. Enschede contains 13 shunt track, ranging in length from 50 meter to 684 meter. Moreover, the station has 4 platforms and only 2 IU-tracks for national railway services, in the direction of Hengelo.

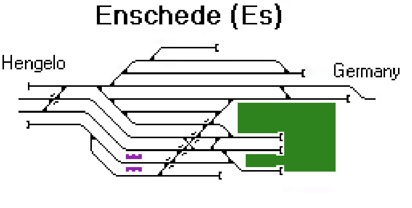


Figure B.4: The layout of the railway station Enschede [Zeegers, 2004].

Table B.3 provides additional information on the timetables that serve as input for the instances in this thesis. These timetables are based on 24 hours period starting at 8:00 either on Tuesday or Saturday. A graphical breakdown of different train unit types at stations Zwolle and Enschede can be found in Figure B.5, respectively Figure B.6.

Instance	Zwolle		Enschede	
	Tue / Wed	Sat / Sun	Tue / Wed	Sat / Sun
# of arriving trains	306	276	81	72
# of departing trains	305	270	79	66
# of arriving train units	538	413	147	111
# of departing train units	538	397	147	101
# of fixed train units	450	352	122	90

Table B.3: Some characteristics of the shunting instances.

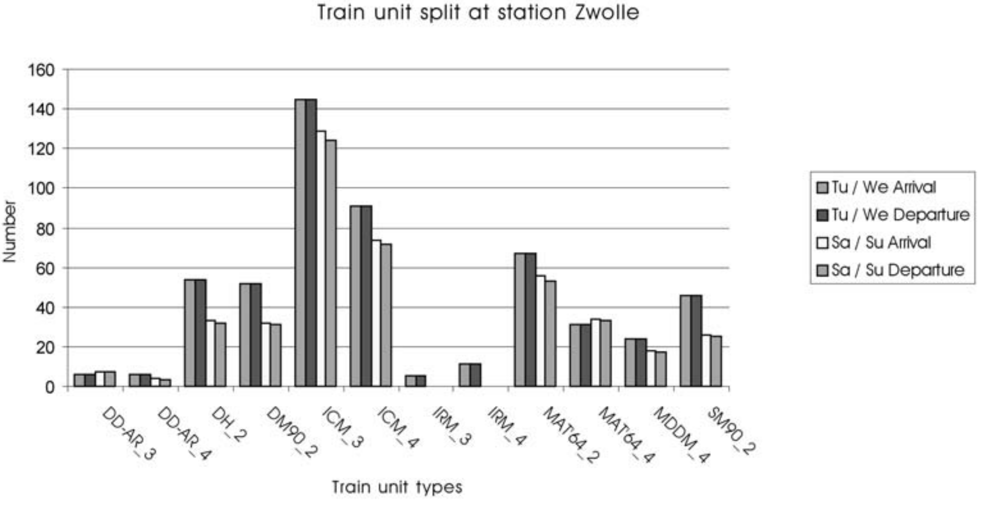


Figure B.5: Train unit split for station Zwolle.

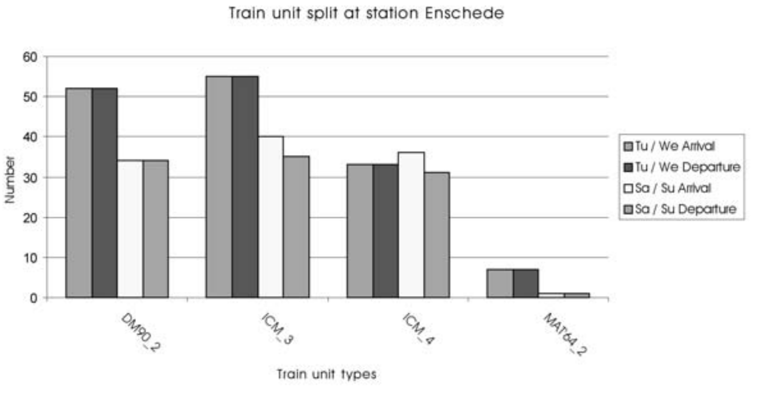


Figure B.6: Train unit split for station Enschede.

B.2 MAIN EXAMPLE OF THE THESIS

The main example of the thesis is based on the real-life situation at station Zwolle. An overview of this infrastructure as well as a detailed figure are presented in the previous paragraph. Table B.4 presents some additional details on the shunt tracks in Figure B.3.

Shunt track	Length (in meters)	Type
2A	185	free track
2B	168	free track
8	236	free track
19	169	free track
18C	200	free track
17	275	free track
100	274	LIFO track
101	200	LIFO track
102	200	LIFO track

Table B.4: Some characteristics of the shunt tracks in Figure B.3.

Moreover, Table B.5 gives the details of the timetable that is used in the main example. The rolling stock in this example is depicted in Figure B.7.

Train ID	Platform	Time	Event	Configuration	Direction	Station side
771	3a/b	Tu 20:46	A	ICM_3 ICM_4 ICM_3 ICM_3	Amf (VA)	A
771	3a/b	Tu 20:49	D	ICM_3 ICM_3	Gn (AB)	B
10771	3a	Tu 20:52	D	ICM_4	Lw (AB)	B
3672	7a/b	Tu 22:09	A	IRM_4 IRM_3	Rsd (OW)	B
3687	7a/b	Tu 22:23	D	IRM_4	Rsd (WO)	B
7984	5b	Tu 23:12	A	DH_2	Es (ZH)	B
584	1a	Tu 23:18	A	ICM_3 ICM_3	Gn (BA)	B
3680	5b	We 0:09	A	IRM_4	Rsd (OW)	B
3623	5a	We 5:50	D	IRM_3 IRM_4	Rsd (WO)	B
516	1a	We 6:18	D	ICM_3 ICM_3	Amf (HA)	A
7917	5b	We 7:21	D	DH_2	Es (ZH)	B
721	3a	We 7:46	A	ICM_3	Amf (VA)	A
10721	3a	We 7:52	D	ICM_3 ICM_3	Lw (AB)	B

Table B.5: The timetable for our example.

B.3 ADDITIONAL COMPUTATIONAL RESULTS

Table B.6 presents computational results for the instances of the Shunt Routing Problem (SRP). These instances represent the Saturday / Sunday instances at station Zwolle and accompany the results in Table 5.1 on page 111. Similarly, Table B.7 report computational results for SRP at station Enschede for Saturday / Sunday instances.

Subsequently, Tables B.8, B.9 and B.10 present additional computational results for the effect of the different instances of the Shunt Unit Cleaning Problem (SUCP) on the Track Assignment Problem (TAP). These tables accompany Tables 6.6 and 6.7 on pages 138 and 140.

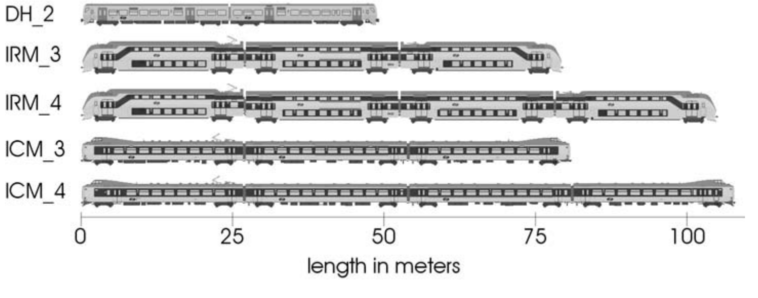


Figure B.7: The different families (3) and types (5) of rolling stock from Table B.5 [Pijpers, 2004].

Finally, Tables B.11 and B.12 present computational results for the Train Unit Shunting Problem (TUSP) for three, respectively 4, virtual tracks where multiple types of train units are allowed. Tables 7.3 and 7.4 report on pages 162 and 163 accompanying results.

Integrated results for 3 and 4 tracks with multiple types of train units allowed.

Instance	ZSAD	ZSBD	ZSCD	ZSAR	ZSBR	ZSCR	ZSAO	ZSBO	ZSCO
Number of route requests	611	610	609	611	610	609	611	610	609
Minimum routing cost	30248	30084	30306	30635	30512	30651	30293	30065	30306
No 2-OPT									
Number of routes not found	4	4	3	3	3	3	6	6	3
Number of changes in direction	37	31	35	26	23	36	35	32	35
Number of start time deviations	29	29	29	34	32	30	31	27	29
Routing cost	34273	33804	33315	33637	33487	33684	36007	35668	33315
Gap with minimal routing cost	11.74%	11.00%	9.03%	8.92%	8.88%	9.00%	15.87%	15.71%	9.03%
Computation time (in sec.)	0.61	0.50	0.06	0.08	0.06	0.14	1.42	1.61	0.05
Apply 2-OPT once									
Number of routes not found	2	1	2	0	1	0	5	3	2
Number of changes in direction	45	43	41	35	28	48	41	44	41
Number of start time deviations	29	29	29	35	32	32	29	27	29
Routing cost	32587	31578	32547	31222	31755	31272	35233	33290	32547
Gap with minimal routing cost	7.18%	4.73%	6.89%	1.88%	3.91%	1.99%	14.02%	9.69%	6.89%
Computation time (in sec.)	19.17	26.77	27.19	12.59	11.25	28.36	48.42	67.69	26.63
Apply 2-OPT twice									
Number of routes not found	2	1	2	0	1	0	5	3	2
Number of changes in direction	45	43	41	35	28	48	41	44	41
Number of start time deviations	29	29	29	35	32	32	29	27	29
Routing cost	32587	31578	32547	31222	31755	31272	35233	33290	32547
Gap with minimal routing cost	7.18%	4.73%	6.89%	1.88%	3.91%	1.99%	14.02%	9.69%	6.89%
Computation time (in sec.)	35.09	50.09	52.77	22.516	20.00	55.13	93.41	137.14	52.11

Table B.6: Computational results for SRP at station Zwolle for Saturday / Sunday instances.

Instance	ES.D	ES.R	ES.O
Number of route requests	153	153	153
Minimum routing cost	18814	18814	18492
No 2-OPT			
Number of routes not found	0	0	1
Number of changes in direction	10	10	9
Number of start time deviations	1	1	1
Routing cost	18837	18837	19542
Gap with minimal routing cost	0.12%	0.12%	5.37%
Computation time (in sec.)	0.00	0.00	0.00
Apply 2-OPT once			
Number of routes not found	0	0	0
Number of changes in direction	10	10	9
Number of start time deviations	1	1	2
Routing cost	18837	18837	18592
Gap with minimal routing cost	0.12%	0.12%	0.54%
Computation time (in sec.)	0.05	0.05	0.06
Apply 2-OPT twice			
Number of routes not found	0	0	0
Number of changes in direction	10	10	9
Number of start time deviations	1	1	2
Routing cost	18837	18837	18592
Gap with minimal routing cost	0.12%	0.12%	0.54%
Computation time (in sec.)	0.09	0.09	0.11

Table B.7: Computational results for SRP at station Enschede for Saturday / Sunday instances.

Use 2-OPT Start?	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Instance **ZTAO**	1		2		5		10		1440	
Number of blocks	68		66		63		63		68	
Comp. time for TAP (in sec.)	488	171	464	215	555	122	508	140	798	247
LP solution value	7987.96	7971.00	7685.50	7588.00	8189.75	8331.64	8180.88	7324.67	8694.83	8716.37
IP solution value	7990	7971	7740	7588	8333	8477	8254	7371	8901	8887
Gap	0.026%	0.000%	0.709%	0.000%	1.719%	1.715%	0.886%	0.629%	2.316%	1.920%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	363	0	389	0	324	0	347	0	394
# Columns generated	8286	4169	8120	3563	7484	3550	6904	3357	8588	4689
# Iterations of col. gen.	48	32	49	35	42	26	40	28	54	37
Instance **ZTCR**	1		2		5		10		1440	
Number of blocks	65		67		65		65		70	
Comp. time for TAP (in sec.)	925	398	843	683	707	497	1592	481	2197	3646
LP solution value	11402.10	11435.60	11823.70	11811.40	11585.20	11646.10	11678.00	11671.40	12322.60	12345.50
IP solution value	11592	11692	12012	12137	11730	12023	11898	11918	12835	13038
Gap	1.638%	2.193%	1.568%	2.683%	1.234%	3.135%	1.849%	2.069%	3.992%	10.481%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	308	0	282	0	307	0	316	0	317
# Columns generated	6235	3013	6381	3205	6219	2700	6215	2954	6899	3319
# Iterations of col. gen.	43	31	39	32	40	25	40	26	45	34

Table B.8: The effect of cleaning on TAP for station Zwolle on Tuesday / Wednesday for instances ZTAO and ZTCR.

Use 2-OPT Start?	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Number of blocks	69		69		67		68		73	
Instance **ZTBD**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	8240	1181	1584	583	3852	767	1138	355	9349	2137
LP solution value	10813.20	10772.60	10812.80	10775.30	10548.40	10536.90	10712.60	10681.40	11257.80	11378.30
IP solution value	11748	11153	11213	11122	10752	10971	10799	10914	12044	11564
Gap	7.957%	3.411%	3.569%	3.117%	1.894%	3.957%	0.800%	2.131%	6.528%	1.606%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	322	0	332	0	337	0	296	0	304
# Columns generated	6455	3118	6245	3529	6804	3005	6137	3106	6647	3447
# Iterations of col. gen.	44	28	46	37	45	31	49	35	46	35
Instance **ZTBO**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	1699	347	1363	423	1176	417	1506	429	2532	1080
LP solution value	8978.60	8814.18	8142.19	8111.20	8110.67	7906.50	8311.40	8135.28	8554.20	8752.70
IP solution value	9433	9354	8529	8518	8252	8178	8622	8396	8725	8972
Gap	4.817%	5.771%	4.535%	4.776%	1.713%	3.320%	3.602%	3.105%	1.958%	2.444%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	410	0	402	0	441	0	393	0	398
# Columns generated	7314	3703	7377	3705	8004	4090	7382	4053	8539	4480
# Iterations of col. gen.	45	31	47	33	52	32	52	32	54	38

Table B.9: The effect of cleaning on TAP for station Zwolle on Tuesday / Wednesday for instances ZTBD and ZTBO.

Use 2-OPT Start?	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Number of blocks	68		66		63		63		68	
Instance **ZTBR**	1		2		5		10		1440	
Comp. time for TAP (in sec.)	917	491	1034	370	2994	446	728	317	2727	995
LP solution value	12254.50	12238.70	12032.50	12030.50	12001.70	12000.80	12211.30	12248.50	12907.30	12770.20
IP solution value	12644	12335	12242	12242	12413	12481	12578	12386	13267	13226
Gap	3.081%	0.781%	1.711%	1.728%	3.313%	3.847%	2.915%	1.110%	2.711%	3.446%
# Blocks not parked	0	0	0	0	0	0	0	0	0	0
# Columns by 2-OPT	0	320	0	293	0	317	0	315	0	289
# Columns generated	5838	3506	6453	3102	6085	2608	6270	2587	6309	3066
# Iterations of col. gen.	41	38	51	37	47	31	53	33	48	37

Table B.10: The effect of cleaning on TAP for station Zwolle on Tuesday / Wednesday for instances ZTBR.

	# pre-assigned	Station Zwolle				Station Enschede			
		A	D	Type	Time	A	D	Type	Time
Model (7.c)	0	0	1	2	10800 (*)	0	0	2	2.17
	1	0	1	2	10800 (*)	0	0	2	1.61
	2	0	1	2	10800 (*)	0	0	2	2.17
	3	0	1	2	10800 (*)	0	0	2	1.28
	4	0	1	2	10800 (*)	—	—	—	—
	5	0	1	2	10800 (*)	—	—	—	—
Model (7.d)	0	0	0	2	10800 (*)	0	0	2	97.69
	1	0	0	2	2982.44	0	0	2	55.45
	2	0	0	2	2914.61	0	0	2	133.00
	3	0	0	2	1302.23	0	0	2	29.66
	4	0	0	2	1754.63	—	—	—	—
	5	0	0	2	1149.59	—	—	—	—

Table B.11: Computational results with three tracks with multiple types of train units.

	# pre-assigned	Station Zwolle				Station Enschede			
		A	D	Type	Time	A	D	Type	Time
Model (7.c)	0	0	1	2	10800 (*)	0	0	2	3.05
	1	0	1	2	10800 (*)	0	0	2	2.50
	2	0	1	2	10800 (*)	0	0	2	2.20
	3	0	1	2	10800 (*)	0	0	2	2.67
	4	0	1	2	10800 (*)	—	—	—	—
	5	0	1	2	10800 (*)	—	—	—	—
Model (7.d)	0	0	0	2	10800 (*)	0	0	2	44.00
	1	0	0	2	10800 (*)	0	0	2	64.17
	2	0	0	2	3418.63	0	0	2	39.50
	3	0	0	2	4775.50	0	0	2	117.23
	4	0	0	2	3618.55	—	—	—	—
	5	0	0	2	4429.86	—	—	—	—

Table B.12: Computational results with four tracks with multiple types of train units.

Samenvatting

Het aanbieden van vervoer per trein in Nederland van een hoge kwaliteit vereist veel coördinatie en een complex planningsproces. Een belangrijk onderdeel van dit planningsproces is de operationele rangeerplanning. Rangeerplanning richt zich op de logistieke processen met rollend materieel in en om een station. Buiten de spits heeft een vervoerder van reizigers een overschot aan rollend materieel omdat de vraag naar transport varieert gedurende de dag. Vaak wordt het overbodige materieel geparkeerd op een rangeerterrein zodat de vrijgekomen infrastructuur kan worden gebruikt door andere treinen.

De hoofddoelstelling van dit proefschrift is "*het ontwikkelen van kwantitatieve modellen en algoritmen om rangeerplanners te ondersteunen*".

Het eerste hoofdstuk bespreekt een aantal algemene ontwikkelingen in de wereld van het Europese treinvervoer van reizigers. Daarnaast bevat het een introductie tot verschillende aspecten van de rangeerplanning, en wordt de rangeerplanning gepositioneerd ten opzichte van de andere onderdelen van het planningsproces van een spoorvervoerder en bij NS Reizigers (NSR) in het bijzonder. Daarnaast wordt de rangeerplanning van NSR gerelateerd aan rangeerproblemen van trams, bussen en treinen van andere openbaar vervoer bedrijven. Tenslotte worden de onderzoeksvragen in dit hoofdstuk geïntroduceerd.

In het tweede hoofdstuk wordt het operationele rangeerprobleem van NSR in detail besproken. Hiermee wordt de basis voor de volgende hoofdstukken gelegd. Bij NSR wordt het rangeerprobleem dag voor dag opgelost en voor één station tegelijkertijd. De belangrijkste doelstelling van de operationele rangeerplanning is om een soepele start van de dienstverlening aan reizigers in de volgende ochtend te faciliteren. Het is hierbij van belang om rekening te houden met de verschillende typen treinstellen en met treinen die bestaan uit meerdere treinstellen. De rangeerplanning is één van de laatste onderdelen van het planningsproces bij NSR. Dit heeft tot gevolg dat elke wijziging in een eerder onderdeel van dit planningsproces zeer waarschijnlijk leidt tot aanpassingen in de rangeerplannen van één of meerdere stations. Vele planners bij NSR zijn bezig met het maken en aanpassen van rangeerplannen.

De belangrijkste onderdelen van de rangeerplanning zijn het toewijzen van aankomende aan vertrekkende treinstellen, het opstellen van treinstellen, het routeren van treinstellen over de infrastructuur van het station, en personeelsplanning voor het rangeerpersoneel. Hoofdstuk 2 bevat ook een initiële beschrijving van een groot voorbeeld. Dit voorbeeld loopt als een rode draad door het proefschrift en wordt op verschillende plaatsen gebruikt voor verduidelijking. In de hoofdstukken 3 tot en met 7 worden kwantitatieve modellen en algoritmen ontwikkeld voor verschillende onderdelen van het rangeerprobleem. Elk hoofdstuk besteedt aandacht aan de complexiteit van het deelprobleem dat wordt bestudeerd en geeft een overzicht van relevante literatuur.

Het onderwerp van het derde hoofdstuk is het toewijzen van aankomende aan vertrekkende treinstellen. De dienstregeling met aankomende en vertrekkende treinen vormt de belangrijkste input data voor dit probleem. In deze dienstregeling wordt ook de exacte configuratie van elke trein voorgeschreven. In het algemeen kunnen treinstellen met hetzelfde type worden uitgewisseld, en moeten de voorgeschreven configuraties van de treinen in de dienstregeling in tact blijven. Een belangrijke doelstelling voor dit probleem is om treinstellen uit dezelfde trein zo veel mogelijk bij elkaar te houden omdat er zo minder productiemiddelen benodigd zijn voor de uitvoering van het plan. Een groot gedeelte van de toewijzing is al in eerdere onderdelen van het planningsproces gemaakt. Dit vaste deel bestaat grotendeels uit treinstellen van treinen die na een korte haltering doorrijden naar volgende stations. Een rangeerplanner kan de gevonden oplossing beïnvloeden door toewijzingen van de resterende treinstellen vast te zetten. Het toewijzingsprobleem is vertaald naar een deterministisch geheeltallig programmeringsprobleem. Verschillende varianten van dit probleem zijn succesvol toegepast op realistische instanties gebaseerd op de stations van Zwolle en Enschede.

Het vierde hoofdstuk heeft als onderwerp het opstellen van treinstellen op opstelsporen. Dit opstellen is gebaseerd op de toewijzing van het vorige hoofdstuk. Hier is het essentieel dat een treinstel een ander treinstel niet in de weg mag staan tijdens de aankomst op of het vertrek van het opstelspoor. Daarnaast is een grote rol weggelegd voor de configuratie van de opstelsporen. Sommige opstelsporen kunnen slechts van één kant worden benaderd. Treinstellen die op zo'n opstelspoor worden geparkeerd moeten voldoen aan het Last-In-First-Out (LIFO) principe. Andere opstelsporen kunnen van beide kanten worden benaderd. Op deze sporen is het zelfs mogelijk om via de ene kant aan te komen en via de andere kant het opstelspoor te verlaten. Deze laatste opstelsporen zijn veel flexibeler, maar maken het opstellen van treinstellen ook complexer. De belangrijkste doelstelling van een ondersteunend algoritme is het opstellen van een maximaal aantal treinstellen. Het ontwikkelde algoritme slaagt hierin voor alle doorgerekende instanties. Andere onderdelen van de doelstelling voor dit probleem zijn het minimaliseren van de geschatte routeringskosten, voorkeuren van planners voor het gebruiken van bepaalde opstelsporen, en het sturen op een robuste oplossing, die bestand is tegen kleine vertragingen. De twee onderdelen die de robuustheid modelleren zijn het combineren van treinstellen voor dezelfde vertrekkende trein in de juiste volgorde op hetzelfde opstelspoor en het opstellen van treinstellen met hetzelfde type op een opstelspoor. Het eerste onderdeel vermindert de operationele complexiteit net voor het vertrek van de trein en draagt zo bij aan een soepele start van de operationele uitvoering van het spoorwegproces in de ochtend. Voor het tweede onderdeel geldt dat de volgorde van treinstellen op een opstelspoor met slechts één type treinstel irrelevant is aangezien treinstellen met hetzelfde type onderling kunnen worden uitgewisseld. In het laatste geval hebben wijzigingen in de volgorde van aankomst of vertrek van treinstellen geen invloed op de opstelling van zo'n spoor. Voor dit probleem is een heuristiek ontwikkeld die is gebaseerd op kolomgeneratie. De kolommen worden gegenereerd met behulp van dynamisch programmeren in een specifiek netwerk. Rekenresultaten tonen aan dat het erg beperkend is om alle opstelsporen te modelleren als LIFO sporen. Daarnaast hebben de maatregelen om de robuustheid te vergroten het gewenste effect.

Hoofdstuk 5 behandelt het routeren van treinstellen over de infrastructuur van het station. Dit probleem wordt op 2 momenten opgelost. Ten eerste levert het een deel van de invoer voor het probleem van het opstellen, zoals beschreven in hoofdstuk 4, namelijk een schatting van de routeringskosten. Ten tweede wordt het opgelost om te bepalen of de routes die volgen uit een gegeven opstelling gezamenlijk kunnen worden ingepland. Conflicten tussen twee routes zijn niet toegestaan omdat deze zullen vertragingen in de operationele uitvoering. De routes van de treinen uit de dienstregeling hebben een hogere prioriteit dan de rangeerroutes van en naar de opstelsporen en daarom worden de routes van de treinen uit de dienstregeling vastgezet voordat de rangeerroutes worden gezocht. Een eigenschap van het resulterende routeringsprobleem is het feit dat het starttijdstip van de route enigszins flexibel is. Ook hier geldt dat het belangrijk is dat het algoritme zo veel mogelijk routes vindt. Daarnaast is het van belang om zo min mogelijk van richting te veranderen tijdens een route binnen het station en om zo min mogelijk af te wijken van de geprefereerde starttijdstippen. Het veranderen van richting tijdens een route kost extra tijd. Het probleem wordt opgelost door het sequentieel toepassen van een uitbreiding van de A* zoekmethode. Deze uitbreiding vindt één route tegelijk. Daarnaast wordt de zoekmethode gebruikt in een 2-OPT toepassing, waarin de volgorde van het inplannen van de routes gewisseld kan worden om een betere oplossing te vinden. Voor het vinden van goede oplossingen voor verschillende instanties is er een actief samenspel nodig tussen de rangeerplanner en het algoritme.

Het reinigen van het interieur van de treinstellen is het onderwerp van hoofdstuk 6. Alleen de treinstellen die overnachten op een rangeerterrein worden gereinigd. Dit reinigen vindt vaak plaats langs een speciaal reinigingsperron, wat leidt tot extra routering van treinstellen van en naar de sporen langs dit perron. Een gevolg hiervan is dat het leidt tot extra flexibiliteit om het opstelspoor van een treinstel te veranderen nadat het is gereinigd. Daarnaast kenmerkt dit reinigingsproces zich door een variërend aantal beschikbare mensen voor het reinigen op verschillende tijdstippen. Dit heeft tot gevolg dat de doorlooptijd van het reinigen van een treinstel ook varieert over tijd. Het belangrijkste aspect van het plannen van dit reinigingsproces is het streven om zo veel mogelijk treinstellen direct na aankomst op het station te reinigen. Het reinigen van treinstellen net voor vertrek conflicteert met de algemene doelstelling van een soepele start van de operationele uitvoering. Het reinigen van treinstellen ongeveer halverwege tussen aankomst- en vertrektijd heeft als nadeel dat deze treinstellen waarschijnlijk twee keer moeten worden opgesteld: voor en na de reiniging van het interieur. Dit is complexer en vereist extra productiemiddelen en is daarom ongewenst. Het probleem wordt wiskundig gemodelleerd als een geheeltallig programmeringsprobleem. De rekenresultaten geven aan dat praktische instanties van het probleem snel kunnen worden opgelost. Uit een planning voor het reinigen volgt een nieuw probleem voor het opstellen van treinstellen, zoals besproken in hoofdstuk 4. Deze resulterende instanties kunnen worden opgelost door een uitbreiding van het opstelalgoritme, zoals in het vierde hoofdstuk is besproken.

In hoofdstuk 7 wordt het effect van het integreren van de problemen uit de hoofdstukken 3 en 4 geanalyseerd, te weten het toewijzen van aankomende aan vertrekkende treinstellen en het opstellen van treinstellen op opstelsporen. Hiervoor is weer een geheeltallig programmeringsmodel geformuleerd. Uit de rekenresultaten voor de praktijkinstanties blijkt dat de kennis van rangeerplanners nodig is om de rekentijd tot aanvaardbare proporties terug te brengen. Echter, het geïntegreerd oplossen van deze problemen leidt tot robuustere oplossingen ten opzichte van de sequentiële aanpak.

In hoofdstuk 8 wordt een samenvatting van het proefschrift gegeven en worden de onderzoeksvragen uit het eerste hoofdstuk beantwoord. Ook passeren enkele mogelijkheden de revue om het toepassingsgebied van de ontwikkelde modellen en algoritmen uit te breiden.

Samenvattend beschrijft dit proefschrift het operationele planningsprobleem van het rangeren van treinstellen. Er worden een aantal deelproblemen geïdentificeerd en de complexiteit van een aantal van deze deelproblemen is geanalyseerd. Vervolgens zijn er kwantitatieve modellen voor deze deelproblemen ontwikkeld, en gebaseerd op deze modellen zijn algoritmen ontwikkeld om de modellen op te lossen. De algoritmen zijn getest met realistische data en vormen een solide basis voor een geavanceerd beslissingsondersteunend systeem voor het ondersteunen van rangeerplanners. De algoritmen zijn flexibel en bieden verschillende mogelijkheden voor rangeerplanners om de gevonden oplossingen te beïnvloeden. Vanuit een praktisch oogpunt kan het gebruik van deze algoritmen in het planningsproces van NSR een interessante vervolgstap zijn. Wetenschappelijk vervolgonderzoek zou zich kunnen richten op het toepassen van de modellen en algoritmen in de operationele uitvoering van het spoorwegproces, waar tijdslimieten voor het vinden van een oplossing beperkter zijn dan in de planning.

Summary

In order to provide train services in the Netherlands with a high quality, much coordination is required and a complex planning process is carried out. An important element of this planning process is operational shunt planning. Shunt planning focuses on the logistics of rolling stock within a station and its surroundings. Since demand for transportation fluctuates over a day, a railway operator typically has a surplus of rolling stock outside the rush hours, and especially during the night. In general, the idle rolling stock is parked at a shunt yard, keeping the main railway infrastructure available for other train services.

The main goal of this thesis is to "*develop quantitative models and algorithms for supporting shunt planners*".

In Chapter 1, some general developments in European passenger railway transportation are discussed. Moreover, it provides an introduction of several aspects of shunt planning. In addition, shunt planning is positioned into the general planning process of a railway passenger operator, and Netherlands Railways Passengers (in Dutch: Nederlandse Spoorwegen Reizigers, NSR) more specifically. This chapter also aligns shunt planning at NSR with other shunt planning problems of trams, buses and trains at other operators of public transport systems. Finally, the research questions of the thesis are introduced.

Chapter 2 lays the base for subsequent chapters by giving an in-depth description of the operational shunt planning problem faced by NSR. Typically, this problem is solved on a day-by-day basis and for one station at a time. The main goal of operational shunt planning is to enable a smooth start-up of the railway operations in the next morning. Important aspects are train units with different types, and several train units combined into one train. Shunt planning is one of the last elements of the planning process at NSR. Every change in a previous step of the planning process is likely to require changes in the shunt plans at one or more stations. Therefore, many planners at NSR are currently involved in shunt planning.

The most important parts of the shunt planning problem include the matching of

arriving train units to departing ones, the parking of train units at shunt tracks, the routing of train units over station infrastructure, the cleaning of train units, and crew planning for the shunting crews. The second chapter also contains the initial description of the main example. This example is used throughout the thesis for clarification. Chapters 3 – 7 continue with the development of quantitative models and algorithms for several subproblems of shunt planning. Each of these chapters pays some attention to the computational difficulty of the subproblem under consideration, and positions the subproblem by discussing related problems.

The subject of the third chapter is the matching of arriving train units to departing train units. The most important input data for this problem consists of a timetable with arriving and departing trains and their configurations. Typically, train units of the same type can be interchanged. In a resulting matching, the train configurations prescribed by the timetable and the order of train units in trains need to be respected. Moreover, keeping units from the same train together as much as possible ensures a minimum resource usage. Therefore, it is an important element of the objective of this problem. A large part of the matching has already been made in previous planning processes. This part typically consists of the trains that continue their service after a short dwell time for alighting and boarding passengers. Moreover, a shunt planner can control the found solution by fixing additional parts of the remaining matching. A deterministic integer programming model for this problem has been developed. Several variants of this model have been applied successfully to real-life instances resulting from the Dutch stations Zwolle and Enschede.

Based on the matchings resulting from Chapter 3, Chapter 4 studies the problem of parking train units at shunt tracks. An important restriction in this problem is the fact that a train unit is not allowed to obstruct the arrival or departure of a different train unit. Here, the configurations of the shunt tracks play an important role. At certain shunt tracks, arrivals and departures are restricted to the same side of the tracks. Train units are parked at such tracks according to the Last-In-First-Out (LIFO) principle. At other tracks, arrivals and departures can occur at both sides, and even arriving at one side and departing from the other side is possible. These tracks offer more flexibility, but also complicate the problem. Obviously, an algorithm for this problem should try to park as many train units as possible. The developed algorithm succeeds in this objective in all tested instances. Additional criteria include estimates of routing cost to and from shunt tracks, preferences of shunt planners for shunt tracks, and robustness measures. The robustness measures consist of combining train units for the same departing train in the right order at the same shunt track, and parking train units of the same type at a shunt track. The first measure reduces the operational complexity just before the departure of a train and therefore contributes to a smooth start-up of the operations. Furthermore, since train units with the same type can be interchanged, the order of train units at a shunt track with only one type of train units is not relevant. Therefore, disruptions resulting in changes of arrival and departure times of timetabled trains do not influence the parking of train units at tracks with only one type of train units. A column generation heuristic for solving this problem has been implemented, where columns are generated by dynamic programming in a network with a distinctive structure. Computational experiments showed that modeling all tracks as LIFO tracks is quite restrictive, while robustness measures have the desired effect on the obtained solutions.

Chapter 5 discusses the problem of routing train units over station infrastructure. The problem is solved in order to produce estimates of the routing cost to be used in the parking subproblem described in Chapter 4 as well as in order to find routes for route requests resulting from a solution to this parking problem. Conflicts between two planned routes are not allowed, because these would result in delays during the operations. Moreover, timetabled trains are prioritized over shunt routes to and from shunt tracks. Therefore the routes of timetabled trains need to be considered fixed when generating shunt routes. A unique characteristic of the routing problem studied in this chapter is that the timing of routing rolling stock to or from shunt tracks is flexible to some extent. The most important objective is to find as many routes as possible. Secondary elements of this objective include the number of changes in direction, which are time consuming, and the number of deviations from preferred start times. The resulting problem is solved by sequentially applying an extension of A* Search for finding a shunt route for one request, and a 2-OPT procedure to improve the overall solution by interchanging the order of in which the train units are routed. In several cases, the solution procedure requires active interaction with shunt planners in order to find good solutions.

The planning of the cleaning of the interior of train units that lay over at a shunt yard is the subject of Chapter 6. This internal cleaning typically takes place along a dedicated cleaning platform. Therefore, this problem introduces additional routing of train units. In addition, it provides more flexibility to change the shunt track at which a specific train unit is parked after it has been cleaned. A distinctive aspect of this problem is that the number of crews available for cleaning train units differs over time. This results in different throughput times for cleaning a train unit at different points in time. Furthermore, one tries to clean as many train units as possible shortly after they have arrived at the station. Cleaning train units just before departure results in high risks of disturbing the start-up in the next morning. Moreover, cleaning train units somewhere in the middle between their arrival and departure might require that these units need to be parked before and after the cleaning process, which is more complex and requires additional resources. These arguments result in a preference for cleaning train units just after their arrival. After modeling the problem as an integer program, computational results showed that the practical instances can be solved fast. Moreover, the resulting instances for the parking of train units can be solved satisfactorily by an extension of the parking algorithm, which was already discussed in Chapter 4.

The goal of Chapter 7 is to study the effect of integrating the matching and parking problems described in Chapters 3 and 4. The computational results of a resulting integer linear program indicate that knowledge of shunt planners is required to reduce the computation times for solving these models. However, high-quality solutions to these models provide considerable improvements over the sequential approach in the robustness measures described in Chapter 4.

Chapter 8 summarizes our work and answers the research questions posed in the first chapter. In addition, some opportunities to extend the scope of the presented models and algorithms are identified.

Summarizing, this thesis describes the operational shunt planning problem. Several subproblems are identified. Some effort is devoted to the computational complexity of each subproblem. Moreover, mathematical models of these problems are described and developed. In turn, algorithms for solving these models are introduced and tested with real-life data. Since planners have many opportunities to influence the obtained solutions, the developed algorithms provide a firm basis for an advanced planning system to support shunt planners. From a practical point of view, a further step could be to use these algorithms in obtaining solutions in the actual planning process. In further research, the scope of the presented models and algorithms can be extended to the operations, where limited time for finding a solution is available.

Curriculum Vitae

Ramon Lentink was born in Apeldoorn, the Netherlands on October 27, 1975. After completing his secondary school education at *De Heemgaard* in Apeldoorn, he started his study Econometrics at the *Free University of Amsterdam* in 1995. In 1999, he graduated with a Master's thesis on train crew scheduling. The thesis was written at ORTEC, one of the key providers of advanced planning systems, and the developed algorithms were applied to real-life problems provided by NS Reizigers, the largest Dutch passenger railway operator.

From September 1999, he has been working as a consultant at ORTEC. Here, he applies Operations Research models and algorithms to problems of clients in the aviation and railway industry. His focus is on crew planning problems. These activities led to a paper in *Annals of Operations Research* and several papers in proceedings of conferences.

In October 2000, he started a part-time Ph.D. study at the Rotterdam School of Management of the Erasmus University Rotterdam, while he remained part-time employed at ORTEC. The subject of his Ph.D. research is the development of mathematical models and algorithms for supporting railway shunt planners. This thesis is the result of the Ph.D. study. The corresponding research was carried out in close cooperation with the logistics department of NS Reizigers.

Besides his articles related to his work at ORTEC, his Ph.D. research resulted in a number of publications. These publications have been published or are forthcoming in scientific journals and a book with as subject a multidisciplinary view on planning. Moreover, he has presented his research work at various international conferences and workshops. In October 2002, Dennis Huisman and Ramon won the first prize in the first edition of the Management Science in Railroad Applications Student Competition, organized by the Rail Applications Special Interest Group of INFORMS and the journal Railway Age. The corresponding paper has been published in *Transportation Science*.

ERIM Ph.D. Series Research in Management

Erasmus Research Institute of Management (ERIM)

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

1. Meer, J.R. van der, **Operational Control of Internal Transport**
   Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. R. Dekker
   ISBN: 90-5892-004-6, URL: http://hdl.handle.net/1765/859

2. Fleischmann, M., **Quantitative Models for Reverse Logistics**
   Promotor(es): Prof.dr.ir. J.A.E.E. van Nunen, Prof.dr.ir. R. Dekker
   ISBN: 3540-417-117, URL: http://hdl.handle.net/1765/1044

3. Romero Morales, D., **Optimization Problems in Supply Chain Management**
   Promotor(es): Prof.dr.ir. J.A.E.E. van Nunen, dr. H.E. Romeijn
   ISBN: 90-9014-078-6, URL: http://hdl.handle.net/1765/865

4. Roodbergen, K.J., **Layout and Routing Methods for Warehouses**
   Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. J.A.E.E. van Nunen
   ISBN: 90-5892-005-4, URL: http://hdl.handle.net/1765/861

6. Chen, Y., **Labour flexibility in China's companies: an empirical study**
   Promotor(es): Prof.dr. A. Buitendam, Prof.dr. B. Krug
   ISBN: 90-5892-012-7, URL: http://hdl.handle.net/1765/307

7. Heugens, P.M.A.R., **Strategic Issues Management: Implications for Corporate Performance**
   Promotor(es): Prof.dr.ing. F.A.J. van den Bosch, Prof.dr. C.B.M. van Riel
   ISBN: 90-5892-009-7, URL: http://hdl.handle.net/1765/358

8. Speklé, R.F., **Beyond Generics; A closer look at Hybrid and Hierarchical Governance**
   Promotor(es): Prof.dr. M.A. van Hoepen RA
   ISBN: 90-5892-011-9, URL: http://hdl.handle.net/1765/357

9. Puvanasvari Ratnasingam, P., **Interorganizational Trust in Business to Business E-Commerce**
   Promotor(es): Prof.dr. K. Kumar, Prof.dr. H.G. van Dissel
   ISBN: 90-5892-017-8, URL: http://hdl.handle.net/1765/356

10. Mol, M.M., **Outsourcing, Supplier-relations and Internationalisation: Global Source Strategy as a Chinese puzzle**
    Promotor(es): Prof.dr. R.J.M. van Tulder
    ISBN: 90-5892-014-3, URL: http://hdl.handle.net/1765/355

11. Wolters, M.J.J., **The Business of Modularity and the Modularity of Business**
    Promotor(es): Prof.mr.dr. P.H.M. Vervest, Prof.dr.ir. H.W.G.M. van Heck
    ISBN: 90-5892-020-8, URL: http://hdl.handle.net/1765/920

12. Oosterhout, J. van, **The Quest for Legitimacy; On Authority and Responsibility in Governance**
    Promotor(es): Prof.dr. T. van Willigenburg, Prof.mr. H.R. van Gunsteren
    ISBN: 90-5892-022-4, URL: http://hdl.handle.net/1765/362

13. Koppius, O.R., **Information Architecture and Electronic Market Performance**
    Promotor(es): Prof.dr. P.H.M. Vervest, Prof.dr.ir. H.W.G.M. van Heck
    ISBN: 90-5892-023-2, URL: http://hdl.handle.net/1765/921

14. Vis, I.F.A., **Planning and Control Concepts for Material Handling Systems**
    Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. R. Dekker
    ISBN: 90-5892-021-6, URL: http://hdl.handle.net/1765/866

15. Bijman, W.J.J., **Essays on Agricultural Co-operatives; Governance Structure in Fruit and Vegetable Chains**
    Promotor(es): Prof.dr. G.W.J. Hendrikse
    ISBN: 90-5892-024-0, URL: http://hdl.handle.net/1765/867

16. Teunter, L.H., **Analysis of Sales Promotion Effects on Household Purchase Behavior**
    Promotor(es): Prof.dr.ir. B. Wierenga, Prof.dr. T. Kloek
    ISBN: 90-5892-029-1, URL: http://hdl.handle.net/1765/868

17. Loef, J., **Incongruity between Ads and Consumer Expectations of Advertising**
    Promotor(es): Prof.dr. W.F. van Raaij, Prof.dr. G. Antonides
    ISBN: 90-5892-028-3, URL: http://hdl.handle.net/1765/869

18. Ganzaroli, A., **Creating Trust between Local and Global Systems**
Promotor(es): Prof.dr. K. Kumar, Prof.dr. R.M. Lee
ISBN: 90-5892-031-3, URL: http://hdl.handle.net/1765/361

19. Fenema, P.C. van, **Coordination and Control of Globally Distributed Software Projects**
Promotor(es): Prof.dr. K. Kumar
ISBN: 90-5892-030-5, URL: http://hdl.handle.net/1765/360

20. Delporte–Vermeiren, D.J.E., **Improving the flexibility and profitability of ICT-enabled business networks: an assessment method and tool.**
Promotor(es): Prof.mr.dr. P.H.M. Vervest, Prof.dr.ir. H.W.G.M. van Heck
ISBN: 90-5892-040-2, URL: http://hdl.handle.net/1765/359

21. Wijk, R.A.J.L. van, **Organizing Knowledge in Internal Networks. A Multilevel Study**
Promotor(es): Prof.dr.ing. F.A.J. van den Bosch
ISBN: 90-5892-039-9, URL: http://hdl.handle.net/1765/347

22. Peeters, L.W.P., **Cyclic Railway Timetable Optimization**
Promotor(es): Prof.dr. L.G. Kroon, Prof.dr.ir. J.A.E.E. van Nunen
ISBN: 90-5892-042-9, URL: http://hdl.handle.net/1765/429

23. Jong, C. de, **Dealing with Derivatives: Studies on the role, informational content and pricing of financial derivatives**
Promotor(es): Prof.dr. C.G. Koedijk
ISBN: 90-5892-043-7, URL: http://hdl.handle.net/1765/1043

24. Waal, T. de, **Processing of Erroneous and Unsafe Data**
Promotor(es): Prof.dr.ir. R. Dekker
ISBN: 90-5892-045-3, URL: http://hdl.handle.net/1765/870

25. Hooghiemstra, R., **The Construction of Reality**
Promotor(es): Prof.dr. L.G. van der Tas RA, Prof.dr. A.Th.H. Pruyn
ISBN: 90-5892-047-X, URL: http://hdl.handle.net/1765/871

26. Dijksterhuis, M., **Organizational dynamics of cognition and action in the changing Dutch and U.S. banking industries**
Promotor(es): Prof.dr. F.A.J. van den Bosch, Prof.dr. H.W. Volberda
ISBN: 90-5892-048-8, URL: http://hdl.handle.net/1765/1037

27. Fok, D., **Advanced Econometric Marketing Models**
Promotor(es): Prof.dr. P.H.B.F. Franses
ISBN: 90-5892-049-6, URL: http://hdl.handle.net/1765/1035

28. Wielemaker, M.W., **Managing Initiatives. A Synthesis of the Conditioning and Knowledge-Creating View**
Promotor(es): Prof.dr. H.W. Volberda, Prof.dr. C.W.F. Baden-Fuller
ISBN: 90-5892-050-X, URL: http://hdl.handle.net/1765/1036

29. Berghe, D.A.F., **Working Across Borders: Multinational Enterprises and the Internationalization of Employment**
Promotor(es): Prof.dr. R.J.M. van Tulder, Prof.dr. E.J.J. Schenk
ISBN: 90-5892-05-34, URL: http://hdl.handle.net/1765/1041

30. Miltenburg, P.R., **Effects of modular sourcing on manufacturing flexibility in the automotive industry. A study among German OEMs**
Promotor(es): Prof.dr. J. Paauwe, Prof.dr. H.R. Commandeur
ISBN: 90-5892-052-6, URL: http://hdl.handle.net/1765/1039

31. Graaf, G., **Tractable Morality. Customer discourses of bankers, veterinarians and charity workers**
Promotor(es): Prof.dr. F. Leijnse, Prof.dr. T. van Willigenburg
ISBN: 90-5892-051-8, URL: http://hdl.handle.net/1765/1038

32. Gilsing, V.A., **Exploration, Exploitation and Co-evolution in Innovation Networks**
Promotor(es): Prof.dr. B. Nooteboom, Prof.dr. J.P.M. Groenewegen
ISBN: 90-5892-05-42, URL: http://hdl.handle.net/1765/1040

33. Flier, B., **Strategic Renewal of European Financial Incumbents**
Promotor(es): Prof.dr. F. van den Bosch, Prof.dr. H. Volberda
ISBN: 90-5892-055-0 , URL: http://hdl.handle.net/1765/1071

34. Langen, P.W. de, **The Performance of Seaport Clusters**
Promotor(es): Prof.dr. B. Nooteboom, Prof.dr. H.W.H. Welters
ISBN: 90-5892-056-9, URL: http://hdl.handle.net/1765/1133

35. Brito, M.P. de, **Managing Reverse Logistics or Reversing Logistics Management?**
Promotor(es): Prof.dr.ir. R. Dekker, Prof.dr. M.B.M. de Koster
ISBN: 90-5892-058-5, URL: http://hdl.handle.net/1765/1132

36. Appelman, J.H., **Governance of Global Interorganizational Tourism Networks; Changing Forms of Co-ordination between the Travel Agency and Aviation Sector**
Promotor(es): Prof.dr. F.M. Go, Prof.dr. B. Nooteboom
ISBN: 90-5892-060-7, URL: http://hdl.handle.net/1765/1199

37. Popova, V.N., **Knowledge Discovery and Monotonicity**
Promotor(es): Prof.dr. A. de Bruin
ISBN: 90-5892-061-5, URL: http://hdl.handle.net/1765/1201

38. Muller, A.R., **The Rise of Regionalism: Core Company Strategies Under The Second Wave of Integration**
Promotor(es): Prof.dr. R.J.M. van Tulder
ISBN: 90-5892-062-3, URL: http://hdl.handle.net/1765/1272

39. Berens, G. A.J.M., **Corporate Branding: The Development of Corporate Associations and their Influence on Stakeholder Reactions**
Promotor(es): Prof.dr. C.B.M. van Riel
ISBN: 90-5892-065-8, URL: http://hdl.handle.net/1765/1273

40. Six, F.E., **Trust and Trouble: Building Interpersonal Trust Within Organizations**
Promotor(es): Prof.dr. B. Nooteboom, Prof.dr. A.M. Sorge
ISBN: 90-5892-064-X, URL: http://hdl.handle.net/1765/1271

41. Slager, A.M.H., **Banking across Borders**
Promotor(es): Prof.dr. D.M.N. van Wensveen, Prof.dr. R.J.M. van Tulder
ISBN: 90-5892-066-6, URL: http://hdl.handle.net/1765/1301

42. Mandele, L.M., van der, **Leadership and the Inflection Point: A Longitudinal Perspective**
Promotor(es): Prof.dr. H.W. Volberda, Prof.dr. H.R. Commandeur
ISBN: 90-5892-067-4, URL: http://hdl.handle.net/1765/1302

43. Kippers, J., **Empirical Studies on Cash Payments**
Promotor(es): Prof.dr. Ph.H.B.F. Franses
ISBN: 90-5892-069-0, URL: http://hdl.handle.net/1765/1520

44. Daniševská, P., **Empirical Studies on Financial Intermediation and Corporate Policies**
Promotor(es): Prof.dr. C.G. Koedijk
ISBN: 90-5892-070-4, URL: http://hdl.handle.net/1765/1518

45. Mulder, A., **Government Dilemmas in the Private Provision of Public Goods**
Promotor(es): Prof.dr. R.J.M. van Tulder
ISBN: 90-5892- 071-2, URL: http://hdl.handle.net/1765/1790

46. Hermans, J.M., **ICT in Information Services, Use and deployment of the Dutch securities trade, 1860-1970**
Promotor(es): Prof.dr. F.H.A. Janszen
ISBN: 90-5892-072-0, URL: http://hdl.handle.net/1765/1793

47. Liang, G., **New Competition; Foreign Direct Investment And Industrial Development In China**
Promotor(es): Prof.dr. R.J.M. van Tulder
ISBN: 90-5892-073-9, URL: http://hdl.handle.net/1765/1795

48. Schweizer, T.S., **An Individual Psychology of Novelty-Seeking, Creativity and Innovation**
Promotor(es): Prof.dr. R.J.M. van Tulder
ISBN: 90-5892-07-71, URL: http://hdl.handle.net/1765/1818

49. Vliet, P. van, **Downside Risk and Empirical Asset Pricing**
    Promotor(es): Prof.dr. G.T. Post
    ISBN: 90-5892-07-55, URL: http://hdl.handle.net/1765/1819

50. Valck, K. de, **Virtual Communities of Consumption: Networks of Consumer Knowledge and Companionship**
    Promotor(es): Prof.dr.ir. G.H. van Bruggen, Prof.dr.ir. B. Wierenga
    ISBN: 90-5892-078-X, URL: http://hdl.handle.net/1765/1

51. Le-Anh, T., **Intelligent Control of Vehicle-Based Internal Transport Systems**
    Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. R. Dekker
    ISBN: 90-5892-079-8, URL: http://hdl.handle.net/1765/1

52. Pouchkarev, I., **Performance Evaluation of Constrained Portfolios**
    Promotor(es): Prof.dr. J. Spronk, dr. W.G.P.M. Hallerbach
    ISBN: 90-5892-083-6

53. Maeseneire, W., de, **Essays on Firm Valuation and Value Appropriation**
    Promotor(es): Prof.dr. J.T.J. Smit
    ISBN: 90-5892-082-8

54. Verheul, I., **Is there a (fe)male approach? Understanding gender differences in entrepreneurship**
    Promotor(es): Prof.dr. A.R. Thurik
    ISBN: 90-5892-080-1, URL: http://hdl.handle.net/1765/2005

55. Jansen, J.J.P., **Ambidextrous Organizations**
    Promotor(es): Prof.dr.ing. F.A.J. Van den Bosch, Prof.dr. H.W. Volberda
    ISBN: 90-5892-081-X

56. Assen, M.F. van, **Empirical Studies in Discrete Parts Manufacturing Management**
    Promotor(es): Prof.dr. S.L. van de Velde, Prof.dr. W.H.M. Zijm
    ISBN: 90-5892-085-2

57. Keizer, A.B., **The Changing Logic of Japanese Employment Practices, A Firm-Level Analysis of Four Industries**
    Promotor(es): Prof.dr. J.A. Stam, Prof.dr. J.P.M. Groenewegen
    ISBN: 90-5892-087-9

58. Moerman, G.A., **Empirical Asset Pricing and Banking in the Euro Area**
    Promotor(es): Prof.dr. C.G. Koedijk
    ISBN: 90-5892-090-9

59. Kotlarsky, J., **Management of Globally Distributed Component-Based Software Development Projects**
    Promotor(es): Prof.dr. K. Kumar
    ISBN: 90-5892-088-7

60. Boer, N.I., **Knowledge Sharing within Organizations: A situated and relational Perspective**
    Promotor(es): Prof.dr. K. Kumar
    ISBN: 90-5892-086-0
61. Pak, K., **Revenue Management: New Features and Models**
    Promotor(es): Prof.dr.ir. R. Dekker
    ISBN: 90-5892-092-5
62. Vromans, M.J.C.M., **Reliability of Railway Systems**
    Promotor(es): Prof.dr. L.G. Kroon, Prof.dr.ir. R. Dekker, Prof.dr.ir. J.A.E.E. van Nunen
    ISBN: 90-5892-089-5
63. Brohm, R., **Polycentric Order in Organizations: a dialogue between Michael Polanyi and IT-consultants on knowledge, morality, and organization**
    Promotor(es): Prof.dr. G.W.J. Hendrikse, Prof.dr. H.K. Letiche
    ISBN: 90-5892-095-X
64. Le-Duc, T., **Design and control of efficient order picking processes**
    Promotor(es): Prof.dr. M.B.M. de Koster
    ISBN: 90-5892-094-1
65. Boer, C.A., **Distributed Simulation in Industry**
    Promotor(es): Prof.dr. A. de Bruin, Prof.dr.ir. A. Verbraeck
    ISBN: 90-5892-093-3
66. Kuilman, J., **The re-emergence of foreign banks in Shanghai: An ecological analysis**
    Promotor(es): Prof.dr. B. Krug
    ISBN: 90-5892-096-8
67. Hartigh, E. den, **Increasing Returns and Firm Performance: An Empirical Study**
    Promotor(es): Prof.dr. H.R. Commandeur
    ISBN: 90-5892-098-4
68. Hagemeijer, R.E., **The Unmasking of the Other**
    Promotor(es): Prof.dr. S.J. Magala, Prof.dr. H.K. Letiche
    ISBN: 90-5892-097-6
69. Govers, R., **Virtual Tourism Destination Image: Glocal identities constructed, perceived and experienced**
    Promotor(es): Prof.dr. F.M. Go, Prof.dr. K. Kumar
    ISBN: 90-5892-107-7
70. Mentink, A., **Essays on Corporate Bonds**
    Promotor(es): Prof.dr. A.C.F. Vorst
    ISBN: 90-5892-100-X
71. Pietersz, R., **Pricing Models for Bermudan-style Interest Rate Derivatives**
    Promotor(es): Prof.dr. A.A.J. Pelsser, Prof.dr. A.C.F. Vorst
    ISBN: 90-5892-099-2

Algorithmic Decision Support for Shunt Planning

In order to provide train services in the Netherlands with a high quality, much coordination is required and a complex planning process is carried out. One of the last elements of this planning process is operational shunt planning. Shunt planning focuses on the logistics within a station and its surroundings. Since demand for transportation fluctuates over a day, a railway operator typically has a surplus of rolling stock outside the rush hours, and especially during the night. In general, the idle rolling stock is parked at a shunt yard, thereby keeping the main railway infrastructure available for other train services. Besides parking of rolling stock, matching of arriving to departing rolling stock, routing over local railway infrastructure, cleaning of rolling stock, and crew planning are part of shunt planning. Every change in a previous step of the planning process is likely to require changes in shunt plans at one or more stations. Therefore, many planners at NS Reizigers are currently involved in shunt planning. In addition, high-quality shunt plans enable a smooth start-up of the railway operations in the morning. A smooth start-up decreases the chances of disturbances in the morning. It is well known that such disturbances spread out easily in time and space. Therefore, the quality of shunt plans influences the quality of the services offered to passengers. The relevance of research on shunt planning from a societal, managerial and scientific point of view is therefore clear. "Algorithmic Decision Support for Shunt Planning" introduces relevant aspects of shunting and provides a first step for quantitative models and algorithms to support shunt planning. The algorithms for solving the models contain algorithms that resemble the current practice of shunt planners as well as algorithms that are somewhat farther away from current practice. Computational tests on real-life data show that high-quality solutions are typically found within minutes of computation time. In addition, these algorithms are designed to interact with shunt planners. They provide a firm basis for an advanced planning system to support shunt planners.

ERIM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are RSM Erasmus University and the Erasmus School of Economics. ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focussed on the management of the firm in its environment, its intra- and inter-firm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced graduate program in Research in Management. Within ERIM, over two hundred senior researchers and Ph.D. candidates are active in the different research programs. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

www.erim.eur.nl

ISBN 90-5892-104-2

Design: B&T Ontwerp en advies www.b-en-t.nl Print: Haveka www.haveka.nl