Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency

Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency

Personeelsplanning bij de Nederlandse Spoorwegen: Het verbeteren van eerlijkheid, kwaliteit en efficiency.

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
Rector Magnificus

Prof.dr. R.C.M.E. Engels

and in accordance with the decision of the Doctorate Board.

The public defence shall be held on

Friday 24 January 2020 at 13:30 hours

by

Thomas Breugem
born in Zoetermeer, the Netherlands.

Erasmus University Rotterdam

Doctoral committee

Promotor:	Prof.dr. D. Huisman
Other members:	Prof.dr. R. Borndörfer
	Prof.dr. J. Larsen
	Prof.dr. A.P.M. Wagelmans
Copromotor:	Dr. T.A.B. Dollevoet

Erasmus Research Institute of Management - ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: www.erim.eur.nl

ERIM Electronic Series Portal: repub.eur.nl

ERIM PhD Series in Research in Management, 494
ERIM reference number: EPS-2020-494-LIS
ISBN 9789058925664

Cover design: PanArt, www.panart.nl

This publication (cover and interior) is printed by Tuijtel on recycled paper, BalanceSilk®.
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Ecolabel, FSC®, ISO14001.
More info: www.tuijtel.com


Acknowledgements

The past years have been a great adventure. The things you learn, people you meet, and places you go, all make a PhD candidacy an unforgettable experience. I cannot think of a better way I could have spent the past years. There are many people that have contributed to this experience, and I would like to use this opportunity to mention some of them. Please note that this 'thank you' is by no means meant to be exhaustive: everybody I have met during the last four years has contributed to this experience, so thank you!

Dennis and Twan, you have been amazing supervisors. Your insight, patience, and rigor have shaped me as an academic and had a great impact on the research we did together. You are always there to help, discuss, and give advice when necessary. I could not have wished for better supervisors. I hope there will be more opportunities to work together in the future.

It was a great privilege to be part of PI while writing my thesis. You are a unique and welcoming group with a large variety of backgrounds and skills. In case you ever lack participants for the midwinter marathon, you know where to find me. I specifically would like to thank Erwin, who, as my NS supervisor, has had great impact on my research with his sharp insights and large amount of experience, and Gábor, Hilbert, and Pieter-Jan, with whom I had the pleasure to work together.

I look back at my time at the Erasmus University with much pleasure. The atmosphere we created among the PhDs and staff felt unique and stimulating. The endless discussions about all kinds of things, the game nights, and the Friday afternoon drinks always were a lot of fun and a source of energy. I enjoyed the conferences we attended, the holidays we spent together, and all the (nonsensical) activities we planned. It was an unforgettable experience. Hopefully, our paths will cross often in the future, I am sure we will have a great time when they do.

I would also like to thank the people from the Zuse Institute in Berlin (ZIB), where I had the pleasure to spend my research visit. The visit to Berlin was a great experience, both on an academic and personal level. I especially would like to thank Christof, Thomas, and Ralf; it was a great experience to work together.

Finally, I would like to thank all the people close to me. I am very happy I have such a supportive, skilled, and fun group of people around me. I am sure we will embark on many new adventures in the future, and I am looking forward to it.

Table of contents

Chapter 1

Introduction

Netherlands Railways (abbreviated as NS, from the Dutch name Nederlandse Spoorwegen) is the major public transport operator in the Netherlands, serving more than 1.3 million passengers on an average working day, and employing a workforce consisting of more than 3,000 drivers and 2,500 guards spread over 28 different crew depots. Each day, these drivers and conductors operate over 5000 timetabled trips, leading to an extremely challenging crew planning problem. As a result, Operations Research (OR) techniques have been used intensively to support (parts of) the crew planning process at NS.

NS is by far the largest operator in the Netherlands: Approximately 90% of the passengers travel with NS. As a result, the operations of NS have a major impact on society: The societal costs of disturbances in the railway system or strikes of NS personnel are estimated to be hundreds of millions of euros*. In both cases, crew planning plays an important role: In the case of disturbances, the re-planning of crew is an important part of an efficient response, whereas strikes can be avoided by incorporating the demands of the employees in the crew planning process. In recent years, decision support has been developed for crew planning at NS (Abbink 2014), yet there remains room for improvement: Parts of the planning process are still solved sequentially (due to their complexity), or even planned by hand. In this thesis, we aim at further improving the crew planning process, making the next step towards decision support for integrated crew planning at NS.

*sources (in Dutch): `https://www.kimnet.nl/mobiliteitsbeeld/mobiliteitsbeeld-2017` and `http://www.seo.nl/pagina/article/ov-staking-kostte-honderden-miljoenen/`.

1.1 Planning Problems and Decision Support

The planning problems at NS can be decomposed based on the involved resources and the considerd planning horizons (Abbink 2014). The three key resources at NS are rail-infrastructure (owned by the government), rolling stock, and crew, leading to planning problems concerning line planning, timetabling, rolling stock allocation and circulation, and crew scheduling and rostering. Each of the planning problems can be roughly categorized into four planning phases, according to different planning horizons: strategic, tactical, and operational planning; and operational control (Abbink 2014). Strategic planning problems, i.e., long term decision making, span numerous years, and include investing in new rolling stock, the construction of the line plan (i.e., where and with which frequency trains are operated), and the hiring and training of new crew. Tactical planning problems have a planning horizon of a few months up to a year. Important tactical planning problems are, for example, the construction of the generic timetable, rolling stock schedule, and crew schedule. Each of these problems is solved in a generic context, i.e., a 'typical' week of work is assumed when solving the problem. The details of these plans are then further developed in the operational planning phase, which has a time horizon of roughly one month. In this phase of the planning process, the finalized plans are obtained. In particular, the generic plans constructed in the tactical planning phase are now transformed into precise plans for each calender day. Finally, operational control focuses on real-time adjustments such as adjusting the operated timetable due to delays or updating the crew schedules whenever a crew member calls in sick.

Every planning phase has unique characteristics and hence calls for different solution approaches. In strategic planning, for example, forecasting is of utmost importance, whereas in many other planning phases parameters can be assumed to be fixed. In the tactical and operational planning phases, on the other hand, we already consider a lot of detail and hence computationally difficult problems arise. It is therefore no surprise that the tactical and operational planning problems are strongly represented in OR literature. Finally, operational control concerns real-time problem solving, and is therefore often approached using heuristics or even simple 'rules of thumb'. As a result, a wide variety of planning methods is applied at NS, ranging from state-of-the-art mathematical programming techniques to simulation-based optimization, all with the goal of operating the Dutch railway network as efficiently as possible. For a thorough discussion of the different planning problems and solution approaches, we refer to Huisman et al. (2005b) and Kroon et al. (2009), and references therein.

1.2 Crew Planning at Netherlands Railways

In this thesis, we focus on crew planning in the tactical and operational planning phases (see, for example, Abbink et al. (2018)). We consider the timetable and rolling stock schedules to be given, i.e., the tasks (indivisible blocks of work) are considered input, and the goal is to assign these tasks to the crew members. In this setting, crew planning at NS is typically decomposed into two phases: crew scheduling and crew rostering. In crew scheduling, the days of work (i.e., duties) have to be constructed, and in crew rostering the constructed duties have to be assigned to the crew members. The construction of duties follows from a centralized process: The duties for all crew bases throughout the Netherlands are constructed simultaneously. The duties are then communicated with the individual crew bases, which then, in a decentralized fashion, each construct the rosters for their crew members.

1.2.1 Crew Scheduling

In crew scheduling, the focus is mainly on operational cost (i.e., the number of necessary crew members), together with the constraints imposed in the collective labor agreement. Among these constraints are a maximum duty length (depending on the start time of the duty) and a proper meal break for every duty exceeding a certain length. Furthermore, each duty should start and end at the same crew base. Figure 1.1 gives an example of three duties, involving the major stations The Hague (Gvc), Zwolle (Zl), Utrecht (Ut), Rotterdam (Rtd), and Groningen (Gn). Note that each duty starts and ends at the same station. Furthermore, a proper meal break (indicated by a star) is specified, and the duty does not exceed 9.5 hours (which is the maximum length for duties starting after 6 in the morning).

The crew schedule implicitly allocates the work over the different crew bases. Modern day duty scheduling at NS is based on the 'Sharing-Sweet-and-Sour' rules, which concern the allocation of work among the different crew bases. Developed around the turn of the century (see Abbink et al. (2005)), this set of rules resolved a seemingly unresolvable conflict between NS and its employees, thereby ending a series of nationwide strikes shutting down the railway operations. The new rules led to improved schedules, both from the crew's and the operator's point of view.

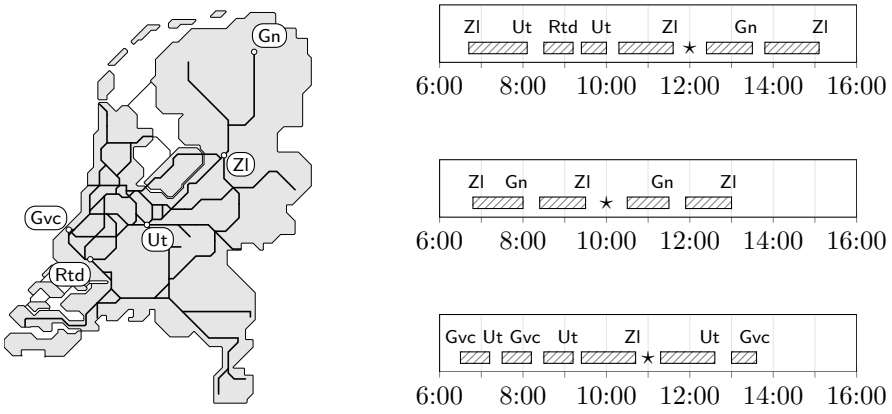


Figure 1.1: Schematic visualization of the railway network operated by NS and examples of three duties traversing this network for employees based at Zwolle (Zl) and The Hague (Gvc). Each block represents a trip. For each trip the departure station (top left) and/or arrival station (top right) are shown. The star indicates a meal break.

1.2.2 Crew Rostering

Crew rostering consists of combining the duties into rosters, which are sequences of duties satisfying numerous labor constraints. Typical constraints consider, for example, days off, rest times, and the variation of work. The rosters at NS are cyclic, i.e., multiple employees are working the same roster in a so-called roster group. Each cyclic roster consists of rows and columns. The rows represent a week of work, and the columns represent each of the week days (Monday to Sunday). The number of rows is always equal to the number of employees in the roster group. Each cell (i.e., intersection of a row and column) represents a single day of work. Furthermore, each cell has a specific type (e.g., early duty or rest day). Figure 1.2 shows an example of a cyclic roster. The numbers indicate the different duties and the top left character indicates a type for each cell: an Early (E), Late (L), or Night (N) duty; or a rest day (R). The first row of the roster starts with two late duties, followed by a rest day, three early duties, and again a rest day.

The employees cycle through the rows of the same roster. Hence, assigning a duty to a given row in the roster means that the duty is performed every week, but each week by a different employee of the roster group. This is illustrated in Figure 1.3, which

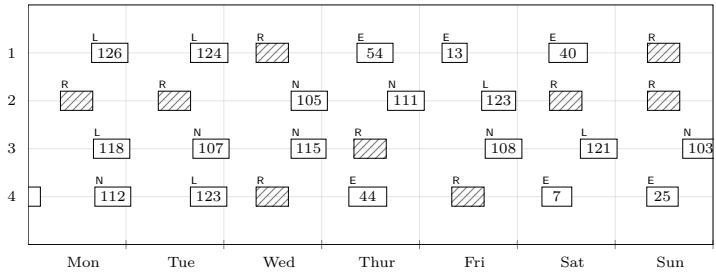


Figure 1.2: Example of a cyclic roster for four employees. The type of each cell (Early, Late, Night, and Rest) is indicated above the cell, and the numbers indicate the assigned duties.

'unfolds' the roster of Figure 1.2 for the first two employees over the period February 11 until February 24. Note that the schedule for the first employee is obtained from the first two rows of the roster, and that of the second employee from the second and third row. The cyclicity implies that duty 105 on Wednesday, which appears in the second row of the roster, is executed by the second employee on February 13, and by the first employee on February 20. In this way, each duty is always covered by one of the employees.

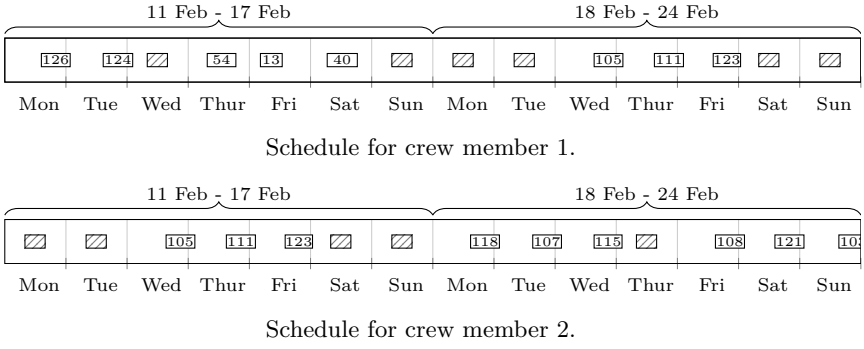


Figure 1.3: Roll-out of the cyclic roster shown in Figure 1.2 for the first two employees. The schedules cover the period from February 11 until February 24.

The quality of the cyclic rosters depends on numerous aspects. These aspects include, for example, the rest times between duties, thereby enforcing a minimum rest time and avoiding, but not forbidding, rest times close to this minimum, and a maximal amount of work within a row (i.e., a working week).

1.2.3 Evaluation and Decision Support

The crew rosters, i.e., the end product of the crew planning phase, can be evaluated based on three criteria: fairness, attractiveness, and efficiency. The efficiency relates mostly to the number of employees necessary, and hence is predominantly considered in the crew scheduling phase. Fairness on the duty level is based on the 'Sharing-Sweet-and-Sour' rules, i.e., an equal distribution of work among the crew bases. These rules naturally extend to crew rostering: The work *per base* should be equally distributed among the roster groups of that base. The attractiveness, on the other hand, relates to the precise rosters, e.g., the rest times and weekly variation within the roster. Although fairness and attractiveness are closely related, they are also substantially different, as we will discuss in Chapter 2.

Decision support for crew planning at NS can be considered both state-of-the-art and non-existing: Duty scheduling is done using a well-designed column generation algorithm, whereas rostering is still a manual process. The 'Sharing-Sweet-and-Sour' rules turned out to be too complex to be taken into account in the manual planning process, which was a major motivator for succesfull implementation of algorithmic support tooling (see Abbink et al. (2005)). For the rostering process such an urgent need did not occur before, although a successful pilot study was conducted in Hartog et al. (2009). As a result, the construction of the rosters is still a manual process.

1.3 Contributions

The contributions of this thesis are fourfold. Firstly, we present novel optimization problems which further integrate the crew planning process at NS. In Chapters 2 and 3 we focus on the combination of fairness and attractiveness in crew rostering, thereby extending the 'Sharing-Sweet-and-Sour' rules to crew rostering. In particular, we analyze the explicit trade-off between fairness and attractiveness, in which we show the trade-off between an equal distribution of work among the roster groups and the overall (utilitarian) quality of the rosters. Furthermore, in Chapter 5, we propose an integrated approach towards crew re-planning (i.e., updating the crew schedules due to planned maintenance), where we exploit additional freedom in the crew rosters to efficiently re-schedule the crew after disruptions. In this chapter we focus on efficiency (i.e., minimizing the number of necessary duties), while explicitly assuring that hard constraints regarding attractiveness (e.g., minimum rest times) are respected.

Secondly, we develop suitable solution methods for each of the presented problems. In Chapters 2 and 4 we develop exact branch-and-price solution methods for crew rostering with and without fairness requirements. To cope with the large instances encountered in practice, we also propose sophisticated heuristics: In Chapter 3 we built upon the exact solution framework proposed in Chapter 2 to obtain efficient heuristics for crew rostering with fairness requirements. In doing so, we combine column generation techniques with state-of-the-art local search techniques. In Chapter 4 we show that good solutions for difficult instances can be found by using heuristic branching strategies, and, in Chapter 5, we propose a sophisticated column generation heuristic able to cope with large re-planning instances. In all cases, we obtain strong lower bounds to assess the solution quality.

Thirdly, we provide rigorous theoretical results to motivate the problems and mathematical formulations presented. In Chapter 2, we analyze a class of resource allocation problems, in which the resource allocation is based on approximate utility functions. We consider a fairness scheme for this class, inspired by the 'Sharing-Sweet-and-Sour' rules, and analyze this scheme in detail. In particular, we show that the proposed scheme, on which the mathematical formulation is based, has optimal properties under the assumption that all employees derive similar utilities. Furthermore, we derive a tight upper bound for the loss of attractiveness for the considered fairness scheme whenever the utility can differ among employees. In Chapter 4, we propose a family of formulations for crew rostering, and derive explicit analytical results regarding the relative strength of the formulations. Furthermore, we show which type of constraints lead to the theoretically strongest formulation possible.

Finally, we evaluate the performance of all developed solution methods using real-world data from NS. In Chapters 2, 3, and 4 we consider data from crew base Utrecht, one of the largest crew bases in the country. In Chapter 5 we consider re-planning instances based on historical data. By using real-world data from NS, the practical benefit of the developed solution methodology can readily be examined.

1.4 Thesis Overview

Figure 1.4 gives a schematic overview of the problems considered in each chapter. In Chapters 2, 3, and 4, we focus on crew rostering on the tactical level (i.e., the allocation of duties to the roster groups and the construction of the rosters per group)

and in Chapter 5 we consider the simultaneous adjustment of duties and rosters (i.e., integrated crew re-planning) in the operational planning phase.

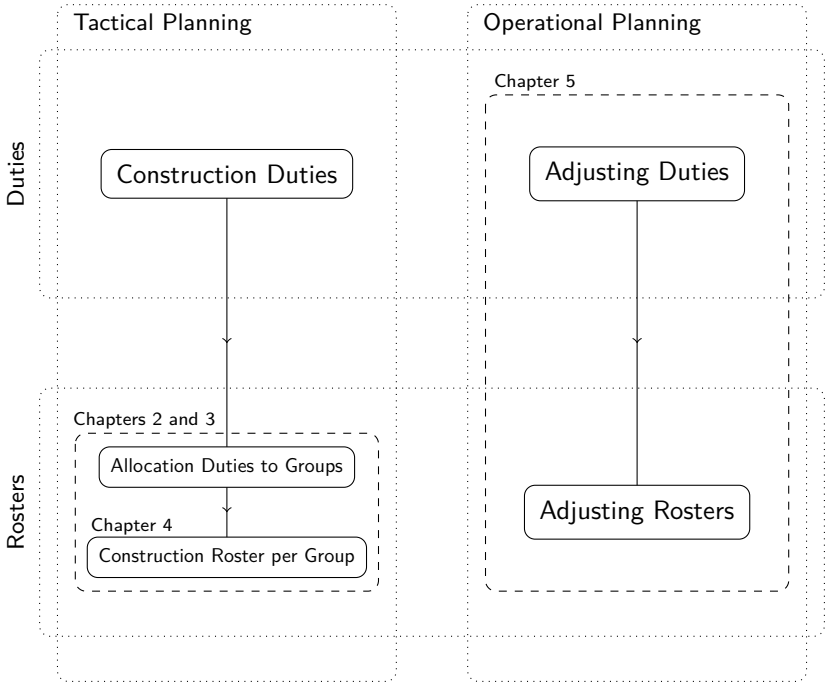


Figure 1.4: Schematic overview of the problems considered in this thesis. The vertical blocks indicate the planning horizon (i.e., tactical and operational planning), and the horizontal blocks indicate the output of each of the problems (i.e., duties and rosters). The arcs indicate the (current) order in which the problems are solved, and for each chapter the considered problems are indicated.

Each chapter in this thesis can be read as a stand-alone work. We recommend, however, to read the chapters in their natural order. Chapters 2 and 3 follow naturally, as Chapter 3 builds upon the research presented in Chapter 2. The work in Chapter 4 consists of an in-depth analysis of the modeling techniques used in Chapters 2 and 3. We therefore strongly recommend to read Chapter 4 after Chapters 2 and 3. The work presented in Chapter 5 has little overlap with the other chapters, and can therefore easily be read as an independent chapter.

Below, we briefly summarize each chapter and re-state the contributions per chapter. The chapters are modifications of papers submitted to academic journals, or papers in the final stage before submission. The work in Chapters 2, 4, and 5 has been

done independently, under close supervision of mentioned co-authors, and the work in Chapter 3 has been done in close collaboration with the mentioned co-authors.

Chapter 2: T. Breugem, T. Dollevoet, and D. Huisman: Is Equality always desirable? Analyzing the Trade-Off between Fairness and Attractiveness in Crew Rostering*, currently under second revision at Management Science. This paper has been awarded the INFORMS RAS 2017 best student paper award.*

In this chapter, we analyze the trade-off between perceived fairness and perceived attractiveness in crew rostering. First, we introduce the Fairness-oriented Crew Rostering Problem (FCRP). In this problem, fair and attractive cyclic rosters have to be constructed for groups of employees. We consider a fairness scheme motivated by a class of resource allocation problems, in which the allocation is based on approximate utility functions. We analyze this fairness scheme in detail and derive a tight upper bound on the relative loss of attractiveness. We then propose a mathematical formulation for the FCRP and develop an exact branch-price-and-cut solution method. We conclude by applying our solution approach to practical instances from NS: We analyze the explicit trade-off curve between fairness and attractiveness, and confirm the loss of attractiveness due to fair allocations. Thus, in order to generate high-quality rosters, the explicit trade-off between fairness and attractiveness should be taken into account.

Chapter 3: T. Breugem, C. Schulz, T. Schlechte, and R. Borndörfer: A Three-Phase Heuristic for Cyclic Crew Rostering with Fairness Requirements*, in preparation for journal submission.*

In this chapter, we consider the Cyclic Crew Rostering Problem with Fairness Requirements (CCRP-FR). In this problem, attractive cyclic rosters have to be constructed for groups of employees, considering multiple, a priori determined, fairness levels. We propose a three-phase heuristic for the CCRP-FR, which combines the strength of column generation techniques with a large-scale neighborhood search algorithm. The design of the heuristic assures that good solutions for all fairness levels are obtained quickly, and can still be further improved if additional running time is available. We evaluate the performance of the algorithm using real-world data from NS, and show that the heuristic finds close to optimal solutions for many of the considered instances. In particular, we show that the heuristic is able to quickly find major improvements upon the current sequential practice: For most instances, the heuristic is able to increase the attractiveness by at least 20% in just a few minutes.

Chapter 4: T. Breugem, T. Dollevoet, and D. Huisman: Analyzing a Family of Formulations for Cyclic Crew Rostering*, in preparation for journal submission.*

In this chapter, we analyze a family of formulations for the Cyclic Crew Rostering Problem (CCRP), in which a cyclic roster has to be constructed for a group of employees. We derive analytical results regarding the relative strength of the different formulations, which can serve as a guideline for formulating a given problem instance. Furthermore, we propose a column generation approach, which we use to develop a branch-and-price solution method. We conclude by applying our proposed solution method to practical instances from NS. In particular, we show that the computation time depends heavily on the selected formulation, and that the column generation approach outperforms a commercial solver on hard instances.

Chapter 5: T. Breugem, T. Dollevoet, and D. Huisman: A Column Generation Approach for the Integrated Crew Re-Planning Problem, *in preparation for journal submission.*

In this chapter, we propose a column generation solution approach for crew re-planning. The problem of re-scheduling the crew has been formalized as the Crew Re-Scheduling Problem (CRSP) in Huisman (2007). In the current practice, the feasibility of the new rosters is 'assured' by allowing the new duties to deviate only slightly from the original ones. In the Integrated Crew Re-Planning Problem (ICRPP) we aim at exploiting exactly this flexibility: The ICRPP considers the re-scheduling of crew for multiple days simultaneously, thereby allowing more flexibility in the re-scheduling. We propose a mathematical formulation for the ICRPP and develop a column generation approach to solve the problem. We apply our solution approach to practical instances from NS, and show the benefit of integrating the re-scheduling process.

Chapter 2

Is Equality always desirable? Analyzing the Trade-Off between Fairness and Attractiveness in Crew Rostering*

2.1 Introduction

In recent years, the Netherlands has seen numerous strikes from personnel of Netherlands Railways (NS). Reasons for these strikes were, for example, little variation in work (adding the infamous 'rondje om de kerk' or 'circling the church', a mocking reference to repetitive work, to the Dutch vocabulary), or demand for higher staffing levels for certain rolling stock. One of such conflicts led to the development of the 'Sharing-Sweet-and-Sour' rules ('Lusten-en-Lasten-Delen' in Dutch), a new set of scheduling rules aimed at increasing the quality of work. As mentioned in Abbink

*This chapter, up to minor modifications, is a direct copy of T. Breugem, T. Dollevoet, and D. Huisman (2017): *Is Equality always desirable? Analyzing the Trade-Off between Fairness and Attractiveness in Crew Rostering.*

et al. (2005), one of the key success factors of the project was the openness and transparency during the development of the new rules. Using state-of-the-art Operations Research techniques, NS was able to generate schedules satisfying the new rules for all employees, leading to a new agreement between NS and the labor unions. These events highlight the importance of a participative approach to crew planning. In particular, they show the importance of incorporating the demands of personnel in the planning process.

The construction of *rosters* (i.e., the precise assignment of duties to the employees) is a major part of the personnel planning process. Unlike many operational problems, the goal when creating rosters is not to minimize expenses. Instead, the rosters are evaluated on two different aspects: *perceived fairness* and *perceived attractiveness*. Perceived fairness considers the distribution of work among personnel. In line with the 'Sharing-Sweet-and-Sour' rules, the aim is to balance certain attributes as fairly as possible over the employees. A first step in achieving this is the use of *roster groups*. That is, to use groups of employees that are assigned the same work. This approach leads to cyclic rosters, as often seen at railway operators and other public transport companies. In a cyclic roster, each employee of the roster group 'cycles' through the same roster, which implies that each employee performs exactly the same work in the long term. Although *within* each group every employee does the same work (and hence the distribution is fair), it is not necessarily the case that the distribution of work *among* roster groups is fair. Perceived fairness therefore takes the distribution among the different groups into account, aiming for an overall fair allocation of duties. Perceived attractiveness, on the other hand, focuses on the rosters on an individual level, that is, on the actual work scheduled. In the attractiveness of the roster, one can take, for example, the workload over the different weeks into account. Furthermore, sufficient rest time and other (un)desirable properties can be incorporated.

It is important to note that fairness and attractiveness are distinct concepts: Fairness considers solely the allocation of work to roster groups, without any regard to the actual quality of the rosters. Attractiveness, on the other hand, considers solely the quality of the rosters, without any regard to the bigger picture (i.e., the distribution of work over the groups). Hence, one can have a very fair, but unattractive, set of rosters, and vice versa. For example, one can have a balanced distribution of the total workload over the groups (i.e., a fair allocation), but at the same time a set of rosters where certain weeks have a very high workload (i.e., unattractive rosters). Hence, crew rostering is a trade-off between perceived fairness and perceived attractiveness.

On the one hand, the allocation of duties should be as fair as possible, while on the other hand, the attractiveness of the rosters should be maximized.

In this chapter, we present a unified approach to crew rostering. This approach allows for a simultaneous optimization of the perceived fairness and the perceived attractiveness. We call this problem the Fairness-oriented Crew Rostering Problem, abbreviated as FCRP. In current approaches, crew rostering is solved in a sequential fashion. First, an assignment of duties to the groups is made. Then, the attractiveness of the separate rosters is optimized. In such a sequential optimization procedure, possible good solutions might be lost: Focusing solely on fairness in the first stage can lead to rosters that turn out disproportionally unattractive in the second stage. As a consequence, the resulting rosters can be perceived undesirable by the employees. In the FCRP, on the other hand, we aim at finding the explicit trade-off between fairness and attractiveness for the rostering problem as a whole.

The contribution of this chapter is fourfold. Firstly, we analyze a class of resource allocation problems, in which the resource allocation is based on approximate utility functions. We consider a fairness scheme for this class, inspired by the 'Sharing-Sweet-and-Sour' rules, and analyze this scheme in detail. In particular, we derive a tight upper bound for the loss of attractiveness for the considered fairness scheme. Secondly, we propose a mathematical formulation to solve the FCRP. The formulation we propose is versatile, and can be easily adapted to different settings. Thirdly, we develop an exact Branch-Price-and-Cut solution method for the FCRP. Finally, we apply the solution method to real life instances at NS, where we show the benefits of our integrated approach. In particular, we generate multiple rosters with a different trade-off between fairness and attractiveness, and confirm that the roster with the highest fairness might not be the most desirable one.

The remainder of this chapter is organized as follows. In Section 2.2, we discuss crew rostering in detail and formalize the FCRP. In Section 2.3, we give an overview of related research, and, in Section 2.4, we analyze the fairness scheme inspired by the 'Sharing-Sweet-and-Sour' rules. Our mathematical model is introduced in Section 2.5, and in Section 2.6, we propose a Branch-Price-and-Cut approach to solve the FCRP. In Section 2.7, we show the benefits of our approach in a case study at NS, and discuss the acquired managerial insights.

2.2 Problem Description

We now discuss crew rostering in more detail. We first give a description of the input for the crew rostering phase and discuss the concept of cyclic rostering. We then discuss how perceived fairness and perceived attractiveness are measured, and end with a formal statement of the FCRP.

The input to the crew rostering phase is a *basic schedule* for each roster group, and a set of *generic duties* (often simply referred to as duties). Each basic schedule specifies the key elements of the roster. This schedule, for example, could specify when employees have a day-off or what type of work should be scheduled on a certain day (the level of detail may vary according to the application). Figure 2.1 shows an example of a set consisting of three basic schedules, each indicated by one of the stacked rectangles. Each basic schedule consists of *rows* and *columns*. The rows represent a week of work, and the columns represent each of the week days (Monday to Sunday). Each *cell* (i.e., intersection of a row and column) represents a single day of work. Note that not all basic schedules in Figure 2.1 have the same number of rows (i.e., the number of employees cycling through this schedule). In this example, the foremost basic schedule specifies that the first row of the roster starts with a late duty, followed by two night duties, and then a day off (indicated by L, N and R in the basic schedule, respectively). In practice, the basic schedules are generally created manually by the planners and based on those used in the previous year. For a more detailed discussion on the construction of basic schedules, we refer to Hartog et al. (2009) and Abbink et al. (2018).

The use of generic duties is a consequence of using cyclic rosters. Recall that in a cyclic roster multiple employees cycle through the rows of the same roster. Hence, assigning a duty to a given cell in the roster means that the duty is performed every week, but each week by a different employee of the roster group. The duties are therefore generic, which means that each duty belongs to, for example, Wednesday, and not to, say, Wednesday the 11th of October. A detailed example of a cyclic roster is shown in Figure 2.2. The numbers in the cells indicate different scheduled duties. The roster consists of four rows, meaning that it belongs to a roster group of four employees. The cyclicity of the roster implies that the second employee starts in row 2, and, after completing this row, he or she carries out the duties of row 3. Similarly, the duty indicated by 118 on Monday, is first carried out by the third employee, and then a week later by the second employee. Note that the cyclicity implies that

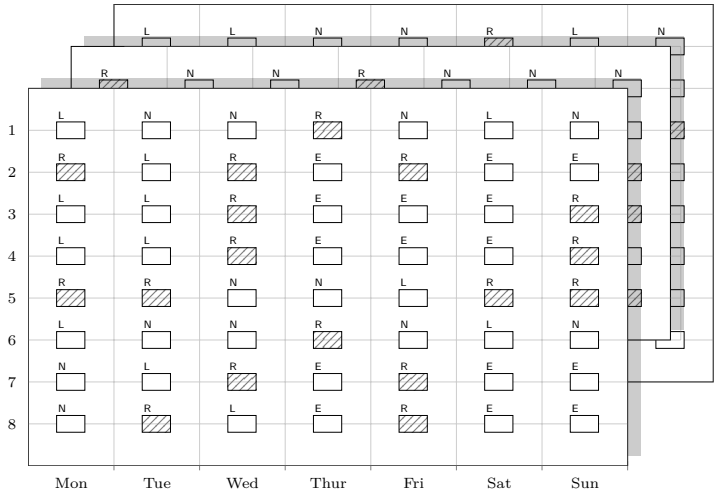


Figure 2.1: Example of basic schedules for three groups. The schedules specify the early (E), late (L), and night (N) duties; and rest days (R).

after four weeks each employee has carried out each duty shown in Figure 2.2 exactly once. Furthermore, note that some overlap between rows (and hence weeks) can be present. For example, the last duty of row 3 starts on Sunday night and ends on Monday morning.

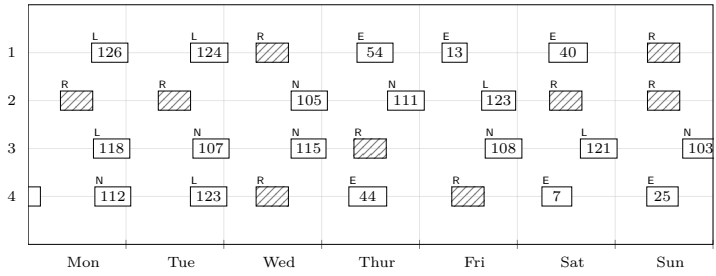


Figure 2.2: Example of a cyclic roster for four employees.

In practice, multiple rosters need to be constructed simultaneously. That is, the given set of duties should be assigned to the cells of the different basic schedules at once. The goal is to find a fair assignment of the duties such that the resulting rosters are feasible and of high quality. This implies that a trade-off has to be made between perceived fairness and perceived attractiveness.

The perceived fairness of the rosters is based on *duty attributes*. Since a duty consists

of multiple tasks, it has certain characteristics. Hence, we can say that some duties are 'desirable' and some are not. Duties with many short trips, for example, are undesirable, while duties with fewer but longer trips are considered desirable, as long trips allow crew members to traverse large parts of the Netherlands at high speed. We refer to these desirable and undesirable characteristics as duty attributes. Other examples of duty attributes at NS are the duty length and the percentage of work on double decker trains. A fair allocation, according to the 'Sharing-Sweet-and-Sour' rules, means that the spread (i.e., the difference between the maximum and minimum average value) over the groups is minimized. Note that a spread of zero implies that all groups are assigned the exact same average values.

The perceived attractiveness and the feasibility of a roster are expressed using *roster constraints*. These constraints impose restrictions on the assignment of the duties to the basic schedule. Rest constraints, for example, are often present in crew rostering. Rest constraints enforce that an employee has a certain minimum time to rest after a duty (if it is a hard constraint), or penalize rest times shorter than a certain threshold (if it is a soft constraint). Another classical example are workload constraints, enforcing a maximum amount of work within a week. The perceived attractiveness is maximized by minimizing the penalty incurred from the soft roster constraints. As we will show in Section 2.5, the developed model allows for a broad range of roster constraints. In Section 2.7, we discuss the duty attributes and roster constraints applied at NS.

The FCRP can now be stated as follows: Given a set of duties, and a set of basic schedules, create cyclic rosters which are perceived both fair and attractive. The goal of the FCRP is to present a set of solutions, each optimal for a different trade-off between fairness and attractiveness. We note that the FCRP, although formulated for a cyclic context, naturally extends to domains where acyclic rostering is common practice (e.g., nurse rostering, airline crew planning). A fair allocation for cyclic rostering means that the duties should be fairly allocated over the different roster groups, as each employee in the group performs the same work. Hence, a fair allocation for acyclic rostering means that the duties should be fairly allocated over the *individual* employees, as each employee is assigned a personal roster. It is not difficult to see acyclic rostering as a special case of cyclic rostering with regard to the FCRP: Each basic schedule could correspond to an individual employee, instead of a group of employees. Hence, although we focus on cyclic rostering, the approach presented in this research can be extended to acyclic rostering problems as well.

2.3 Literature Review

In this section we give an overview of related literature. We first discuss the literature related to crew planning, thereby focusing on crew rostering in particular. Then, we give a brief overview of the literature related to perceived fairness. Finally, we discuss the research that integrates some form of fairness concept in crew rostering, i.e., the research that relates most closely to our work.

Crew planning appears in a wide variety of contexts, such as the railway sector, the airline industry and the healthcare sector. The railway sector focuses mainly on cyclic rosters, whereas the airline industry almost exclusively works with acyclic rosters. Ernst et al. (2004) and Van den Bergh et al. (2013) give extensive overviews of different staff scheduling problems, discussing both applications and solution methods. Kohl and Karisch (2004) give a detailed overview of applications and techniques for crew planning at an airline operator. Burke et al. (2004) present the state-of-the-art for nurse rostering, i.e., the rostering of personnel at medical facilities. Huisman et al. (2005b), Caprara et al. (2007), and Abbink et al. (2018) review popular models and solution methods for crew planning at a railway operator.

Crew planning is often decomposed into two consecutive planning problems: crew scheduling, and crew rostering. The crew scheduling problem consists of constructing the duties or pairings, given the tasks, whereas the crew rostering problem consists of constructing the rosters, given the duties/pairings (which are the output of the crew scheduling problem). The crew scheduling problem is well-studied, and appears in the literature in many variants (see, for example, Stojković and Soumis (2001) and Abbink et al. (2005) for an operational variant and a strategic variant of the problem, respectively).

In crew rostering, many complex labor rules have to be taken into account (e.g., maximum total workload and sufficient rest time for each employee). The (railway) crew rostering problem occurs in roughly two variants: either the basic schedules are considered input, or the construction of the basic schedules is part of the optimization problem (i.e., the rosters are constructed without a pre-specified basic schedule). Proposed algorithms for crew rostering vary according to the considered objective and constraint structure (see Van den Bergh et al. (2013) for a detailed overview).

Hartog et al. (2009) propose an assignment model with side constraints to solve the crew rostering problem at NS. They first optimize the basic schedules, and then the assignment of the duties to the schedules (as first proposed in Sodhi and Norris

(2004)). Their results were (blindly) presented to the NS workforce, which preferred their generated rosters over manually constructed ones (see Hartog et al. (2009) for the details of this experiment). Xie and Suhl (2015) propose a multi-commodity flow formulation for the cyclic and non-cyclic crew rostering problem. They consider both the sequential approach of Sodhi and Norris (2004) and an integrated approach (i.e., constructing the rosters directly without first constructing basic schedules). The developed methods are applied to practical instances of a German bus company, for which reasonable sized instances could be solved using a commercial solver. Mesquita et al. (2013) develop a model for the integrated vehicle scheduling and crew rostering problem, for which they develop a non-exact Benders decomposition approach.

Caprara et al. (1997) propose alternative formulations for crew rostering. They develop a multi-commodity flow model and a set partitioning model. The usefulness of the models is related to the constraint set (e.g., a set partitioning model is preferred when many high level constraints are imposed). The developed models are applied to crew rostering at an Italian railway operator. Freling et al. (2004) develop a flexible Branch-Price-and-Cut algorithm based on a set covering formulation. The performance of their approach is evaluated on different practical instances. Borndörfer et al. (2015) discuss both a network flow model and a set partitioning model. They propose a heuristic solution method based on the well-known Lin-Kernighan heuristic (Lin and Kernighan 1973). Their algorithm is evaluated for cyclic rostering in public transport and for the rostering of toll enforcement inspectors.

Perceived fairness as considered here closely relates to *distributive justice* (see e.g., Greenberg (1990)). This concept originates from equity theory, and relates to the equitability of resource distributions. Although often limited to monetary compensation, the concept readily extends to other domains (e.g., desirable and undesirable work). As mentioned before, a similar concept was applied in Abbink et al. (2005) in the context of crew scheduling, thereby ending a seemingly unresolvable conflict between NS and the labor unions. The effects of distributive justice are well-established in the literature. In different meta-analytical studies (Colquitt et al. (2001), Colquitt et al. (2013)) a significant relation is found between distributive justice and, for example, task performance and organizational trust. Similar results are presented in Rhoades and Eisenberger (2002), where a significant relation is found between distributive justice and perceived organizational support.

The joint analysis of fairness, i.e., the equitability of the work allocation, and efficiency (e.g., operational costs) has received considerable attention in recent literature.

Recent work considering fairness includes Bertsimas et al. (2013) with regard to organ allocation for kidney transplantation, and Barnhart et al. (2012) and Bertsimas and Gupta (2015) with respect to air traffic flow management. Bertsimas et al. (2011) derive tight bounds for the price of fairness, i.e., the efficiency loss due to a fair allocation, for well-known fairness schemes. Finally, Bertsimas et al. (2012) derive bounds regarding the trade-off between fairness and efficiency, and argue how to apply this to practical problems.

Fairness in crew rostering, however, remains relatively unexplored. Hartog et al. (2009) propose to allocate the duties using an assignment model with side constraints. The allocation is determined a priori creating the rosters, hence no direct trade-off is made between fairness and attractiveness. Their fairness concept is identical to the one considered in the FCRP. Borndörfer et al. (2015) incorporate fairness in their solution approach by penalizing undesirable sequences of duties (e.g., changing starting times on consecutive days). Especially interesting is their incorporation of fatigue measures, a highly non-linear concept, in a rostering application for airline traffic. Nishi et al. (2014) propose a decomposition approach for crew rostering with fair working conditions. Their concept of fairness solely concerns the distribution of workload. Their decomposition approach splits the problem into assigning the duties to groups and creating the rosters (similar to Hartog et al. (2009)). The resulting problem is solved using Benders decomposition. Finally, Maenhout and Vanhoucke (2010) propose a hybrid scatter search heuristic for crew rostering with fair working conditions.

Summarizing, crew rostering and perceived fairness are both well studied in the literature. The integration of these two concepts, however, is, to the best of our knowledge, relatively unexplored in this context. Although some approaches incorporate (parts of) a fairness concept in crew rostering, none of these approaches consider a joint optimization of the attractiveness of the rosters and the equitability of the duty allocation, so that an explicit trade-off can be made.

2.4 'Sharing-Sweet-and-Sour' Rules: A Theoretical Analysis

In this section, we analyze the 'Sharing-Sweet-and-Sour' rules in more detail. We do this by introducing a class of resource allocation problems, where the resource

allocation is based on approximate utility functions. To assure that our terminology is consistent with the literature, we will refer to the roster groups as *players* and to the duty attributes as *resources* throughout this section.

We focus on resource allocation problems traditionally solved in two phases: In the first phase, an allocation of resources to the players has to be determined, and, in the second phase, each player derives a utility from his or her allocated resources. The allocation of resources in the first phase, however, is based on *approximate* utility functions, instead of *exact* utility functions. This setting is typical for sequential optimization problems: The allocation influences derived utility (or cost) in later stages of the planning process. This relationship, however, is not easily established, due to, e.g., the complexity of the different planning stages. As a result, the quality of the allocation is evaluated based on simpler indicators. In other words, the utility in later stages is approximated when allocating the resources. The use of approximate utilities could also be motivated by a desire for transparency and rules for allocating the resources that are easy to explain. We refer to this type of problem as an Approximate Resource Allocation Problem (ARAP).

The current sequential approach for crew rostering at NS fits the ARAP paradigm. First, the duties are allocated over the groups, based on the duty attributes. The allocation to the groups is according to the 'Sharing-Sweet-and-Sour' rules, i.e., it assures that each group has roughly the same average amount of each attribute. Here, it is implicitly assumed that all groups derive a similar utility from each attribute (e.g., all groups derive the same utility from a given average duty length). The exact utility (i.e., attractiveness), on the other hand, may differ among groups, due to e.g., the size and structure of the basic schedules. We show that the 'Sharing-Sweet-and-Sour' rules are a natural expression of fairness for the ARAP. To be more precise, we show that 'Sharing-Sweet-and-Sour' rules lead to efficient and fair allocations with respect to a large class of approximate utility functions. Furthermore, we bound the loss of overall (exact) utility due to applying these rules.

The remainder of this section is organized as follows. In Sections 2.4.1 and 2.4.2, we develop the necessary notation and terminology, and in Section 2.4.3, we formalize the fairness scheme based on the 'Sharing-Sweet-and-Sour' rules, and derive analytical results regarding this scheme.

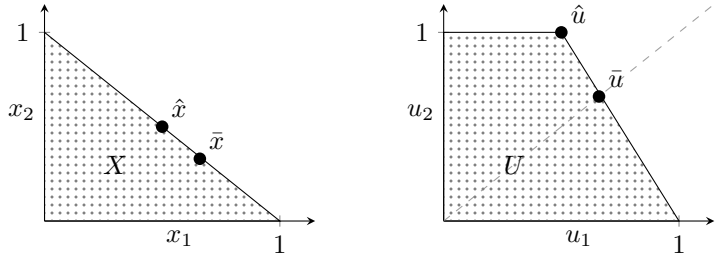
2.4.1 Resource Allocation Problems and Fairness

The ARAP can be seen as an extension of the Resource Allocation Problem (RAP), a well-studied problem in the economic literature. In the RAP, a set of k resources has to be allocated among n different players. The resource set specifies all possible allocations of resources to the players. This set is represented by $X \subseteq \mathbb{R}_+^{nk}$. For notational convenience, we denote, for some allocation $x \in X$, the resource allocation of the i-th player by $x_i \in \mathbb{R}_+^k$, and the amount of resource j allocated to the i-th player by $x_{ij} \in \mathbb{R}_+$. Figure 2.3a gives an example of the resource set for two players and one resource. The resource set, in this case, is given by all $x \in \mathbb{R}_+^2$ satisfying $x_1 + x_2 \leq 1$.

Each player is assigned a utility function $f_i : X \to \mathbb{R}_+$, i.e., $f_i(x)$ represents the utility the i-th player derives from allocation $x \in X$. The utility set U consists of all achievable utilities with respect to the utility functions f_i. That is, the utility set U is defined as

$$U = \left\{ u \in \mathbb{R}_+^n \mid \exists x \in X : u_i = f_i(x),\ i = 1, \ldots, n \right\}.$$

For notational convenience, let $U(x) \in U$ denote the utility derived from the allocation $x \in X$. Figure 2.3b shows the utility set U for $f_1(x) = x_1$, and $f_2(x) = \min\{1, 2x_2\}$, where X, as shown in Figure 2.3a, is the underlying resource set.



(a) Resource allocations $\bar{x}$ and $\hat{x}$. (b) Utilities $\bar{u}$ and $\hat{u}$.

Figure 2.3: Resource set $X \subseteq \mathbb{R}_+^2$ restricted by $x_1 + x_2 \leq 1$, and corresponding utility set $U \subseteq \mathbb{R}_+^2$, for $f_1(x) = x_1$ and $f_2(x) = \min\{1, 2x_2\}$. The points $\bar{u} = (2/3, 2/3)$ and $\hat{u} = (1/2, 1)$ correspond to the max-min fair and utilitarian allocation, respectively. The allocations $\bar{x} = (2/3, 1/3)$ and $\hat{x} = (1/2, 1/2)$ show the corresponding resource allocations.

The goal of the RAP is to find a resource allocation which is 'good' with respect to the derived utilities. One could for example allocate the resources such that the sum

over all utilities is maximized. This is commonly known as the *utilitarian* allocation. Figure 2.3 shows the utilitarian allocation for the utility set U and the corresponding resource allocation in X, indicated by $\hat{u}$ and $\hat{x}$, respectively. Note, that a utilitarian allocation might be perceived unfair by the players, as one player might derive much more utility compared to the other players.

The concept of fairness is formalized using fairness schemes: A fairness scheme is a function $S : 2^{\mathbb{R}^n_+} \to \mathbb{R}^n_+$ which maps a utility set to an element of that set. The utility vector $S(U) \in \mathbb{R}^n_+$ is said to be the 'fair' allocation, with respect to the fairness scheme S. Bertsimas et al. (2011) give a detailed discussion on the axioms that a fairness scheme should (ideally) satisfy. Resulting from these axioms are two well-established fairness schemes: *max-min* fairness, where the utility allocation is lexicographically maximized and which generalizes the Kalai-Smorodinsky solution for the two player bargaining problem (see Kalai and Smorodinsky (1975)), and *proportional* fairness, which generalizes the Nash solution for the two player bargaining problem (see Nash (1950)). Figure 2.3 shows the max-min fair allocation for the utility set U and the corresponding resource allocation in X, indicated by $\bar{u}$ and $\bar{x}$, respectively.

An important characteristic of a fairness scheme is the 'loss' of overall utility due to applying this scheme. To analyze this loss of utility, we use the following definitions, which were introduced in Bertsimas et al. (2011). Let $\text{SYSTEM}(U)$ denote the utility derived from the utilitarian allocation, i.e.,

$$\text{SYSTEM}(U) = \max_{u \in U} \sum_{i=1}^{n} u_i. \tag{2.1}$$

Furthermore, let $\text{FAIR}(U, S)$ denote the utility derived under the fairness scheme S, i.e.,

$$\text{FAIR}(U, S) = \sum_{i=1}^{n} S(U)_i. \tag{2.2}$$

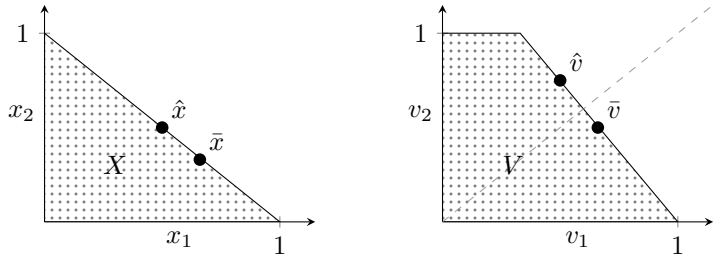
The Price of Fairness (POF) of a fairness scheme S, given a utility set U, represents the relative amount of utility lost due to enforcing S. Formally, the POF for S, given U, denoted by $\text{POF}(U, S)$ is defined as

$$\text{POF}(U, S) = \frac{\text{SYSTEM}(U) - \text{FAIR}(U, S)}{\text{SYSTEM}(U)}. \tag{2.3}$$

Bertsimas et al. (2011) derived upper bounds on the POF for both max-min and proportional fairness, and showed that the derived bounds are tight (i.e., can be achieved for certain classes of problems).

2.4.2 Approximate Utility Functions

The ARAP extends the RAP by considering not one but two utility functions per player: an approximate utility function $f_i : X \to \mathbb{R}_+$, and an exact utility function $g_i : X \to \mathbb{R}_+$. The resource allocation is based on the approximate utilities, but each player's derived utility follows from the exact utilities. Note that the ARAP collapses to the RAP whenever the approximate and exact utilities are equal. For any RAP and ARAP, the derived utility corresponds directly with one or multiple resource allocations. The key question in the ARAP is how a resource allocation based on the approximate utilities f_i influences the final utility derived from the exact utilities g_i.



(a) Resource allocations $\bar{x}$ and $\hat{x}$. (b) Utilities $\bar{v}$ and $\hat{v}$.

Figure 2.4: Resource set X, similar to 2.3a, and corresponding utility set V, for $g_1(x) = x_1$ and $g_2(x) = \min\{1, (3/2)x_2\}$. The points $\bar{v} = (2/3, 1/2)$ and $\hat{v} = (1/2, 3/4)$ show the derived utilities for $\bar{x}$ and $\hat{x}$. In this case, $\bar{v}$ and $\hat{v}$ do not correspond to a max-min fair and utilitarian allocation in V, respectively.

It is clear that differences between the approximate and exact utilities could lead to 'sub-optimal' behavior. In particular, the utility derived by the players might be skewed whenever some utility functions are approximated either too optimistically or too pessimistically, and the overall derived utility could be lower than possible. The above is illustrated in Figure 2.4. Suppose U was only an approximation of the utility set, e.g., suppose that the exact utility of the second player was given by $\min\{1, (3/2)x_2\}$, instead of $\min\{1, 2x_2\}$. In other words, the exact utility of the second player was modeled too optimistically. The resulting utility set, denoted by V, is depicted in Figure 2.4b. In this case, the resource allocations $\bar{x}$ and $\hat{x}$ no

longer correspond to max-min fair and utilitarian allocations. In particular, the max-min fair allocation, which normally guarantees a balanced utility when possible, is skewed, and the utilitarian allocation, normally guaranteeing the largest overall utility, is sub-optimal.

2.4.3 Analytical Results

We now turn to a specific class of ARAPs. We first introduce and analyze a fairness scheme, which relates closely to the notion of perceived fairness defined in Section 2.2. We conclude by proving a tight upper bound on the POF for this fairness scheme. We do this under the following assumptions.

Assumption 2.4.1. *The exact utility function $g_i(x)$ of the i-th player depends only on x_i, i.e., the utility function $g_i(x)$ can be written as $g_i(x_i)$, with $g_i : \mathbb{R}_+^k \to \mathbb{R}_+$. The function g_i is assumed to be concave, bounded, and continuous over X.*

Assumption 2.4.1 states that the utility of the i-th player depends only on the allocation of resources to the i-th player. In other words, the derived utility depends only on the player's own allocation. This assumption, including the concavity of the utility function, is similar to that in Bertsimas et al. (2011), and common in the literature.

Assumption 2.4.2. *The resource set $X \subseteq \mathbb{R}_+^{nk}$ consists of all $x \in \mathbb{R}_+^{nk}$, such that $\sum_{i=1}^{n} x_i = \gamma$. Here $\gamma \in \mathbb{R}_+^k$ specifies the available amount of each resource.*

Assumption 2.4.2 states that each resource is non-disposable, i.e., a fixed amount should be divided among the players. Note that this is the case for the FCRP, where a fixed set of duties needs to be allocated among the groups.

Assumption 2.4.3. *The approximate utility function $f_i(x)$ of the i-th player is of the form $f_i(x) = \phi_i f(x_i/\phi_i)$, with $f : \mathbb{R}_+^k \to \mathbb{R}_+$ the same for each player. Here $\phi_i \in \mathbb{N}_+$ represents the 'size' of the player. The function f is assumed to be concave, bounded, and continuous over X.*

Assumption 2.4.3 states that the approximate utility of each player is based on the allocated resources (divided by the player's size), multiplied by the size of the player. This type of utility function is motivated by practice. At NS, for example, the allocation of duties follows from a centralized process. During this process, the

allocation per roster group is evaluated based on the 'average work' allocated, e.g., two 'bad' duties for a group of four employees is considered equivalent to four of such duties for a group of eight employees. Furthermore, it is assumed that there is little distinction between groups when it comes to the evaluation of this allocation, i.e., f is assumed to be the same for each group. The multiplication with the group size assures that all employees are weighed equally.

Under Assumptions 2.4.2 and 2.4.3, the optimal resource allocation regarding the approximate utilities can be derived.

Theorem 2.4.1. *Consider an* ARAP *satisfying Assumptions 2.4.2 and 2.4.3. The vector* $\hat{x} \in X$ *with*

$$\frac{\hat{x}_i}{\phi_i} = \frac{\gamma}{\sum_{j=1}^{n} \phi_j}, \tag{2.4}$$

for all $i = 1, \ldots, n$*, is a utilitarian allocation regarding the approximate utilities* $\phi_i f(x_i/\phi_i)$*.*

Proof. See Appendix 2.A.1. □

Theorem 2.4.1 has an interesting interpretation: The overall approximate utility is maximized by focusing on a fair balance of the resources among the players. In other words, assigning each player the same average value assures that the sum of the approximate utilities is maximized, independent of the (possibly unknown) function f. This motivates the 'Sharing-Sweet-and-Sour' model applied at NS, which focuses on a balanced distribution of resources among different groups. Note that this solution can also be considered fair, as each employee is assigned exactly the same average for each attribute. In particular, it is readily seen that, under Assumptions 2.4.2 and 2.4.3, the solution specified by (2.4) coincides with the max-min fairness allocation with regard to the non-weighted approximate utilities $f(x_i/\phi_i)$.

Corollary 2.4.1. *Consider an* ARAP *satisfying Assumptions 2.4.2 and 2.4.3. The vector* $\hat{x} \in X$ *satisfying* (2.4) *is a max-min fairness allocation regarding the non-weighted approximate utilities* $f(x_i/\phi_i)$*.*

Proof. See Appendix 2.A.1. □

Under Assumptions 2.4.2 and 2.4.3, Theorem 2.4.1 and Corollary 2.4.1 advocate to allocate the resources according to (2.4). The effect of this allocation can be analyzed by considering a normal RAP with the exact utilities, and the 'Sharing-Sweet-and-Sour' fairness scheme $S^{\mathrm{S\&S}}$, where each player is assigned the utility derived from the resource allocation specified by (2.4).

Definition 2.4.1. *Consider a* RAP *satisfying Assumption 2.4.2. The fairness scheme* $S^{\mathrm{S\&S}}$ *is given by:*

$$S^{\mathrm{S\&S}} : U \mapsto U\left(\hat{x}\right). \tag{2.5}$$

with $\hat{x} \in X$ *as defined by* (2.4).

The fairness scheme $S^{\mathrm{S\&S}}$ can be seen as special case of the perceived fairness measure introduced in Section 2.2, namely when a spread of zero can be achieved over all roster groups. We now derive a tight upper bound on the price of fairness for the scheme $S^{\mathrm{S\&S}}$.

Theorem 2.4.2. *Let* U *be the utility set for a given* RAP *satisfying Assumptions 2.4.1 and 2.4.2, and let* $S^{\mathrm{S\&S}}$ *be the fairness scheme defined by* (2.5)*. It holds that*

$$\mathrm{POF}\left(U, S^{\mathrm{S\&S}}\right) \leq 1 - \frac{\phi^{\star}}{\sum_{i=1}^{n} \phi_i}, \tag{2.6}$$

with $\phi^{\star} = \min_{i \in \{1,\ldots,n\}} \phi_i$*. Furthermore, this bound is tight for all* $\phi \in \mathbb{N}_{+}^{n}$*.*

Proof. See Appendix 2.A.2. □

Interestingly, the derived bound on the POF is independent of the utility functions. Equation (2.6) can be equivalently expressed in a more intuitive way by means of the *standardized* minimum player size ρ, defined as the minimum player size divided by the mean player size, i.e.,

$$\mathrm{POF}\left(U, S^{\mathrm{S\&S}}\right) \leq 1 - \frac{\rho}{n}, \tag{2.7}$$

where

$$\rho = \frac{n\phi^{\star}}{\sum_{i=1}^{n} \phi_i}. \tag{2.8}$$

Note that, by definition, ρ is between zero and one, and equals one whenever all players have equal size. Based on (2.7), it is readily seen that the upper bound increases due to both a large number of players, and a small standardized minimum player size (i.e., the smallest size is much smaller than the average size). Figure 2.5 shows the upper bound on the price of fairness as a function of the standardized minimum player size ρ, for various numbers of players n.

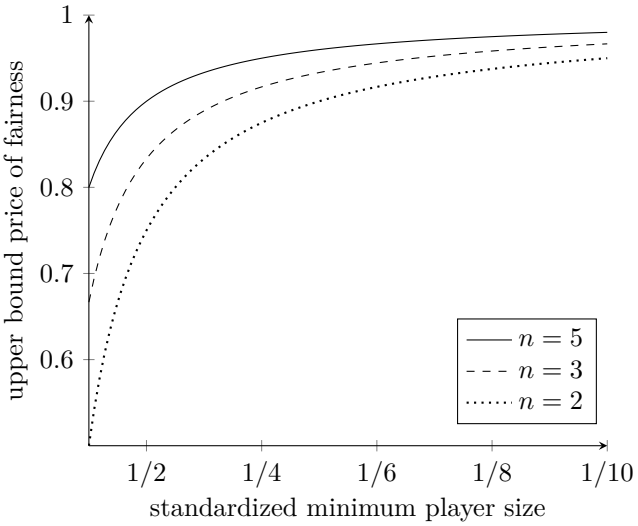


Figure 2.5: Upper bound on the price of fairness as a function of the standardized minimum player size. Each line represents a different number of players.

Theorem 2.4.2 suggests how employees should be divided over roster groups. If we consider an instance with 50 employees, for example, then grouping them in two groups of 25 will give a worst case attractiveness loss of 50%, whereas grouping them in a group of 10 and a group of 40 or in five groups of 10 will both result in a worst case loss of 80%. Hence, in order to safeguard against a large loss of attractiveness, it is recommended to limit the number of roster groups, and to assure that each group is of roughly the same size, whenever this is practically possible.

2.5 Mathematical Model

In the remainder of this chapter, we analyze the trade-off between fairness and attractiveness empirically, by developing an exact solution method for the FCRP. In

order to do so, we first give a mathematical formulation. In Section 2.5.1, we introduce the concept of roster sequences, and in Section 2.5.2 we discuss roster constraints in detail. We introduce the necessary notation in Section 2.5.3, and we conclude by giving a mathematical formulation for the FCRP in Section 2.5.4.

2.5.1 Roster Sequences

We develop a mathematical model based on the *rows* of each roster. That is, we simultaneously assign a number of duties to a row, instead of assigning a duty to each cell separately. Constraints regarding the rows occur naturally, as each row represents a week of work (consider, for example, a maximum workload over a week). Many roster constraints at NS, for example, consider the rows of the roster (we will discuss these in detail in Section 2.7). Hence, modeling the rostering problem based on the rows often leads to a strong formulation, as it implies that many of the roster constraints can be modeled implicitly. Initial experiments showed that modeling based on the rows of the roster improved the performance of the algorithm substantially compared to modeling based on the cells of the roster. In Chapter 4 we give a detailed analysis of different modeling approaches for crew rostering.

The key concept of our modeling approach is the use of *roster sequences*. Modeling the problem based on the rows of a roster implies we need to simultaneously assign multiple duties to all cells in a row. We call such an assignment of duties a roster sequence (or simply a *sequence*, if there is no ambiguity). Note that the roster sequences should always satisfy the basic schedule.

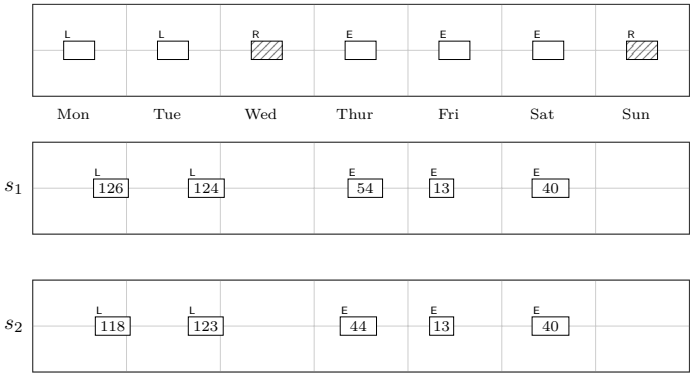


Figure 2.6: Example Sequences.

Figure 2.6 shows two examples of possible roster sequences. The first row of the basic schedule shown in Figure 2.2 is depicted, together with two possible roster sequences s_1 and s_2 for this row. The first roster sequence is obtained from the assigned duties in Figure 2.2, while the second roster sequence is obtained by swapping some duties currently assigned to the first row with those assigned to the third and fourth row. Note that both sequences satisfy the given basic schedule, and that the sequences only specify the assigned duties (i.e., the rest days are not specified in the sequences, as these are input to the problem).

2.5.2 Roster Constraints

Roster constraints impose restrictions on the assignment of duties in the roster, and are used to quantify the attractiveness of the roster. We consider both *hard* roster constraints (i.e., constraints that must always be satisfied) and *soft* roster constraints (i.e., constraints that may be violated against a certain penalty). Each roster constraint is fully specified by a coefficient for each possible assignment of a duty to a cell in the basic schedule, a *threshold value*, and a *violation interval*. It is assumed that each violation interval is a closed interval on the real line. This assumption is important when developing the Branch-Price-and-Cut approach, as will be noted in Section 2.6.2.

A roster constraint restricts the possible assignment of duties by enforcing that *if* the sum of coefficients of assigned duties exceeds the threshold value (i.e., the constraint is violated), *then* the difference between the sum and the threshold lies within the violation interval. Note that this general form allows both for hard and soft constraints. Hard constraints can be modeled by setting the violation interval equal to $\{0\}$ (i.e., no violation is allowed), whereas soft constraints can be modeled by picking a suitable violation interval, such as the interval $[0, 1]$ (i.e., a violation is allowed, but of at most one unit). Each soft constraint has a specified penalty, which is used to penalize deviations above the threshold value.

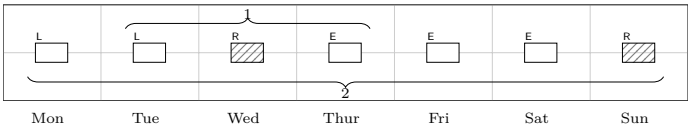


Figure 2.7: Example Roster Constraints.

A wide variety of practical constraints can be expressed in this general form (for a thorough discussion we refer to Hartog et al. (2009)). To illustrate this, consider the row depicted in Figure 2.7, which corresponds to the first row shown in Figure 2.2. Two different roster constraints are indicated.

- The first constraint considers the rest time between the duties scheduled on Tuesday and Thursday. A certain minimal rest time is required between these two days. This is done by introducing a roster constraint for *each* possible duty assigned to Tuesday in this week. Consider d_{124}, a possible duty for Tuesday. The assignment of d_{124} to Tuesday in this row, and all assignments of duties to Thursday that would violate the rest time with respect to d_{124} are given coefficient 1. All other assignments are given coefficient 0. By setting the threshold value equal to 1 and the violation interval equal to $\{0\}$ we model the rest time as a hard constraint. An alternative choice would be the violation interval $[0, 1]$, thereby modeling the rest time as a soft constraint.
- The second constraint considers the average duty length measured over the row. We penalize those assignments of duties where the average duty length in the row exceeds the average duty length measured over all duties. This can be done as follows. The coefficient for each assignment to a day in this row equals the length of the assigned duty, divided by 5. All other assignments are given coefficient 0. The scaling factor 5 follows from the number of duties that need to be assigned to this row (note that this number is known a priori, as the basic schedules are considered input). The threshold value is the average duty length measured over all duties, and the violation interval is $[0, \infty)$.

Roster constraints concerning solely a single row of the basic schedule are modeled implicitly. That is, these roster constraints are incorporated in the penalties associated with the sequences, instead of explicitly modeled as a constraint in the model. Note that this holds for both roster constraints depicted in Figure 2.7.

2.5.3 Notation

We are now ready to introduce the notation for the mathematical formulation. Let D denote the set of duties, and let R denote the set of basic schedules. A basic schedule r is defined by a set of cells T_r, where for each cell $t \in T_r$ it is known which duties can be assigned. An assignment of a duty d to a cell t in a basic schedule will be denoted by the pair (t, d). We define n_r as the total number of duties to be assigned

to basic schedule r.

The set K_r denotes the set of rows for basic schedule $r \in R$, and the set K denotes the set of all rows. We define the set S_k as the set of all roster sequences for row $k \in K$. Each roster sequence can be seen as a sequence of assignments (t, d). We define $S_k^d \subseteq S_k$ as the set of all roster sequences for row k that contain duty d (i..e, the duty d appears in one of the assignments describing the roster sequence). Finally, we define c_s^k as the penalty associated with sequence $s \in S_k$.

The set of duty attributes, used to define the fairness of the constructed rosters, is denoted by A. The non-negative parameter g_{ad} denotes the value of attribute a for duty d. Each attribute has a lower bound l_a and an upper bound u_a, which are considered input. Furthermore, each attribute has an associated weight w_a, representing the relative importance of the different duty attributes when calculating the fairness level. Finally, we introduce a budget parameter ζ for the fairness level, used to compute the trade-off between fairness and attractiveness.

The set of roster constraints, used to define the attractiveness and feasibility of the roster, is denoted by P. The coefficient for the assignment (t, d) for roster constraint p is denoted by f_{td}^p. The threshold value for p is denoted by b_p, and the violation interval for p is denoted by Δ_p. The penalty corresponding to roster constraint p is denoted by c_p. Let $P_K \subseteq P$ denote the set of roster constraints fully contained in one of the rows $k \in K$, and let P_k denote those fully contained in row k. The constraints in P_K are exactly those that are modeled implicitly using the sequence penalties. Hence, the penalty c_s^k associated with sequence $s \in S_k$ is the sum of all violations in the sequence s, restricted to the roster constraints P_k. Note that the roster constraints in $P \setminus P_K$ need to be modeled explicitly.

2.5.4 Mathematical Formulation

We now formalize the FCRP. We introduce the following decision variables.

- x_s^k for all $k \in K$ and $s \in S_k$. The binary variable x_s^k takes value 1 if sequence s is assigned to row k, and value 0 otherwise.
- δ_p for all $p \in P \setminus P_K$. The variable δ_p models the violation of the roster constraint p. The variable is restricted to the violation interval Δ_p.
- z_a for all $a \in A$. The variable z_a expresses the maximum average value of duty attribute a among all roster groups. This variable is used to determine the

perceived fairness of the solution.

- v_a for all $a \in A$. Similar to z_a, the variable v_a models the minimum average value among all roster groups with respect to attribute a.

The FCRP can be expressed as a mixed integer linear program as follows.

$$\min \quad \sum_{k \in K} \sum_{s \in S_k} c_s^k x_s^k + \sum_{p \in P \setminus P_K} c_p \delta_p \tag{2.9}$$

$$\text{s.t.} \quad \sum_{a \in A} w_a \left(z_a - v_a \right) \leq \zeta \tag{2.10}$$

$$\sum_{s \in S_k} x_s^k = 1 \qquad \forall k \in K \tag{2.11}$$

$$\sum_{k \in K} \sum_{s \in S_k^d} x_s^k = 1 \qquad \forall d \in D \tag{2.12}$$

$$\sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f_{td}^p x_s^k \leq b_p + \delta_p \qquad \forall p \in P \setminus P_K \tag{2.13}$$

$$\sum_{k \in K_r} \sum_{s \in S_k} \sum_{(t,d) \in s} g_{ad} x_s^k \leq n_r z_a \qquad \forall a \in A, r \in R \tag{2.14}$$

$$\sum_{k \in K_r} \sum_{s \in S_k} \sum_{(t,d) \in s} g_{ad} x_s^k \geq n_r v_a \qquad \forall a \in A, r \in R \tag{2.15}$$

$$z_a \leq u_a \qquad \forall a \in A \tag{2.16}$$

$$v_a \geq l_a \qquad \forall a \in A \tag{2.17}$$

$$x_s^k \in \mathbb{B} \qquad \forall k \in K, s \in S_k \tag{2.18}$$

$$\delta_p \in \Delta_p \qquad \forall p \in P \setminus P_K \tag{2.19}$$

$$v_a, z_a \in \mathbb{R}_+ \qquad \forall a \in A. \tag{2.20}$$

The objective (2.9) expresses that we minimize the sum of the sequence penalties, together with the cost of all explicitly modeled roster constraints. The fairness level is modeled as a budget constraint using (2.10). As discussed in Section 2.2, the perceived fairness is calculated as a weighted sum of the spread (i.e., the difference between the maximum and minimum average values) of the duty attributes with respect to the roster groups. By varying the budget parameter ζ, the entire trade-off curve between fairness and attractiveness can be computed (see, e.g., Ehrgott (2000) for an overview of bi-objective optimization techniques). Constraints (2.11) and (2.12) assure that the duties are assigned correctly to the basic schedules. That is, each row is assigned exactly one roster sequence, and each duty is assigned exactly

once. Constraints (2.13) consider the roster constraints. Note that (2.13) assures that δ_p takes the value of the constraint violation. Constraints (2.14) and (2.15) assure that the variables v_a and z_a are set to the minimum and maximum value, respectively, while (2.16) and (2.17) enforce the lower and upper bounds on the attribute values. Finally, Constraints (2.18)-(2.20) express the domains of the decision variables.

2.6 Solution Approach

We develop an exact solution method for the FCRP based on Branch-Price-and-Cut. In Branch-Price-and-Cut algorithms, the solution space is searched using a Branch-and-Bound procedure, in which the linear relaxation is solved using column generation in each node of the Branch-and-Bound tree. Furthermore, valid inequalities (i.e., cuts) are added to strengthen the linear relaxation. The concept of column generation has been applied succesfully to a large variety of problems (see e.g., Desaulniers et al. (1997), Löbel (1998), Potthoff et al. (2010)). For detailed surveys on column generation we refer to Barnhart et al. (1998), Lübbecke and Desrosiers (2005), Desaulniers et al. (2006) and Lübbecke (2011). An extensive overview of Branch-Price-and-Cut algorithms is given in Desrosiers and Lübbecke (2011).

The lay-out of this section is as follows. In Sections 2.6.1 and 2.6.2, we discuss the master problem and the resulting pricing problems, respectively. In Section 2.6.3, we discuss the branching strategy. We conclude in Section 2.6.4 with a discussion on how we obtain all efficient solutions of the FCRP.

2.6.1 Master Problem

In Branch-Price-and-Cut algorithms the problem is decomposed in a master problem and a set of pricing problems. The Master Problem is obtained from (2.9)–(2.20) by relaxing the integrality condition for the x_s^k variables. That is, Constraints (2.18) are replaced with $x_s^k \geq 0$, for all $k \in K$ and $s \in S_k$. Note that (2.11) assures that $x_s^k \leq 1$. Because the number of sequences can be very large, we use column generation to solve the Master Problem. This means that only a subset of the sequences is considered instead of the entire set of sequences. As this might lead to suboptimal solutions, profitable sequences are added based on their reduced cost. Deciding whether such sequences exist is done in the pricing problems, where sequences with

negative reduced cost are generated. The linear relaxation of (2.9)–(2.20) is tightened by adding valid inequalities (see Appendix 2.B).

2.6.2 Pricing Problem

The pricing problems can be modeled as resource constrained shortest path problems (RCSPP) with surplus variables on dedicated graphs (see e.g., Irnich and Desaulniers (2005) for a detailed survey on this topic). For each row $k \in K$, we construct a directed graph in which each vertex represents an assignment of a duty to a cell in the basic schedule. An arc in the graph indicates a possible follow-up duty.

An example of such a graph is shown in Figure 2.8. Here we show the pricing graph for the row depicted in Figure 2.7. Each vertex specifies a combination of a cell and a duty. The duties are obtained from the roster in Figure 2.2. The cell on Monday of type L, for example, can be assigned either duty d_{126} or d_{118}. Clearly, each $s-t$ path in such a graph corresponds to a roster sequence.

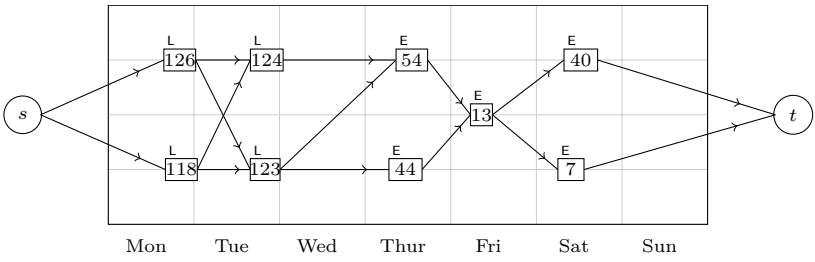


Figure 2.8: Example Pricing Graph.

The reduced cost of a sequence can be modeled using the arc set and the resource constraints. The dual multipliers are readily incorporated in the arc costs (see Appendix 2.C). The difficulty lies in expressing the primal cost, that is, in expressing the cost related to the different roster constraints that are implicitly modeled. This is done using the resource constraints. Each roster constraint p relates to a resource with consumption f^p_{td} at node (t, d) and consumption limit b_p. The violation interval Δ_p is modeled using the domain of the surplus variables. Here we use the fact that each Δ_p represents a closed interval on the real line. We note that binary constraints (i.e., constraints involving two subsequent duties) are a special case, as they can be directly incorporated into the arcs. An example of this is the 'missing' arc between the duties d_{124} and d_{44} on Tuesday and Thursday, respectively. The time between

duties d_{124} and d_{44} is insufficient for a proper rest day on Wednesday, hence no sequence can have these two duties scheduled sequentially.

We solve the RCSPP using an enumerative depth first search approach, where we use *completion bounds* (i.e., the remaining shortest path costs from the current node to the sink) to prune provably suboptimal paths. Similar depth first search approaches have been successfully applied to the RCSPP (see e.g., Beasley and Christofides (1989), Grötschel et al. (2003), Dumitrescu and Boland (2003)). We note that this approach closely resembles the well-known A* algorithm, introduced by Hart et al. (1968).

2.6.3 Obtaining Integer Solutions

To obtain integer solutions we apply a bi-level branching strategy. In the first branching rule, we consider the assignment of duties to the basic schedules, i.e., pairs (r, d). We branch if a duty is only partially assigned to a basic schedule. That is, we construct a branch where the duty is assigned only to this schedule, and a branch where the duty is excluded from this schedule. Note that partially assigning a duty to a schedule is equivalent to assigning a duty to multiple schedules (a direct consequence of (2.11) and (2.12)). If multiple branching possibilities are present, we branch on the most fractional value.

The assignment of duties to the basic schedules is determined whenever no branching opportunities of this form are available. It is possible, however, that the solution is still fractional, since a duty could be assigned to multiple cells *within* one basic schedule. The second branching rule therefore considers the assignment of duties to cells, i.e., pairs (t, d). That is, we branch if a duty is only partially assigned to a cell in the basic schedule. Also for the second branching rule we branch on the most fractional value, whenever multiple branching possibilities are present. Note that no branching opportunities exist only if each duty is assigned to a unique cell, and hence an integer solution is found.

The branching rules are always executed in a sequential fashion. This means that the second branching rule is applied *only* if there are no branching possibilities for the first rule. Since the first branching rule divides the solution space more evenly than the second, this sequential strategy leads to a more balanced search tree.

2.6.4 Finding Pareto-optimal Solutions

We want to identify all Pareto-optimal solutions for the FCRP. A solution is Pareto-optimal if there exists no solution that is better in all criteria. In the FCRP this means that there can be no solution with higher attractiveness *and* higher fairness level.

All Pareto-optimal solutions to a bi-objective optimization problem can be found by imposing a bound on one objective, while optimizing the other. In the FCRP, this bound is the desired fairness level ζ in (2.10). By varying ζ iteratively, we obtain all Pareto-optimal solutions (up to a certain precision). That is, we solve the problem for a sequence of strictly decreasing fairness levels, until a feasible solution can no longer be found.

2.7 Case Study at NS

To show the practical benefits of the FCRP we apply our solution method to a set of rostering instances of NS. The instances are based on roster groups of guards at depot Utrecht, one of the major crew depots in the Netherlands. In Section 2.7.1, we discuss our experimental set-up, and all attributes and roster constraints that are taken into account. In Section 2.7.2, we present the result of our experiments. We conclude in Section 2.7.3 with the practical insights obtained from the case study.

2.7.1 Rostering at Base Utrecht

Each rostering instance specifies a set of basic schedules and a set of duties. The instances in our experiments are based on a set of roster groups and their corresponding basic schedules. The duties are obtained from the original rosters, to assure compatibility with the basic schedules. We consider three different types of duties: early, late and night. These are denoted by E, L and N, respectively. Furthermore, a day in the roster can be an official rest day (denoted by R), or an unofficial rest day. For the latter, no constraints have to be taken into account. The basic schedules specify one of these types for each day in the roster.

The roster groups are divided into three categories, based on their basic schedules. The first category consists of groups performing solely early duties, the second cat-

egory of groups performing solely late and night duties, and the third category of groups performing all three types of duties. The first two instances consist of two large groups of the first two categories, and a relatively small group of the third category. The other two instances consist of one group of each category, each representing roughly the same number of employees. The details are shown in Table 2.1. The group size refers to the number of employees in the group (i.e., the number of rows that need to be rostered).

Table 2.1: Rostering Instances Base Utrecht.

Instance	Groups	Group sizes	Duties	E/L/N
U1	3	14/12/4	113	55/29/29
U2	3	12/12/4	115	58/33/24
U3	3	6/6/6	71	38/22/11
U4	3	8/6/6	69	37/17/15

NS considers a large variety of duty attributes and roster constraints, leading to a complex rostering problem. A total of five duty attributes have to be taken into account.

- *Duty length.* For each of the duties, the length is defined as the difference between the start and end time of a duty.
- *Percentage of type-A work.* Trips with type-A work are desirable, and hence this work needs to be fairly distributed. Generally, type-A work consists of work on Intercity trains, which are trains that only stop at major stations. This type of work mainly contains long trips.
- *Percentage of aggression work.* Certain trips have a higher chance of passenger aggression (due to, e.g., passengers not having a ticket). Trips where such situations frequently occur are undesirable and therefore need to be fairly distributed.
- *Percentage of work on double decker trains.* Work on double decker trains is considered undesirable, as it is physically more demanding (due to the high amount of stairs an employee has to climb during work on such a train). We therefore want to distribute this type of work equally.
- *RWD values.* The Repetition Within Duty (RWD) values are defined as the total number of routes divided by the total number of *distinct* routes in the duty. From an employee point of view, variation is desirable (reflected by a

low RWD value), and hence we want to balance the RWD values among the groups.

Table 2.2: Specification Duty Attributes.

Attribute	Av. U1	Av. U2	Av. U3	Av. U4	LB	UB	Weight
Duty length (hours)	7.97	7.97	7.89	7.98	-	8	30
Type-A	40.28	44.91	42.07	42.10	35	-	1
Aggression	14.08	15.17	12.02	9.40	-	18	1
Double Decker	34.36	34.79	35.24	38.54	-	40	1
RWD	2.06	2.09	2.05	2.12	-	2.5	25

Each attribute has a pre-specified lower or upper bound which should be respected for each group (see Table 2.2). Furthermore, each attribute is given a weight to model the relative contribution to the fairness measure. For example, a spread of 5% in aggression work and a spread of 10 minutes in duty length are penalized equally. Table 2.2 also shows the average value for each of the instances, displaying the tightness of the imposed bounds.

The roster constraints considered at NS can be roughly classified into two categories: those concerning pairs of cells and those concerning entire rows.

- *Rest Constraints.* After completing a duty it is required that an employee has a certain minimum time to rest. A rest time of 14 hours is required after a night duty. For the other types of duties the required rest time is 12 hours. Rest times shorter than 16 hours are undesirable, and are therefore penalized with a penalty of 30. This penalty is independent of the shortage in rest time.
- *Rest Day Constraints.* If two duties have a rest day in between, there should be at least 30 hours between the end of the first duty and the start of the second. Furthermore, for each additional rest day that is between the duties, another 24 hours are required.
- *Workload Constraints.* We require that the total workload in one row should not exceed 45 hours. Here, workload is measured as the difference between start and end time (i.e., including the meal break). The workload is always measured from Monday to Sunday (i.e., a 'regular week').
- *Variation Constraints.* It is considered desirable if e.g., aggression work is distributed evenly over the rows. Therefore, variation constraints penalize positive deviations from the average (measured over all duties) for each of the duty at-

tributes. Furthermore, we take duties with a duration longer than 9 hours and a set of (un)desirable routes into account. This gives a total of 10 variation constraints per row. Similar to the workload constraints, the variation is always measured from Monday to Sunday.

By assigning roster sequences to the rows, we can model all workload and variation constraints implicitly using the sequence penalties. Furthermore, most rest and rest day constraints can also be modeled implicitly. The set of explicitly modeled roster constraints consists of only those rest and rest day constraints spanning two rows (e.g., constraints modeling the rest time from Sunday to Monday).

2.7.2 Computational Results

We solved each of the instances using the Branch-Price-and-Cut approach. We first discuss the computation times for the different instances. Thereafter, we present the Pareto-optimal curves, and we conclude with a detailed analysis of the found solutions.

Computation Times

All experiments were done on a computer with a 1.6 GHz Intel Core i5 processor. We used the LP solver embedded in CPLEX 12.7.1 to solve the master problems. Recall from Section 2.6.4 that the Pareto-optimal curve is obtained by solving (2.9) – (2.20) for different values of ζ. Hence, the overall computation time depends both on the number of Pareto-optimal points found, and on the computation time per point. Table 2.3 shows for each instance the number of Pareto-optimal points found, and the average computation time per point (in minutes). In our experiments, we used a step-size of 0.1 for iteratively updating the fairness level.

Table 2.3: Computation times per instance.

Instance	Number of Points	Average Time (min)
U1	23	14.6
U2	12	21.2
U3	17	7.6
U4	21	36.6

As Table 2.3 shows, each of the Pareto-optimal curves could be determined in reasonable time. The Pareto-optimal curve for instance U4 took most time to compute (about 13 hours in total), due to the relatively large number of points and high average computation time. The other three Pareto-optimal curves were found in substantially less time (at most 6 hours in total). From a practical point of view, these computation times can be considered reasonable whenever optimality is required. Note, however, that when optimality is not a prerequisite, heuristic solution approaches might be better suited, as we will further discuss in Chapter 3.

Pareto-optimal Curves

The Pareto-optimal curves for the four instances are depicted in Figure 2.9. From hereon we refer to the objective value (i.e., the measure of perceived attractiveness) as the *attractiveness penalty* and to ζ as the *fairness penalty*.

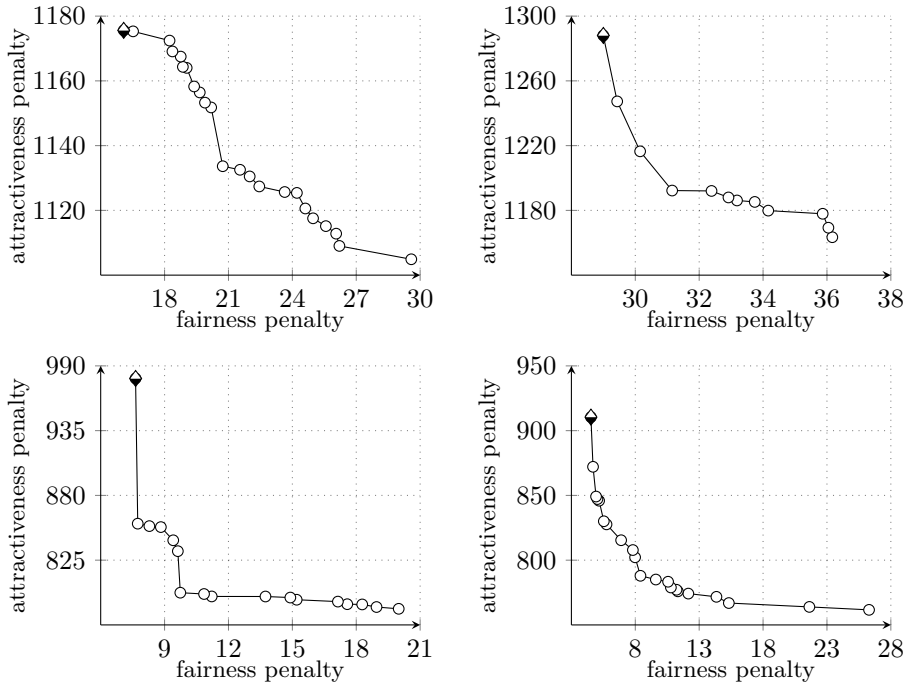


Figure 2.9: Pareto curve for instances U1 (top left), U2 (top right), U3 (bottom left) and U4 (bottom right).

Figure 2.9 shows that many Pareto-optimal solutions are lost in a sequential ap-

proach. Note that a sequential approach, where perceived fairness is optimized first, results (at best) in the solution represented by the leftmost point in the curve (those highlighted in Figure 2.9). The set of Pareto-optimal points, however, covers a broad spectrum of solutions, each of which might be optimal depending on the given preferences. The curve of instance U1, for example, ranges from 1104 to 1175 (a relative spread of about 6%) with respect to the attractiveness penalty and from 16 to 30 (a relative spread of about 85%) with respect to the fairness penalty. Note that each of the instances exhibits a clear price of fairness, in line with the results of Section 2.4. The differences in group size, however, does not seem to enlarge this loss of attractiveness.

By computing a large set of 'best' solutions, the decision maker gets a comprehensive image of the problem. In particular, it gives a better indication of the consequences of managerial decisions, such as imposed fairness bounds. A striking example of such a consequence is given by the two leftmost Pareto-optimal points in the curve of instance U3, where a small decrease of the fairness penalty leads to a disproportionally large increase of the attractiveness penalty. As mentioned above, this leftmost point would be found using a sequential approach. It is clear that this solution (and hence the sequential approach) is optimal only when fairness is extremely highly valued.

Trade-Off Analysis

The Pareto curves in Figure 2.9 exhibit a clear 'tail-off' effect. That is, decreasing the fairness penalty close to the minimum leads to a disproportionally large increase in the attractiveness penalty, and vice versa. To highlight this behavior, we standardize each solution based on the attained minimum and maximum penalties. For example, if some solution attains a fairness penalty of ζ, we compute the *relative decrease* of ζ. That is, we compute the quantity

$$\frac{\zeta^{\max} - \zeta}{\zeta^{\max} - \zeta^{\min}} \in [0, 1], \tag{2.21}$$

where $\zeta^{\min}$ and $\zeta^{\max}$ denote the lowest and highest observed fairness penalties, respectively. Note that $\zeta^{\min}$ and $\zeta^{\max}$ correspond to the left- and rightmost points in the curve. We transform the observed attractiveness penalties similarly. The result is shown in Figure 2.10. The horizontal axis shows the relative decrease of the fairness penalty and the vertical axis shows the relative increase of the attractiveness penalty.

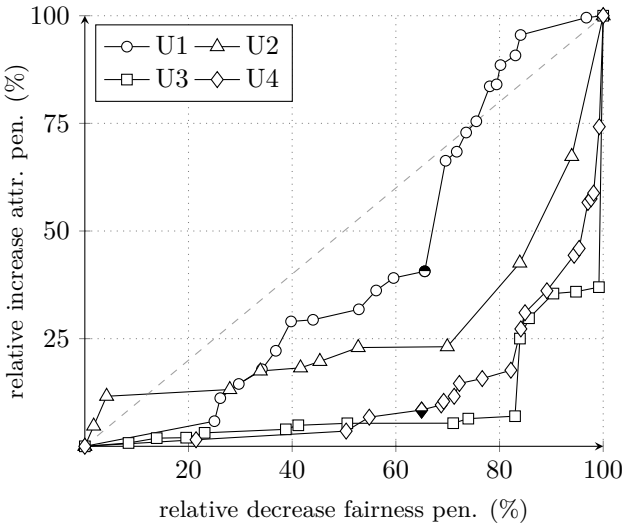


Figure 2.10: Relative Trade-Off between Fairness and Attractiveness.

The transformed curves clearly show an increasing rate (the gray line indicates an identical growth rate). We see, for example, that 65% of the fairness gap, i.e., the difference between the minimum and maximum, can be closed against a relative small increase of the attractiveness penalty. This 65% decrease leads to a relative increase of the attractiveness penalty of at most 41% (for instance U1) and sometimes even of as little as 8% (for instance U4). These points are highlighted in Figure 2.10. Similarly, we see that for almost all instances the first 25% can be closed at almost no additional increase in attractiveness penalty. On the other hand, Figure 2.10 also shows that the final improvements of the fairness penalty often come at a disproportionally large increase of the attractiveness penalty.

Attractiveness Per Group

Using an integrated approach allows to analyze the change in attractiveness per group for the different levels of fairness. This implies we can analyze the trade-off between fairness and attractiveness on a *local* level, i.e., for each group separately, instead of only on a *global* level, i.e., seeing the three groups as a whole. We compute for each group the *average* incurred attractiveness penalty per row. The groups are classified based on the types of duties they contain (recall the classification discussed in Section 2.7.1). The result is shown in Figure 2.11.

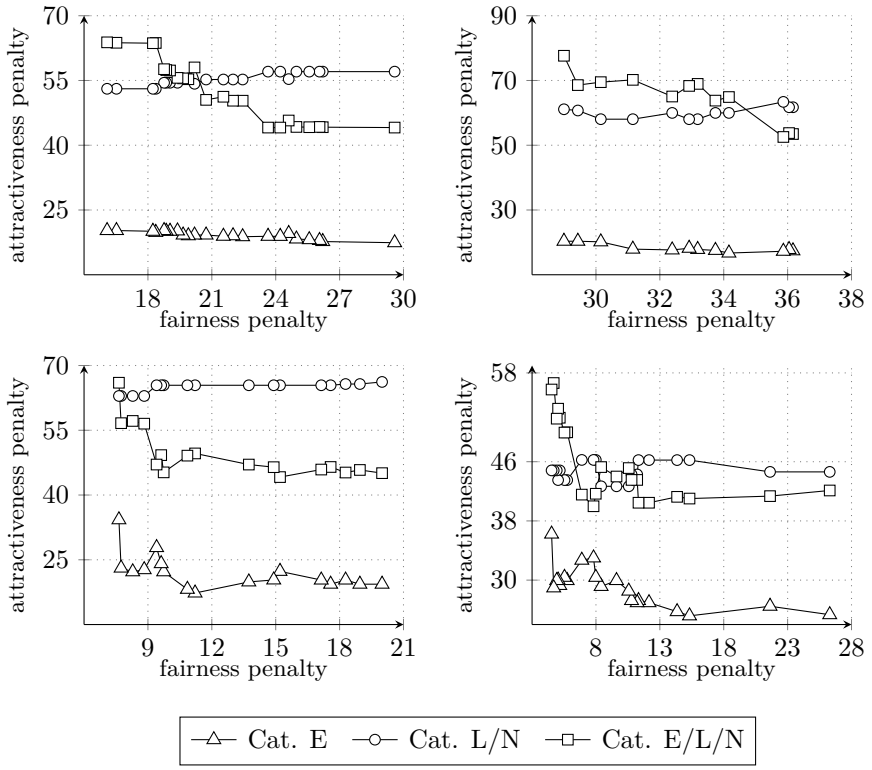


Figure 2.11: Average attractiveness penalty per group for instances U1 (top left), U2 (top right), U3 (bottom left) and U4 (bottom right).

Figure 2.11 shows that solutions with a low fairness penalty might be perceived as 'skewed' with respect to the attractiveness penalty. When the attractiveness penalty is averaged and decomposed per group, we see that the different types of groups have, in general, a different attractiveness. This can be attributed to the different duty types defining each group (e.g., rosters alternating between late and night duties often have more rest time violations). When considering solutions with a low fairness penalty, we see a clear increase in the average attractiveness penalty for groups of the third category, and, for the equally sized instances, also for the groups of the first category. On the other hand, we see that for the groups of the second category, the average attractiveness penalty hardly increases, or even decreases. Especially for groups of the third category, those most flexible with respect to the duty allocation, we observe a disproportionally large increase.

2.7.3 Managerial Insights

Our experiments indicate the necessity of a simultaneous optimization of fairness and attractiveness. The sequential approach is inadequate for analyzing the trade-off between fairness and attractiveness. By using an integrated approach, we could analyze this trade-off in detail. The analysis in Section 2.7.2 led to the following important (empirical) insights.

Firstly, decision makers should be careful not to 'over-optimize' fairness. The trade-off analysis showed that tight fairness levels lead to a rapid decrease of the attractiveness of the rosters. In our experiments, we observed cases where an almost negligible increase in fairness led to a major decrease in attractiveness. Hence, fully prioritizing fairness over attractiveness should be considered only when fairness is extremely highly valued. This is line with the analytical results derived in Section 2.4.

Secondly, it is questionable whether a set of rosters with high perceived fairness will be perceived as desirable by all employees. Our analysis showed that the decrease in attractiveness is unevenly distributed over the different roster groups. That is, for some groups the attractiveness of the roster deteriorates rapidly, while for other groups the attractiveness of the roster might actually improve. This skewed distribution of attractiveness is an implicit consequence of minimizing perceived fairness. The decrease of attractiveness occurs most rapidly for those roster groups that are willing to accept an irregular roster (i.e., those groups flexible with respect to the work that can be assigned). A rapid decrease in attractiveness for these roster groups is highly undesirable from a practical point of view, as the employees in these groups already 'accept' a loss of attractiveness by working an irregular roster.

Our experiments show that the current practice should be revised. In particular, the trade-off between fairness and attractiveness should be incorporated in the construction of the rosters. Furthermore, the current metric for perceived fairness, as discussed in Section 2.2, should be reconsidered. For example, the skewed distribution of attractiveness could be incorporated in a new fairness metric, or a different fairness scheme (e.g., max-min fairness) could be used. Note, however, that such adjustments heavily depend on whether the 'Sharing-Sweet-and-Sour' rules capture the true essence of fairness or not. Or, to put it differently, whether a skewed distribution of attractiveness is considered unfair by the employees. Furthermore, adjustments in the fairness metric might lead to less transparent allocation rules, compared to the 'Sharing-Sweet-and-Sour' rules, which would be more difficult to communicate

to the employees. In any case, adjustments in the fairness metric would have a great impact in practice, as fairness plays a major role in the negotiations with the labor unions.

2.8 Conclusion

In this chapter, we introduced the Fairness-oriented Crew Rostering Problem (FCRP). In the FCRP, fair and attractive cyclic rosters have to be constructed. The goal of the FCRP is to make an explicit trade-off between fairness and attractiveness. That is, to present a set of solutions, where each solution is optimal for a different trade-off between fairness and attractiveness.

We analyzed a class of resource allocation problems, in which the resource allocation is based on approximate utility functions. We considered a fairness scheme for this class, based on the 'Sharing-Sweet-and-Sour' rules, and we derived a tight upper bound on the price of fairness for this scheme, which showed that the price of fairness is highest for instances with a large number of groups, and instances with large differences in group sizes.

Furthermore, we developed an exact Branch-Price-and-Cut solution method, based on a novel mathematical formulation. By partitioning the days of a basic schedule in weeks, and assigning sequences of duties to these weeks, we obtain a flexible model in which different roster constraints can be easily incorporated in the sequence penalties.

We applied our solution approach to practical instances of NS, where we showed that our integrated approach leads to a diverse set of solutions. The analysis of these solutions led to two important insights. Firstly, decision makers should be careful not to 'over-optimize' fairness. We observed that by loosening the fairness requirements slightly, the attractiveness could be greatly improved, thereby showing the possible suboptimality of a sequential approach. Secondly, we found that the decrease in attractiveness caused by a tight fairness level is unevenly distributed over the different roster groups.

Based on our findings we recommend to revise the current approach. The trade-off between fairness and attractiveness should be incorporated in the construction of the rosters. Both the theoretical and empirical results show the existence of a loss of attractiveness due to fair allocations. Hence, by explicitly considering the trade-off between fairness and attractiveness, a more comprehensive image of the problem can

be obtained. Furthermore, it is recommended to consider additional properties (e.g., the distribution of attractiveness over the groups) when determining the fairness level of a solution, and to explore different fairness schemes. Whether or not this will lead to an improved allocation, however, depends on the preferences of the employees and the value of transparent and simple allocation rules.

A promising avenue for further research would be to extend our approach to large scale instances (e.g., containing hundreds of duties). Such instances should most likely be solved using a heuristic approach, as the problems will become too difficult to solve to optimality. From a theoretical point of view, exactness is necessary to analyze the trade-off between fairness and attractiveness. From a practical point of view, however, a set of 'good', not necessarily optimal, solutions can already improve the decision process greatly.

Appendix

2.A Proofs Section 2.4

In this section, we give the proofs omitted in Section 2.4. In Section 2.A.1, we give the proofs of Theorem 2.4.1 and Corollary 2.4.1 concerning the utilitarian and max-main fairness allocations for an ARAP satisfying Assumptions 2.4.2 and 2.4.3. In Section 2.4.2, we give the proof of Theorem 2.4.2 regarding the price of fairness of the ''Sharing-Sweet-and-Sour' fairness scheme.

2.A.1 Proof of Theorem 2.4.1 and Corollary 2.4.1

To prove Theorem 2.4.1, we first prove the following lemma.

Lemma 2.A.1. *Consider a* RAP *satisfying Assumption 2.4.2, where the utility function of the i-th player is given by* $f_i(x_i) = f(x_i)$*, for some* $f : \mathbb{R}_+^k \to \mathbb{R}_+$*, which is concave, bounded, and continuous over* X*. The resource allocation* $\hat{x} \in X$ *satisfying*

$$\hat{x}_i = \frac{\gamma}{n},$$

for all $i = 1, \ldots, n$*, maximizes the overall utility.*

Proof. Let $\bar{x} \in X$ be an allocation maximizing utility, i.e., $\bar{x} = \arg\max_{x \in X} \sum_{i=1}^{n} f(x_i)$. The existence of $\bar{x}$ follows from the fact that X is compact (by Assumption 2.4.2), and f continuous. Note that, by Assumption 2.4.2, the resource allocation $\hat{x}_i$ can be written as

$$\hat{x}_i = \frac{\gamma}{n} = \frac{\sum_{j=1}^{n} \bar{x}_j}{n}.$$

It holds that

$$\sum_{i=1}^{n} f(\hat{x}_i) = nf\left(\frac{\gamma}{n}\right) = nf\left(\frac{1}{n}\sum_{i=1}^{n} \bar{x}_i\right) \geq n\sum_{i=1}^{n} \frac{1}{n} f(\bar{x}_i) = \sum_{i=1}^{n} f(\bar{x}_i),$$

where the inequality follows since f is concave. This, however, shows that $\hat{x}$ maximizes the overall utility. The result follows. □

Theorem 2.4.1 can now be proven as follows.

Theorem 2.4.1. *Consider an* ARAP *satisfying Assumptions 2.4.2 and 2.4.3. The vector $\hat{x} \in X$ with*

$$\frac{\hat{x}_i}{\phi_i} = \frac{\gamma}{\sum_{j=1}^{n} \phi_j}, \tag{2.4}$$

for all $i = 1, \ldots, n$, is a utilitarian allocation regarding the approximate utilities $\phi_i f(x_i/\phi_i)$.

Proof. The problem of maximizing the utility can be written as

$$\max \quad \sum_{i=1}^{n} \phi_i f(x_i/\phi_i) \tag{2.22}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i = \gamma \tag{2.23}$$

$$x_i \in \mathbb{R}^k_+ \qquad \forall i = 1, \ldots, n. \tag{2.24}$$

Using the change of variable $y_i = x_i/\phi_i$, this is equivalent to

$$\max \quad \sum_{i=1}^{n} \phi_i f(y_i) \tag{2.25}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \phi_i y_i = \gamma \tag{2.26}$$

$$y_i \in \mathbb{R}^k_+ \qquad \forall i = 1, \ldots, n. \tag{2.27}$$

We now consider a relaxation of the above problem with $m = \sum_{i=1}^{n} \phi_i$ players. Let z_i denote the resource allocation of the i-th player in this new setting. The corresponding problem reads as follows.

$$\max \quad \sum_{i=1}^{m} f(z_i) \tag{2.28}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} z_i = \gamma \tag{2.29}$$

$$z_i \in \mathbb{R}^k_+ \qquad \forall i = 1, \ldots, m. \tag{2.30}$$

Note that the above problem satisfies the assumptions of Lemma 2.A.1. It follows that $\hat{z}_i = \gamma/m = \gamma/\sum_{j=1}^{n} \phi_j$ is an optimal solution for (2.28)–(2.30). Note that the solution $\hat{y}_i = \gamma/\sum_{j=1}^{n} \phi_j$ is feasible for (2.25)–(2.27), and achieves the same objective value. Since (2.28)–(2.30) is a relaxation of (2.25)–(2.27), $\hat{y}_i$ must be optimal for (2.25)–(2.27). The result follows. □

It remains to prove Corollary 2.4.1.

Corollary 2.4.1. *Consider an* ARAP *satisfying Assumptions 2.4.2 and 2.4.3. The vector $\hat{x} \in X$ satisfying* (2.4) *is a max-min fairness allocation regarding the non-weighted approximate utilities $f(x_i/\phi_i)$.*

Proof. Suppose $\hat{x}$ does not correspond to max-min fairness allocation. Since $f(\hat{x}_i/\phi_i) = f(\hat{x}_j/\phi_j)$, for all i, j, this implies that there exists an $\bar{x} \in X$ such that

$$f(\bar{x}_i/\phi_i) \geq f(\hat{x}_i/\phi_i),$$

for all $i = 1, \ldots, n$, and, furthermore, that the inequality is strict for at least one i.

This, however, implies that

$$\sum_{i=1}^{n} \phi_i f(\bar{x}_i/\phi_i) > \sum_{i=1}^{n} \phi_i f(\hat{x}_i/\phi_i),$$

contradicting Theorem 2.4.1. □

2.A.2 Proof of Theorem 2.4.2

Theorem 2.4.2 can be proven as follows.

Theorem 2.4.2. *Let U be the utility set for a given* RAP *satisfying Assumptions 2.4.1 and 2.4.2, and let $S^{\text{S\&S}}$ be the fairness scheme defined by* (2.5)*. It holds that*

$$\text{POF}\left(U, S^{\text{S\&S}}\right) \leq 1 - \frac{\phi^\star}{\sum_{i=1}^{n} \phi_i}, \tag{2.6}$$

with $\phi^\star = \min_{i\in\{1,\dots,n\}} \phi_i$. Furthermore, this bound is tight for all $\phi \in \mathbb{N}_+^n$.

Proof. For notational convenience, define $h(x) = \sum_{i=1}^{n} g_i(x_i)$, and $\Phi = \sum_{i=1}^{n} \phi_i$. Let $\hat{x}$ correspond to an allocation maximizing utility, and let $x^\star$ be defined by $x_i^\star = (\phi_i/\Phi)\gamma$.

We consider the line from $\hat{x}$, passing through $x^\star$. When $\hat{x} = x^\star$ the price of fairness equals zero, hence, we can assume, without loss of generality, that $\hat{x} \neq x^\star$. Let $\bar{x}$ denote the last feasible allocation on this line, i.e., $\bar{x} + \delta(x^\star - \hat{x})$ is not feasible for arbitrary small $\delta > 0$. Consider $\pi : [0,1] \to \mathbb{R}_+$, defined by

$$\pi(\alpha) = h((1-\alpha)\bar{x} + \alpha\hat{x}).$$

Note that $\pi(0) = h(\bar{x})$, and $\pi(1) = h(\hat{x})$. Furthermore, $\pi(\alpha^\star) = x^\star$ for some $\alpha^\star \in (0,1)$. Since $\bar{x} + \delta(\hat{x} - x^\star)$ is not feasible for any $\delta > 0$, it must hold that $\bar{x}_{ij} = 0$ for some i and j. We have

$$\frac{\phi_i \gamma_j}{\Phi} = x_{ij}^\star = (1-\alpha^\star)\bar{x}_{ij} + \alpha^\star \hat{x}_{ij} = \alpha^\star \hat{x}_{ij}.$$

Furthermore, since $\hat{x}_{ij} \leq \gamma_j$, it follows that

$$\alpha^\star \geq \frac{\phi_i}{\Phi} \geq \frac{\phi^\star}{\Phi},$$

with $\phi^\star = \min_{i\in\{1,\dots,n\}} \phi_i$. By the concavity of π, we know that

$$\pi(\alpha^\star) \geq \pi(0) + \alpha^\star(\pi(1) - \pi(0)) \geq \alpha^\star \pi(1) \geq \frac{\phi^\star \pi(1)}{\Phi},$$

where the second inequality follows since $1 - \alpha^\star \geq 0$ and $\pi(0) \geq 0$. It follows that

$$\frac{h(\hat{x}) - h(x^\star)}{h(\hat{x})} = \frac{\pi(1) - \pi(\alpha^\star)}{\pi(1)} \leq \frac{\pi(1) - (\phi^\star/\Phi)\pi(1)}{\pi(1)} = 1 - \frac{\phi^\star}{\Phi}.$$

It remains to show that the bound is tight for all $\phi \in \mathbb{N}_+^n$. For $n = 1$ this is trivial (note that there is only one feasible solution in this case). Consider some $\phi \in \mathbb{N}_+^n$ with $n \geq 2$. Assume, without loss of generality, that $\phi_1 = \phi^\star$, and $k = 1$. Set $g_1(x_1) = x_1$ and $g_i(x_i) = 0$, for $i \geq 2$. Clearly, the maximum achievable utility is γ in this case, which is achieved by assigning the entire resource to the first player. The fairness scheme $S^{\text{S\&S}}$, however, will lead to an overall utility of $(\phi_1/\Phi)\gamma = (\phi^\star/\Phi)\gamma$, which shows that (2.6) is tight. □

2.B Valid Inequalities

The linear relaxation of (2.9)–(2.20) can be tightened by adding valid inequalities. We consider the set of roster constraints $P \setminus P_K$. Recall from Section 2.7 that in our experiments the set $P \setminus P_K$ contains the rest constraints (both the strict version regarding the 12/14 hour rest period and the penalized version regarding the 16 hour rest period) and rest day constraints involving more than one week. We derive a set of valid inequalities to bound the penalty incurred from the rest constraints regarding the 16 hour rest period. For each $Q \subseteq R$ let $P_Q \subseteq P \setminus P_K$ be the set of 16 hour rest constraints related to the set of basic schedules Q. We derive a lower bound for the incurred penalty by minimizing

$$\sum_{p\in P_Q} c_p \delta_p, \tag{2.31}$$

subject to (2.11)–(2.12) and

$$\sum_{k\in K} \sum_{s\in S_k} \sum_{(t,d)\in s} f_{td}^p x_s^k \leq b_p + \delta_p \tag{2.32}$$

for all $p \in P_Q$. That is, we ignore the sequence penalties, and hence the roster constraints P_K, thereby greatly simplifying the problem. Many roster sequences can be aggregated in the above problem, as the rest patterns in P_Q depend solely on the first and last duty scheduled in each sequence.

The above optimization problem can be solved 'efficiently' for practical instances using a commercial solver. In our experiments, for example, the valid inequalities could be obtained in only a fraction of the time needed to obtain the Pareto-optimal curves, and they strengthened the formulation greatly: For instances U1 and U2, none but a few Pareto-optimal points could be found within two hours without using the valid inequalities. This threshold of two hours per point was set to avoid extremely long computation times, and is about a factor 6 larger than the average computation times for U1 and U2 reported in Table 2.3. For instances U3 and U4 the valid inequalities did not lead to a significant improvement in average computation time.

We note that the above procedure leads to $2^{|R|} - 1$ valid inequalities, and hence should only be applied when the number of basic schedules is relatively small. In case of a large set of basic schedules, a more economical alternative is to derive the above bound only for all basic schedules simultaneously (i.e., $Q = R$). Although this can lead to a slightly weaker formulation, we observed that this valid inequality still strengthened the formulation greatly. For instance U1 adding only the single valid inequality performed equally well as adding all valid inequalities, whereas for instance U2 it performed only a factor 1.3 worse on average compared to adding all $2^{|R|} - 1$ valid inequalities .

2.C Modeling Reduced Cost

In this section we show how to model the reduced cost of a sequence. We first introduce the necessary notation. Let μ_k denote the dual variables corresponding to (2.11), η_d those corresponding to (2.12) and σ_p those corresponding to (2.13). Furthermore, let γ_{ar} and θ_{ar} denote the dual variables corresponding to (2.14) and (2.15), respectively. The reduced cost $\bar{c}_s^k$ of sequence $s \in S_k$, with $k \in K_r$, is given by

$$\bar{c}_s^k = c_s^k - \mu_k - \sum_{(t,d)\in s} \eta_d - \sum_{p\in P\setminus P_K} \sum_{(t,d)\in s} f_{td}^p \sigma_p - \sum_{a\in A} \sum_{(t,d)\in s} g_{ad}(\gamma_{ar} - \theta_{ar}).$$

Note that we can aggregate the dual variables on the vertex level and rewrite the above expression as

$$\bar{c}_s^k = c_s^k - \mu_k - \sum_{(t,d)\in s} \lambda_{td}^k,$$

where the aggregated dual variables λ_{td}^k are defined as

$$\lambda_{td}^k = \eta_d + \sum_{p\in P\setminus P_K} f_{td}^p \sigma_p + \sum_{a\in A} g_{ad}(\gamma_{ar} - \theta_{ar}).$$

Chapter 3

A Three-Phase Heuristic for Cyclic Crew Rostering with Fairness Requirements*

3.1 Introduction

The scheduling of personnel is one of the most challenging planning problems for a public transport operator. This is partly due to the large-scale nature of the problem, but also due to the two conflicting objectives: On the one hand, the operator must minimize cost from an operational point of view, yet on the other hand the operator must also maximize the quality of work from an employees' point of view. The Netherlands, for example, has a history of strikes from employees of Netherlands Railways (NS), expressing their discontent regarding the scheduled work. Reoccurring themes during these conflicts are, for example, the irregularity of scheduled work, and the distribution of work among the different crew bases. Hence, it is clear that, in order to avoid such conflicts, a public transport operator should incorporate the demands of employees in the planning process.

The assignment of work to the employees is traditionally decomposed into crew

*This chapter, up to minor modifications, is a direct copy of T. Breugem, C. Schulz, T. Schlechte, and R. Borndörfer (2019): *A Three-Phase Heuristic for Cyclic Crew Rostering with Fairness Requirements.*

scheduling and crew rostering. In the former, the duties (i.e., working days) are constructed, and, in the latter, these duties are assigned to the employees. The focus in crew rostering lies on *perceived fairness* and *perceived attractiveness*, leading to a bi-objective decision problem. This problem was formalized in Chapter 2 as the Fairness-oriented Crew Rostering Problem (FCRP). Roughly speaking, allocations are perceived fair whenever each crew member performs similar work (measured over a given number of attributes). Perceived attractiveness, on the other hand, focuses on the structure of the rosters, taking, for example, the workload in each week and the rest time between consecutive working days into account. Although both objectives have overlapping components, there is a clear trade-off, as shown in Chapter 2.

It is evident that perceived fairness and attractiveness of rosters are far more ambiguous concepts than, for example, the cost of a rolling stock schedule. As a result, the crew rostering problem is generally solved multiple times for varying parameter settings, thereby steering towards a desired solution. This implies that, even during the tactical planning phase, it is desirable that high quality solutions can be obtained quickly. Furthermore, exact methods can be intractable for some of the instances encountered in practice: The exact approach developed in Chapter 2 is able to solve instances of about three roster groups and 100 duties (obtained from crew base Utrecht) in at most a few hours, but fails to obtain good solutions for instances of, say, six groups and about 200 duties in reasonable time. The latter is the size of crew base Amersfoort, where NS conducted a pilot study regarding decision support for crew rostering in the beginning of 2018. The pilot sparked interest and enthusiasm for decision support, but also highlighted the need for a heuristic algorithm for the larger instances, and is therefore a driving force behind the research in this chapter.

In this chapter, we consider a variant of the FCRP, which we will call the Cyclic Crew Rostering Problem with Fairness Requirements (CCRP-FR). This problem deviates from the FCRP by assuming a fixed, a priori known, set of fairness levels for which rosters must be constructed. This differs from the FCRP, where we aim at finding the entire trade-off curve and hence determining the relevant fairness levels is part of the solution process. In practice, the former setting is often encountered, as planners generally have some fairness levels (e.g., high, medium, and low fairness) in mind for which they want to construct rosters. One key difference with the FCRP is that the solution with maximum fairness does not necessarily have to be computed, which

can be time consuming for large instances.

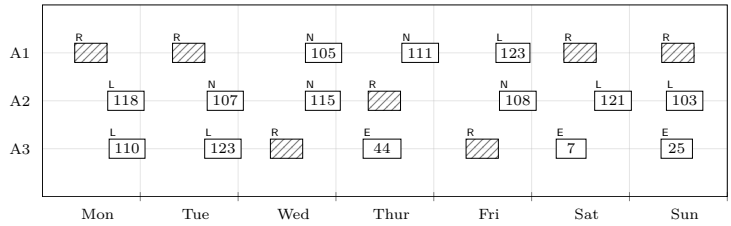
The main contribution of this chapter is a three-phase heuristic for the CCRP-FR, which combines the strengths of the exact approach for the FCRP developed in Chapter 2 with a large-scale neighborhood search algorithm. The design of the heuristic assures that good solutions for all fairness levels are obtained quickly, and can still be further improved if additional running time is available. We evaluate the performance of the proposed solution approach using real-world instances from NS. In particular, we show that the three-phase heuristic finds close to optimal solutions for most instances, and achieves a major improvement (up to 40%) over the current (sequential) approach.

The remainder of this chapter is organized as follows. In Section 3.2 we discuss the CCRP-FR in detail, and in Section 3.3, we give an overview of related work. In Section 3.4 we discuss the row-based formulation, followed by a detailed description of the three-phase heuristic in Section 3.5. Section 3.6 evaluates the performance of the solution method on practical instances from NS, and the chapter is concluded in Section 3.7.

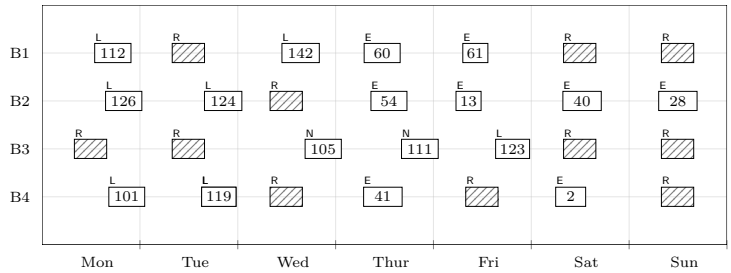
3.2 Cyclic Crew Rostering with Fairness Requirements

The goal of crew rostering is to construct rosters for the employees whilst taking both perceived fairness and perceived attractiveness into account. The crew is partitioned into *roster groups*, and each of these groups has to be assigned a roster. Figure 3.1 shows an example of two possible rosters, one for group A, consisting of three employees, and one for group B, consisting of four employees.

The roster groups operate in *cyclic rosters*, i.e., the rosters are executed by multiple employees in a periodic fashion. These rosters are constructed for a period of one year, and the work for this period is assumed to be cyclic. In other words, Monday the 22th of April is identical to Monday the 29th of April. Each roster has an underlying structure, known as the *basic schedule*. The basic schedule specifies the type of work of each day (e.g., a late duty or day-off). Each basic schedule consists of *cells* (i.e., elements to which duties must be assigned), grouped into *rows* (i.e., weeks of work) and *columns* (i.e., generic weekdays). The duty types, as specified in the basic schedules, for both rosters in Figure 3.1 is shown at the top left of the cells,



Roster Group A



Roster Group B

Figure 3.1: Example of two rosters for roster group A, consisting of three employees, and B, consisting of four employees. The duty types are indicated by the types (Early, Late, Night, and Rest) above the cells, and the numbers indicate the assigned duties.

and the numbers indicate the assigned duties. Here, the basic schedule of group B, for example, specifies that the first row starts with a late duty (L), followed by a day-off (R), and then again a late duty. Similar to Chapter 2, we assume the basic schedules to be input to the problem.

The crew rostering phase consists of allocating the duties, i.e., days of work, to the basic schedules of the different roster groups. Note that, due the the cyclic nature, also the duties are generic, i.e., the duties are specified on the weekday level. Consider, for example, duty 112, scheduled on Monday in the first row of the roster for group B, as shown in Figure 3.1. The cyclicity of the roster implies that this duty is executed by the first employee of the group in the first week, by the fourth employee in the second week, and by the third employee in the third week. This process is also known as *rolling out the roster*. Note that the work for an employee of group B repeats itself every four weeks and for an employee of group A every three weeks, since the rosters have has four and three rows, respectively.

The perceived fairness and perceived attractiveness both depend on the way the du-

ties are allocated to the cells of the basic schedules. The perceived fairness relates to the distribution of work among the roster groups (e.g., one roster group should not have much nicer work compared to another roster group), and the perceived attractiveness relates to the structure of these rosters (e.g., sufficient rest time, average workload).

The perceived fairness of the rosters is based on different characteristics of the duties. Since every duty represents a work day, some can be considered more desirable than others. Duties with many tasks on out-dated rolling stock, for example, are generally considered undesirable. These characteristics are referred to as *duty attributes*. The perceived fairness is measured by the spread (i.e., the difference between the maximal and minimal average value of the duties in the roster) over the groups for the different duty attributes. The smaller this spread, the higher the perceived fairness.

The perceived attractiveness considers the structural characteristics of the rosters, and is defined by so-called *roster constraints*, which forbid or penalize assignments of duties. Well-known examples of roster constraints are rest constraints, enforcing a certain minimum time to rest between consecutive working days, and workload constraints, enforcing a maximum amount of work within a week. The smaller the penalty incurred from the roster constraints, the higher the perceived attractiveness.

Improving both the perceived fairness and perceived attractiveness is not always possible, despite the fact that both metrics aim at improving the quality of the scheduled work. This difference is best illustrated with a simple example, based on the rosters of Figure 3.1. Suppose the current average duty length is 7 hours and 50 minutes for group A, and 7 hours and 45 minutes for group B. Consider the Monday duties 110 and 112, assigned to groups A and B, respectively. Duty 110 starts at 13:15 and ends at 21:15, hence has a length of 8 hours, and duty 112 starts at 16:30 and ends at 00:10, having a length of 7 hours and 40 minutes. By swapping duties 110 and 112 we would reduce the spread in average duty length, and hence improve the perceived fairness. This swap, however, also implies that the rest period on Tuesday in row B1 becomes shorter, which makes the roster possibly less attractive.

The CCRP-FR can now be stated as follows: Given as input the basic schedules and duties, and a set of fairness levels, determine attractive rosters for each fairness level. That is, determine for each fairness level a roster for each group that minimizes the penalties incurred from the roster constraints, whilst enforcing the desired fairness level.

3.3 Related Work

Crew planning is a widely studied problem in the literature, dating back as far as Dantzig (1954). The applications range from health care (e.g., nurse rostering) to transportation (e.g., airline, railway, and bus planning), and the solution methodology covers well-known exact and heuristic methods, such as branch-and-price, simulated annealing, and tabu search. In this section we focus mainly on crew planning within the transportation sector: We refer to Ernst et al. (2004) and Van den Bergh et al. (2013) for more general overviews.

Crew planning is commonly decomposed into two sequential planning phases. In the first phase, known as the crew scheduling or crew pairing problem, the days of work (i.e., duties or pairings) are constructed. This phase mainly focuses on operational cost (i.e., the number of necessary crew members), together with other key factors, such as the fairness of the work allocation and the constraints resulting from the collective labor agreements. The crew scheduling problem is a well-studied problem in the literature (see, for example, Desrochers and Soumis (1989), Hoffman and Padberg (1993), Grötschel et al. (2003), Abbink et al. (2005), among others), and has been considered in numerous variants, such as rescheduling whenever the underlying tasks are modified (e.g., Lettovskỳ et al. (2000), Potthoff et al. (2010)) and in conjuction with other planning problems, such as aircraft routing and vehicle scheduling (e.g., Cordeau et al. (2001), Huisman et al. (2005a)).

The second planning phase, known as crew rostering, consists of combining the duties (or pairings) into rosters, which are sequences of duties (or pairings) satisfying numerous labor constraints. Typical constraints consider, for example, days off, rest times, variation of work, and personal preferences. Rosters are generally classified as cyclic, i.e., multiple employees working the same roster, and acyclic, i.e., each individual employee working his or her own roster. The latter type is common in the healthcare sector and airline industry (see, for example, Kohl and Karisch (2004) and De Causmaecker and Vanden Berghe (2011), and references therein), whereas the former type is often used in railway operations and mass transit (see Huisman et al. (2005b), Caprara et al. (2007)). Cyclic crew rostering is well-studied in the literature and a variety of formulations have been proposed to model the problem, including a generalized assignment formulation (e.g., Hartog et al. (2009)), a multi-commodity flow formulation (e.g., Caprara et al. (1997), Xie and Suhl (2015), Borndörfer et al. (2015)), and a set covering or set partitioning formulation (e.g., Caprara et al. (1997),

Freling et al. (2004), Borndörfer et al. (2015)). The integration of both crew scheduling and rostering has been considered in Mesquita et al. (2013) and **Borndorfer2017** which both propose a solution method based on Benders decomposition. Note that most of aforementioned work focuses on exact methods or heuristics based on mathematical programming techniques, such as column generation. For large-scale and highly complex rostering problems, heuristic methods are sometimes better suited. We refer to Van den Bergh et al. (2013), Table 13, for a detailed overview of solution approaches. In this paper, we build upon the variable-depth neighborhood search heuristic proposed in Borndörfer et al. (2015).

The incorporation of fairness measures in combinatorial optimization problems is, to the best of our knowledge, a relatively young field of research. The fairness of utility allocations, however, has a long history in the economic literature, dating back to the work on bargaining problems by Nash (1950). Recent work in the field of Operations Research has focused on the trade-off between efficiency (e.g., minimizing cost) and fairness (e.g., maximizing the lowest derived utility). This work includes, among others, the work of Bertsimas et al. (2012) and Bertsimas and Gupta (2015) in the context of air traffic flow management, Bertsimas et al. (2013) on organ allocation, and, in Chapter 2, on the trade-off between fairness and attractiveness in crew rostering.

The three-phase heuristic extends the exact branch-price-and-cut approach of Chapter 2 to a solution approach for large-scale instances. We showed in Chapter 2 that the exact method is able to solve practically sized instances in reasonable time. For some of the large instances encountered in practice, however, these computation times are considered too high, and quickly found high-quality solutions, although possibly suboptimal, are preferred. We therefore develop a heuristic taking the exact method as basis, thereby inheriting the strong points of this method, while avoiding excessive computation times. This chapter aims at bridging the gap between the exact method developed in Chapter 2 and the current sequential practice, where first the duties are assigned to the roster groups, and then the rosters per group are optimized separately.

3.4 Mathematical Formulation

In this section we discuss the mathematical formulation underlying the heuristic. We consider the *row-based* formulation introduced in Chapter 2 for the FCRP, in which

a variable represents the simultaneous assignment of multiple duties to all cells in a row of the basic schedules. The main benefit of this formulation is that all weekly roster constraints, such as weekly variation and maximum workload constraints, can be modeled implicitly in the definition of the variables. In Section 3.4.1, we introduce the necessary notation and terminology, and in Section 3.4.2, we present the row-based formulation.

3.4.1 Notation and Terminology

The row-based formulation models the assignment of duties to the cells by means of *roster sequences*, which specify a simultaneous assignment of duties to a row. In the case of Figure 3.1, for example, a roster sequence for row A1 has to specify a duty for all three cells (recall that rest days are assumed fixed). In this specific roster, the selected roster sequence specifies the assignment of duty 105 for Wednesday, 111 for Thursday, and 123 for Friday.

Let D denote the set of duties, and let R denote the set of basic schedules. Each basic schedule r is defined by a set of cells T_r, and the set of all cells is denoted by T. An assignment of a duty d to a cell t in a basic schedule will be denoted by the pair (t, d). We define n_r as the total number of duties to be assigned to basic schedule r. Let K denote the set of all rows, and K_r denote the set of rows for basic schedule $r \in R$. We define S_k as the set of all roster sequences for row $k \in K$, where each roster sequence is formally defined as a sequence of assignments (t, d) for the cells in k. The parameter h^k_{ds} indicates whether sequence $s \in S_k$ contains duty d.

The row-based formulation models the assignment of duties to the cells using the decision variables x^k_s, for all $k \in K$ and $s \in S_k$, indicating whether roster sequence $s \in S_k$ is assigned to row k. The perceived fairness and perceived attractiveness are modeled as follows.

Perceived Fairness

The perceived fairness is expressed in terms of the maximum and minimum average values of different duty attributes (e.g., duty length) among the roster groups. Let A denote the set of duty attributes, and let g_{ad} denote the value of attribute $a \in A$ for duty $d \in D$. Each attribute has a specified lower bound ℓ_a and upper bound u_a, representing the minimum and maximum allowed average values for a basic schedule.

In case of the duty length, for example, one could enforce that no roster group works more than 8 hours on average. Besides these bounds, each duty attribute has an associated weight w_a, representing the relative importance of the different duty attributes when calculating the perceived fairness.

The calculation of the perceived fairness is based on the variables v_a and z_a, representing the minimum and maximum average value of duty attribute a among all roster groups, respectively. These variables are linked to the x_s^k variables by means of the constraints

$$\sum_{k \in K_r} \sum_{s \in S_k} \sum_{(t,d) \in s} g_{ad} x_s^k \leq n_r z_a \qquad \forall a \in A, r \in R \tag{3.1}$$

$$\sum_{k \in K_r} \sum_{s \in S_k} \sum_{(t,d) \in s} g_{ad} x_s^k \geq n_r v_a \qquad \forall a \in A, r \in R, \tag{3.2}$$

assuring that v_a and z_a attain the minimum and maximum average value for each duty attribute. Given the values v_a and z_a, the fairness level is calculated as

$$\sum_{a \in A} w_a (z_a - v_a), \tag{3.3}$$

and a fairness budget ζ is enforced by assuring that (3.3) does not exceed ζ.

Perceived Attractiveness

Each roster constraint penalizes, or forbids, certain combinations of assignments of duties to the cells of the basic schedules. Let P denote the set of roster constraints and let f_{td}^p denote the coefficient for assigning duty d to cell t for roster constraint $p \in P$. For any given assignment of duties to the cells, the violation of the roster constraint is given by the sum of these coefficients minus some allowed threshold value b_p, and this violation is restricted to the interval $\Delta_p = [0, m_p]$. The interval $\Delta_p = [0, 1]$, for example, would allow a violation of at most one. Each roster constraint $p \in P$ can be written as

$$\sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f_{td}^p x_s^k \leq b_p + \delta_p, \tag{3.4}$$

where $\delta_p \in \Delta_p$ represents the violation. The perceived attractiveness is maximized by minimizing the sum of roster constraint violations $c_p \delta_p$, where the cost coefficients

c_p regulate the relative importance of the different roster constraints. The row-based formulation allows every roster constraint that is *contained* in a row, i.e., it only has non-zero coefficients f_{td}^p for the cells in one row, to be modeled implicitly. Let $P_K \subseteq P$ denote the set of roster constraints that are contained in the rows $k \in K$ (and can therefore be modeled implicitly), and let c_s^k denote the cost associated with sequence $s \in S_k$, that is, c_s^k is the sum of all roster constraint violations in the sequence s.

3.4.2 Row-Based Formulation

The concepts introduced in Section 3.4.1 can be integrated to obtain the row-based formulation for the CCRP-FR, similar to the FCRP formulation introduced in Chapter 2. For a given fairness budget ζ, the model reads as follows.

$$\min \quad \sum_{k \in K} \sum_{s \in S_k} c_s^k x_s^k + \sum_{p \in P \setminus P_K} c_p \delta_p \tag{3.5}$$

$$\text{s.t.} \quad \sum_{a \in A} w_a \left(z_a - v_a \right) \leq \zeta \tag{3.6}$$

$$\sum_{s \in S_k} x_s^k = 1 \qquad \forall k \in K \tag{3.7}$$

$$\sum_{k \in K} \sum_{s \in S_k} h_{ds}^k x_s^k = 1 \qquad \forall d \in D \tag{3.8}$$

$$\sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f_{td}^p x_s^k \leq b_p + \delta_p \qquad \forall p \in P \setminus P_K \tag{3.9}$$

$$\sum_{k \in K_r} \sum_{s \in S_k} \sum_{(t,d) \in s} g_{ad} x_s^k \leq n_r z_a \qquad \forall a \in A, r \in R \tag{3.10}$$

$$\sum_{k \in K_r} \sum_{s \in S_k} \sum_{(t,d) \in s} g_{ad} x_s^k \geq n_r v_a \qquad \forall a \in A, r \in R \tag{3.11}$$

$$z_a \leq u_a \qquad \forall a \in A \tag{3.12}$$

$$v_a \geq \ell_a \qquad \forall a \in A \tag{3.13}$$

$$x_s^k \in \mathbb{B} \qquad \forall k \in K, s \in S_k \tag{3.14}$$

$$\delta_p \in \Delta_p \qquad \forall p \in P \setminus P_K \tag{3.15}$$

$$v_a, z_a \in \mathbb{R} \qquad \forall a \in A. \tag{3.16}$$

The objective (3.5) expresses that we minimize the penalties incurred from the roster constraints, partly expressed by the roster sequence costs and partly expressed by the cost of the explicitly modeled roster constraints. The fairness budget is enforced by (3.6). Constraints (3.7) and (3.8) assure that the duties are assigned correctly to the basic schedules: Each row is assigned exactly one roster sequence, and each duty appears in exactly one roster sequence. Constraints (3.9) model the roster constraint violations, as discussed in Section 3.4.1. Furthermore, Constraints (3.10) and (3.11) assure that the variables v_a and z_a are set to the minimum and maximum value, respectively, while (3.12) and (3.13) enforce the lower and upper bounds on the attribute values. Finally, Constraints (3.14)–(3.16) express the domains of the decision variables.

3.5 Three-Phase Heuristic

In this section we present the three-phase heuristic. We first give a general overview of the heuristic and discuss the key components. We then discuss the three different phases separately: In Section 3.5.1, we discuss the first phase of the heuristic, in which a feasible solution is obtained using a sequential decomposition, and in Sections 3.5.2, and 3.5.3, we discuss the paiwise and global improvements phases, respectively.

Recall that for the CCRP-FR the set of fairness levels is considered input. That is, the goal is to determine a solution maximizing attractiveness for a given set of fairness levels, represented by the ordered fairness budgets $\zeta_1 < \ldots < \zeta_n$. In this case, ζ_1 corresponds to the highest fairness level (i.e., smallest budget), and ζ_n to the lowest fairness level (i.e., largest budget). For the sake of explanation, we will often consider the three fairness levels high, medium, and low, but note that the approach does not rely on any assumption regarding the number of fairness levels, nor the size of their corresponding budgets.

The three-phase heuristic aims at quickly finding high-quality solutions for each fairness level. Figure 3.2 schematically visualizes the algorithm. First, we obtain an initial allocation of the duties to the basic schedules for the highest fairness level, i.e., the lowest fairness budget. This is done by solving a mixed-integer linear program. The roster per basic schedule, given the allocation of the duties, is then optimized using the exact branch-price-and-cut approach developed in Chapter 2.

The second and third phase of the algorithm both aim at finding profitable re-

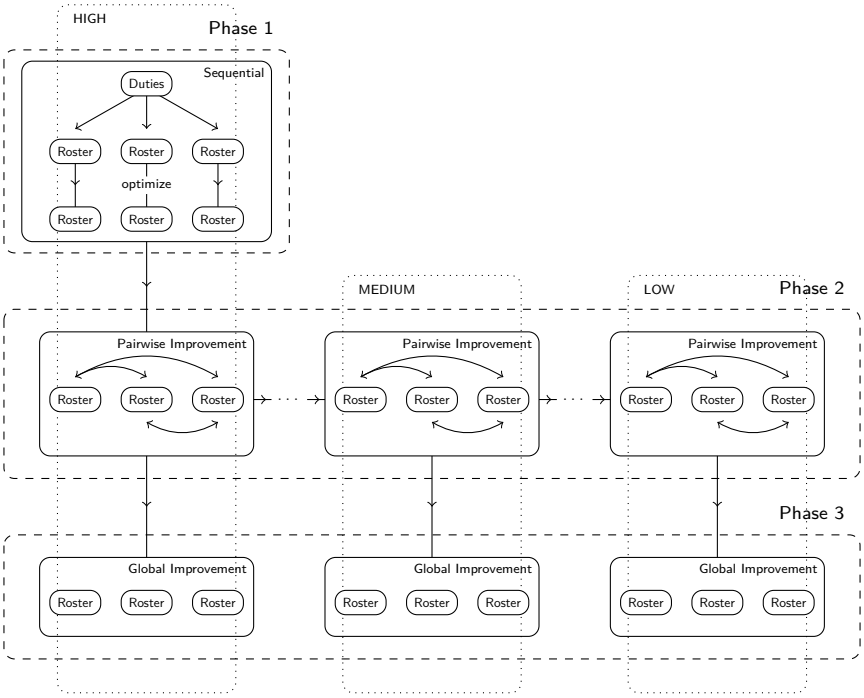


Figure 3.2: Schematic visualization of the three-phase heuristic. For illustrative purposes, three fairness levels (high, medium, and low) are highlighted. The algorithm starts by finding a feasible solution for the tightest fairness level using a sequential approach (Phase 1). This solution is taken as starting point of a pairwise improvement phase (Phase 2), where we obtain a feasible solution for each fairness level. Finally, the separate solutions are further improved in a global improvement step (Phase 3).

allocations of duties among the roster groups. In the second phase, we look for pairwise improvements, i.e., we try to find a profitable re-allocation of duties between pairs of roster groups. This is done by solving a reduced problem, based on the linear relaxation of (3.5)–(3.16). The key aim of the second phase is to quickly find good solutions for each fairness level, which is achieved by using the found solutions for high levels as starting point for the lower levels.

In the third and final phase of the heuristic we invoke a more time-consuming large-scale neighborhood search algorithm to further improve the solutions. Note that the solutions found in the third phase are not used as starting solutions for the second phase. This assures that, even upon early termination, the algorithm returns

a solution for each fairness level, thereby not having to wait for the third, most time consuming, phase to finish. Furthermore, this explicit decoupling would also allow the third phase to be executed in parallel.

3.5.1 Phase 1: Sequential Decomposition

In the first phase of the algorithm we consider a natural, yet heuristic, decomposition: First, a fair and feasible allocation of the duties to the roster groups is determined, and, secondly, the roster per group is optimized given the allocated duties. This sequential decomposition closely resembles the current practice. The solution found in this phase can therefore be considered as a well-motivated benchmark solution. The allocation of the duties to the groups is obtained by solving a feasibility problem. We solve this problem using a *cell-based* formulation: Let the binary variable π_{td}, for all $t \in T$ and $d \in D$ indicate whether duty d is assigned to cell t, and let $D_t \subseteq D$ denote the subset of duties compatible with cell t, i.e., those duties for which the weekday and duty type match. Similarly, let $T_d \subseteq T$ denote the cells to which duty d can be assigned. We obtain a feasible allocation by solving the following system of inequalities.

$$\sum_{a \in A} w_a(z_a - v_a) \leq \zeta \tag{3.17}$$

$$\sum_{d \in D_t} \pi_{td} = 1 \qquad \forall t \in T \tag{3.18}$$

$$\sum_{t \in T_d} \pi_{td} = 1 \qquad \forall d \in D \tag{3.19}$$

$$\sum_{t \in T} \sum_{d \in D} f^p_{td} \pi_{td} \leq b_p + m_p \qquad \forall p \in P \tag{3.20}$$

$$\sum_{t \in T_r} \sum_{d \in D} g_{ad} \pi_{td} \leq n_r z_a \qquad \forall a \in A, r \in R \tag{3.21}$$

$$\sum_{t \in T_r} \sum_{d \in D} g_{ad} \pi_{td} \geq n_r v_a \qquad \forall a \in A, r \in R \tag{3.22}$$

$$z_a \leq u_a \qquad \forall a \in A \tag{3.23}$$

$$v_a \geq \ell_a \qquad \forall a \in A \tag{3.24}$$

$$\pi_{td} \in \mathbb{B} \qquad \forall t \in T, d \in D \tag{3.25}$$

$$v_a, z_a \in \mathbb{R} \qquad \forall a \in A. \tag{3.26}$$

Constraints (3.18) and (3.19) assure each cell is assigned exactly one duty and (3.20) guarantees the feasibility with respect to the roster constraints. Note that, compared to (3.4), we set δ_p equal to the upper bound m_p. The remaining constraints model the fairness budget, as discussed in Section 3.4.1. The model (3.17) – (3.26) without fairness constraints is similar to the assignment model proposed in Hartog et al. (2009) for cyclic crew rostering.

Given a feasible allocation, the roster for each group is optimized. This sequential decomposition greatly simplifies the problem, as it eliminates the connection between fairness and attractiveness: For a known allocation of duties to the roster groups, the fairness constraints are either satisfied or not, and hence the problem decomposes into a set of (simpler) cyclic crew rostering problems, one for each roster group. For practical roster group sizes, these problems can be solved efficiently using the branch-price-and-cut approach proposed in Chapter 2.

3.5.2 Phase 2: Pairwise Improvement

In the second phase of the algorithm we obtain a feasible solution for each fairness budget. First we improve the solution obtained in Phase 1 to obtain a solution for ζ_1, and then we proceed in an iterative fashion: The solution for the $(i+1)$-th fairness budget ζ_{i+1}, is obtained by searching a neighborhood around the solution for the i-th fairness budget ζ_i, i.e., we exploit the increase of the fairness budget to find profitable re-allocations of duties.

We search for a profitable re-allocation of duties between pairs of roster groups. Note that profitable re-allocations might exist since we increase the fairness budget, hence allocations previously not feasible become feasible. We restrict the re-allocation to pairs of roster groups, to assure that the running time scales well with the instance size. The pairs of roster groups are ordered such that the small groups are considered first. To illustrate this, consider four roster groups A, B, C, and D, of 8, 9, 10, and 12 employees, respectively. We first look for a profitable re-allocation between A and B, then between A and C, then A and D, then between B and C, then B and D, and finally between C and D, each time updating the rosters when a profitable re-allocation has been found. Hence, given k groups, we check $k(k-1)/2$ pairs, after which the improvement step terminates.

The re-allocation is determined by solving the row-based formulation for a reduced set of roster sequences, where the reduction is based on the solution to the linear

relaxation: Consider a given feasible integer solution $\hat{x}$ and an optimal (fractional) solution of the linear relaxation $\bar{x}$. For each cell $t \in T$, we allow only a subset $\bar{D}_t \subseteq D_t$ of duties to be assigned. The set $\bar{D}_t$ is initialized by the duty assigned to t in the solution $\hat{x}$, to assure the integer solution remains feasible, and is then enriched based on $\bar{x}$: For each $k \in K$ and $t \in k$, we enlarge $\bar{D}_t$ by adding the duties $d \in D_t$ for which

$$\sum_{\substack{s \in S_k: \\ s \ni (t,d)}} \bar{x}_s^k > 0,$$

i.e., we add those duties to $\bar{D}_t$ that have a non-zero coefficient for t in $\bar{x}$. Figure 3.3 illustrates this reduction. Generally, not too many duties are selected this way.

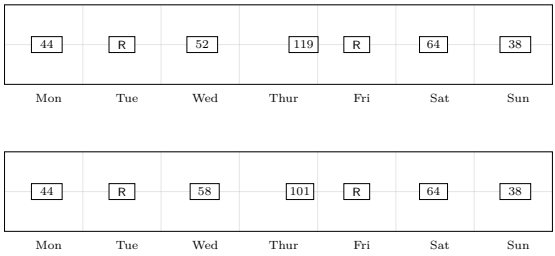


Figure 3.3: An example of the problem reduction. Suppose the shown roster sequences are assigned positive values in $\bar{x}$ for some given row. The allowed duties obtained from $\bar{x}$ are given by $\bar{D}_{\text{Mon}} = \{44\}$, $\bar{D}_{\text{Wed}} = \{52, 58\}$, $\bar{D}_{\text{Thur}} = \{101, 119\}$, $\bar{D}_{\text{Sat}} = \{64\}$, and $\bar{D}_{\text{Sun}} = \{38\}$. In this example, two additional roster sequences will be generated leading to a total of four roster sequences.

The reduced set of roster sequences $\bar{S}_k \subseteq S_k$ for each row $k \in K$, consists of exactly those roster sequences assigning only duties in $\bar{D}_t$ to each $t \in k$. This set is determined by complete enumeration. The resulting reduced model is solved using a commercial solver to obtain a possible improved allocation of the duties. In this phase of the heuristic, we also solve the linear relaxation of (3.5)– (3.16) for all roster groups simultaneously to obtain a lower bound on the optimal solution value.

3.5.3 Phase 3: Global Improvement

In the third and final step of the algorithm, each of the solutions is further improved using a sophisticated local search algorithm. In this phase of the heuristic we aim at finding improving duty exchanges between *all* roster groups (hence, the name *global*

improvement), as opposed to Phase 2, where we only consider pairs of roster groups. This is done using local search.

The proposed local search algorithm is based on variable-depth neighborhood search (VDNS), a neighborhood search technique belonging to the family of so-called very large-scale neighborhood search algorithms (see e.g., Ahuja et al. (2002), Pisinger and Ropke (2010) for a detailed overview). The key idea behind VDNS is to (heuristically) explore a large neighborhood by constructing a *chain* of moves, with each move belonging to a smaller neighborhood. In this way, a large part of the solution space is searched, including moves involving a large number of elements, whilst avoiding excessive computation times. This simple yet successful idea dates back to Lin and Kernighan (1973), who used it to construct an efficient heuristic for the Traveling Salesman Problem (TSP). The concept of VNDS has been applied to a wide range of difficult combinatorial optimization problems ever since.

To apply VDNS one first needs to define a parametrized neighborhood N_k, satisfying

$$N_1 \subseteq \ldots \subseteq N_i \subseteq \ldots \subseteq N_n.$$

Typical examples of such neighborhoods are the k-permutation neighorhood, i.e., permuting k elements of the solution, or the k-arc exchange neighborhood considered in Lin and Kernighan (1973). The key idea is to first pick a suitably sized neighborhood N_i, large enough to escape local optima and small enough to be searched efficiently, and heuristically explore N_n by chaining together moves in the smaller neighborhood N_i. The chain is constructed by making a profitable move in N_i, fixing the involved elements, and repeating this until no more profitable moves exist. As noted in Johnson and McGeoch (1997), this chaining of moves can be seen as a special type of tabu search, using a flexibly sized tabu list. Figure 3.4 gives a schematic visualization of VDNS.

Crucial for VDNS to work is a suitably picked neighborhood. In particular, the neighborhood should (*i*) be able to escape local optima, (*ii*) be searchable in reasonable time, and (*iii*) guarantee that the solution remains feasible, or at least can be easily made feasible. The latter depends heavily on the structure of the underlying problem. In case of a fixed basic schedule, for example, duties cannot be freely exchanged (e.g., a late duty on Monday cannot be exchanged with an early duty on Monday nor a late duty on Tuesday). We therefore propose two neighborhoods for the VDNS algorithm, based on two different duty exchange operations: *vertical* and

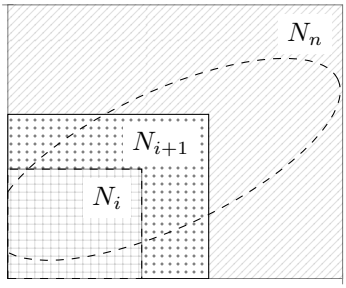


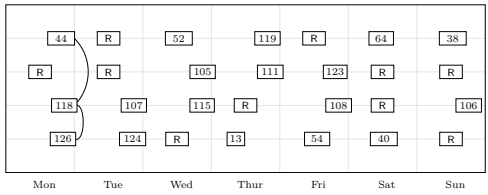
Figure 3.4: Schematic visualization of VDNS. The parametrized neighborhood N_k is heuristically searched (indicated by the dashed area) by chaining N_i moves. The resulting move can be part of N_i, N_{i+1}, or even N_n, implying that the 'depth' of the move is not fixed.

horizontal duty exchanges.

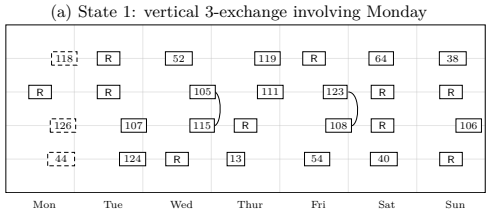
Vertical k-exchanges are exchanges between k duties of the same type and in the same column, i.e., they are vertically aligned. This assures that the involved duties can always be exchanged without violating the basic schedule, i.e., the structure of the solution is not affected by a vertical exchange. The feasibility with respect to the roster constraints can be readily checked when performing an exchange, hence the feasibility of the solution can always be assured. Note that the vertical k-exchange neighborhood can be searched in $\mathcal{O}\left(|D|^k\right)$ time. Figure 3.5a gives an example of a vertical 3-exchange. We will denote the vertical k-exchange neighborhood by V_k.

Horizontal k-exchanges are a more involved type of exchange, affecting duties of different types and in different columns. The horizontal exchange neighborhood aims at complementing the vertical exchange neighborhood, which can get stuck in local optima due to the restriction to one single column. Formally, a horizontal k-exchange, for k even, is a sequence of $k/2$ vertical 2-exchanges, where each 2-exchange shares at least one row with its predecessor. Figure 3.5b gives an example of a horizontal 4-exchange. By considering a sequence of multiple vertical 2-exchanges we allow duties of different weekdays and types to be affected within a single exchange, i.e., one exchange can involve multiple duties in one row. Furthermore, we limit the search time by only considering sequences where consecutive exchanges share a row: It is not difficult to show that the horizontal k-exchange neighborhood can be searched in $\mathcal{O}\left(|D|^{k/2+1}\right)$ time. We will denote the horizontal k-exchange neighborhood by H_k.

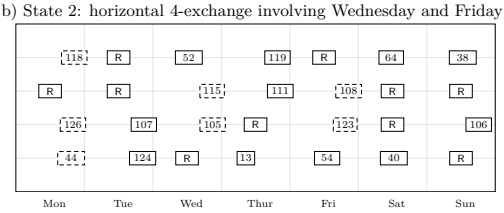
The proposed VDNS algorithm combines chains of horizontal 4- and vertical 3-



(a) State 1: vertical 3-exchange involving Monday



(b) State 2: horizontal 4-exchange involving Wednesday and Friday



(c) State 3: final roster

Figure 3.5: Example of horizontal and vertical exchanges. First, a vertical 3-exchange is performed on the the Monday duties 44, 118 and 126. Then, a horizontal 4-exchange is performed on the Wednesday duties 105 and 115, and the Friday duties 108 and 123.

exchanges, searchable in $\mathcal{O}\left(|D|^3\right)$ time, with horizontal 6- and vertical 4-exchanges, searchable in $\mathcal{O}\left(|D|^4\right)$ time. This is done similar to the Dynamic Depth-EXchange (DEX) algorithm, introduced in Borndörfer et al. (2015): We combine H_4+V_3 chains, i.e., chains of horizontal 4- and vertical 3-exchanges, with single $H_6 + V_4$ moves to escape local optima. These neighborhoods are large enough to escape local optima, and small enough to be considered efficient.

The final VDNS algorithm is shown in Figure 3.6. Starting from an initial solution, we construct improving $H_4 + V_3$ chains. Here we allow for some small deterioration in the chaining process, to add more flexibility to the search. Once the chaining procedure finishes, i.e., the best $H_4 + V_3$ move among the non-fixed duties leads to too much of a cost increase, the solution is updated and all duties are removed from the set of fixed duties. If an improving chain was found, we again search for an improving chain for the updated solution. Otherwise, we try to escape the current local optimum by using one (strictly profitable) $H_6 + V_4$ move. If this succeeds, we

repeat the chaining procedure for the updated solution. Otherwise the algorithm terminates and the final solution is found.

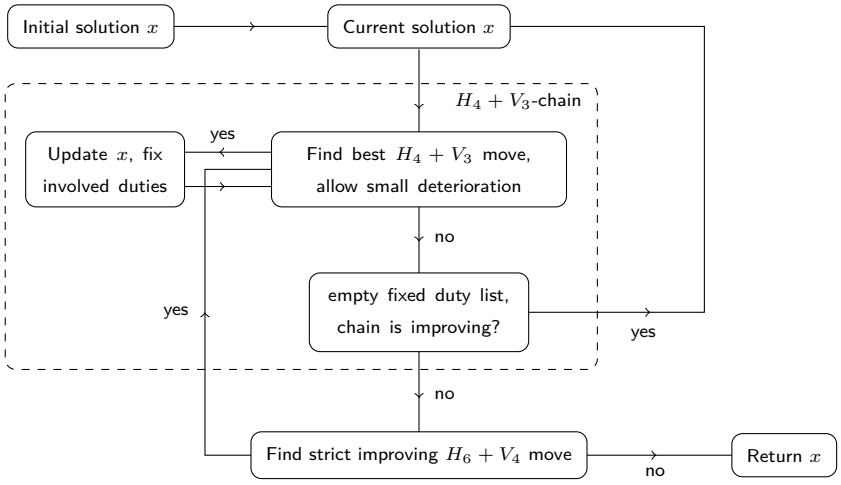


Figure 3.6: Schematic visualization of the VDNS algorithm. An initial solution is improved using $H_4 + V_3$ chains. If no improving chain can be found, a single $H_6 + V_4$ move is considered. In case no profitable $H_6 + V_4$ move exists, the algorithm terminates. Otherwise, the algorithm continues the search for improvements using $H_4 + V_3$ chains.

3.6 Computational Experiments

To evaluate the performance of the heuristic, we apply our solution approach to different instances based on data from NS. In Section 3.6.1 we discuss the experimental set-up and in Section 3.6.2 we show the computational results.

3.6.1 Experimental Set-Up

We consider a total of 10 instances. The first four instances are those considered in Chapter 2, and can be used to validate the performance of the heuristic, as the optimal solutions are known for these instances. For the remaining six (larger) instances, no optimal solutions are known.

The instances each consist of basic schedules and duties that need to be assigned to

these schedules. The basic schedules specify a type, i.e., a duty type or a day off, for each cell. The considered duty types are early, late, and night. Based on the types of the duties that need to be assigned, each roster group can be (roughly) categorized in one of three categories: early, late-night, and mixed (i.e., all three types). We refer to these categories as E, LN, and M, respectively.

	Duties				Employees			
	E	L	N	Total	E	LN	M	Nr Groups
1	55	29	29	113	14	12	4	3
2	58	33	24	115	12	12	4	3
3	38	22	11	71	6	6	6	3
4	37	17	15	69	6	8	6	3
5	74	62	55	191	12	12/12	12	4
6	88	36	31	155	14	8	12/6	4
7	86	36	37	159	12/6	12	12	4
8	74	30	16	120	6/6	8	12	4
9	126	70	54	250	14/6	12/8	12/12	6
10	142	63	61	266	12/12	12/8	12/12	6

Table 3.1: Characteristics of the instances. For each instance, the number of duties of type early (E), late (L), and night (N) is shown, together with the total number of duties, and the number of employees of each category early (E), late-night (LN), and mixed (M). Finally, the number of roster groups per instance is shown.

For each instance, Table 3.1 shows the number of duties of each type, and the size of the groups per category (i.e., the entry 12/8 for LN means the instance contains one LN group of 12 employees and one LN group of 8 employees). The first four instances all consist of three groups, one of each type, of varying sizes. The second four instances all consist of four groups of varying type and varying size. The fifth instance is the largest among these four, with approximately 190 duties, whereas the eighth instance is the smallest, with roughly 120 duties. The final two instances both consist of six groups, with two groups of each type. The total number of duties for both instances is roughly 250. The size of these instances corresponds to the size of the pilot study mentioned in Section 3.1.

The perceived fairness is based on five different duty attributes. These include three attributes concerning the scheduled work: the percentages of *high quality work* (e.g., Intercity work), *aggression work* (e.g., trips where passengers are less likely to have a ticket, and hence might become aggressive), and *double decker work* (as work on

double decker trains is physically more demanding). Furthermore, there are two attributes regarding the entire duty: the *duty length*, and the *repetition within duty* (as duties with many of the same trips are repetitive, which is considered undesirable).

The perceived attractiveness is determined by four types of roster constraints. These types can be divided into two classes. The first class contains 'binary' roster constraints, i.e., roster constraints linking exactly two cells in the basic schedule. This class contains the so-called *rest time* end *rest day* constraints: It is required that an employee has a minimum rest time after each duty. After a night duty this rest time should be 14 hours and after any other type of duty it should be 12 hours. Furthermore, rest times shorter than 16 hours are penalized. In addition, the length of each scheduled rest period has to be sufficient. This implies that there is a minimal time enforced between duties scheduled before and after rest days. The enforced rest time is 6 hours plus 24 hours for each rest day. The second class of roster constraints consists of 'row-based' roster constraints. This class contains *workload* constraints, i.e., the total workload in a row is not allowed to exceed 45 hours, and a large collection of *variation* constraints. This latter type of constraint aims at balancing different duty attributes (e.g., duty length, percentage double decker work) equally over the rows, which is achieved by penalizing positive deviations from the average (measured over all duties) for each row in the roster. In total we consider variation constraints for 9 different attributes. Note that the stronger bound obtained from the row-based formulaton (compared to the cell-based formulation) results from capturing the row-based roster constraints directly in the roster sequence costs.

3.6.2 Computational Results

We evaluated the heuristic solution method on each of the ten instances using different fairness budgets. The cell-based formulation used in the first phase is solved using CPLEX 12.7.1 (from hereon simply referred to as CPLEX), with a time limit of 30 minutes, i.e., whenever no feasible allocation is found within 30 minutes the fairness budget is considered infeasible. The optimization problem per roster group, when a feasible allocation is found, is solved to optimality using the branch-price-and-cut approach developed in Chapter 2. For the pairwise improvement step in the second phase, we allow a solving time of two minutes per pair. The linear relaxation of the row-based formulation is solved using the column generation approach proposed in Chapter 2, and the reduced problem is again solved using CPLEX.

The results for the first four instances are shown in Table 3.2. For each instance, we consider five different fairness levels: extreme, high, moderate, low, and poor. The corresponding fairness budgets are obtained as follows. We first determine the exact trade-off curve between perceived fairness and attractiveness, using the branch-price-and-cut algorithm of Chapter 2. We then pick the fairness budgets for each level based on the quantiles of the trade-off curve: extreme corresponds to the leftmost (i.e., fairest) point, high to the 10% quantile, moderate to the 25% quantile, low to the 50% quantile, and, finally, poor to the 75% quantile of the curve. In this way, we assure that the fairness levels cover the entire spectrum of possible solutions.

	Fairness	Phase 1	Phase 2	Phase 3	Optimal	Root	Gain (%)	Gap (%)
	Extreme	1175.8	1175.8	1175.8	1175.8	1155.6	0.0	0.0
	High		1172.6	1172.6	1169.2	1133.4	0.3	0.3
1	Moderate		1172.3	1172.3	1164.2	1128.5	0.3	0.7
	Low		1172.3	1165.8	1133.8	1115.4	0.8	2.7
	Poor		1143.1	1142.7	1120.8	1104.8	2.8	1.9
	Extreme	1288.0	1288.0	1288.0	1288.0	1172.3	0.0	0.0
	High		1288.0	1248.6	1216.4	1158.3	3.1	2.6
2	Moderate		1257.3	1226.3	1192.2	1149.3	4.8	2.8
	Low		1257.3	1210.2	1186.1	1135.8	6.0	2.0
	Poor		1204.5	1196.7	1177.8	1126.6	7.1	1.6
	Extreme	979.3	979.3	979.3	979.3	800.9	0.0	0.0
	High		979.3	979.3	853.8	793.4	0.0	12.8
3	Moderate		859.0	859.0	830.3	788.4	12.3	3.3
	Low		802.4	802.4	794.0	781.5	18.1	1.0
	Poor		791.3	787.5	787.5	777.6	19.6	0.0
	Extreme	910.5	910.5	910.5	910.5	776.7	0.0	0.0
	High		872.1	872.1	847.2	767.3	4.2	2.9
4	Moderate		872.1	872.1	827.7	761.3	4.2	5.1
	Low		808.4	808.4	788.0	746.8	11.2	2.5
	Poor		774.2	774.2	774.2	742.7	15.0	0.0

Table 3.2: Results for the first four instances for five different fairness levels. For each instance and each fairness level, we show the objective values, i.e., perceived attractiveness penalty, obtained in the different phases of the heuristic, the optimal solution value, the root bound, the gain compared to the benchmark solution (i.e., Phase 1), and the gap with the optimal solution.

Table 3.2 shows that the heuristic finds high-quality, close to optimal, solutions for all but a few instances. For the extreme fairness level, the solution found in the first phase coincides with the optimal solution for all four instances. As Table 3.2 shows, the major improvement is often achieved in the second phase of the algorithm, i.e., the pairwise improvement phase. The third phase generally improves only slightly upon this solution. There are, however, also some cases (e.g., instance 2) in which the third phase greatly improves upon the second phase. Note that for most of the

instances, there is a substantial integrality gap.

The computation times for each instance and each fairness level are shown in Table 3.3. The overall computation times are decomposed per fairness level and phase of the algorithm. Furthermore, we show the total computation times for the solutions for each fairness level found in Phases 2 and 3: The sequential nature of Phase 2 implies that solutions for low fairness levels are found only after the solutions for the higher fairness levels are already computed. The total computation times take these additional computations into account.

	Fairness	Phase 1	Phase 2	Total Phase 2	Phase 3	Total Phase 3	Optimal
	Extreme	21	25	46	171	217	99
	High		13	59	172	231	979
1	Moderate		28	87	168	255	1102
	Low		25	112	191	303	507
	Poor		15	127	193	320	1009
	Extreme	83	77	160	132	292	479
	High		77	237	172	409	460
2	Moderate		76	313	174	487	533
	Low		77	390	163	553	721
	Poor		35	425	172	597	2769
	Extreme	0	1	1	24	25	2315
	High		1	2	24	26	1093
3	Moderate		0	2	26	28	686
	Low		0	2	25	27	50
	Poor		0	2	24	26	28
	Extreme	2	3	5	23	28	5346
	High		3	8	23	31	2498
4	Moderate		1	9	23	32	5044
	Low		0	9	25	34	1306
	Poor		0	9	25	34	629

Table 3.3: Computation times for the first four instances for five different fairness levels. For each instance and each fairness level, the computation times for each phase are shown (in seconds). Furthermore, we show the total computation time for Phase 2 and 3, since Phase 2 is solved sequentially, and hence the computation times are cumulative. Finally, the computation times of the branch-price-and-cut algorithm are shown.

Table 3.3 shows that in almost all cases the computation time of the heuristic is an order of magnitude smaller than the time necessary for the exact approach. Only for a few fairness levels, the computation time of the branch-price-and-cut algorithm is comparable with those of the heuristic. For instances 3 and 4, we observe that the heuristic only needs half a minute, wheres the exact approach can take up to more than one hour. Finally, Table 3.3 shows that the time necessary for the pairwise improvement step in Phase 2, is substantially smaller than the time necessary for

Phase 3.

For instances 5 to 10, computing the exact trade-off curve is computationally intractable. We therefore solve each instance for four fairness levels, and compare with the lower bound based on the linear relaxation of the row-based formulation. We consider the fairness levels extreme, high, moderate, and low. These fairness levels correspond to the a priori determined fairness budgets 2, 3, 5 and 10. To intuitively understand and relate the chosen fairness budgets, consider that, for groups of about 10 employees, a two unit difference in fairness budget could imply that one group is assigned one full Intercity duty less than the other groups, or has to work about half an hour longer in one row of the schedule compared to the other groups. From a practical point of view, such differences are substantial and undesirable, also because these differences persist throughout the entire year. The results are shown in Table 3.4.

	Fairness	Phase 1	Phase 2	Phase 3	Root	Gain (%)	Gap (%)
5	Extreme	2041.5	2041.5	2039.5	1542.5	0.1	24.4
	High		1780.2	1778.2	1529.0	12.9	14.0
	Moderate		1668.0	1630.5	1514.5	20.1	7.1
	Low		1546.2	1544.1	1491.8	24.4	3.4
6	Extreme	1811.7	1651.6	1650.4	1296.4	8.9	21.4
	High		1513.2	1512.0	1291.5	16.5	14.6
	Moderate		1412.4	1411.1	1282.3	22.1	9.1
	Low		1358.2	1356.9	1265.7	25.1	6.7
7	Extreme	1956.3	1559.0	1557.8	1328.4	20.4	14.7
	High		1510.4	1503.7	1318.0	23.1	12.3
	Moderate		1469.5	1465.1	1299.2	25.1	11.3
	Low		1412.1	1410.7	1262.5	27.9	10.5
8	Moderate	1502.0	1397.4	1396.9	1120.4	7.0	19.8
	Low		1292.7	1207.9	1097.8	19.6	9.1
9	Moderate	2877.0	2252.1	2249.2	1774.0	21.8	21.1
	Low		2194.0	2182.3	1768.2	24.1	19.0
10	Moderate	3009.0	1863.7	1859.2	1526.0	38.2	17.9
	Low		1759.6	1759.6	1486.5	41.5	15.5

Table 3.4: Results for instances 5 to 10. Each instance is solved for four different fairness levels: extreme, high, moderate, and low. For each instance and each fairness level, we show the objectives obtained in the different phases of the heuristic, the gain compared to the benchmark solution (i.e., Phase 1), and the gap with the root bound obtained from the row-based formulation. Omitted rows indicate that no feasible allocation could be found within the time limit of 30 minutes.

Table 3.4 shows that the heuristic increases the attractiveness greatly: For 12 out

of 18 instances, the improvement was more than 20%, and only for one instance the improvement was not substantial. The improvement for the largest two instances is especially large. Furthermore, the largest gains are obtained for the less strict fairness levels. This is intuitive, as the focus on fairness is less, and hence the loss of not taking attractiveness into account when allocating the duties over the groups will be larger. Note that a major part of the improvement is in the pairwise improvement phase, i.e., the second phase, and only a small further improvement is obtained in the third phase. For some instances, however, the third phase allows for a substantial improvement (e.g., for instance 8). The gap with the root bound indicates that the found solutions are of high quality, although in some cases a substantial gap is still present. We note, however, that a substantial integrality gap can be expected, especially for tight fairness budgets, as was also noted in Table 3.2.

The computation times for the second set of instances are shown in Table 3.5. Similar to Table 3.3, we show the computation times per fairness level and phase of the algorithm, together with the total computation times for Phases 2 and 3.

	Fairness	Phase 1	Phase 2	Total Phase 2	Phase 3	Total Phase 3
5	Extreme	167	301	468	1020	1488
	High		214	682	1079	1761
	Moderate		25	707	1232	1939
	Low		15	722	1068	1790
6	Extreme	56	201	257	731	988
	High		47	304	767	1071
	Moderate		11	315	740	1055
	Low		8	323	815	1138
7	Extreme	512	92	604	695	1299
	High		11	615	728	1343
	Moderate		30	645	871	1516
	Low		8	653	746	1399
8	Moderate	7	46	53	309	362
	Low		4	57	361	418
9	Moderate	23	328	351	4594	4945
	Low		26	377	5452	5829
10	Moderate	127	58	185	6335	6520
	Low		38	223	5546	5769

Table 3.5: Computation times for the instances 5 to 10, for the four different fairness levels. For each instance and each fairness level, the computation times for each phase are shown (in seconds). Furthermore, we show the total computation time for the solutions found in Phase 2 and 3, to account for the sequential approach used in Phase 2. Omitted rows indicate that no feasible allocation could be found within the time limit of 30 minutes.

The running times shown in Table 3.5 can be considered well-suited for practice: The second phase has a running time of a few minutes, the same order of magnitude as the first phase, and the third phase stays within two hours for all instances. Note that, for instances 8, 9, and 10, the running time for Phase 1 does not include the time for fairness levels Extreme and High, for which no solution could be found within the time limit of half an hour.

Summarizing, our experiments show that the heuristic approach complements the exact branch-and-price method developed in Chapter 2. The algorithm is able to find close-to-optimal solutions for the first set of instances, and greatly improves upon the sequential approach for the second set of instances. Furthermore, a major part of this improvement is due to the pairwise improvement in the second phase of the algorithm, which runs orders of magnitude faster than the branch-and-price approach. The third phase is more time consuming, but can realize a substantial improvement.

3.7 Conclusion

In this chapter, we proposed a heuristic method for the Cyclic Crew Rostering Problem with Fairness Requirements (CCRP-FR), a variant of the Fairness-oriented Crew Rostering Problem (FCRP), introduced in Chapter 2. In this problem, attractive rosters have to be constructed for a fixed, a priori known, set of fairness levels. The development of the heuristic solution approach is motivated by practice: The crew rostering problem is generally solved multiple times for varying parameter settings, implying that, even during the tactical planning phase, it is desirable that high quality solutions can be obtained quickly. Also, the underlying complexity of the problem implies that exact methods are incapable of coping with some of the large instances encountered in practice.

The developed three-phase heuristic combines the strengths of the exact apprach for the FCRP developed in Chapter 2 with a large-scale neighborhood search algorithm. The design of the heuristic assures that good solutions for all fairness levels are obtained quickly, and can still be further improved if additional running time is available.

We evaluated the heuristic on real-world data from NS. We showed that the heuristic finds close to optimal solutions for many of the considered instances. Furthermore,

the computation times are generally an order of magnitude smaller than the time necessary for the branch-price-and-cut approach. In particular, the computational results show that the heuristic is able to quickly find major improvements upon the sequential approach: For most instances the heuristic is able to reduce the attractiveness penalty by at least 20% in just a few minutes. Furthermore, the heuristic also provides a lower bound which gives an estimate of the solution quality. This bound indicates that the heuristic finds high-quality solutions, also for the instances for which no optimal solution is known. The running time of the heuristic can be considered well-suited for practice, even for the largest instances: The second phase has a running time of a few minutes, similar to the first phase, and the third phase stays within two hours for all instances.

Chapter 4

Analyzing a Family of Formulations for Cyclic Crew Rostering*

4.1 Introduction

The Fairness-oriented Crew Rostering Problem (FCRP) and the Cyclic Crew Rostering Problem with Fairness Requirements (CCRP-FR), introduced in Chapters 2 and 3, are both extensions of the Cyclic Crew Rostering Problem (CCRP). Hence, tight formulations for the CCRP are crucial for the development of efficient solution methods for both the FCRP and the CCRP-FR. In this chapter we provide an in-depth analysis of different formulations for the CCRP and, in doing so, we show that the row-based formulations of Chapters 2 and 3 are the best-suited formulations for the roster constraints at Netherlands Railways (NS).

The CCRP in itself is well-studied in the Operations Research literature, with most applications focusing on railway and mass transit transportation (see, e.g., Huisman et al. (2005b), Caprara et al. (2007), and Abbink et al. (2018)). By focusing on the attractiveness of the rosters, the operator tries to incorporate the demands of the

*This chapter is partially based on T. Breugem, T. Dollevoet, and D. Huisman (2018): *Analyzing a Family of Formulations for Cyclic Crew Rostering.*

employees in the planning process. The possible (practical) benefits from this are apparent from Abbink et al. (2005), which discuss how incorporating attractiveness in crew planning resolved conflicts between the labor unions and NS, and Borndörfer et al. (2017) which mentions the importance of attractive work for bus drivers at BVG (Berlin's public transport company): The bus drivers at BVG have an average age of around 50 years. The focus on attractiveness, in this case, can be a key instrument for recruiting new (younger) drivers, as the possibility for higher salaries is generally limited.

Models for the CCRP generally belong to one of three categories: generalized assignment, multi-commodity flow, and set partitioning models. Caprara et al. (1997) and Borndörfer et al. (2015) consider both a multi-commodity flow model and a set partitioning model for crew rostering. Furthermore, both argue which formulation is more suitable, given the constraint set: Caprara et al. (1997) stress that flow-based formulations are well-suited for problems where the main focus is on the follow-up of duties, whereas a set partitioning formulation is better suited for problems where the feasibility and cost depend on the overall duty sequence. Similarly, Borndörfer et al. (2015) note that the set partitioning formulation is preferred when many difficult roster constraints have to be taken into account. Sodhi and Norris (2004) and Hartog et al. (2009) propose an assignment model with side constraints. They solve the problem using a two-phase decomposition, in which first the 'skeleton' of the roster is optimized (e.g., days-off are determined), and then the duties are assigned. Xie and Suhl (2015) propose a multi-commodity flow formulation for both cyclic and acyclic crew rostering, and apply both models to practical instances from a German bus company. Finally, Mesquita et al. (2015) considers both assignment and multi-commodity flow models for the bus driver rostering problem with day-off patterns, and provide theoretical results regarding the relative strength of the models.

In this chapter, we provide an in-depth analysis of modeling techniques for the CCRP. We propose a family of formulations, and derive analytical results regarding the relative strength of the proposed formulations. The family of formulations can be seen as a generalization of the typical assignment and set partitioning formulations, and is motivated by the poor performance of assignment formulations on difficult instances. Furthermore, we discuss modeling techniques and provide tightness results for an important class of roster constraints. Finally, we adapt the branch-price-and-cut approach of Chapter 2 to the family of formulations, and show the benefit of a suitably picked formulation using practical instances from NS.

The remainder of this chapter is organized as follows. In Section 4.2, we discuss the general modeling framework for the CCRP. The family of formulations is presented in Section 4.3, and, in Section 4.4, we derive analytical results regarding the tightness of the different formulations. Finally, we discuss the branch-and-price approach in Section 4.5, and apply this approach to practical instances of NS in Section 4.6. The chapter is concluded in Section 4.7.

4.2 Modeling the Cyclic Crew Rostering Problem

In the CCRP, cyclic rosters have to be constructed for groups of employees. Each cyclic roster consists of rows (i.e., generic work weeks), columns (i.e., weekdays), and cells (i.e., the intersection of a row and a column). The roster consists of two components: the type specification of each cell (e.g., an early, late or night duty, or a day-off), known as the basic schedule, and an allocation of the duties to the cells. Similar to Chapters 2 and 3, we assume the basic schedules to be given (see, e.g., Hartog et al. (2009) for a discussion on the construction of basic schedules).

Two important aspects have to be taken into account when constructing the rosters. Firstly, the roster should be feasible with respect to the labor regulations. For example, there should be sufficient rest time between consecutive duties, and the total amount of work in a row (i.e., in a week of work) cannot be too large. Secondly, the roster should be perceived attractive by the employees. Short, although legal, rest times, for example, make the roster unattractive, as employees prefer a proper rest period between two duties. The feasibility and perceived attractiveness of a roster are expressed using *roster constraints*, which are (linear) constraints depending on the assigned duties: Feasibility (e.g., minimum rest times, maximum workload) is modeled using hard constraints, whereas attractiveness (e.g., undesirable rest times, variation of work) is modeled using soft constraints, thereby penalizing unattractive assignments of duties. Figure 4.1 shows a cyclic roster for four employees, and highlights a few roster constraints. The first roster constraint, for example, requires that a scheduled rest period (here scheduled on Wednesday), is sufficiently long. In other words, the difference between the end of duty 124 on Tuesday and the start of duty 54 on Thursday should be sufficiently large. The second constraint specifies the minimum time between consecutive duties, assuring that the crew members can have a sufficient rest. Finally, the third constraint considers the work scheduled in an entire row, and could, for example, enforce a maximum workload over a working

week.

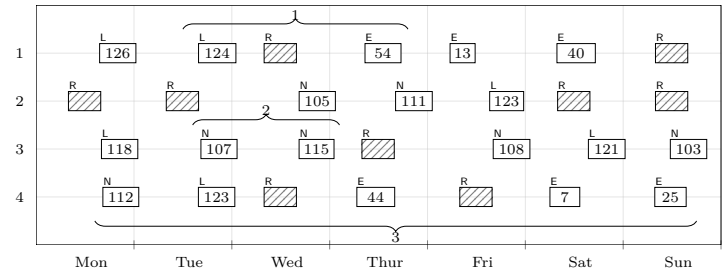


Figure 4.1: Example of different roster constraints. The first roster constraint requires that a scheduled rest period is sufficiently long and the second constraint specifies the minimum time between consecutive duties. The third constraint enforces a maximum workload over a working week.

To obtain a strong formulation for the CCRP, it is important to analyze the types of roster constraints that are present. That is, many roster constraints have a similar structure which should be taken into account when modeling the problem. In Figure 4.1, for example, the first two roster constraints can be classified as *linking constraints*, i.e., those linking exactly two cells in the basic schedule (note that the rest days are assumed fixed), whereas the third constraint can be classified as a *row-based constraint*. Given such a classification, an efficient modeling of the constraints can be determined, and a strong formulation can be obtained.

In the remainder of this section we discuss the modeling of roster constraints in detail: In Section 4.2.1, we discuss modeling techniques and tightness results for linking constraints, and, in Section 4.2.2, we propose a general framework, that allows to model many practical roster constraints, and which generalizes the framework for roster constraints posed in Chapter 2.

4.2.1 Modeling Linking Constraints

Linking constraints often occur in crew rostering problems, hence a strong formulation for such constraints can lead to major efficiency gains. In this section we analyze the polyhedron resulting from an 'isolated' linking constraint. We assume that the linking constraints are binary, i.e., the constraint is either satisfied or not, and hence that the penalty incurred is independent of the size of the violation.

Consider a linking constraint between two cells t_1 and t_2. Let D and F denote the

respective sets of feasible duties for these cells. Furthermore, let $E \subseteq D \times F$ denote the *violation set* for the linking constraint, i.e., all pairs $(d, f) \in D \times F$ such that assigning d to t_1 and f to t_2 violates the constraint. The linking constraint can be naturally modeled as a bipartite graph. Consider the seven duties depicted in Table 4.1, and suppose we require a 12 hour rest between two consecutive duties. The corresponding bipartite graph is shown in Figure 4.2. In this graph, the vertex sets represent the sets of feasible duties D and F, and the violation set is represented by the edges in the graph.

Duty	Cell	Start	End
d_1	t_1	14:00	22:10
d_2	t_1	14:30	22:00
d_3	t_1	13:00	21:00
d_4	t_1	12:00	20:00
f_1	t_2	07:00	14:30
f_2	t_2	08:00	15:30
f_3	t_2	09:15	16:00

Table 4.1: Seven possible duties for two consecutive cells.

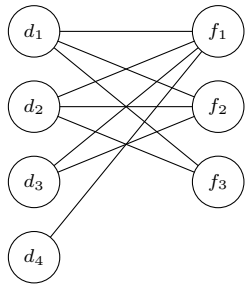


Figure 4.2: Visualization of a linking constraint between cells t_1 and t_2. The constraint is represented as a bipartite graph, where the vertices represent the duties D (left) and the duties F (right). An edge in the graph indicates that a combination of duties (as represented by the vertices) violates the linking constraint.

Let $\delta \in \mathbb{B}$ represent the violation of the linking constraint, and define the decision variables $\pi_{t_1 d}$, for $d \in D$, and $\pi_{t_2 f}$, for $f \in F$, indicating whether duty d, respectively f, is assigned to cell t_1, respectively t_2. The linking constraint is readily expressed

by the following system of equations.

$$\sum_{d \in D} \pi_{t_1 d} = 1 \tag{4.1}$$

$$\sum_{f \in F} \pi_{t_2 f} = 1 \tag{4.2}$$

$$\pi_{t_1 d} + \pi_{t_2 f} \leq 1 + \delta \qquad \forall (d, f) \in E \tag{4.3}$$

$$\delta, \pi_{t_1 d}, \pi_{t_2 f} \in \mathbb{B} \qquad \forall d \in D, f \in F. \tag{4.4}$$

Constraints (4.1) and (4.2) state that exactly one duty should be assigned to both cells and Constraints (4.3) assure that the constraint violation is modeled correctly.

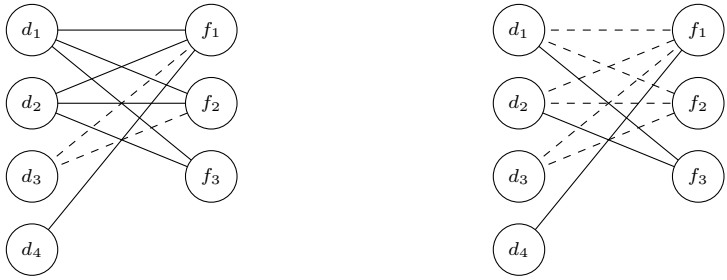
Note that (4.1)–(4.4) is a relatively weak formulation, as a single constraint is introduced for each possible violation. To obtain a stronger formulation, these constraints can be aggregated. One way of doing this is based on the duties in D: Let $F_d \subseteq F$ denote all duties incident with d, i.e., F_d contains all $f \in F$ for which $(d, f) \in E$. Constraints (4.3) can be replaced by

$$\pi_{t_1 d} + \sum_{f \in F_d} \pi_{t_2 f} \leq 1 + \delta \qquad \forall d \in D. \tag{4.5}$$

We will refer to (4.5) as *flow-based* constraints, as each single constraint sums over the out-going arcs of a single $d \in D$ (Figure 4.3a visualizes such a constraint). This type of aggregation has been previously used in Hartog et al. (2009) and in Chapters 2 and 3. The correctness of (4.5) is readily seen, as each arc (i.e., violation) appears in exactly one constraint. That is, each combination $d \in D$ and $f \in F$ such that $(d, f) \in E$ appears in exactly one constraint, and hence is penalized if both duties are selected.

Another way of aggregating (4.3), is based on bicliques in the graph-representation. This type of modelling has been considered in Ernst et al. (2001) and Mesquita et al. (2015). To the best of our knowledge, however, no theoretical results (i.e., regarding the size and strength of these formulations) have been provided.

To formulate the clique-based constraints, we introduce the following additional notation. For a given $d \in D$, let $D_d \subseteq D$ denote all $d' \in D$ for which $F_d \subseteq F_{d'}$, i.e., D_d contains those duties in D connected with at least all duties d is connected with. Note that it always holds that $d \in D_d$. In the case of Figure 4.3, we have, for example, $D_{d_3} = \{d_1, d_2, d_3\}$, since d_1 and d_2 are also connected with f_1 and f_2, and thus, with



(a) Flow-based constraint (b) Clique-based constraint

Figure 4.3: Example strengthened linking constraints. The dashed edges indicate the variables included in the constraint (either flow- or clique-based) for duty d_3.

all neighbors of d_3. In the case of rest time constraints, D_{d_3} boils down to exactly those duties in D that end at the same time or later than d_3. The clique-based constraints now read as follows.

$$\sum_{d' \in D_d} \pi_{t_1 d'} + \sum_{f \in F_d} \pi_{t_2 f} \leq 1 + \delta \qquad \forall d \in D. \tag{4.6}$$

The clique-based constraints are illustrated in Figure 4.3b. Note that replacing (4.3) by (4.6) is allowed since, by definition of D_d and F_d, it holds that $(d, f) \in E$ for all $d' \in D_d$ and $f \in F_d$. Furthermore, every violation appears in at least one constraint, since $d \in D_d$.

By definition, the clique-based constraints are at least as strong as the flow-based constraints. Assuming the violation set is of a particular structure, it can even be shown that the clique-based constraints give rise to a polyhedron with the integrality property.

Theorem 4.2.1. *Assume the violation set E is such that, for every $d, d' \in D$, either $F_d \subseteq F_{d'}$ or $F_d \supseteq F_{d'}$. Then, the coefficient matrix obtained from* (4.1), (4.2), *and* (4.6) *is totally unimodular.*

Proof. We prove total unimodularity by showing that the coefficient matrix (up to multiplication of columns with -1) has the consecutive ones property: The rows can be permuted such that each column consists of a sequence of consecutive ones, and zeros otherwise. Note that the assumption implies that we can construct an order $d_1, \ldots, d_n$ of the duties in D such that $F_{d_i} \supseteq F_{d_j}$ whenever $i < j$. Note that, by

definition, this implies that $D_{d_i} \subseteq D_{d_j}$ whenever $i < j$.

We now arrange the rows as follows: the first row corresponds to (4.2), the $(i+1)$-th row corresponds to (4.6) for d_i, and the last row corresponds to (4.1). Figure 4.4 shows the resulting coefficient matrix for the graph shown in Figure 4.2.

$$\begin{array}{cccccccc} d_1 & d_2 & d_3 & d_4 & f_1 & f_2 & f_3 & \delta & \\ \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right] & \begin{array}{c} F \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ D \end{array} \end{array}$$

Figure 4.4: The rearranged coefficient matrix resulting from the clique-based constraints for the graph shown in Figure 4.2. For each column the corresponding variable is indicated. Furthermore, for each row the constraint is specified: The rows specify the assignment constraints for D and F, respectively, and the clique-based constraint for each d_i.

It is readily seen that the resulting coefficient matrix satisfies the consecutive ones property. To see this, we note that for any $d \in D$, if $d \in D_{d_i}$, then $d \in D_{d_j}$ with $j > i$. Similarly, for $f \in F$, if $f \in F_{d_i}$, then $f \in F_{d_j}$ with $j < i$. The result follows. □

The assumption underlying Theorem 4.2.1 holds for many practical linking constraints, such as rest time and rest day constraints. Note that these constraints depend solely on the start and end time of the duties, hence we have $F_d \subseteq F_{d'}$ whenever d ends before d', and vice versa. For these constraints, the clique-based constraints provide a strong formulation, and improve over the flow-based constraints.

4.2.2 General Modeling Framework

We now discuss a general modeling framework for roster constraints, which is a generalization of the framework posed in Chapters 2 and 3. This generalization is necessary to accommodate the biclique-constraints discussed in Section 4.2.1.

Let D denote the set of duties, T the set of cells, and let D_t denote the duties that can be assigned to cell $t \in T$. Let Q denote the set of roster constraints. Each roster constraint q is modeled using a set of linear constraints $p \in P_q$ (this generalizes the framework posed in Chapters 2 and 3, where each roster constraint is assumed to be

modeled using a single linear constraint). Each linear constraint $p \in P_q$ is specified by a coefficient for each assignment of a duty to a cell in the basic schedule, and a scalar called the *threshold value*. The coefficient for the assignment (t, d) for linear constraint p is denoted by f^p_{td} and the threshold value for p is denoted by b_p.

Let δ_q denote the violation of roster constraint q, and let c_q be the corresponding penalty variable. The roster constraints enforce that *if* the sum of coefficients of assigned duties exceeds the threshold value for one of the linear constraints, *then* the difference between the sum and the threshold value is penalized *and* lies within the violation interval, given by $\Delta_q = [0, u_q]$. In other words, the roster constraint is modeled by enforcing each linear constraint $p \in P_q$:

$$\sum_{t \in T} \sum_{d \in D_t} f^p_{td} \pi_{td} \leq b_p + \delta_q, \tag{4.7}$$

and assuring $\delta_q \in \Delta_q$. Note that (4.7) assures that δ_q is equal tot the maximum violation, calculated over all $p \in P_q$. It is readily seen that both the flow- and clique-based linking constraints fit this framework, and that also many other constraints, such as workload and variation constraints, can be modeled in this fashion.

4.3 Family of Mathematical Formulations

In this section we propose a family of mathematical formulations for the CCRP. In Section 4.3.1, we define the concept of clusters and roster sequences, and we conclude with a family of mathematical formulations for the CCRP in Section 4.3.2.

4.3.1 Clusters and Roster Sequences

The family of formulations is based on different partitions of the basic schedule. That is, we develop a mathematical formulation under the assumption that each basic schedule is partitioned into disjoint subsets, which we call *clusters*. This partition will be referred to as a *clustering* for the respective basic schedule, and should be picked a priori solving the model.

The formulation will have a different structure for each possible clustering, giving rise to the family of formulations. Figure 4.5 gives an example of two possible clusterings for a basic schedule of four rows. In the cell clustering each cluster contains exactly

one of the cells in the basic schedule. The row clustering, on the other hand, assigns all cells in the same row (i.e., Monday to Sunday) to the same cluster. Note that many more clusterings are possible. One could, for example, also consider a 'weekend' clustering, in which each cluster relates to either Friday to Monday (the 'weekend' days), or Tuesday to Thursday (the 'week' days). Such a clustering can be a good choice when, e.g., the rest time over the weekend is of utmost importance. Generally, cells in a cluster do not need to be consecutive.

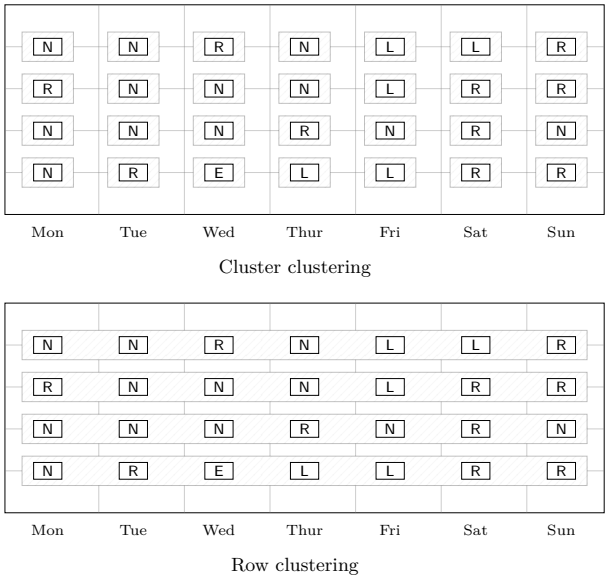


Figure 4.5: Example of different clusterings. Each highlighted area represents a cluster.

Each cluster is assigned a number of duties simultaneously. Each possible assignment of duties to a cluster is called a *roster sequence*. Formally, a roster sequence specifies a duty or rest day for each cell in the cluster, such that the assignment is compatible with the basic schedule, and such that no duty is assigned twice (within the same cluster).

To illustrate the use of roster sequences, consider the cluster depicted in Figure 4.6, together with two possible roster sequences. Note that the roster sequences contain different duties (indicated by the numbers). In this case the second roster sequence has a shorter rest period than the first roster sequence (as duty 124 ends later than duty 126, and duty 58 starts earlier than duty 54), which might be considered un-

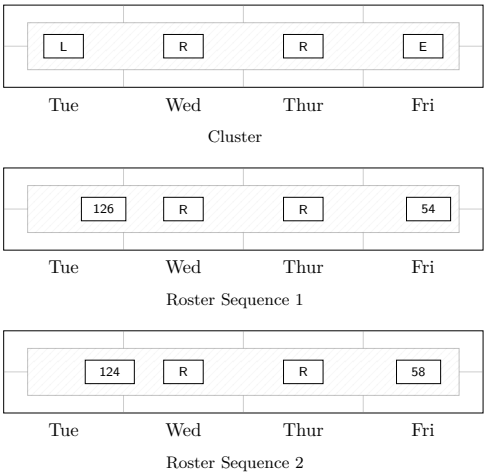


Figure 4.6: Cluster from Tuesday to Friday, together with two possible roster sequences.

desirable.

The goal of a clustering is to model constraints implicitly using the roster sequences. That is, ideally each constraint considers the cells in *solely* one of the clusters, and can therefore be taken care of when generating the roster sequences. As an example, consider a constraint in which an employee can have only a maximum amount of work per row. In this case, the row clustering of Figure 4.5 allows to model these constraints implicitly using the roster sequences (i.e., a roster sequence is feasible only if it does not exceed the maximum working time). On the other hand, for the cell clustering these constraints have to be modeled explicitly in the mathematical formulation.

4.3.2 Mathematical Formulation

We are now ready to formalize the family of formulations. The set K denotes the set of all clusters (note that these are determined a priori formulating the mathematical model). We define the set S_k as the set of all roster sequences for cluster $k \in K$. Each roster sequence can be seen as a sequence of assignments (t, d). The parameter h^k_{ds} indicates whether roster sequence $s \in S_k$ contains duty d (i.e., duty d appears in one of the assignments describing the roster sequence s). Finally, we define c^k_s as the penalty associated with roster sequence $s \in S_k$ for cluster $k \in K$.

Let $Q_k \subseteq Q$ denote the set of roster constraints fully contained in cluster $k \in K$, and define $Q_K = \bigcup_{k \in K} Q_k$. The constraints in Q_K are exactly those that are modeled implicitly using the roster sequences. The penalty c_s^k associated with roster sequence $s \in S_k$ is the sum of all violations in the roster sequence s, restricted to the roster constraints Q_k. Note that the roster constraints in $Q \setminus Q_K$ need to be modeled explicitly.

The CCRP, given a clustering K, can now be formulated as follows. We introduce the following decision variables.

- x_s^k, for all $k \in K$ and $s \in S_k$. The binary variable x_s^k indicates whether roster sequence $s \in S_k$ is assigned to cluster $k \in K$.

- δ_q, for each $q \in Q \setminus Q_K$. The variable $\delta_q \in \Delta_q$, expresses the violation of the roster constraint $q \in Q \setminus Q_K$.

The formulation now reads as follows.

$$\min \quad \sum_{k \in K} \sum_{s \in S_k} c_s^k x_s^k + \sum_{q \in Q \setminus Q_K} c_q \delta_q \tag{4.8}$$

$$\text{s.t.} \quad \sum_{s \in S_k} x_s^k = 1 \qquad \forall k \in K \tag{4.9}$$

$$\sum_{k \in K} \sum_{s \in S_k} h_{ds}^k x_s^k = 1 \qquad \forall d \in D \tag{4.10}$$

$$\sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f_{td}^p x_s^k \leq b_p + \delta_q \qquad \forall q \in Q \setminus Q_K, p \in P_q \tag{4.11}$$

$$x_s^k \in \mathbb{B} \qquad \forall k \in K, s \in S_k \tag{4.12}$$

$$\delta_q \in \Delta_q \qquad \forall q \in Q \setminus Q_K. \tag{4.13}$$

The Objective (4.8) expresses that we minimize the sum of the roster sequence costs, together with the cost of all explicitly modeled roster constraints. Constraints (4.9) and (4.10) assure that the duties are assigned correctly to the basic schedules. That is, each cluster is assigned exactly one roster sequence, and each duty is assigned exactly once to a cell in the basic schedule. Constraints (4.11) represent the roster constraints that are modeled explicitly. Finally, Constraints (4.12) and (4.13) specify the domains of the decision variables. The family of formulations for the CCRP is now obtained by taking (4.8)–(4.13) for all possible clusterings K.

4.4 Theoretical Comparison Clusterings

Intuitively, the implicit modeling of the roster constraint violations leads to a tighter linear relaxation. In this section, we prove this rigorously. From hereon, we consider two clusterings K and L such that $Q_K \supseteq Q_L$. An example of such clusterings is given in Figure 4.5, where K and L are the row and cell clustering, respectively. Let S_k, for all $k \in K$, and G_ℓ, for all $\ell \in L$, denote the respective sets of roster sequences for both clusters. For notational convenience, define Ω as the set of all feasible assignments (t, d), with $t \in T$ and $d \in D$, of duties to the cells in the basic schedule. Furthermore, we define the operator $[\cdot]^+$ as $[a]^+ = \max\{0, a\}$. Throughout this section, a solution refers to a solution to the linear relaxation.

We first prove the following lemma. Intuitively, this lemma states that, given a solution for K, we can construct a solution for L such that each duty is assigned to the same cell in both solutions.

Lemma 4.4.1. *Let $\bar{x}$ be a solution for clustering K. There exists a feasible solution $\bar{z}$ for clustering L, such that*

$$\sum_{k \in K} \sum_{\substack{s \in S_k: \\ s \ni (t,d)}} \bar{x}_s^k = \sum_{\ell \in L} \sum_{\substack{q \in G_\ell: \\ q \ni (t,d)}} \bar{z}_g^\ell \tag{4.14}$$

for each $(t, d) \in \Omega$.

Proof. We consider an auxiliary clustering O, defined as the largest clustering which is fully contained in both K and L (see Figure 4.7). Formally, O is uniquely defined by taking all non-empty subsets $k \cap \ell$, for all $k \in K$ and $\ell \in L$. Let R denote the set of feasible roster sequences for this clustering, and let R_o denote the feasible roster sequences for $o \in O$.

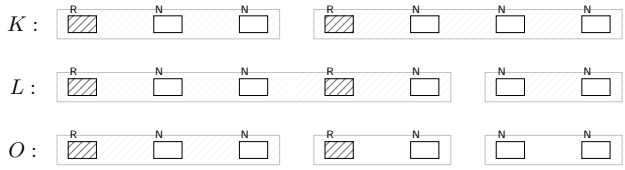


Figure 4.7: Example of the clustering O, which is the largest clustering contained in both K and L.

Since each cluster $o \in O$ is fully contained in some $k \in K$, we can readily obtain a solution $\bar{y}$ for O satisfying

$$\sum_{k \in K} \sum_{\substack{s \in S_k: \\ s \ni (t,d)}} \bar{x}_s^k = \sum_{o \in O} \sum_{\substack{r \in R_o: \\ r \ni (t,d)}} \bar{y}_r^o \tag{4.15}$$

by splitting up each roster sequence for K into smaller roster sequences for O. Furthermore, since each $o \in O$ is also fully contained in some $\ell \in L$, we can obtain a solution $\bar{z}$ satisfying

$$\sum_{\ell \in L} \sum_{\substack{q \in G_\ell: \\ g \ni (t,d)}} \bar{z}_g^\ell = \sum_{o \in O} \sum_{\substack{r \in R_o: \\ r \ni (t,d)}} \bar{y}_r^o \tag{4.16}$$

by greedily constructing roster sequences for L given those for O. To be more precise, let $O_\ell \subseteq O$ denote the clusters contained in $\ell \in L$. For each $\ell \in L$, we pick the roster sequence r with smallest non-zero value $\bar{y}_r^o$, say v, over all clusters in O_ℓ. This roster sequence is then combined with a roster sequence for each of the other clusters in O_ℓ, to obtain a roster sequence g for cluster ℓ. We set $\bar{z}_g^\ell = v$, reduce $\bar{y}_r^o$ for all involved roster sequences by v, and repeat the procedure until all roster sequences are assigned.

It follows that we can construct a solution $\bar{z}$ that satisfies (4.14). It remains to show that a solution constructed in this fashion is feasible with respect to the roster constraints. By combining (4.11) and (4.14), and doing some algebraic manipulations, it can be shown that $\bar{z}$ is feasible with respect to $Q \setminus Q_L$ (see Appendix 4.A).

To show that $\bar{z}$ is feasible with respect to the roster constraints in Q_L we make the following crucial observation: Since $Q_K \supseteq Q_L$ it must hold that $Q_O \supseteq Q_L$, and hence $Q_O = Q_L$. Suppose that this would not be true, then there must be a roster constraint $q \in Q_L$ and linear constraint $p \in P_q$ with non-zero coefficient f_{td}^p for multiple clusters in O. By definition of O, however, this would imply that $Q_L \setminus Q_K \neq \emptyset$, as O is the largest clustering contained in both K and L. This contradicts the assumption that $Q_K \supseteq Q_L$. Hence, if the constructed solution $\bar{y}$ is feasible with respect to Q_O, then a solution $\bar{z}$ created by combining these roster sequences must be feasible with respect to Q_L. The feasibility of $\bar{y}$ with respect to Q_O, however, follows directly from the feasibility of $\bar{x}$, since $Q_O \subseteq Q_K$. This concludes the proof. □

It is important to note that Lemma 4.4.1 does not hold in the other direction. That

is, given a solution $\bar{z}$ it is not always possible to construct a solution $\bar{x}$ satisfying (4.14). We are now able to prove the following theorem.

Theorem 4.4.1. *Let K and L be two clusterings such that $Q_K \supseteq Q_L$. Furthermore, let v_K denote the optimal value of the* LP *relaxation using clustering K, and define v_L similarly. Let $\bar{x}$ be an optimal solution corresponding to v_K. It holds that*

$$v_K \geq v_L + \sum_{q \in Q_K \setminus Q_L} c_q \phi_q(\bar{x}),$$

where the non-negative coefficients $\phi_q(\bar{x})$ are given by

$$\phi_q(\bar{x}) = \sum_{k \in K} \sum_{s \in S_k} \bar{x}_s^k \cdot \max_{p \in P_q} \left[\sum_{(t,d) \in s} f_{td}^p - b_p \right]^+ - \max_{p \in P_q} \left[\sum_{k \in K} \sum_{s \in S_k} \bar{x}_s^k \left(\sum_{(t,d) \in s} f_{td}^p \right) - b_p \right]^+ .$$

Proof. Let $\bar{z}$ be a feasible solution for clustering L satisfying (4.14), obtained using the construction heuristic described in the proof of Lemma 4.4.1. Note that $\bar{z}$ is feasible for L and hence the objective value of $\bar{z}$ is an upper bound for v_L. Furthermore, note that, by the construction of $\bar{z}$, the cost incurred for the roster constraints Q_L is identical for $\bar{x}$ and $\bar{z}$. As a consequence, the difference in objective value between $\bar{x}$ and $\bar{z}$ is exactly the penalty incurred by the roster constraints in $Q_K \setminus Q_L$. Hence, the difference in the penalty incurred by these constraints is a lower bound on $v_K - v_L$.

First, consider the solution $\bar{x}$. Recall that the constraint violations for each pattern $q \in Q_K \setminus Q_L$ are modeled implicitly in the roster sequence cost for clustering K. The penalty incurred from roster constraint $q \in Q_K \setminus Q_L$ is therefore given by

$$c_q \sum_{k \in K} \sum_{s \in S_k} \bar{x}_s^k \cdot \max_{p \in P_q} \left[\sum_{(t,d) \in s} f_{td}^p - b_p \right]^+ .$$

Next, consider the solution $\bar{z}$. Note that for L the constraint violations for all $q \in Q_K \setminus Q_L$ are modeled explicitly using (4.11). Hence, the penalty incurred from roster constraint $q \in Q_K \setminus Q_L$ is given by

$$c_q \max_{p \in P_q} \left[\sum_{\ell \in L} \sum_{g \in G_\ell} \bar{z}_g^\ell \left(\sum_{(t,d) \in g} f_{td}^p \right) - b_p \right]^+ .$$

Using that

$$\sum_{\ell \in L} \sum_{g \in G_\ell} \sum_{(t,d) \in g} f_{td}^p \bar{z}_g^\ell = \sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f_{td}^p \bar{x}_s^k,$$

(see (4.17) in Appendix 4.A), it follows that the difference in incurred penalty is given by

$$c_q \sum_{k \in K} \sum_{s \in S_k} \bar{x}_s^k \cdot \max_{p \in P_q} \left[\sum_{(t,d) \in s} f_{td}^p - b_p \right]^+ - c_q \max_{p \in P_q} \left[\sum_{k \in K} \sum_{s \in S_k} \bar{x}_s^k \left(\sum_{(t,d) \in s} f_{td}^p \right) - b_p \right]^+ .$$

The result now follows from summing over all $q \in Q_K \setminus Q_L$. □

The value $\phi_q(\bar{x})$ represents the error incurred from modeling roster constraint q explicitly (note that $\phi_q(\bar{x})$ is zero if $\bar{x}$ is integer), opposed to modeling it implicitly (i.e., correctly). Theorem 4.4.1 can be used as a guideline to pick the 'ideal' set of clusters. In particular, the proof of Theorem 4.4.1 leads to the following two key insights. First, it shows that switching from clustering L to K with K coarser than L, i.e, every $\ell \in L$ is a subset of some $k \in K$, but $P_K = P_L$ is never beneficial, i.e., will not increase the LP bound. This implies that the roster constraints should be explicitly considered when enlarging the cluster size. Secondly, it shows that switching from L to K is likely to be beneficial whenever $P_K \setminus P_L$ contains 'weak' roster constraints, where the weakness is represented by the value of $c_q \phi_q(\bar{x})$. Note that, although this value is not known a priori, it is often possible to estimate these values based on, e.g., passed experience or expert knowledge.

The above insights are illustrated in the following example. Consider the three clusterings depicted in Figure 4.8, each partitioning the cells of the basic schedule differently. The first and second clustering are similar to the clusterings of Figure 4.5, and the third clustering partitions the cells based on the rest days, i.e., a new cluster starts after each rest period. Note that for this clustering the last cell of the second row is part of the first cluster. Consider a roster constraint with regard to the duty length, and assume that all duties have a length of 8 hours, except for one of the Monday duties, which has a length of 9.5 hours. We consider two cases. In the first case, the constraint specifies that the average duty length over the week is penalized whenever it exceeds 8 hours and 10 minutes. Note that in this case it is optimal to assign the long duty to the first row, and incur a penalty of 8 minutes (since the average duty length in the first row will be 8 hours and 18 minutes). Consider

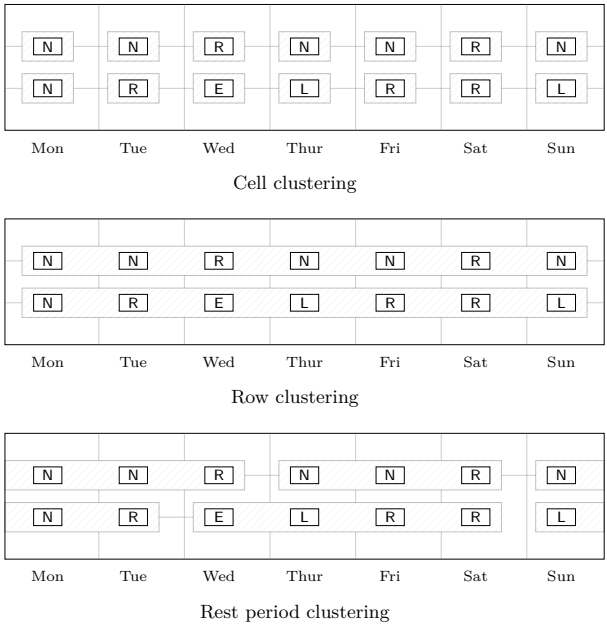


Figure 4.8: Three possible clusterings for a basic schedule of two rows. Depending on the roster constraints, one of the clusterings is preferred over the others.

now the LP relaxation of the three clusterings. For the row-based clustering this constraint is implicit, and hence it is readily seen that the LP value will equal the optimal solution. For the two other formulations, however, the constraint is explicit, and by assigning the long duty to the first row with value 5/9 and to the second row with value 4/9 a penalty can be avoided. In this case the strength of the row-based formulation is expected given the second insight, and the lack of gain in LP bound between the cell and rest period formulation is evident from the first insight.

The second case considers the same set-up, but the roster constraint now specifies that the average duty length between rest periods (instead of the rows) is penalized whenever it exceeds 8 hours and 20 minutes. Note that in this case it is still optimal to assign the long duty to the first row, and incur a penalty of 10 minutes (since the average duty length over the first three duties will be 8 hours and 30 minutes). Consider now the LP relaxation of the three clusterings. For the rest period clustering this constraint is implicit, and hence it is readily seen that the LP value will equal the optimal solution. For the two other formulations, however, the constraint is explicit, and by assigning the long duty to the first row with value 6/9 and to the second row

with value 3/9 a penalty can be avoided. In this case the strength of the formulation resulting from the rest period clustering is expected given the second insight, and the lack of gain in LP bound between the cell and row-based formulation is evident from the first insight.

4.5 Branch-and-Price Framework

We adapt the branch-price-and-cut framework of Chapter 2 to develop a branch-and-price approach for the CCRP formulation (4.8)–(4.13). We purposely omit the 'cuts', i.e., the valid inequalities proposed in Chapter 2, as, depending on the clustering, obtaining these cuts can be as difficult as the original problem.

The master problem is obtained from (4.8)–(4.13) by relaxing the integrality constraints on the x_s^k variables. The reduced cost γ_s^k of a roster sequence $s \in S_k$, for a given cluster $k \in K$ can be expressed as follows. Let μ_k denote the dual variables corresponding to (4.9), ϕ_d those corresponding to (4.10), and θ_{qp} those corresponding to (4.11). The reduced cost γ_s^k can now be expressed as

$$\gamma_s^k = c_s^k - \mu_k - \sum_{d \in D} h_{ds}^k \phi_d - \sum_{q \in Q \setminus Q_K} \sum_{p \in P_q} \sum_{(t,d) \in s} f_{td}^p \theta_{qp}.$$

The pricing problem for each $k \in K$ can be modeled as discussed in Chapter 2: For each $k \in K$, the pricing problem can be modeled as a resource constrained shortest path problem (RCSPP) with surplus variables on a directed layered graph $G_k = (V_k, A_k)$, where each vertex corresponds to an assignment (t, d) of a duty to a cell in k and each arc corresponds to a feasible follow-up of two assignments. Note that the implicit roster constraint penalties have to be taken into account slightly different compared to Chapter 2: for each $q \in Q_K$, we take the maximum over all $p \in P_q$.

Similar to Chapter 2, we first branch on the assignment of the duties to the roster groups. If this is not possible, we branch based on the assignments of the duties to the cells. That is, we branch if some duty is assigned to multiple cells in the basic schedule. By first branching on the roster groups, we obtain a more balanced branching tree. Whenever multiple branching decisions exist, we branch on the one with the most fractional value. Note that whenever no branching decisions are present, each duty is assigned to exactly one cell, and hence an integer solution is

found.

We consider two different node selection strategies. The first node selection strategy is based on the lowest bound, to improve the best-known (i.e., highest) lower bound as quickly as possible. For large instances concerning possibly many roster groups this strategy might not be well-suited as it may not lead to an integer solution in reasonable time. We therefore also consider a (heuristic) branching procedure in which we aim at quickly diving towards a good feasible solution by fixing the assignments of all duties to roster groups that exceed a certain threshold value.

4.6 Computational Experiments

In this section we discuss the computational results. We first discuss the experimental set-up in Section 4.6.1. That is, we discuss the roster constraints that are taken into account, and the different instances considered. We then present the computational results in Section 4.6.2.

4.6.1 Experimental Set-Up

We apply our solution approach to different instances based on data from NS. For each instance, the basic schedule specifies the days off. Furthermore, for each duty that is to be scheduled a type is given. The considered types are Early, Late, and Night. The type of each duty is based on the start time of the duty. The following roster constraints are taking into account.

- *Rest Time.* After completing a duty it is required that an employee has a certain minimum time to rest. After a night duty this rest time should be at least 14 hours, otherwise it should be at least 12 hours. Furthermore, we penalize rest times shorter than 16 hours with a penalty of 30.
- *Rest Day.* When rest days are scheduled in the roster, the length of the rest period has to be sufficient. This implies that there is a minimal time enforced between duties scheduled before and after the rest days. The enforced rest time is 6 hours plus 24 hours for each rest day.
- *Red Weekend.* At least once every three rows of the roster there should be a weekend which has a consecutive period of 60 hours off. These so-called red

weekends can be determined given the basic schedule. The 60 hour rest period can then be enforced using the roster constraints.

- *Workload.* The total workload in a row is not allowed to exceed 45 hours. Here, the workload of a duty is the difference between the start and end time (i.e., including the meal break).

- *Variation.* The variation constraints assure that the different attributes of work (e.g., duty length, percentage double decker work) are divided equally over the rows. These constraints penalize a positive deviation from the average (measured over all duties) for each row in the roster. In total we consider 10 different variation constraints.

We consider a total of 10 different instances: four 'small' instances of 12 employees and roughly 50 duties, four 'medium' instances of 24 employees and roughly 100 duties, and two 'large' instances of about 50 employees and 200 duties. Each of the instances is obtained by combining multiple roster groups as operated at NS. The properties of the instances are summarized in Table 4.2.

	Groups	Employees	Early	Late	Night	Total
1	1	12	23	11	15	49
2	1	12	21	12	16	49
3	1	12	49	0	1	50
4	1	12	49	0	1	50
5	2	24	21	36	35	92
6	2	24	23	35	37	95
7	2	26	101	0	2	103
8	2	24	97	0	2	99
9	4	54	118	47	52	217
10	4	50	198	0	4	202

Table 4.2: Characteristics of the instances. For each instance the number of groups and number of employees (i.e., the number of rows) is specified, along with the number of Early, Late, and Night duties, and the total number of duties.

The instances can be categorized into one of two categories. The instances 1, 2, 5, 6, and 9 represent instances in which all three duty types have to be scheduled. For the other instances the duties consist almost exclusively of early duties. The former category of instances provide more structure compared to the latter ones, since (i) less roster sequences are possible (as the duties are divided over different types), and

(ii) the rest time and rest day constraints are expected to be more important for these instances (i.e., if all duties start early, the chance of having a rest time violation is small). The second category is therefore expected to be more difficult to solve if the formulation is not chosen carefully.

4.6.2 Computational Results

In this section the computational results are discussed in detail. We first compare the performance of different clusterings and evaluate the modeling of linking constraints, and we conclude by comparing the Branch-and-Price approach with a commercial solver. All experiments are done on a computer with a 1.6 GHz Intel Core i5 processor. We use the LP solver embedded in CPLEX 12.7.1 (simply referred to as CPLEX from hereon) to solve the master problems.

Comparison Root Bounds

To illustrate the effect of different clusterings (for the given constraints) and the modeling of linking constraints, we solve the root node relaxation for four clusterings and both the flow- and clique-based linking constraints. We consider clusterings where each cluster contains a single cell, three cells, six cells, and seven cells (i.e., a cluster per row). We denote these clusterings by C_1, C_3, C_6, and C_7, respectively. Each clustering leads to a different formulation. In particular, the clustering C_1 results in the assignment formulation proposed in Hartog et al. (2009), and the clustering C_7 leads to the row-based formulation used in Chapters 2 and 3.

Table 4.3 shows for each clustering and each instance, the root bound for the flow- and clique-based constraints, together with the percentage of constraints that can be modeled implicitly (the non-zero percentage for C_1 and instance 6 is due to one row in which only one duty has to be assigned). The results in Table 4.3 are in line with Theorem 4.4.1. That is, there is a clear relation between the percentage of implicit constraints and the bound obtained from the linear relaxation. The benefit of a suitable clustering is most apparent for the instances with mostly early duties (i.e., instances 3, 4, 7, 8, and 10). For these instances the main challenge is to capture the cost incurred from the variation constraints, which only clustering C_7 is able to do. Furthermore, we see that the clique-based linking constraints improve the root bound substantially for the mixed instances (i.e., those with relative many rest time

		1	2	3	4	5	6	7	8	9	10
	Flow	558.0	654.4	192.4	274.0	671.3	618.3	181.0	250.4	916.9	221.4
C_1	Clique	570.1	681.8	192.8	286.5	833.1	843.1	302.8	345.9	1213.2	330.5
	Impl. (%)	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0
	Flow	569.3	660.9	192.5	289.5	762.6	800.5	196.7	297.9	1103.7	251.9
C_3	Clique	571.4	687.4	192.8	299.6	850.0	889.8	309.6	370.6	1245.8	351.4
	Impl. (%)	29.1	20.2	31.6	38.2	37.7	39.7	42.4	50.7	48.6	53.6
	Flow	581.3	705.5	206.4	313.5	834.1	825.8	256.9	340.4	1192.6	301.5
C_6	Clique	584.1	705.8	206.4	318.0	875.3	914.8	335.7	396.9	1278.6	390.6
	Impl. (%)	49.3	59.0	49.3	59.4	61.3	56.0	64.5	68.0	67.6	73.4
	Flow	609.8	713.8	280.0	370.2	873.4	943.2	447.8	523.4	1312.7	598.0
C_7	Clique	609.8	713.8	280.0	370.2	942.7	1004.0	447.8	523.4	1435.0	598.0
	Impl. (%)	98.0	94.7	94.5	98.6	92.3	93.2	91.4	94.6	93.8	92.0

Table 4.3: Comparison of different clusterings and the modeling of linking constraints. For each clustering and each instance, the root bound for the flow- and clique-based constraints are shown, together with the percentage of constraints that can be modeled implicitly.

violations). If we consider C_7, for example, we see that these constraints substantially improve the root bound for almost all instances with mixed duty types, namely for instances 5,6, and 9. Only for the smaller mixed instances 1 and 2 no improvement is found. Note that this improvement is expected for the mixed instances, as opposed to the non-mixed instances, where rest time violations hardly occur.

Comparison Cell- and Row-based Formulation

We compare the performance of the Branch-and-Price approach for the row-based formulation (i.e., the formulation resulting from C_7) with CPLEX for the cell-based formulation (i.e., the formulation resulting from C_1). The Branch-and-Price approach is also considered in a heuristic setting, by aggressively fixing the assignments of duties to the basic schedules (as discussed in Section 4.5). Here we use a threshold of 0.8, i.e., if a duty is assigned to a basic schedule with a value more than 0.8, this assignment is fixed in the remainder of the Branch-and-Bound tree.

The results for the different instances are shown in Table 4.4. For each method and each instance, Table 4.4 shows the root bound resulting from the used clustering, the best obtained lower bound for each method, the objective value of the best found solution, the gap with respect to the best-known lower bound, and the overall computation time (limited to at most one hour). Bold entries indicate the best found

solutions among all methods, and omitted entries imply that no solution was found within the set time limit. Note that the Branch-and-Price approach with aggressive fixing does not improve the root bound throughout the process, as no backtracking is considered for the fixed variables.

		1	2	3	4	5	6	7	8	9	10
	Root Bound	570.1	681.8	192.8	286.5	833.1	843.1	302.8	345.9	1213.2	330.5
	Best Bound	609.8	713.8	283.5	373.9	950.2	1008.4	325.4	401.7	1344.5	353.9
CPLEX	Best Solution	**609.8**	**713.8**	**283.5**	**373.9**	**950.2**	**1008.4**	503.6	545.9	1584.6	961.5
	Gap (%)	0.0	0.0	0.0	0.0	0.0	0.0	10.6	3.8	9.4	37.8
	Time (s)	1	1	2093	1519	87	93	3600	3600	3600	3600
	Root Bound	609.8	713.8	280.0	370.2	942.7	1004.0	447.8	523.4	1435.0	598.0
	Best Bound	609.8	713.8	283.5	373.9	950.2	1008.4	450.0	525.1	1436.3	598.3
B&P	Best Solution	**609.8**	**713.8**	**283.5**	**373.9**	**950.2**	**1008.4**	-	-	-	-
	Gap (%)	0.0	0.0	0.0	0.0	0.0	0.0	-	-	-	-
	Time (s)	1	1	7	13	20	23	3600	3600	3600	3600
	Root Bound	609.8	713.8	280.0	370.2	942.7	1004.0	447.8	523.4	1435.0	598.0
	Best Bound	609.8	713.8	280.0	370.2	942.7	1004.0	447.8	523.4	1435.0	598.0
B&P-FIX	Best Solution	**609.8**	**713.8**	**283.5**	**373.9**	950.9	1013.7	**465.3**	**544.2**	**1476.5**	**677.5**
	Gap (%)	0.0	0.0	0.0	0.0	0.1	0.5	3.3	3.5	2.7	11.7
	Time (s)	1	1	7	13	14	7	124	30	602	1674

Table 4.4: Comparison of CPLEX, Branch-and-Price (B&P), and Branch-and-Price with heuristic fixing (B&P-FIX). For each method and each instance, the root bound, best bound obtained after termination, the best found solution, the gap with the best known bound, and the overall computation time is shown. Each run is limited to at most one hour. Bold entries indicate the best found solutions among all methods, and omitted entries imply that no solution was found within the set time limit.

Table 4.4 shows that the Branch-and-Price approach outperforms CPLEX for the smaller instances consisting of mainly early duties. Both approaches quickly solve instances 1 and 2, but the Branch-and-Price approach is much more efficient for the difficult instances 3 and 4. This can be attributed to the strong linear relaxation obtained using clustering C_7. Both instance 5 and 6 could be solved to optimality within the set time limit using the Branch-and-Price approach and CPLEX. We see that CPLEX is about a factor four slower for these instances, but the differences do not seem substantial.

The benefit of the row-based formulation is again clearly visible when we consider the final four instances, which are considered the most difficult to solve. For these instances the (exact) branch-and-price approach stagnates, and CPLEX is not able to obtain a proper gap within the time limit. The branch-and-price approach with aggressive fixing, on the other hand, is generally able to obtain a good solution for each of the instances and substantially improves upon the solution found using

CPLEX: Both the found lower and upper bound are better than those found with CPLEX.

Summarizing, the column generation approach allows to efficiently obtain good solutions to each of the ten instances. The smaller instances can be solved to optimality using the exact branch-and-price approach, and for the larger instances a good feasible solution can be obtained using the heuristic variant.

4.7 Conclusion

In this chapter, we analyzed formulations for the Cyclic Crew Rostering problem (CCRP), in which attractive cyclic rosters have to be constructed for groups of employees. We proposed a family of formulations, motivated by the poor performance of traditional assignment models for difficult instances. Each formulation has a different structure, which implies that a suitable variant can be picked for a given problem instance. We derived analytical results regarding the relative strength of the different formulations, which can be used as a guideline to pick a suitable formulation for a given problem instance. Furthermore, we discussed modeling techniques and provided tightness results for linking constraints, a frequently occurring class of roster constraints.

We developed a branch-and-price approach to solve the CCRP, suitable for each formulation in the family. The pricing of columns in this approach is done by solving a resource constrained shortest path problem (RCSPP) with surplus variables. To cope with large instances, we proposed a heuristic branching strategy, which dives towards an integer solution by fixing the allocation of duties to the roster groups.

We applied both the exact and heuristic branch-and-price approach to practical instances from NS. Our experiments showed the importance of picking a suitable formulation for a given problem instance. In particular, we show that a suitable formulation is better able to capture the penalty incurred from the roster constraints. Furthermore, we showed that the clique-based modeling of linking constraints improves the root bound substantially.

We showed that branch-and-price approach outperforms a commercial solver using the traditional assignment model. For the small instances the exact approach is much faster in finding an optimal solution. For the large instances, the branch-and-price approach combined with the heuristic branching framework greatly outperforms the

solutions found with the assignment model, both in terms of the lower bound and the found feasible solution. The experiments also showed that the performance of the different methods depends on the structure of the problem instances, which is in line with the derived analytical results.

Appendix

4.A Remainder Proof Lemma 4.4.1

To complete the proof of Lemma 4.4.1 it remains to show that $\bar{z}$ is feasible for the roster constraints in $Q \setminus Q_L$. Consider some $q \in Q \setminus Q_L$ and fixed $p \in P_q$. Recall that u_p is the upper bound of the violation interval Δ_p. Using (4.14) we have

$$\sum_{\ell \in L} \sum_{g \in G_\ell} \sum_{(t,d) \in g} f^p_{td} \bar{z}^\ell_g = \sum_{(t,d) \in \Omega} \sum_{\ell \in L} \sum_{\substack{g \in G_\ell: \\ g \ni (t,d)}} f^p_{td} \bar{z}^\ell_g \tag{4.17a}$$

$$= \sum_{(t,d) \in \Omega} \sum_{k \in K} \sum_{\substack{s \in S_k: \\ s \ni (t,d)}} f^p_{td} \bar{x}^k_s \tag{4.17b}$$

$$= \sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f^p_{td} \bar{x}^k_s. \tag{4.17c}$$

Hence, for $q \in Q \setminus Q_K$ and $p \in P_q$, the feasibility of $\bar{x}$ implies that

$$\sum_{\ell \in L} \sum_{g \in G_\ell} \sum_{(t,d) \in g} f^p_{td} \bar{z}^\ell_g - b_p = \sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f^p_{td} \bar{x}^k_s - b_p \leq u_p, \tag{4.18a}$$

Next, consider some $q \in Q_K \setminus Q_L$ and $p \in P_q$. Since $q \in Q_K$, there is a cluster $k' \in K$ such that the coefficients f^p_{td} are non-zero only for this cluster. It follows that

$$\sum_{\ell \in L} \sum_{g \in G_\ell} \sum_{(t,d) \in g} f^p_{td} \bar{z}^\ell_g - b_p = \sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f^p_{td} \bar{x}^k_s - b_p \tag{4.19a}$$

$$= \sum_{s \in S_{k'}} \bar{x}^{k'}_s \sum_{(t,d) \in s} f^p_{td} - b_p \tag{4.19b}$$

$$= \sum_{s \in S_{k'}} \bar{x}^{k'}_s \left(\sum_{(t,d) \in s} f^p_{td} - b_p \right), \tag{4.19c}$$

where (4.19c) follows from (4.9). Using the feasibility of the roster sequence s, we have

$$\sum_{s \in S_{k'}} \bar{x}_s^{k'} \left(\sum_{(t,d) \in s} f_{td}^p - b_p \right) \leq \sum_{s \in S_{k'}} \bar{x}_s^{k'} u_p \tag{4.20a}$$

$$= u_p, \tag{4.20b}$$

were (4.20b) follows from (4.9). It follows that $\bar{z}$ is feasible for all $q \in Q_K \setminus Q_L$, and thus for all $q \in Q \setminus Q_L$.

Chapter 5

A Column Generation Approach for the Integrated Crew Re-Planning Problem

5.1 Introduction

Large maintenance and construction projects are crucial for heavily used railway networks to cope with the ever increasing demand. In 2019, for example, there were around 150 planned maintenance activities for the Dutch railway network that led to rerouting and/or the necessity of buses as alternative mode of transport*. These activities are inconvenient for the passengers and have a large impact on the crew schedules: Many crew members traverse large parts of the railway network, and hence their duties (i.e., days of work) become infeasible due to the maintenance activities. As a result, the crew schedule needs to be updated, preferably with as few modifications as possible.

The crew duties have to satisfy numerous rules, expressing, for example, a maximum on the length of the duty or the occurrence of a proper meal break. At NS, it is also required that each duty starts and ends at the same crew base. Figure 5.1 gives an example of a duty, passing the major stations The Hague (Gvc), Utrecht (Ut),

*source (in Dutch): `https://www.ns.nl/reisinformatie/werk-aan-het-spoor`.

and Zwolle (Zl). Note that the duty starts and ends at The Hague. Furthermore, a proper meal break of half an hour (indicated by a star) is specified, and the duty does not exceed 9.5 hours (which is the maximum length for duties starting after 6 in the morning).

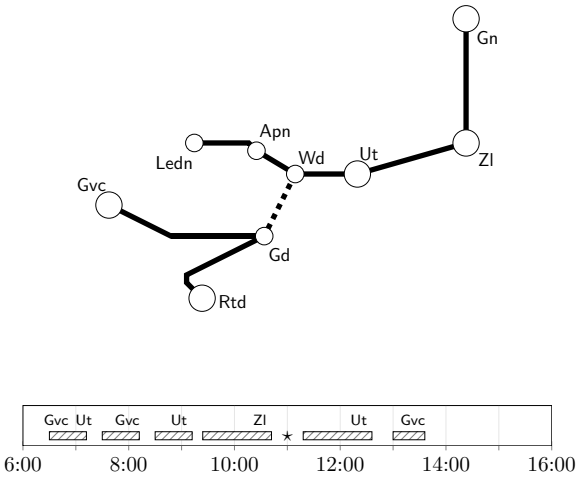


Figure 5.1: Schematic visualization of a part of the Dutch railway network and an example of a duty traversing this network for an employee based at The Hague (Gvc). Each block represents a trip. For each trip the departure station (top left) and/or arrival station (top right) are shown. The star indicates a meal break. The dashed line indicates scheduled maintenance between the two stations.

Besides rules on the duty level, the scheduled duties should also satisfy complex global rules related to the distribution of work among the crew bases. These rules, known as the 'Sharing-Sweet-and-Sour' rules (see Abbink et al. (2005)), assure that each crew base gets roughly the same type of work, i.e., the work allocation is fair. This implies, for example, that the duties for each crew base have roughly the same average length and the same percentage of work on high-quality rolling stock. These rules also include spatial variations: Trips marked as attractive are divided equally over all crew bases, to the extent to which this is possible. As a result, the 'Sharing-Sweet-and-Sour' rules 'enforce' that a crew member traverses different parts of the railway network within each duty. Hence, maintenance activities in a single part of the railway network can affect many different duties.

The impact of maintenance activities on the duties can be illustrated using the following example. Suppose that a maintenance activity is scheduled between Gouda

(Gd) and Woerden (Wd), as indicated by the dashed line in Figure 5.1. This implies that the duty shown in Figure 5.1 is no longer valid, due to the trips between The Hague and Utrecht being canceled (as a result of the maintenance activities). As a consequence, the trip between Utrecht and Zwolle, originally covered by this duty, now has to be covered by a different (or even an additional) duty.

The problem of re-scheduling the crew has been formalized as the Crew Re-Scheduling Problem (CRSP) in Huisman (2007). The CRSP differs from the traditional Crew Scheduling Problem (CSP) in the sense that the operational costs are of much less importance, a consequence of the fact that the original duties, and hence the amount of crew needed, are input to the CRSP. Instead, the CRSP aims at finding new duties for the already scheduled crew members such that all tasks are covered. If necessary, additional duties can be scheduled (i.e., an additional crew member will be asked to work) but this should be avoided whenever possible.

The newly constructed duties should assure that the roster of each employee (see Chapter 2) remains feasible, implying that additional rules, such as a maximal workload and a minimum rest time between duties, have to be taken into account. In the current practice, the feasibility of the new rosters is 'assured' by allowing the new duties to deviate only slightly from the original ones: At NS, for example, the newly constructed duties are not allowed to start more than half an hour earlier than the original duties, or end more than half an hour later. We refer to this as *day-by-day* re-scheduling, as it allows the re-scheduling problem to be solved for each day separately.

It is clear that day-by-day re-scheduling limits the possible new duties that can be assigned, and hence a possible better solution, feasible with respect to the roster rules, might not be found. In the Integrated Crew Re-Planning Problem (ICRPP) we aim at capturing exactly this flexibility in start and end times: The ICRPP considers the re-scheduling of crew for multiple days simultaneously, thereby allowing more flexibility in the re-scheduling. The ICRPP further complicates the CRSP due to (*i*) the size of the problem, as multiple days need to be re-scheduled at once, and (*ii*) the possibility of larger shifts in start and end time, implying that the feasibility of the rosters should be taken into account explicitly.

The contribution of this chapter is twofold. Firstly, we propose a mathematical formulation for the ICRPP and develop a column generation approach to solve the problem. Secondly, we apply our solution approach to practical instances from NS, and show the benefit of integrating the re-scheduling process. In particular, we show

that the additional flexibility in the start and end times of the re-scheduled duties allows for a reduction in the number of additional duties compared to the day-by-day approach.

The remainder of this chapter is organized as follows. In Section 5.2, we formalize the ICRPP and, in Section 5.3, we discuss related work. We then propose a mathematical formulation for the ICRPP in Section 5.4, and propose a column generation based solution approach in Section 5.5. In Section 5.6, we show the benefit of the newly developed solution approach using practical instances from NS, and we conclude the chapter in Section 5.7.

5.2 Problem Description

The timetable and the rolling stock schedule together specify the work for a given planning day. The scheduled work is represented by *tasks* (i.e., indivisible blocks of work). Different types of tasks exist, such as driving and passenger tasks (i.e., riding a train as a passenger), but also operational tasks such as shunting and deadheading are specified in the plan. Certain tasks must always be covered (e.g., trip tasks, shunting tasks) to assure a correct execution of the plan, whereas other tasks are optional tasks (e.g., passenger tasks, taxi trips), and are solely added to the plan for additional flexibility.

In the crew planning phase, the planned tasks are assigned to the employees in the form of *duties*, i.e., sequences of tasks. Each duty has to satisfy certain rules, relating to the transfer times between tasks, the possibility of a meal break, the duration of the duty, among other things. These constraints follow from labor laws and the collective labor agreement. Furthermore, each duty should start and end at the same crew base. The duties are assigned to the crew members in the form of *rosters*: The roster of each employee specifies which duties to perform on which day. Furthermore, the roster specifies on which days the employee has a day-off. Similar to the duties, the rosters should satisfy numerous rules as agreed upon in the collective labor agreement. Typical roster rules include a minimum rest time between duties, a maximum workload over the week, and a sufficient number of days-off. We refer to Abbink et al. (2005), Hartog et al. (2009), and the previous chapters for an overview of the duty and roster rules at NS.

In crew re-planning the *original* duties and rosters are replaced by *alternative* duties

and rosters, such that all tasks are covered in the disrupted situation. This implies that, after re-planning, each employee is assigned an alternative roster consisting of alternative duties. The alternative duties and rosters have to satisfy a number of rules, yet the rules in re-planning are generally less stringent than the rules for the initial planning phase. The 'Sharing-Sweet-and-Sour' rules, for example, are not explicitly taken into account in the re-planning process. The rules for the alternative duties are as follows.

- *Connection Times.* Between every two scheduled tasks there should be a sufficient connection time. This connection time depends on whether or not both tasks are on the same rolling stock unit. Furthermore, possible travel time (e.g., due to a passenger task) should be taken into account.
- *Duty Length.* A duty is not allowed to exceed a certain length. This maximum length depends on the start and end time of the duty. The duty length rules are specified by a start and/or end interval, and a maximum length for duties within these intervals.
- *Meal Break.* Each duty should allow for a proper meal break after a given amount of time. The meal break should always take place at one of the dedicated train stations, generally the larger stations containing a canteen.
- *Route and Rolling Stock Knowledge.* A crew member is only allowed to perform a certain trip if he or she has sufficient route knowledge. This knowledge is generally regional, i.e., it is assumed to be equal for all crew belonging to the same crew base.

Furthermore, the alternative rosters have to satisfy the following roster rules.

- *Rest Time.* After completing a duty it is required that an employee has a certain minimum time to rest. This implies that there should be a minimum amount of time between the end of a duty and the start of the duty on the next day.
- *Workload.* The total workload of a roster is not allowed to exceed a given maximum value. This maximum value depends on the work scheduled in the original roster: Ideally, the workload in the alternative roster should not deviate too much from the workload in the original roster.

Finally, each alternative duty has an associated cost. This cost can be decomposed into two parts: a fixed cost and a cost based on each connection (i.e., follow-up of

two tasks). The fixed costs regulate the trade-off between different types of duties. Scheduling an additional alternative duty (i.e., an additional crew member has to perform it) is associated with a high fixed cost, as such a solution is costly and should be avoided. On the other hand, it is desirable to give some employees a day-off if possible, as the maintenance activities often mean that less tasks have to be covered. Hence, empty duties (i.e., duties without any tasks) are associated with a low fixed cost. Furthermore, there are costs associated with the connections, related to, for example, the actual cost of the connection (e.g., the cost of a taxi trip).

The alternative rosters are linked to the original rosters by means of *time windows*, specifying for each alternative duty an interval in which the start and end time of the duty must lie. The main purpose of the time windows is to assure that the newly constructed rosters remain feasible with the work scheduled outside of the re-planning period. At NS, for example, during re-planning a shift of at most half an hour in start and end time is always considered feasible with respect to the roster rules. As a result, the start and the end of the alternative roster can be at most half an hour earlier (respectively, later) than the original roster. Within each roster, however, the time windows can be loosened, by taking the roster rules into account explicitly. This is illustrated in Table 5.1, where we compare the time windows for the day-by-day approach with a fully integrated approach (i.e., allowing arbitrary shifts in start and end times). By varying the maximum allowed shift, one can analyze the trade-off between the number of necessary duties and the deviation from the original schedule.

	Original	Day-By-Day	Integrated
Day 1	12:30 - 21:00	12:00 - 21:30	12:00 - 08:30
Day 2	11:15 - 20:30	10:45 - 21:00	04:00 - 21:00

Table 5.1: Allowed start and end times for a re-scheduling period of two days. At NS, duties start after 04:00 and end the latest at 08:30 the next day. The column Original shows the originally planned start and end time of each duty, i.e., the interval in which the duty is scheduled. The column Day-by-Day shows the resulting interval according to the day-by-day approach, i.e., the interval is extended by half an hour on both sides. The column Integrated shows the possible additional flexibility: The first duty is allowed to end at any time before 08:30 the next day, and, similarly, the second duty can start already as early as 04:00. In this case, the rest time between the two duties should be explicitly enforced.

The Integrated Crew Re-Planning Problem (abbreviated ICRPP) can now be stated

as follows: Given the original rosters, and the new set of tasks, determine an alternative roster (consisting of alternative duties) for each employee such that all tasks are covered. When necessary, additional duties can be scheduled, but at a high cost. The new rosters should be feasible with respect to the duty and roster constraints, and the alternative duties should respect the given time windows. The ICRPP aims at capturing the additional flexibility in the start and end times of the alternative duties, by having much looser requirements on the time windows compared to day-by-day rescheduling.

5.3 Literature Review

Crew planning is a well-established field of research in the Operations Research literature. In railway and mass transit optimization, the planning problem is generally decomposed into crew scheduling and crew rostering: Both crew scheduling (see, for example, Desrochers and Soumis (1989), Hoffman and Padberg (1993), Kroon and Fischetti (2001), Grötschel et al. (2003), and Abbink et al. (2005)) and crew rostering (see, e.g., Sodhi and Norris (2004), Hartog et al. (2009), Mesquita et al. (2013), and Borndörfer et al. (2015)) are well-studied problems. We refer to Kohl and Karisch (2004), Huisman et al. (2005b), Caprara et al. (2007), Abbink et al. (2018), and Heil et al. (2019) for general overviews of crew planning in railway and airline optimization.

Only little research considers the integration of crew scheduling and rostering in public transport. Ernst et al. (2001) propose an integrated model able to construct both cyclic and acyclic rosters. The solution approach relies on the complete enumeration of all possible duties for each day. As noted by the authors, this is tractable for the considered sparse railway network in Australia, but for larger railway networks (e.g., the Dutch railway network) approaches such as column generation should be considered to deal with the large number of possible duties. Mesquita et al. (2013) integrate the construction of the vehicle and duty schedules with crew rostering, and propose a Benders decomposition approach to solve the resulting problem. The benefit of the approach is shown using practical instances based on the urban bus systems of Lisbon and Porto. Finally, Borndörfer et al. (2017) consider the integration of crew scheduling and rostering. The proposed mathematical formulation links the duties and rosters through *duty templates*: coarse representations of duties expressing only the key characteristics (see Borndörfer et al. (2013)). By picking suitable templates,

the number of linking constraints can be reduced drastically. The resulting model is solved using Benders decomposition, and it is shown that substantially improved rosters can be constructed without increasing the costs of the duty scheduling phase.

Railway re-scheduling and recovery has received considerable attention in recent years. We refer to Cacchiani et al. (2014) for a detailed overview. Huisman (2007) considers re-scheduling due to planned maintenance, similar to this work. Rezanova and Ryan (2010), Potthoff et al. (2010), and Sato and Fukumura (2012), on the other hand, focus on operational re-scheduling (or recovery), i.e., the re-scheduling of personnel during operations. Note that additional difficulties arise in this case, as crew members might already have started their duties at the moment of re-scheduling. Another key difference is the time available for re-scheduling: The re-scheduling due to planned maintenance is generally done numerous days in advance, and hence allows for multiple hours of computation time, whereas in operational re-scheduling the new duties should be obtained in a few minutes (or even seconds). Integrated approaches are proposed in Walker et al. (2005) and Veelenturf et al. (2012), where (part of) the timetable can be modified when re-scheduling. Finally, Veelenturf et al. (2014) propose a quasi-robust approach towards re-scheduling to cope with the uncertain length of the disruption period. The proposed formulation assures that a percentage of the re-scheduled duties should be recoverable. As a result, a subset of tasks is guaranteed to be covered for every disruption scenario.

Research regarding re-scheduling in the field of airline optimization precedes railway re-scheduling by numerous years (see, for example, Stojković et al. (1998), Lettovskỳ et al. (2000), Stojković and Soumis (2001), Petersen et al. (2012), among others). Clausen et al. (2010) gives a detailed overview of disruption management in airline optimization. The focus in airline recovery is generally on the operational planning phase, i.e., only short computation times are allowed. It is important to note that the railway and airline re-scheduling problem fundamentally differ: The railway re-scheduling problem deals with a single day in which many duties are to be re-scheduled, whereas the airline re-scheduling deals with pairings (i.e., a sequence of duties spanning multiple days), implying that multiple days are to be taken into account. In airline optimization, the duties can generally be enumerated (see, e.g., Lavoie et al. (1988), Stojković et al. (1998)), implying that all rules related to the duties can be taken care of implicitly. The pairings should, however, satisfy numerous rules regarding, e.g., rest times, making the problem more closely related to the crew rostering problem.

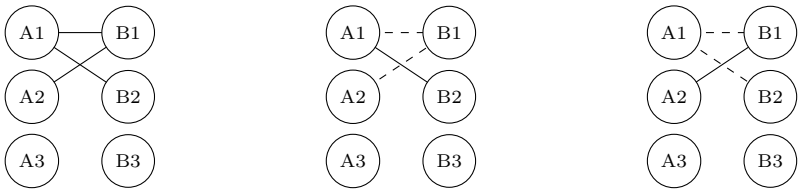
In this chapter, we add to the literature by integrating railway crew scheduling and rostering in the re-planning phase, i.e., we simultaneously re-schedule the duties for multiple days, thereby taking the feasibility of the individual rosters into account. This problem extends the work on re-scheduling in railway optimization, where traditionally only one day is considered. Furthermore, it differs from current research on the integration of crew scheduling and rostering, as the original duties are considered input (similar to the way crew re-scheduling differs from crew scheduling). The resulting problem resembles the re-scheduling of crew pairings in airline optimization, yet differs fundamentally in the sense that the number of possible duties per day is huge, and hence a different solution approach is necessary.

5.4 Mathematical Formulation

We propose to formulate the ICRPP on the duty level, i.e., we propose a formulation in which each variable indicates whether a possible alternative duty is selected or not. This implies that the duty constraints are readily taken care of in the variable definitions (that is, only feasible duties are considered). The roster constraints, on the other hand, need to be modeled explicitly.

Let R denote the set of original rosters (one for each employee) and let T denote the set of days in the re-planning period. The set K_t, for all $t \in T$, denotes the set of tasks that need to be covered on day T. The set of alternative duties for day $t \in T$ is denoted by Δ_t, and the set of duties that can be assigned to roster r on day t is given by $\Delta_t^r \subseteq \Delta_t$. The set Δ_t^r directly incorporates the specified time window and restrictions due to route knowledge. The binary parameter a_{tk}^{δ} indicates whether the duty $\delta \in \Delta_t$ covers task $k \in K_t$. The cost of assigning duty $\delta \in \Delta_t^r$ to roster $r \in R$ is given by c_{rt}^{δ} and the cost of selecting $\delta \in \Delta_t$ as additional duty is given by f_t^{δ}.

The rest time constraints are modeled using violation graphs, as previously proposed in Chapter 4. Let N denote the nights in the re-planning period, i.e., the transitions between two consecutive days in T. For each roster $r \in R$ and each night $n \in N$ we consider a bipartite graph where each edge represents a rest time violation between an end task at the beginning of n and a start task at the end of n. Figure 5.2 shows the violation graph for three possible end tasks A1, A2, and A3, and three possible start tasks B1, B2, and B3 for a given roster r. In this case, ending with A1 is incompatible with starting with B1 or B2, and ending with A2 is incompatible with starting with B1.



Rest time violation graph Maximal biclique 1 Maximal biclique 2

Figure 5.2: Example of clique-constraints, defined on the rest time violation graph. The dashed arcs indicate the maximal bicliques corresponding to the constraint: The first maximal biclique involves tasks A1, A2, and B1, and the second maximal biclique involves tasks A1, B1, and B2.

The rest time constraints for roster $r \in R$ and night $n \in N$ are obtained from the set of maximal bicliques (i.e., maximal complete bipartite subgraphs) in the violation graph, denoted by Q_n^r. Figure 5.2 depicts the two maximal bicliques in a violation graph. It has been shown in Chapter 4 that the size of Q_n^r is bounded by the number of nodes in the violation graph. Furthermore, it has been shown that the biclique-based constraints lead to the strongest possible formulation for the rest time constraints. We introduce the binary parameter $o_{rnt}^{\delta q}$, indicating whether the begin or end task of duty $\delta \in \Delta_t^r$ is part of the biclique $q \in Q_n^r$. In the first maximal biclique in Figure 5.2, for example, the parameter will equal 1 for all duties on the first day ending with either task A1 or A2 and all duties on the second day starting with B1. Note that, by definition, the parameter $o_{rnt}^{\delta q}$ is zero whenever t is not starting (or ending) before (or after) night n.

To model the workload constraints, let ℓ_{rt}^{δ} denote the length of duty $\delta \in \Delta_t^r$ and let w_r denote the maximum workload for roster R over the re-planning period T. We introduce the following decision variables.

- x_{rt}^{δ} for all $r \in R$, $t \in T$, and $\delta \in \Delta_t^r$. The binary decision variables x_{rt}^{δ} indicate whether duty $\delta \in \Delta_t^r$ is assigned to roster R.

- y_t^{δ} for all $t \in T$ and $\delta \in \Delta_t$. The binary decision variables y_t^{δ} indicate whether duty $\delta \in \Delta_t$ is scheduled as additional duty.

The ICRPP can now be formulated as follows.

$$\min \quad \sum_{r\in R}\sum_{t\in T}\sum_{\delta\in\Delta_t^r} c_{rt}^{\delta}x_{rt}^{\delta} + \sum_{t\in T}\sum_{\delta\in\Delta_t} f_t^{\delta}y_t^{\delta} \tag{5.1}$$

$$\text{s.t.} \quad \sum_{r\in R}\sum_{\delta\in\Delta_t^r} a_{tk}^{\delta}x_{rt}^{\delta} + \sum_{\delta\in\Delta_t} a_{tk}^{\delta}y_t^{\delta} \geq 1 \qquad \forall t\in T, k\in K_t \tag{5.2}$$

$$\sum_{\delta\in\Delta_t^r} x_{rt}^{\delta} = 1 \qquad \forall r\in R, t\in T \tag{5.3}$$

$$\sum_{t\in T}\sum_{\delta\in\Delta_t^r} o_{rnt}^{\delta q}x_{rt}^{\delta} \leq 1 \qquad \forall r\in R, n\in N, q\in Q_n^r \tag{5.4}$$

$$\sum_{t\in T}\sum_{\delta\in\Delta_t^r} \ell_{rt}^{\delta}x_{rt}^{\delta} \leq w_r \qquad \forall r\in R \tag{5.5}$$

$$x_{rt}^{\delta} \in \mathbb{B} \qquad \forall r\in R, t\in T, \delta\in\Delta_t^r \tag{5.6}$$

$$y_t^{\delta} \in \mathbb{B} \qquad \forall t\in T, \delta\in\Delta_t. \tag{5.7}$$

The Objective (5.1) expresses that we minimize the cost of the selected duties, consisting of the cost of the duties assigned to the rosters and the cost of the additional duties. Constraints (5.2) and (5.3) assure that each task is covered and that each roster is assigned exactly one duty for each day, respectively. Constraints (5.4) and (5.5) represent the roster constraints: (5.4) assure the minimum rest time is respected, and (5.5) enforce a maximum workload for each roster. Finally, the domains of the decision variables are specified in (5.6) and (5.7).

5.5 Solution Approach

We propose a column generation approach to solve the ICRPP. This type of approach has been succesfully applied to similar crew re-scheduling problems in railway optimization (see, for example, Huisman (2007) and Potthoff et al. (2010)), and is generally considered a state-of-the-art solution approach (for detailed surveys, see Barnhart et al. (1998), Lübbecke and Desrosiers (2005), Desaulniers et al. (2006) and Lübbecke (2011)). The main idea behind column generation is to solve a linear program with a huge amount of columns (i.e., variables) by only considering a subset of all possible columns. In each iteration, it is checked if possible profitable, i.e., negative reduced cost, columns are present among the columns not yet included. If this is the case, the procedure continues. Otherwise, the found solution is optimal, and the algorithm

terminates.

We propose a solution method in which we iteratively select alternative rosters for the crew members. The algorithm continues until all employees are assigned an alternative roster, and, when necessary, selects additional duties to cover the remaining tasks. To select good alternative rosters, we base the selection of alternative rosters on the solution to the linear relaxation of (5.1)–(5.7). In particular, we select the alternative rosters that appear in the solution with a high (fractional) value. The linear relaxation is solved using column generation.

The remainder of this section is structured as follows. In Section 5.5.1 we give a global overview of the iterative selection procedure, and in Section 5.5.2, we discuss the selection procedure for alternative rosters in detail. Then, in Section 5.5.3, we discuss the pricing problem for alternative duties, which underlies the column generation algorithm. We conclude in Section 5.5.4 with an overview of the acceleration strategies used to speed-up the column generation procedure.

5.5.1 Iterative Selection Procedure

We obtain an integer solution for the ICRPP by iteratively selecting an alternative roster for one of the crew members. This type of approach is generally referred to as a diving heuristic, i.e., a depth-first search heuristic in the branching tree (see, for example, Joncour et al. (2010), and references therein). The iterative procedure is schematically visualized in Figure 5.3.

The selection procedure consists of two phases: Firstly, alternative rosters are selected for the crew members, and, secondly, additional duties are selected to cover possibly still uncovered tasks. The main motivation of selecting complete rosters, as opposed to separate duties, is to assure feasibility with respect to the roster rules. Note that the roster rules do not apply to the additional duties. Each time an alternative roster or additional duty is selected, the set of still available rosters (i.e., crew members) and uncovered tasks is updated. Once all tasks are covered, the algorithm terminates and the found solution is returned.

As is common for diving heuristics, both the alternative rosters and additional duties are selected based on the *highest value* rule, i.e., all rosters or duties with integer value are selected, together with one roster or duty assigned the highest (non-integer) value in the fractional solution (the latter assuring that the solution to the linear relaxation

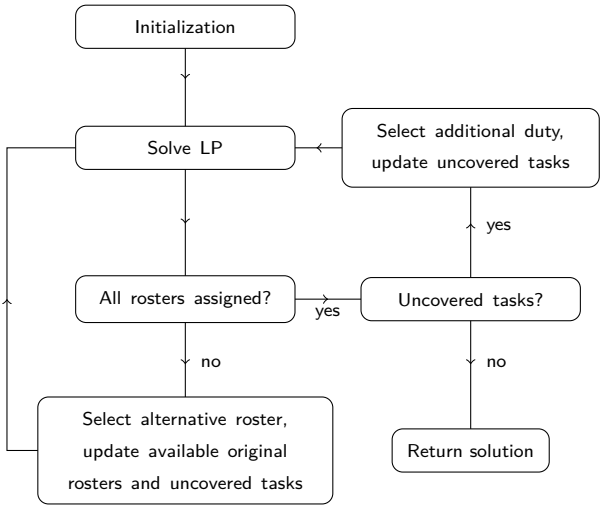


Figure 5.3: Iterative selection procedure for the ICRPP, consisting of two phases: Firstly, alternative rosters are selected for the crew members, and, secondly, additional duties are selected to cover possibly still uncovered tasks.

changes). Whereas defining the highest value is trivial for the additional duties (i.e., the value is directly obtained from the corresponding variable), this is not the case for the alternative rosters. In Section 5.5.2 we discuss the selection of alternative rosters in more detail.

5.5.2 Selecting Alternative Rosters

The roster rules imply that an arbitrary selection of alternative duties for an original roster can lead to infeasible solutions. Consider, for example, the fractional solution depicted in Table 5.2. In this specific example, selecting one duty per day, based on the highest value rule, will lead to an infeasibility: Selecting duties 2 and 3 will lead to an alternative roster which violates the minimum rest time of 12 hours. Note that the fractional solution *does* satisfy (5.4).

To assure the feasibility of the constructed rosters, even when multiple alternative duties are selected in each iteration, we directly select *all* alternative duties for a single roster, thereby taking the roster constraints directly into account. Consider again the example of Table 5.2. The duties give rise to five alternative rosters: $\{1,3\}, \{1,4\}, \{1,5\}, \{2,4\}$, and $\{2,5\}$. Note that the roster $\{2,3\}$ is not considered

	Day	Start	End	Value
1	1	12:00	20:00	0.4
2	1	12:30	21:00	0.6
3	2	08:00	16:00	0.4
4	2	09:00	17:00	0.3
5	2	09:00	17:30	0.3

Table 5.2: Example of a fractional solution for a given original roster. Each row indicates an alternative duty, and specifies the start and end time, and the (fractional) solution value.

here, as it violates the rest time constraint. The highest value rule is applied by taking the minimum over all assigned alternative duties. That is, the value corresponding to, for example, roster $\{1, 3\}$ equals $\min\{0.4, 0.4\} = 0.4$, and the value corresponding to roster $\{2, 4\}$ equals $\min\{0.6, 0.3\} = 0.3$. In this specific example, the highest value rule would select roster $\{1, 3\}$. For each original roster, the alternative roster with highest value can be determined and the alternative with overall highest value can be added to the solution.

The above procedure can be formalized as follows. Let (x, y) be a solution to the linear relaxation of (5.1)–(5.7). Given x, we determine the feasible alternative rosters $P_r(x)$ for each $r \in R$, consisting only of alternative duties with a non-zero solution value. This set is determined by complete enumeration. For an alternative roster $\rho_r \in P_r(x)$, let $\rho_{rt} \in \Delta_t^r$ denote the alternative duty assigned to day $t \in T$. The value $\pi(x, \rho_r)$ for each $\rho_r \in P_r(x)$ is defined as

$$\pi(x, \rho_r) = \min_{t \in T} \{x_{rt}^{\rho_{rt}}\},$$

i.e., the value of the alternative roster is based on the minimum taken over all assigned alternative duties. In each iteration, we determine for each original roster $r \in R$ a candidate $\rho_r^\star$ maximizing $\pi(x, \rho_r)$ and add all alternative rosters $\rho_r^\star$ for which $\pi(x, \rho_r^\star) = 1$ to the solution, together with the alternative roster with highest non-integer value (i.e., the highest value below 1). The involved original rosters and tasks are removed from the pool of available original rosters and uncovered tasks, respectively, and the algorithm continues.

5.5.3 Pricing Problem

The pricing of alternative and additional duties can be modeled as a series of Resource Constrained Shortest Path Problems (RCSPPs) on suitably defined graphs. Let B denote the set of crew bases. For every day $t \in T$ and crew base $b \in B$, we consider a dedicated digraph $G_t^b = (V_t^b, A_t^b)$, referred to as a pricing graph, where the nodes correspond to the tasks in K_t and the arcs represent feasible connections between tasks. The graph contains an additional source and sink node, representing the departure and arrival at the crew base.

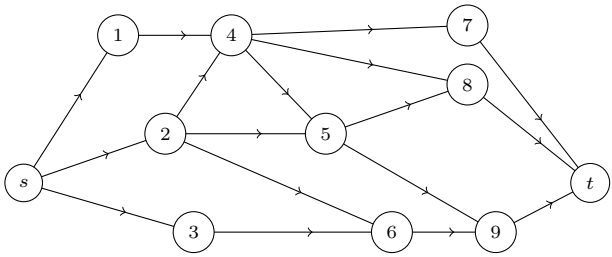


Figure 5.4: Pricing graph for alternative and additional duties. Each node corresponds to a task and each arc represents a feasible connection between two tasks. In this graph, a total of nine tasks are shown. Each path from the source s to the sink t corresponds to a possible duty.

Figure 5.4 gives a stylized example of a pricing graph. Note that any passenger tasks and taxi trips can be directly incorporated in the arc set. Furthermore, the route knowledge is readily incorporated by adding a node to the graph only for those tasks compatible with crew base b. Finally, the travel times from and to the crew base can be set accordingly using the source and sink arcs.

Let R_b denote the original rosters corresponding to base $b \in B$. For each base $b \in B$ and day $t \in T$, we solve a pricing problem for each original roster $r \in R_b$ to price out any negative reduced cost alternative duties for r on day t, and, furthermore, we solve one pricing problem to price out negative reduced cost additional duties for base b on day t. Note that the time window specified for original roster $r \in R_b$ on day $t \in T$ can be easily incorporated in the pricing graph G_t^b by discarding all tasks which cannot be covered within the specified time interval.

The reduced cost of the alternative and additional duties can be expressed as follows. Let λ_{tk} denote that dual variables corresponding to the coverage constraints (5.2),

μ_{rt} to the assignment constraints (5.3), γ^q_{rn} to the rest time constraints (5.4), and, finally, ϕ_r to the workload constraints (5.5). The reduced cost of x^δ_{rt} is given by

$$c^\delta_{rt} - \sum_{k \in K_t} \lambda_{tk} a^\delta_{tk} - \mu_{rt} - \sum_{n \in N} \sum_{q \in Q^r_n} \gamma^q_{rn} o^{\delta q}_{rnt} - \phi_r \ell^\delta_{rt}, \tag{5.8}$$

and the reduced cost of y^δ_t by

$$f^\delta_t - \sum_{k \in K_t} \lambda_{tk} a^\delta_{tk}. \tag{5.9}$$

As discussed in Section 5.2, the costs c^δ_{rt} and f^δ_t consist of a fixed cost and a cost per connection, and can therefore readily be modeled using the arc costs. As a result, the reduced cost can be modeled using appropriate arc costs, as well: The dual variables regarding the coverage constraints can be modeled as arcs costs using the incoming arcs of each task. Furthermore, the assignment constraint duals can be incorporated in the arc costs of arcs leaving the source. For the rest time constraints, we observe that membership of a biclique in the violation graph depends *only* on the start and end times of the alternative duty: The duals corresponding to the rest time constraints can therefore be readily modeled using arc costs on the arcs leaving and entering the source and sink, respectively. Finally, the length of the alternative duty decomposes over the length of the visited arcs and nodes, and hence can also be modeled using the arc costs.

The duty length and meal break rules are incorporated in the pricing problem in a similar way as proposed in Huisman (2007). Recall that the duty length rules are expressed by a start and/or end interval, and a maximum length, e.g., a duty length rule could specify that a duty starting between 05:00 and 06:00 or ending between 02:30 and 08:30 can have a length of at most 8.5 hours, or could specify that a duty starting after 06:00 and ending before 02:30 the next day can have a length of 9.5 hours. All these rules together can be captured by a suitably picked maximum ending time for each possible start task (see Figure 5.5). Hence, the pricing problem, given a pricing graph (either for an alternative or an additional duty), can be modeled as a RCSPP for each start task, with a resource related to the meal break constraint.

Solving the RCSPP for each possible start node allows to incorporate the duty length constraint. This is done as follows: Given a start task s, we enforce the maximum duty length constraint by allowing only those end tasks e such that the duty starting with s and ending with e respects the maximum duty length constraint. This

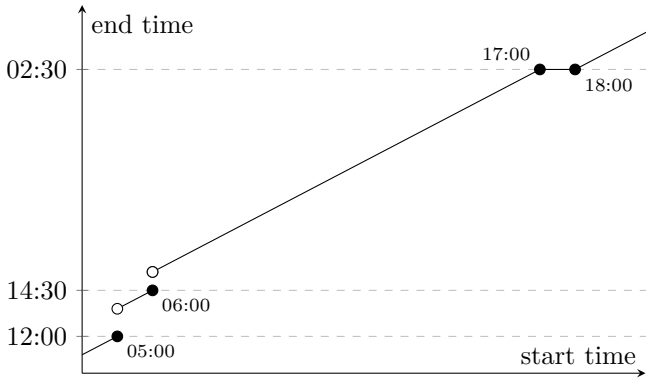


Figure 5.5: Duty length function, mapping each start time to the maximum end time. Duties starting between 04:00 and 05:00 are at most 7 hours, those starting between 05:00 and 06:00 at most 8.5 hours, and those starting after 06:00 at most 9.5 hours. Furthermore, duties ending after 02:30 in the night cannot exceed 8.5 hours.

procedure is illustrated in Figure 5.6. For task s_1, starting at 05:15, we remove end tasks e_3, e_4, and e_5 from the set of possible end task, as the length of the resulting duty would be too long. For start task s_2 all the end tasks except e_5 are feasible.

We solve the RCSPP using a labeling algorithm incorporating *completion bounds* (similar to Chapter 2): lower bounds on the minimum cost of completing a path to the sink node. The underestimation follows from the fact that the lower bounds are based on the shortest path costs, i.e., the meal break constraint is not taken into account. We solve the RCSPP heuristically by using Breadth First Search (BFS), thereby limiting the label set to at most k labels, based on the lowest estimated cost (i.e., the lowest lower bound). Note that this labeling strategy considers at least the k shortest paths in the graph, which, as argued in Huisman (2007), are often feasible for the meal break constraint and hence represent the set of most negative reduced columns. Whenever optimality is required, we switch to a Depth First Search (DFS) procedure (to limit the size of the label set), and terminate whenever either the k most negative reduced cost paths are found, or we prove that no more negative reduced cost paths are present.

5.5.4 Acceleration Strategies

The performance of column generation based algorithms depends heavily on the strategy for pricing negative reduced columns and the precise interaction between the

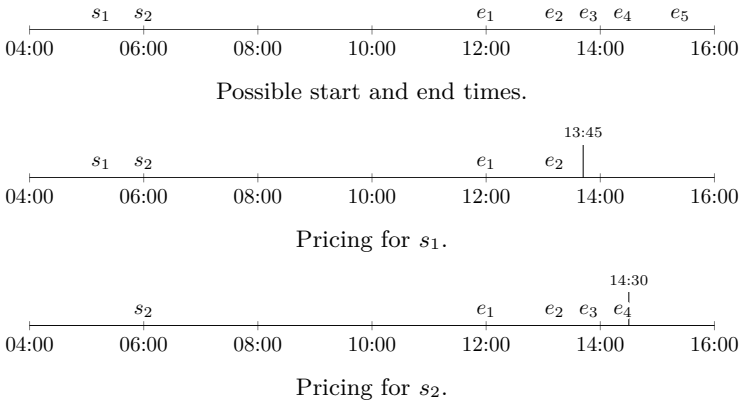


Figure 5.6: Two pricing problems resulting from the start tasks s_1 and s_2. For a given start task s, the duty length constraint is enforced by removing all start tasks before s and all end tasks e such that the time between s and e violates the duty length constraint.

master and pricing problems. This led to the development of acceleration strategies: strategies, often specified for classes of problems, to improve the column generation procedure (see e.g., Desaulniers et al. (2002) for a detailed overview). We consider three acceleration strategies for the ICRPP, generally referred to as *pre-processing* (i.e., reducing the number or size of the RCSPPs that need to be solved), *partial pricing* (i.e., solving only a subset of pricing problems) and the enforcement of *task-disjoint* columns.

Pre-Processing Start Tasks

Firstly, we limit the number of RCSPPs to be solved for a given pricing graph, by avoiding all 'redundant' start tasks, i.e., by avoiding all start tasks which can be skipped without losing optimality. This can be done as follows. Firstly, we note that all tasks leading to the same start time can be aggregated into one pricing problem: Since the decomposition per start task functions to model the duty length constraint, this can be done while still assuring exactness of the pricing problem. Secondly, we note that, in a similar fashion, any two start tasks leading to the same set of end tasks can be aggregated into one pricing problem. Here, the possible end tasks depend on (*i*) the duty length constraint and (*ii*) the specified time window. Given these two rules, the possible start tasks can be aggregated into pricing problems using a simple

greedy procedure. In Section 5.6, we show that this pre-processing framework greatly reduces the number of possible start tasks per pricing graph, and hence the number of RCSPP that need to be solved.

Partial Pricing of Alternative Duties

Secondly, we consider a partial pricing strategy for the alternative duties: For each original duty, we order the set of possible start tasks randomly, and restrict the generation of alternative duties to only those pricing problems corresponding to one of the n first tasks in this list, where n is an a priori set control parameter. Recall that, when solving the pricing problem for start task s, we also allow start tasks starting later than s, i.e., in Figure 5.6 task s_2 is allowed as start task when pricing for task s_1. This implies that many start tasks are considered even if we price only for a small amount of start tasks, a desirable property when applying partial pricing. Note, however, that the duties generated for the additional start tasks will be shorter than technically possible (since the duty length constraint is based on an earlier start task), hence, it is still necessary to price for all start tasks to assure no negative reduced columns exist.

The partial pricing strategy is applied in two ways: *heuristically* and *exact*. In heuristic partial pricing, we stop solving pricing problems after the first n start tasks have been considered. The found negative reduced cost columns are returned, and the column generation algorithm continues. The advantage of heuristic partial pricing is that it allows easy control of the time spent on pricing in each iteration. The major downside, however, is that possibly not all negative reduced cost columns are found, i.e., the linear relaxation will most likely not be solved to optimality. Exact partial pricing aims at avoiding the latter: Whenever no negative reduced cost columns are found using heuristic partial pricing, we continue iterating through the list of possible start tasks for each of the original duties, until either the complete list has been considered, or a column with negative reduced cost has been found. This assures that the linear relaxation will be solved to optimality. In heuristic partial pricing, we solve the RCSPP using the heuristic approach (as discussed in Section 5.5.3), wheres in exact partial pricing we also switch to the exact approach for the RCSPP.

In the iterative selection procedure we apply both heuristic and exact partial pricing to efficiently solve the ICRPP: Exact partial pricing is applied in the root node, to obtain a good initial solution and valid lower bound, and heuristic partial pricing is

used in the remaining nodes of the tree, i.e., each time we select an alternative roster and the linear relaxation is resolved. This avoids that we spend much time on exact partial pricing throughout the diving procedure.

Selecting Task-Disjoint Columns

The third and final acceleration strategy focuses on the structure of the column pool. For the ICRPP, and other set-covering type of problems, an optimal solution will most likely have columns with little overlap, i.e., the columns have little tasks in common. By enforcing the returned columns to have little overlap (referred to as being *task-disjont* in Desaulniers et al. (2002)), the column pool will have an 'optimal' structure, without flooding the master problem with a huge number of columns. The level of overlap allowed controls the trade-off between the number of columns in the column pool and the maximum overlap among columns.

Given a set of negative reduced columns, a close to task-disjoint subset is picked as follows. For any two columns x_1 and x_2 a *similarity score* can be computed based on the tasks covered: The similarity score of column x_1 with column x_2 is defined as the number of tasks covered by both columns divided by the total number of tasks covered by x_1. The higher this score, the more x_1 is similar to (or contained in) x_2. Given the generated columns and an a priori set similarity threshold, we greedily obtain a subset of columns based on most negative cost, whilst assuring that the similarity score with the already selected columns does not exceed the similarity threshold. Note that this procedure assures that the column with most negative reduced cost is always selected. The resulting subset of columns will have little overlap.

5.6 Computational Experiments

To illustrate the benefit of the integrated approach, we apply our solution approach to practical instances from NS. In Section 5.6.1, we give a detailed description of the case study. In Sections 5.6.2 and 5.6.3 we analyze the computation results: In Section 5.6.2 we analyze the effect of the acceleration strategies discussed in Section 5.5.4 and in Section 5.6.3 we discuss the results of the iterative selection procedure proposed in Section 5.5.2.

5.6.1 Case Study

We consider the re-scheduling of crew over a weekend (i.e., Saturday and Sunday), in April 2019, when large-scale maintenance was scheduled around station Leiden (Ledn). The situation is depicted in Figure 5.7. We construct different instances based on the crew bases Amsterdam (Asd), Lelystad (Lls), The Hague (Gvc), and Rotterdam (Rtd), four major crew bases for which many original rosters became infeasible due to the maintenance activities.

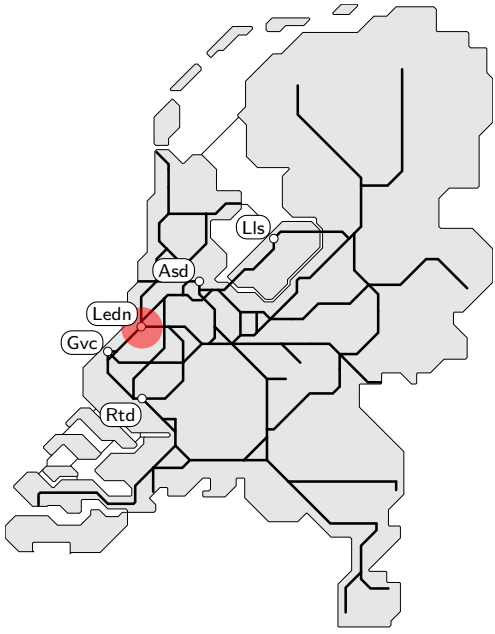


Figure 5.7: Case study considering scheduled maintenance at Leiden (Ledn), shown together with the crew bases located at The Hague (Gvc), Rotterdam (Rtd), Amsterdam (Asd), and Lelystad (Lls). The black lines show the lines operated by NS.

We construct four re-planning instances, each corresponding to a re-planning problem for three of the four crew bases. The characteristics of the four instances are shown in Table 5.3. The original duties and updated timetable are obtained from the second Saturday in April 2019, a day in which the planned timetable was substantially

altered: About 20 duties became infeasible at each of the four crew bases. We extend this to data to the whole weekend by assuming the timetable and original duties for Sunday to be the same. For each instance, we aim at covering all tasks in the (now infeasible) original duties which are still in the plan. The result are four challenging instances, where more than 500 tasks per day are to be covered. Furthermore, to assure sufficient flexibility, the remaining tasks are also added to the problem. This implies that for each of the four instances, there are 2519 possible tasks (i.e., tasks that are not necessarily to be covered but can be used as passenger tasks) and 87514 possible connections per day.

Instance	Bases	Rosters	Tasks To Cover Per Day	Tasks Per Day	Connections Per Day
1	3	75	649	2519	87514
2	3	64	539	2519	87514
3	3	81	713	2519	87514
4	3	74	640	2519	87514

Table 5.3: Description of the four re-planning instances. For each instance, the number of involved crew bases and rosters (i.e., employees) is shown. Furthermore, the total number of tasks and possible connections per day are shown, together with the number of tasks per day that need to be covered.

We construct original rosters covering all of the original duties. When constructing the rosters, we assume that each employee works both days in the weekend. The rosters are obtained by solving a stylized rostering problem, where we minimize the occurrence of short rest times and high workload, while assuring that the roster rules (as discussed in Section 5.2) are satisfied. Furthermore, we assure that each roster remains feasible whenever a duty is shifted by at most half an hour.

5.6.2 Analysis of Acceleration Strategies

The acceleration strategies discussed in Section 5.5.4 aim at (substantially) improving the time spent in the column generation algorithm. In this section, we discuss the gain from pre-processing the possible start tasks for each original duty, and we analyze the effect of the three control parameters: the number of paths returned per RCSPP, the similarity threshold, and the percentage of pricing problems solved in each iteration.

Table 5.4 shows the effect of the pre-processing discussed in Section 5.5.4, i.e., the removal of all pricing problems corresponding to 'redundant' start tasks. For each instance and each maximum allowed shift, Table 5.4 shows the average number of

considered start tasks (averaged over all original duties), and hence RCSPPs to be solved, without pre-processing (Original) and the average number of start tasks with pre-processing, i.e., the average number of non-redundant start tasks (Reduced).

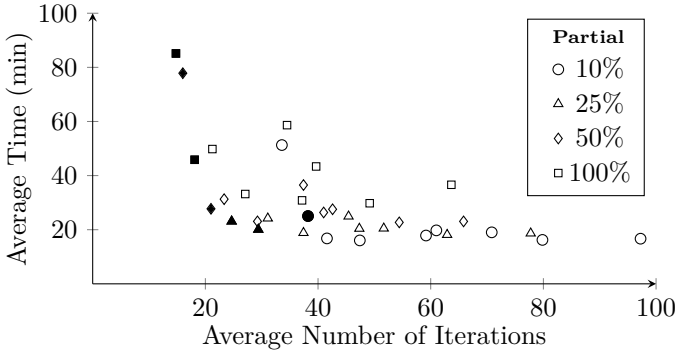
	Instance 1		Instance 2		Instance 3		Instance 4	
Shift	Original	Reduced	Original	Reduced	Original	Reduced	Original	Reduced
00:30	822.1	9.5	551.3	7.4	701.2	8.3	754.3	8.9
01:00	836.5	13.0	562.2	9.3	713.2	11.5	766.3	12.3
02:00	862.7	21.2	582.3	14.0	735.0	18.8	786.7	20.3
05:00	928.8	44.5	633.8	27.4	790.2	39.5	836.4	42.4

Table 5.4: The effect of pre-processing the possible start tasks. For each instance and each shift, the average number of considered start tasks (averaged over the original duties) before and after pre-processing are shown (indicated by Original and Reduced, respectively).

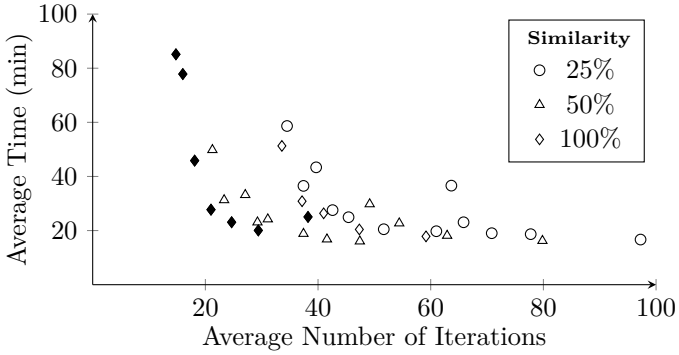
The benefit of pre-processing is clearly visible from Table 5.4. For all instances and all shift sizes, the reduction greatly reduces the number of start tasks: The reduction ranges from about 95% to 99%. Hence, the number of pricing problems can be reduced by more than a factor 20 if the start tasks are checked for redundancy a prori starting the column generation procedure. For the larger shifts, the number of original start tasks (and the number of start tasks after pre-processing) increases, as expected. As can be seen from Table 5.4, the effectiveness of the pre-processing slowly decreases when the shift size increases (from about 99% to 95%), which can most likely be explained by the fact that redundancy of a start task is less likely when many different start and end tasks (and hence start and end times) are possible.

We analyze the effect of the different acceleration strategies by considering the solution time for the linear relaxation for different parameter configurations. We consider returning 10, 100, and 250 columns for each RCSPP, a similarity threshold of 25%, 50%, and 100% and the percentage of pricing problems to be solved in partial pricing to be either 10%, 25%, 50%, or 100%. Furthermore, we average the results over three runs, each using a different random seed. This leads to 432 experiments in total. For each individual run, we limit the maximum computation time to 6 hours. Whenever a run exceeds 6 hours, it is not taken into account when computing the average. The results are visualized in Figure 5.8. For completeness, the full computational results are given in Appendix 5.A.

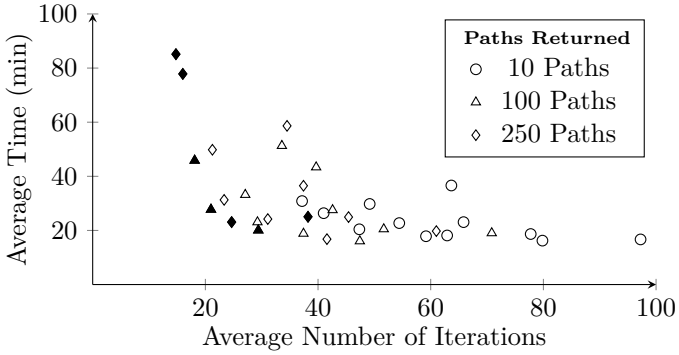
Figure 5.8a clearly highlights the benefit of partial pricing: The shortest average computation times are all achieved for partial pricing parameters of 10% and 25%, and there seems to be a clear increasing relation between the percentage of pricing



(a) Results grouped based on the percentage of pricing problems solved in partial pricing.



(b) Results grouped based on the similarity threshold.



(c) Results grouped based on the number of paths returned per RCSPP.

Figure 5.8: For each combination of parameters, the average computation time in minutes and the average number of iterations are shown, taken over the four shift lengths (00:30, 01:00, 02:00, and 05:00) and three random seeds. Each subfigure groups the results based on one of the parameters. Filled markers indicate parameters settings for which some runs were not finished within the 6 hour time limit.

problems solved in each iteration and the average computation time. The gain of using task-disjointness to select columns for the master problem is also visible in Figure 5.8b: The best performance is achieved for parameter settings using a similarity threshold of less than 100%. Note that a threshold of 100% implies that all generated columns are added to the master problem. As a result, the size of the master problem grows quickly. Whenever the added columns are 'good', the column generation algorithm terminates in only a few iterations. Whenever the columns are 'bad', however, the time spent in the master problem grows quickly. Following this reasoning, it is no surprise that each setting for which not all runs terminated in time, used a similarity threshold of 100%. Figure 5.8b also seems to indicate that a threshold of 50% outperforms a threshold of 25%. In this case, a 25% threshold might be too strict, implying that too many generated (and useful) columns will not be added to the master problem, and hence more column generation iterations are needed. Finally, the effect of the (maximum) number of paths returned per RCSPP, shown in Figure 5.8c, seems to be less clear, although returning 10 or 100 paths seems to outperform returning 250 paths. Note, however, that the effect of this parameter is highly intertwined with the other two parameters, e.g., returning 250 paths can perform well accompanied with a low similarity threshold, but can also perform badly whenever accompanied with a 100% threshold and/or a partial pricing parameter of 100%.

Summarizing, the results shown in Figure 5.8 give insight in the effective usage and possible gain of the acceleration strategies discussed in Section 5.5.4. Clearly, it is not possible to select the 'best' configuration based on the experiments, as the number of runs per configuration are limited and the running times vary substantially (see Appendix 5.A). It does, however, give insight in 'good' configuration settings. In particular, the experiments highlight the benefit of partial pricing and the focus on task-disjoint columns, together with a suitably picked number of maximum paths to be returned for each solved pricing problem.

5.6.3 Results Iterative Selection Procedure

In this section, we analyze the benefit of the integrated approach compared to the day-by-day approach. As discussed in Section 5.2, the start and end of the alternative rosters can shift at most half an hour, but the start and end times of duties within the roster can be changed freely. The maximum allowed shift regulates the latter: Allowing a shift of two hours implies that the end of the alternative duty

on Saturday and the beginning of the alternative duty on Sunday can deviate up to two hours from their corresponding original duty. By varying the allowed shift, the trade-off between the number of necessary duties and the deviation from the original rosters can be analyzed. In particular, the solution found using the day-by-day approach (corresponding to a maximum shift of half an hour) can be compared with the solutions when larger shifts are allowed. Note that every duty should still satisfy the general start and end time rules (i.e., duties start from 04:00 and end before 08:30 the next day), even if the maximum shift would allow otherwise.

We consider four maximum shifts for each instance: 00:30, 01:00, 02:00 and 05:00. Note that larger shifts are also possible, yet we found that these shifts did not lead to a decrease in the root bound (compared to 05:00), and are therefore omitted. The problem for a maximum shift of 00:30 is equivalent to the day-by-day approach, i.e., the problem is decomposable per day. Hence, in this case, we solve two separate CRSPs (instead of the ICRPP). For each instance and each shift size, we average the results over five runs, each for a different random seed. The results are shown in Table 5.5. The major cost components are a cost of 2100 for a scheduled duty, a cost of 1 for an empty duty, and an additional penalty of 10000 for an additional duty (roughly five times the cost of a normal duty), together with some (small) cost on the connections. In other words, we avoid scheduling additional duties, and try to assign empty duties whenever possible (as assigning an empty duty to an original duty reduces the fixed cost for this duty from 2100 to 1) . We used a partial pricing parameter of 10%, similarity threshold of 50%, and return 100 paths per RCSPP.

The results in Table 5.5 show a clear benefit from the integrated approach: The solutions found using time shifts larger than 00:30 all improve on the day-by-day approach. Furthermore, in almost all cases, the best found solution is below the root bound for the half hour shift, i.e., the best found solutions are provably better than the best possible solution using a day-by-day approach. The benefit of larger time shifts is most clearly expressed in the root bounds, where often a major decrease (often about 10%) is visible. Table 5.5 also shows, however, that the solution quality varies over the different runs. In particular, the objective value of the found solutions are not monotonically decreasing (or non-increasing) in the allowed shift size, something which is clearly the case would the problem have been solved to optimality. This variability seems to be mostly contained in the number of reserve duties in the found solution, the major driver behind the objective value. Table 5.5 does show that the number of scheduled empty duties tend to increase with larger shift sizes,

	Shift	Objective		Gap (%)		Best Found	Root Bound	Additional		Empty		Time (s)	
1	00:30	343286.0	(4711.9)	4.3	(1.3)	340506.0	328406.0	3.2	(0.4)	6.0	(0.0)	560.0	(65.6)
	01:00	328529.4	(11253.5)	5.5	(3.3)	310009.0	310009.0	2.6	(1.0)	9.4	(0.8)	1127.0	(211.8)
	02:00	330233.2	(7944.9)	8.0	(2.3)	315812.0	303712.0	3.4	(0.8)	13.2	(1.2)	1969.6	(253.5)
	05:00	**327294.6**	(4435.3)	7.8	(1.2)	323714.0	301613.0	3.4	(0.5)	14.6	(0.8)	5281.4	(464.5)
2	00:30	270574.2	(8396.6)	7.8	(2.8)	261314.0	249214.0	1.8	(0.7)	14.2	(0.4)	418.2	(51.0)
	01:00	**258375.4**	(7032.9)	5.0	(2.6)	247115.0	245365.8	1.0	(0.6)	15.4	(0.5)	812.2	(81.8)
	02:00	265437.8	(5837.7)	8.4	(2.0)	257116.0	242917.0	2.0	(0.6)	17.8	(1.2)	965.8	(145.5)
	05:00	259758.2	(8472.8)	6.4	(3.2)	242917.0	242917.0	1.6	(0.8)	18.2	(0.7)	1983.0	(174.4)
3	00:30	375570.2	(8396.6)	5.6	(2.1)	366310.0	354210.0	3.8	(0.7)	10.2	(0.4)	677.4	(39.4)
	01:00	360274.2	(9348.4)	8.0	(2.4)	345814.0	331086.9	3.2	(0.7)	14.2	(1.0)	2936.8	(462.7)
	02:00	**351197.2**	(1570.7)	6.8	(0.4)	349518.0	327417.0	3.0	(0.0)	17.2	(0.7)	3070.4	(625.3)
	05:00	354876.6	(8887.3)	7.7	(2.4)	339517.0	327417.0	3.2	(0.7)	16.6	(0.5)	5924.0	(303.4)
4	00:30	368077.8	(9504.9)	7.9	(2.4)	350718.0	338618.0	6.4	(0.8)	17.8	(0.4)	641.6	(55.3)
	01:00	**334178.4**	(11083.5)	7.5	(3.0)	322918.0	308719.0	4.0	(1.1)	18.4	(1.4)	1923.8	(286.6)
	02:00	337240.4	(7725.3)	9.0	(2.0)	330820.0	306620.0	4.6	(0.8)	20.4	(1.0)	2406.4	(161.9)
	05:00	342080.4	(13671.5)	10.2	(3.5)	328721.0	306620.0	5.0	(1.1)	20.4	(0.5)	5018.2	(308.6)

Table 5.5: Computational results for the iterative selection procedure. For each instance and each maximum shift, the results are averaged over five different runs, each for a different random seed. The average found solution value, average root gap, best found solution, and root bound are shown. Furthermore, the average number of scheduled additional and empty duties is depicted, together with the average computation time (in seconds). The numbers within brackets denote the standard deviation, when applicable.

an indication that the original duties can be used more efficiently, i.e., more tasks can be placed into a single duty.

The computation times shown in Table 5.5 are in line with the expectations: The running time for the half hour shift (i.e., the combined time of two CRSPs) is substantially lower than the time needed for the ICRPP, since there is no synchronization between the two days in this case. Furthermore, the running times increase in the shift size. Note, however, that the running times for the ICRPP can be considered reasonable: The average computation time never exceeds two hours, and in most cases stays well below one hour.

Summarizing, the integrated approach shows clear potential over the day-by-day approach: Allowing larger shifts leads to provably better solutions. The trade-off between the efficiency (i.e., number of necessary duties) versus the deviation from the original roster, however, is not clear, as the performance of the heuristic varies among runs. From a practical point of view, a small increase in shift (e.g., one hour) seems therefore the most profitable strategy: The solution value decreases substantially, the running time stays within one hour, and the newly scheduled duties stay relatively close to the original duties.

5.7 Conclusion

In this chapter, we introduced the Integrated Crew Re-Planning Problem (ICRPP), an integrated approach for crew re-scheduling over multiple days. In doing so, we extend the Crew Re-Scheduling Problem (CRSP), by allowing more flexibility in the newly assigned duties. The additional complexity of the ICRPP resides in the problem size, as multiple days need to be re-scheduled at once, and in the fact that the feasibility of the rosters should be taken into account explicitly.

We proposed a mathematical formulation for the ICRPP and developed a column generation based heuristic to solve the problem. We applied the approach to four instances, based on data from NS. We analyzed the benefit of an integrated approach and considered the trade-off between the number of necessary duties and the deviation from the original plan. The results show a clear gain from integrating the solution process, yet also show that the performance of the heuristic varies among different runs. From a practical point of view, the integrated approach accompanied with a slightly larger flexibility in the start and end times seems most profitable: The solution value decreases substantially, the increase in running time is limited, and the deviation from the original schedule will be relatively small.

For further research, the (primal) performance of the column generation heuristic seems most interesting. It is well-known that the incorporation of sophisticated local search methods within the column generation algorithm can substantially improve the performance. Similarly, exact branch-and-price methods could further highlight the potential of the integrated approach. Finally, heuristic approaches that iteratively enlarge the allowed time shift, either in an integrated or sequential way, could be an effective way of obtaining good solutions quickly.

Appendix

5.A Overview Computational Results

Tables 5.6 and 5.7 show the computation results used for Figure 5.8. Table 5.6 shows the overall computation time and the computation time per shift size, averaged over three random seeds. Table 5.7 shows the average number of iterations and the average number of iterations per shift size, averaged over three random seeds.

Parameters			Computation Time (s)															
Nr.	Sim.	Perc.	Shift 00:30			Shift 01:00			Shift 02:00			Shift 05:00			Sum			
10	0.25	0.1	268.0	(32.0)	3/3	947.3	(93.1)	3/3	776.0	(176.3)	3/3	2011.3	(157.8)	3/3	4002.7	(105.2)	12/12	
10	0.25	0.25	307.7	(2.5)	3/3	1019.3	(134.3)	3/3	1002.3	(111.1)	3/3	2151.7	(231.7)	3/3	4481.0	(246.4)	12/12	
10	0.25	0.5	398.3	(9.0)	3/3	1088.3	(40.7)	3/3	1201.7	(184.0)	3/3	2848.0	(63.4)	3/3	5536.3	(279.2)	12/12	
10	0.25	1.0	537.3	(4.0)	3/3	1671.7	(232.2)	3/3	1955.3	(254.3)	3/3	4621.0	(455.8)	3/3	8785.3	(430.6)	12/12	
10	0.5	0.1	270.3	(91.8)	3/3	786.0	(106.4)	3/3	759.0	(114.0)	3/3	2080.3	(226.1)	3/3	3895.7	(75.5)	12/12	
10	0.5	0.25	285.3	(18.1)	3/3	863.0	(126.4)	3/3	962.3	(123.8)	3/3	2242.3	(182.9)	3/3	4353.0	(187.4)	12/12	
10	0.5	0.5	338.7	(5.3)	3/3	1127.0	(114.6)	3/3	1220.7	(165.7)	3/3	2764.3	(143.3)	3/3	5450.7	(212.4)	12/12	
10	0.5	1.0	469.0	(7.8)	3/3	1493.7	(219.0)	3/3	1632.7	(273.9)	3/3	3552.7	(59.2)	3/3	7148.0	(132.3)	12/12	
10	1.0	0.1	188.0	(7.5)	3/3	671.3	(66.3)	3/3	730.7	(65.3)	3/3	2694.7	(199.3)	3/3	4284.7	(174.4)	12/12	
10	1.0	0.25	224.7	(2.5)	3/3	784.7	(110.3)	3/3	896.0	(147.4)	3/3	2994.3	(21.8)	3/3	4899.7	(58.0)	12/12	
10	1.0	0.5	282.0	(4.3)	3/3	977.0	(121.3)	3/3	1352.0	(253.7)	3/3	3718.7	(135.7)	3/3	6329.7	(485.1)	12/12	
10	1.0	1.0	407.7	(6.8)	3/3	1322.7	(150.2)	3/3	1450.3	(137.2)	3/3	4222.0	(122.0)	3/3	7402.7	(144.1)	12/12	
100	0.25	0.1	342.0	(24.9)	3/3	827.0	(65.5)	3/3	710.3	(44.2)	3/3	2687.0	(342.2)	3/3	4566.3	(401.9)	12/12	
100	0.25	0.25	371.0	(12.2)	3/3	1004.0	(156.3)	3/3	1001.3	(19.7)	3/3	2542.0	(18.4)	3/3	4918.3	(173.9)	12/12	
100	0.25	0.5	488.3	(6.9)	3/3	1235.3	(171.1)	3/3	1472.3	(89.4)	3/3	3417.0	(80.8)	3/3	6613.0	(270.1)	12/12	
100	0.25	1.0	715.3	(6.1)	3/3	1696.7	(219.0)	3/3	2197.3	(25.4)	3/3	5799.3	(385.0)	3/3	10408.7	(385.7)	12/12	
100	0.5	0.1	244.3	(19.6)	3/3	707.0	(117.3)	3/3	753.0	(25.6)	3/3	2145.7	(171.4)	3/3	3850.0	(171.3)	12/12	
100	0.5	0.25	276.7	(21.1)	3/3	849.3	(111.5)	3/3	868.7	(30.6)	3/3	2528.7	(269.6)	3/3	4523.3	(347.1)	12/12	
100	0.5	0.5	369.3	(18.2)	3/3	955.0	(106.8)	3/3	1163.7	(35.2)	3/3	3050.0	(95.6)	3/3	5538.0	(124.0)	12/12	
100	0.5	1.0	543.7	(7.0)	3/3	1362.0	(160.5)	3/3	1703.3	(78.1)	3/3	4357.0	(101.3)	3/3	7966.0	(164.9)	12/12	
100	1.0	0.1	222.0	(25.4)	3/3	870.3	(32.8)	3/3	883.3	(96.6)	3/3	10332.3	(7221.3)	3/3	12308.0	(7283.0)	12/12	
100	1.0	0.25	349.3	(35.6)	3/3	1114.0	(125.9)	3/3	1167.7	(57.0)	3/3	4141.0	(0.0)	1/3	4011.3	(1825.9)	10/12	
100	1.0	0.5	500.7	(9.0)	3/3	1109.3	(151.6)	3/3	1581.7	(65.0)	3/3	7058.0	(0.0)	1/3	5544.3	(3222.4)	10/12	
100	1.0	1.0	892.0	(32.6)	3/3	1522.7	(63.8)	3/3	3745.7	(1953.3)	3/3	9026.0	(0.0)	1/3	9169.0	(3709.4)	10/12	
250	0.25	0.1	406.7	(88.0)	3/3	1062.0	(173.3)	3/3	896.0	(34.0)	3/3	2376.7	(193.4)	3/3	4741.3	(281.2)	12/12	
250	0.25	0.25	531.0	(54.4)	3/3	1220.0	(109.1)	3/3	1247.0	(17.9)	3/3	2986.7	(216.0)	3/3	5984.7	(236.6)	12/12	
250	0.25	0.5	730.3	(57.1)	3/3	1654.3	(277.3)	3/3	1887.0	(63.9)	3/3	4491.0	(144.5)	3/3	8762.7	(413.3)	12/12	
250	0.25	1.0	1184.0	(114.1)	3/3	2385.0	(282.9)	3/3	3083.3	(73.6)	3/3	7421.7	(500.4)	3/3	14074.0	(772.4)	12/12	
250	0.5	0.1	274.0	(45.7)	3/3	819.3	(85.1)	3/3	792.0	(48.0)	3/3	2143.0	(123.3)	3/3	4028.3	(131.4)	12/12	
250	0.5	0.25	396.0	(43.8)	3/3	887.0	(117.9)	3/3	1124.3	(36.3)	3/3	3401.0	(937.8)	3/3	5808.3	(1048.3)	12/12	
250	0.5	0.5	523.0	(63.1)	3/3	1119.3	(88.7)	3/3	1408.3	(16.3)	3/3	4458.3	(1268.9)	3/3	7509.0	(1332.2)	12/12	
250	0.5	1.0	815.3	(93.3)	3/3	1690.0	(227.7)	3/3	2188.0	(69.3)	3/3	7260.7	(2559.7)	3/3	11954.0	(2769.1)	12/12	
250	1.0	0.1	419.7	(58.7)	3/3	1198.3	(53.2)	3/3	2889.7	(2208.5)	3/3	0.0	(0.0)	0/3	4507.7	(2211.6)	9/12	
250	1.0	0.25	732.7	(16.4)	3/3	1477.3	(91.1)	3/3	1945.0	(151.4)	3/3	0.0	(0.0)	0/3	4155.0	(109.6)	9/12	
250	1.0	0.5	1545.7	(173.6)	3/3	2741.3	(480.1)	3/3	6157.0	(4181.5)	3/3	15372.0	(0.0)	1/3	15568.0	(6431.1)	10/12	
250	1.0	1.0	2972.7	(80.1)	3/3	4218.3	(519.8)	3/3	8133.7	(2195.0)	3/3	0.0	(0.0)	0/3	15324.7	(2628.5)	9/12	

Table 5.6: Solution time for the root node for different parameter settings. For each configuration and each shift, we considered three different random seeds, and show the average and standard deviation (in brackets) of the computation time over these seeds (taken over the runs that terminated in time), together with the number of runs that finished within 6 hours. The parameters are: the number of paths returned per RCSPP (Nr), the similarity threshold (Sim), and the percentage of pricing problems solved in partial pricing (Perc).

Parameters			Iterations														
Nr.	Sim.	Perc.	Shift 00:30			Shift 01:00			Shift 02:00			Shift 05:00			Sum		
10	0.25	0.1	109.3	(7.6)	3/3	156.0	(7.3)	3/3	67.7	(3.1)	3/3	56.0	(1.6)	3/3	389.0	(1.4)	12/12
10	0.25	0.25	95.0	(0.8)	3/3	115.7	(3.7)	3/3	55.3	(2.5)	3/3	45.0	(0.8)	3/3	311.0	(2.9)	12/12
10	0.25	0.5	85.0	(1.4)	3/3	88.0	(1.6)	3/3	48.7	(1.7)	3/3	41.7	(0.9)	3/3	263.3	(4.5)	12/12
10	0.25	1.0	80.3	(1.2)	3/3	83.0	(2.2)	3/3	47.3	(0.5)	3/3	44.0	(1.6)	3/3	254.7	(1.7)	12/12
10	0.5	0.1	96.3	(19.0)	3/3	122.7	(0.9)	3/3	55.7	(1.7)	3/3	44.7	(1.2)	3/3	319.3	(20.3)	12/12
10	0.5	0.25	81.3	(4.5)	3/3	89.7	(3.3)	3/3	44.7	(1.7)	3/3	36.0	(1.4)	3/3	251.7	(2.5)	12/12
10	0.5	0.5	67.0	(0.0)	3/3	79.3	(2.1)	3/3	39.3	(1.2)	3/3	32.0	(0.8)	3/3	217.7	(3.8)	12/12
10	0.5	1.0	63.3	(0.5)	3/3	67.7	(0.5)	3/3	35.0	(2.2)	3/3	30.7	(0.5)	3/3	196.7	(3.3)	12/12
10	1.0	0.1	67.3	(2.6)	3/3	95.0	(6.4)	3/3	41.3	(0.9)	3/3	33.0	(4.2)	3/3	236.7	(8.6)	12/12
10	1.0	0.25	58.7	(1.7)	3/3	73.0	(2.4)	3/3	32.3	(0.9)	3/3	25.3	(0.5)	3/3	189.3	(4.0)	12/12
10	1.0	0.5	50.7	(0.5)	3/3	61.0	(0.8)	3/3	28.7	(1.2)	3/3	23.7	(1.2)	3/3	164.0	(1.4)	12/12
10	1.0	1.0	47.0	(0.8)	3/3	54.0	(1.6)	3/3	25.7	(1.2)	3/3	22.0	(0.8)	3/3	148.7	(1.2)	12/12
100	0.25	0.1	83.0	(4.9)	3/3	119.7	(8.3)	3/3	44.3	(2.6)	3/3	36.3	(1.2)	3/3	283.3	(7.8)	12/12
100	0.25	0.25	63.3	(3.4)	3/3	78.0	(2.4)	3/3	37.3	(0.5)	3/3	28.0	(0.8)	3/3	206.7	(5.9)	12/12
100	0.25	0.5	54.7	(0.5)	3/3	58.7	(0.5)	3/3	31.0	(0.8)	3/3	26.0	(1.4)	3/3	170.3	(1.9)	12/12
100	0.25	1.0	50.7	(0.9)	3/3	51.3	(0.9)	3/3	30.0	(0.0)	3/3	26.7	(1.2)	3/3	158.7	(1.2)	12/12
100	0.5	0.1	52.3	(4.2)	3/3	83.3	(1.9)	3/3	32.0	(0.8)	3/3	22.0	(0.8)	3/3	189.7	(3.8)	12/12
100	0.5	0.25	42.7	(3.9)	3/3	65.7	(3.7)	3/3	22.3	(0.5)	3/3	19.0	(0.8)	3/3	149.7	(6.2)	12/12
100	0.5	0.5	37.7	(3.1)	3/3	41.0	(1.4)	3/3	20.7	(0.5)	3/3	17.7	(0.5)	3/3	117.0	(4.2)	12/12
100	0.5	1.0	34.3	(0.5)	3/3	36.7	(0.5)	3/3	20.0	(0.8)	3/3	17.3	(0.9)	3/3	108.3	(1.2)	12/12
100	1.0	0.1	35.0	(4.5)	3/3	64.3	(3.8)	3/3	21.0	(2.4)	3/3	14.0	(0.8)	3/3	134.3	(8.1)	12/12
100	1.0	0.25	34.3	(6.6)	3/3	45.0	(6.7)	3/3	15.0	(0.8)	3/3	11.0	(0.0)	1/3	98.0	(12.8)	10/12
100	1.0	0.5	23.7	(0.5)	3/3	29.0	(1.4)	3/3	13.7	(0.9)	3/3	11.0	(0.0)	1/3	70.0	(3.6)	10/12
100	1.0	1.0	21.3	(0.5)	3/3	23.7	(1.7)	3/3	12.0	(0.0)	3/3	10.0	(0.0)	1/3	60.3	(6.2)	10/12
250	0.25	0.1	63.0	(7.8)	3/3	112.3	(7.0)	3/3	39.7	(1.7)	3/3	29.0	(0.8)	3/3	244.0	(9.9)	12/12
250	0.25	0.25	53.0	(5.0)	3/3	76.3	(4.1)	3/3	29.3	(0.9)	3/3	23.0	(0.8)	3/3	181.7	(7.8)	12/12
250	0.25	0.5	45.3	(2.1)	3/3	56.7	(4.1)	3/3	26.3	(1.2)	3/3	21.3	(0.5)	3/3	149.7	(4.5)	12/12
250	0.25	1.0	42.7	(0.5)	3/3	47.3	(1.2)	3/3	25.7	(0.9)	3/3	22.3	(0.5)	3/3	138.0	(0.8)	12/12
250	0.5	0.1	38.0	(3.6)	3/3	86.0	(12.1)	3/3	24.7	(0.5)	3/3	17.7	(0.9)	3/3	166.3	(14.1)	12/12
250	0.5	0.25	36.0	(5.0)	3/3	53.7	(3.3)	3/3	20.0	(1.4)	3/3	14.7	(0.5)	3/3	124.3	(6.8)	12/12
250	0.5	0.5	29.0	(0.8)	3/3	36.0	(2.9)	3/3	14.7	(0.5)	3/3	13.7	(0.9)	3/3	93.3	(4.7)	12/12
250	0.5	1.0	27.0	(0.0)	3/3	29.3	(1.2)	3/3	15.0	(0.8)	3/3	13.7	(0.5)	3/3	85.0	(1.4)	12/12
250	1.0	0.1	32.7	(2.5)	3/3	62.3	(6.1)	3/3	19.7	(5.4)	3/3	0.0	(0.0)	0/3	114.7	(10.2)	9/12
250	1.0	0.25	25.3	(2.6)	3/3	37.3	(3.1)	3/3	11.3	(1.2)	3/3	0.0	(0.0)	0/3	74.0	(5.4)	9/12
250	1.0	0.5	19.0	(0.0)	3/3	21.7	(0.5)	3/3	9.3	(0.5)	3/3	10.0	(0.0)	1/3	53.3	(4.0)	10/12
250	1.0	1.0	17.0	(0.8)	3/3	18.3	(0.9)	3/3	9.0	(0.0)	3/3	0.0	(0.0)	0/3	44.3	(1.2)	9/12

Table 5.7: Number of column generation iterations when solving the root node for different parameter settings. For each configuration and each shift, we considered three different random seeds, and show the average and standard deviation (in brackets) of the number of iterations over these seeds (taken over the runs that terminated in time), together with the number of runs that finished within 6 hours. The parameters are: the number of paths returned per RCSPP (Nr), the similarity threshold (Sim), and the percentage of pricing problems solved in partial pricing (Perc).

Chapter 6

Summary and Conclusions

In this thesis, we considered novel optimization problems aimed at further integrating the crew planning process at Netherlands Railways (NS), thereby setting the next step towards decision support for integrated crew planning at NS. In Chapters 2 and 3, we focused on the combination of fairness and attractiveness, thereby extending the 'Sharing-Sweet-and-Sour' rules to crew rostering. In Chapter 4, we gave an in-depth analysis of modeling techniques for crew rostering and provided insight in the solution approaches used in Chapters 2 and 3. Finally, in Chapter 5, we proposed an integrated approach towards crew re-planning (i.e., updating the crew schedules due to planned maintenance), where we exploit additional freedom in the crew rosters to efficiently re-schedule the crew after disruptions.

6.1 Main Findings

In Chapter 2, we introduced the Fairness-oriented Crew Rostering Problem (FCRP). In this problem, fair and attractive cyclic rosters have to be constructed for groups of employees. We analyzed a class of resource allocation problems, in which the resource allocation is based on approximate utility functions. We considered a fairness scheme for this class, based on the 'Sharing-Sweet-and-Sour' rules, and we derived a tight upper bound on the price of fairness for this scheme, which showed that the price of fairness is highest for instances with a large number of groups, and instances with large differences in group sizes. Furthermore, we developed an exact Branch-

Price-and-Cut solution method, based on a novel mathematical formulation. We applied our solution approach to practical instances of NS, where we showed that our integrated approach leads to a diverse set of solutions. We concluded that, in order to generate high-quality rosters, the explicit trade-off between fairness and attractiveness should be taken into account. In particular, the analysis of these solutions led to two important insights. Firstly, decision makers should be careful not to 'over-optimize' fairness. We observed that by loosening the fairness requirements slightly, the attractiveness could be greatly improved, thereby showing the possible suboptimality of a sequential approach. Secondly, we found that the decrease in attractiveness caused by a tight fairness level is unevenly distributed over the different roster groups.

In Chapter 3, we considered the Cyclic Crew Rostering Problem with Fairness Requirements (CCRP-FR), a variant of the FCRP in which the rosters have to be constructed for a fixed, a priori known, set of fairness levels. We proposed a three-phase heuristic for the CCRP-FR, which is complementary to the exact approach developed in Chapter 2: The exact approach is able to solve reasonably sized instances to optimality in a few hours, whereas the heuristic quickly finds good solutions for the larger instances. The three-phase heuristic combines the strengths of the exact approach with a large-scale neighborhood search algorithm. The design of the heuristic assures that good solutions for all fairness levels are obtained quickly, and can still be further improved if additional running time is available. We evaluated the performance of the heuristic using real-world instances from NS, found close to optimal solutions for most instances, and achieved a major improvement (up to 40%) over the current (sequential) approach.

In Chapter 4, we provided an in-depth analysis of different formulations for the CCRP, the problem underlying both the FCRP and CCRP-FR. In doing so, we showed that the row-based formulations of Chapters 2 and 3 are the best-suited formulations for the roster constraints at NS. We proposed a family of formulations, and derived analytical results regarding the relative strength of the proposed formulations. The family of formulations can be seen as a generalization of the typical assignment and set partitioning formulations, and is motivated by the poor performance of assignment formulations on difficult instances. Furthermore, we discussed modeling techniques and provided tightness results for an important class of roster constraints. We adapted the branch-price-and-cut approach of Chapter 2 to the family of formulations, and showed the benefit of a suitably picked formulation using

practical instances from NS.

Finally, we introduced the Integrated Crew Re-Planning Problem (ICRPP) in Chapter 5: an integrated approach for crew re-scheduling over multiple days. In doing so, we extended the Crew Re-Scheduling Problem (CRSP), by allowing more flexibility in the newly assigned duties. The additional complexity of the ICRPP resides in the problem size, as multiple days need to be re-scheduled at once, and in the fact that the feasibility of the rosters should be taken into account explicitly. We proposed a heuristic diving framework based on column generation, where alternative rosters are selected in a sequential fashion. We applied the solution method to four instances, based on real-world data from NS. We analyzed the benefit of an integrated approach and considered the trade-off between the number of necessary duties and the deviation from the original plan. The results showed a clear gain from integrating the solution process, yet also showed that the performance of the heuristic varies among different runs.

6.2 Practical Implications and Recommendations

The research presented in this thesis has been successfully applied at NS. Around March 2018, crew base Amersfoort lacked the necessary experience to construct the rosters for the coming year, due to the retirement of senior employees responsible for constructing the rosters. As a result, the rostering process took much longer and the quality of the final rosters was worse, implying both higher operational costs and less satisfied personnel. These circumstances triggered a sense of urgency to automate and optimize the rostering process, and hence the department Process quality and Innovation of NS started the development of a decision support tool for crew rostering (in collaboration with crew base Amersfoort). The algorithm underlying the decision support tool builds upon the exact approach developed in Chapter 2, and assured that the tool provided good operational rosters before the planners would start the rostering process.

The decision support tool led to a win-win situation for both the crew and the operator. By implementing the 'Sharing-Sweet-and-Sour' rules on the crew base level, we were able to construct high-quality rosters, while simultaneously taking the allocation of work into account. As with duty scheduling, this would be too complex to incorporate in the manual process. The planning tool not only reduced the time needed for the rostering process by a factor three, from three weeks to a single week,

but also produced rosters which were considered to be of very high quality, according to both planners and managers. The key to success was a combination of state-of-the-art Operations Research techniques with a highly participative approach, in which the feedback of the planners was constantly used to improve the algorithm. The rosters constructed using the tool are currently used in practice, and NS is considering to standardize and develop the planning tool for all crew bases in the Netherlands.

Based on the research in this thesis and the success of the pilot study, we recommend to incorporate the solution methods presented in this thesis in full-fledged decision support tools. Such tools should provide quick and transparent feedback to the user (e.g., a roster committee), and, to assure the former, should most likely be based on heuristics. The subjectivity of fairness and attractiveness implies that the tool should be easily adaptable to the users' need, without requiring too much knowledge about the underlying algorithm. The exact design of such a tool should be determined in extensive collaboration with the end users. The exact solution methods developed in this thesis should be applied on a 'tactical' level: They should provide insights in the effect of different roster rules and fairness metrics, fuel the discussion on a suitable trade-off between fairness and attractiveness, and allow for a rigorous performance evaluation of heuristics.

6.3 Further Research

The problems in this thesis pose interesting challenges and directions for further research. The integration of fairness in crew rostering, for example, leads to a challenging optimization problem. Although we developed exact and heuristic methods for the integrated problem, other solution strategies could possibly lead to improvements. In particular, non-trivial incorporation of the balancing aspects of fairness into a branch-and-bound framework, using e.g., constraint propagation or fixing strategies with provable bounds, could lead to more efficient algorithms.

The general modeling framework presented in Chapter 4 could be used to develop, and identify, efficient algorithms for classes of crew rostering problems, leading to a unified framework of exact solution methods for crew rostering in public transportation.

Finally, the integrated crew re-planning problem seems well-suited for a 'decompose

and synchronize' approach in which the re-scheduling problem is decomposed per day and 'synchronization' constraints (e.g., Benders cuts) assure the feasibility of the rosters. An advantage of such an approach is that it closely resembles practice, where problems are often solved in parallel, yet it should be evaluated whether such an approach is competitive with an integrated approach.

References

Abbink, E. (2014). "Crew Management in Passenger Rail Transport". PhD thesis. Erasmus Research Institute of Management (ERIM).

Abbink, E., M. Fischetti, L. Kroon, G. Timmer and M. Vromans (2005). "Reinventing crew scheduling at Netherlands Railways". *Interfaces* 35.5, pp. 393–401.

Abbink, E., D. Huisman and L. Kroon (2018). "Railway Crew Management". *Handbook of Optimization in the Railway Industry*. Ed. by R. Borndörfer, T. Klug, L. Lamorgese, C. Mannino, M. Reuther and T. Schlechte. Cham: Springer International Publishing, pp. 243–264.

Ahuja, R.K., Ö. Ergun, J.B. Orlin and A.P. Punnen (2002). "A survey of very large-scale neighborhood search techniques". *Discrete Applied Mathematics* 123.1-3, pp. 75–102.

Barnhart, C., D. Bertsimas, C. Caramanis and D. Fearing (2012). "Equitable and efficient coordination in traffic flow management". *Transportation Science* 46.2, pp. 262–280.

Barnhart, C., 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, pp. 316–329.

Beasley, J.E. and N. Christofides (1989). "An algorithm for the resource constrained shortest path problem". *Networks* 19.4, pp. 379–394.

Bertsimas, D., V.F. Farias and N. Trichakis (2011). "The price of fairness". *Operations Research* 59.1, pp. 17–31.

Bertsimas, D., V.F. Farias and N. Trichakis (2012). "On the efficiency-fairness trade-off". *Management Science* 58.12, pp. 2234–2250.

Bertsimas, D., V.F. Farias and N. Trichakis (2013). "Fairness, efficiency, and flexibility in organ allocation for kidney transplantation". *Operations Research* 61.1, pp. 73–87.

Bertsimas, D. and S. Gupta (2015). “Fairness and collaboration in network air traffic flow management: an optimization approach”. *Transportation Science* 50.1, pp. 57–76.

Borndörfer, R., A. Langenhan, A. Löbel, C. Schulz and S. Weider (2013). “Duty scheduling templates”. *Public Transport* 5.1-2, pp. 41–51.

Borndörfer, R., M. Reuther, T. Schlechte, C. Schulz, E. Swarat and S. Weider (2015). “Duty Rostering in Public Transport-Facing Preferences, Fairness, and Fatigue”. *CASPT*.

Borndörfer, R., C. Schulz, S. Seidl and S. Weider (2017). “Integration of duty scheduling and rostering to increase driver satisfaction”. *Public Transport* 9.1, pp. 177–191.

Breugem, T., R. Borndörfer, T. Schlechte and C. Schulz (2019). *A Three-Phase Heuristic for Cyclic Crew Rostering with Fairness Requirements*. eng. Tech. rep. 19-43. Takustr. 7, 14195 Berlin: ZIB.

Breugem, T., T. Dollevoet and D. Huisman (2017). *Is Equality always desirable? Analyzing the Trade-Off between Fairness and Attractiveness in Crew Rostering*. Tech. rep. EI2017-30. Econometric Institute.

Breugem, T., T. Dollevoet and D. Huisman (2018). *Analyzing a Family of Formulations for Cyclic Crew Rostering*. Tech. rep. EI2018-35. Econometric Institute.

Burke, E.K., P. De Causmaecker, G. Vanden Berghe and H. Van Landeghem (2004). “The state of the art of nurse rostering”. *Journal of Scheduling* 7.6, pp. 441–499.

Cacchiani, V., D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf and J. Wagenaar (2014). “An overview of recovery models and algorithms for real-time railway rescheduling”. *Transportation Research Part B: Methodological* 63, pp. 15–37.

Caprara, A., M. Fischetti, P. Toth, D. Vigo and P.L. Guida (1997). “Algorithms for railway crew management”. *Mathematical Programming* 79.1-3, pp. 125–141.

Caprara, A., L. Kroon, M. Monaci, M. Peeters and P. Toth (2007). “Passenger railway optimization”. *Handbooks in Operations Research and Management Science* 14, pp. 129–187.

Clausen, J., A. Larsen, J. Larsen and N.J. Rezanova (2010). “Disruption management in the airline industry-Concepts, models and methods”. *Computers & Operations Research* 37.5, pp. 809–821.

Colquitt, J.A., D.E. Conlon, M.J. Wesson, C.O. Porter and K. Yee Ng (2001). “Justice at the Millennium: A Meta-Analytic Review of 25 Years of Organizational Justice Research”. *Journal of Applied Psychology* 86.3, pp. 425–445.

Colquitt, J.A., B.A. Scott, J.B. Rodell, D.M. Long, C.P. Zapata, D.E. Conlon and M.J. Wesson (2013). "Justice at the millennium, a decade later: a meta-analytic test of social exchange and affect-based perspectives." *Journal of Applied Psychology* 98.2, pp. 199–236.

Cordeau, J.F., G. Stojković, F. Soumis and J. Desrosiers (2001). "Benders decomposiion for simultaneous aircraft routing and crew scheduling". *Transportation science* 35.4, pp. 375–388.

Dantzig, G.B. (1954). "Letter to the editor-A comment on Edie's "Traffic delays at toll booths"". *Journal of the Operations Research Society of America* 2.3, pp. 339–341.

De Causmaecker, P. and G. Vanden Berghe (2011). "A categorisation of nurse rostering problems". *Journal of Scheduling* 14.1, pp. 3–16.

Desaulniers, G., J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis (1997). "Daily aircraft routing and scheduling". *Management Science* 43.6, pp. 841–855.

Desaulniers, G., J. Desrosiers and M.M. Solomon (2002). "Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems". *Essays and surveys in metaheuristics*. Springer, pp. 309–324.

Desaulniers, G., J. Desrosiers and M.M. Solomon (2006). *Column Generation*. Vol. 5. Springer Science & Business Media.

Desrochers, M. and F. Soumis (1989). "A column generation approach to the urban transit crew scheduling problem". *Transportation Science* 23.1, pp. 1–13.

Desrosiers, J. and M.E. Lübbecke (2011). "Branch-Price-and-Cut Algorithms". *Wiley Encyclopedia of Operations Research and Management Science*.

Dumitrescu, I. and N. Boland (2003). "Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem". *Networks* 42.3, pp. 135–153.

Ehrgott, M. (2000). *Multicriteria optimization*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.

Ernst, A.T., H. Jiang, M. Krishnamoorthy, H. Nott and D. Sier (2001). "An integrated optimization model for train crew management". *Annals of Operations Research* 108.1-4, pp. 211–224.

Ernst, A.T., H. Jiang, M. Krishnamoorthy and D. Sier (2004). "Staff scheduling and rostering: A review of applications, methods and models". *European Journal of Operational Research* 153.1, pp. 3–27.

Freling, R., 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 algorithm”. *Annals of Operations Research* 127.1-4, pp. 203–222.

Greenberg, J. (1990). “Organizational justice: Yesterday, today, and tomorrow”. *Journal of Management* 16.2, pp. 399–432.

Grötschel, M., R. Borndörfer and A. Löbel (2003). “Duty scheduling in public transit”. *Mathematics-Key Technology for the Future*. Springer, pp. 653–674.

Hart, P.E., 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, pp. 100–107.

Hartog, A., D. Huisman, E. Abbink and L. Kroon (2009). “Decision support for crew rostering at NS”. *Public Transport* 1.2, pp. 121–133.

Heil, J., K. Hoffmann and U. Buscher (2019). “Railway crew scheduling: Models, methods and applications”. *European Journal of Operational Research*.

Hoffman, K.L. and M. Padberg (1993). “Solving airline crew scheduling problems by branch-and-cut”. *Management Science* 39.6, pp. 657–682.

Huisman, D. (2007). “A column generation approach for the rail crew re-scheduling problem”. *European Journal of Operational Research* 180.1, pp. 163–173.

Huisman, D., R. Freling and A.P.M. Wagelmans (2005a). “Multiple-depot integrated vehicle and crew scheduling”. *Transportation Science* 39.4, pp. 491–502.

Huisman, D., L.G. Kroon, R.M. Lentink and M.J.C.M. Vromans (2005b). “Operations Research in Passenger Railway Transportation”. *Statistica Neerlandica* 59.4, pp. 467–497.

Irnich, S. and G. Desaulniers (2005). “Shortest path problems with resource constraints”. *Column Generation*. Springer, pp. 33–65.

Johnson, D.S. and L.A. McGeoch (1997). “The traveling salesman problem: A case study in local optimization”. *Local search in combinatorial optimization* 1.1, pp. 215–310.

Joncour, C., S. Michel, R. Sadykov, D. Sverdlov and F. Vanderbeck (2010). “Column generation based primal heuristics”. *Electronic Notes in Discrete Mathematics* 36, pp. 695–702.

Kalai, E. and M. Smorodinsky (1975). “Other solutions to Nash’s bargaining problem”. *Econometrica: Journal of the Econometric Society*, pp. 513–518.

Kohl, N. and S.E. Karisch (2004). “Airline crew rostering: Problem types, modeling, and optimization”. *Annals of Operations Research* 127.1-4, pp. 223–257.

Kroon, L. and M. Fischetti (2001). "Crew scheduling for Netherlands railways destination: customer". *Computer-aided scheduling of public transport*. Springer, pp. 181–201.

Kroon, L., D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, A. Schrijver, A. Steenbeek and R. Ybema (2009). "The New Dutch Timetable: The OR Revolution ". *Interfaces* 39.1, pp. 6–17.

Lavoie, S., M. Minoux and E. Odier (1988). "A new approach for crew pairing problems by column generation with an application to air transportation". *European Journal of Operational Research* 35.1, pp. 45–58.

Lettovskỳ, L., E.L. Johnson and G.L. Nemhauser (2000). "Airline crew recovery". *Transportation Science* 34.4, pp. 337–348.

Lin, S. and B.W. Kernighan (1973). "An effective heuristic algorithm for the traveling-salesman problem". *Operations Research* 21.2, pp. 498–516.

Löbel, A. (1998). "Vehicle scheduling in public transit and Lagrangean pricing". *Management Science* 44.12-part-1, pp. 1637–1649.

Lübbecke, M.E. (2011). "Column generation". *Wiley Encyclopedia of Operations Research and Management Science*.

Lübbecke, M.E. and J. Desrosiers (2005). "Selected topics in column generation". *Operations Research* 53.6, pp. 1007–1023.

Maenhout, B. and M. Vanhoucke (2010). "A hybrid scatter search heuristic for personalized crew rostering in the airline industry". *European Journal of Operational Research* 206.1, pp. 155–167.

Mesquita, M., M. Moz, A. Paias and M. Pato (2013). "A decomposition approach for the integrated vehicle-crew-roster problem with days-off pattern". *European Journal of Operational Research* 229.2, pp. 318–331.

Mesquita, M., M. Moz, A. Paias and M. Pato (2015). "A decompose-and-fix heuristic based on multi-commodity flow models for driver rostering with days-off pattern". *European Journal of Operational Research* 245.2, pp. 423–437.

Nash, J. (1950). "The bargaining problem". *Econometrica: Journal of the Econometric Society*, pp. 155–162.

Nishi, T., T. Sugiyama and M. Inuiguchi (2014). "Two-level decomposition algorithm for crew rostering problems with fair working condition". *European Journal of Operational Research* 237.2, pp. 465–473.

Petersen, J.D., G. Sölveling, J. Clarke, E.L. Johnson and S. Shebalov (2012). "An optimization approach to airline integrated recovery". *Transportation Science* 46.4, pp. 482–500.

Pisinger, D. and S. Ropke (2010). “Large neighborhood search”. *Handbook of metaheuristics*. Springer, pp. 399–419.

Potthoff, D., D. Huisman and G. Desaulniers (2010). “Column generation with dynamic duty selection for railway crew rescheduling”. *Transportation Science* 44.4, pp. 493–505.

Rezanova, N.J. and D.M. Ryan (2010). “The train driver recovery problem-a set partitioning based model and solution method”. *Computers & Operations Research* 37.5, pp. 845–856.

Rhoades, L. and R. Eisenberger (2002). “Perceived Organizational Support: A Review of the Literature”. *Journal of Applied Psychology* 87.4, pp. 698–714.

Sato, K. and N. Fukumura (2012). “Real-time freight locomotive rescheduling and uncovered train detection during disruption”. *European Journal of Operational Research* 221.3, pp. 636–648.

Sodhi, M. and S. Norris (2004). “A flexible, fast, and optimal modeling approach applied to crew rostering at London Underground”. *Annals of Operations Research* 127.1-4, pp. 259–281.

Stojković, M. and F. Soumis (2001). “An optimization model for the simultaneous operational flight and pilot scheduling problem”. *Management Science* 47.9, pp. 1290–1305.

Stojković, M., F. Soumis and J. Desrosiers (1998). “The operational airline crew scheduling problem”. *Transportation Science* 32.3, pp. 232–245.

Van den Bergh, J., J. Beliën, P. De Bruecker, E. Demeulemeester and L. De Boeck (2013). “Personnel scheduling: A literature review”. *European Journal of Operational Research* 226.3, pp. 367–385.

Veelenturf, L.P., D. Potthoff, D. Huisman and L.G. Kroon (2012). “Railway crew rescheduling with retiming”. *Transportation research part C: emerging technologies* 20.1, pp. 95–110.

Veelenturf, L.P., D. Potthoff, D. Huisman, L.G. Kroon, G. Maróti and A.P.M. Wagelmans (2014). “A quasi-robust optimization approach for crew rescheduling”. *Transportation Science* 50.1, pp. 204–215.

Walker, C.G., J.N. Snowdon and D.M. Ryan (2005). “Simultaneous disruption recovery of a train timetable and crew roster in real time”. *Computers & Operations Research* 32.8, pp. 2077–2094.

Xie, L. and L. Suhl (2015). “Cyclic and non-cyclic crew rostering problems in public bus transit”. *OR Spectrum* 37.1, pp. 99–136.

Nederlandse samenvatting (Summary in Dutch)

Er zullen binnen Nederland weinig bedrijven zijn die meer invloed hebben op het nationale debat dan de Nederlandse Spoorwegen (NS). En dit is ook niet zo gek: Nederland reist volop met de trein en ongeveer 90% van de treinreizigers reist met NS. Zoals besproken in hoofdstuk 1, kunnen als gevolg van dit veelvuldig gebruik de sociale kosten van verstoringen, werkzaamheden en stakingen van NS-personeel oplopen tot in de honderden miljoenen euro's. Personeelsplanning speelt een belangrijke rol in het beperken van deze kosten: Bij verstoringen en werkzaamheden dient NS op een zo slim mogelijke manier het geplande werk van het personeel aan te passen om zoveel mogelijk treinen te kunnen blijven rijden, terwijl stakingen voortkomen uit onvrede onder het personeel en verholpen kunnen worden door hun wensen mee te nemen in het personeelsplanningsproces. NS heeft de afgelopen jaren dan ook veel tijd en energie gestoken in de ontwikkeling van beslissingsondersteunende software voor personeelsplanning. Desondanks blijft er altijd ruimte voor verbetering.

In dit proefschrift zetten we de volgende stap in beslissingsondersteuning voor personeelsplanning bij NS. Hierbij kijken we naar personeelsplanning op de middellange termijn, waarbij het doel is diensten (werkdagen) en roosters (toewijzingen van werkdagen aan personeel) te maken zodat uiteindelijk al het geplande werk toegewezen is aan een machinist of conducteur. Hierbij nemen we drie criteria in acht: eerlijkheid, kwaliteit en effectiviteit. De effectiviteit van de roosters heeft betrekking op het aantal personeelsleden dat nodig is om het plan uit te voeren. Eerlijkheid, daarentegen, heeft betrekking op de verdeling van het werk. Bij NS wordt eerlijkheid uitgedrukt aan de hand van 'Lusten-en-Lasten-Delen', een lijst van regels die ervoor zorgt dat de 'lusten' en 'lasten' van het werk (zoals bijvoorbeeld leuke ritten of lange diensten)

evenredig verdeeld worden over de personeelsleden. De kwaliteit, ten slotte, relateert aan aspecten zoals de rusttijd tussen diensten en werkvariatie binnen het rooster. Hierin dient een veelvoud aan complexe regels, afkomstig uit de cao, in acht genomen te worden, wat ervoor zorgt dat het maken van een kwalitatief hoogstaand rooster een complexe aangelegenheid is.

In de verschillende hoofdstukken bekijken we nieuwe optimalisatievraagstukken en bijbehorende oplosmethoden die verscheidene aspecten van het personeelsplanningsproces verder integreren. In hoofdstukken 2 en 3 kijken we naar de combinatie van eerlijkheid en kwaliteit bij het maken van roosters, stellen we onszelf de vraag "is eerlijkheid altijd gewenst?" en laten we zien dat het antwoord hierop afhangt van de expliciete afweging tussen eerlijkheid en kwaliteit. In hoofdstuk 5 combineren we het aanpassen van de diensten en roosters om zo op een slimmere manier te kunnen reageren wanneer de dienstregeling aangepast moet worden door geplande werkzaamheden aan het spoor. De gepresenteerde optimalisatieproblemen en wiskundige formuleringen zijn zorgvuldig onderbouwd met theoretische resultaten. In hoofdstuk 2, bijvoorbeeld, laten we zien dat 'Lusten-en-Lasten-Delen' optimale eigenschappen heeft onder redelijke aannames en laten we ook het theoretisch hoogst mogelijke verlies van kwaliteit zien door het gebruik van deze regels. In hoofdstuk 4 duiken we in de wiskundige modellen voor personeelsroostering en laten we wiskundig zien welke modellen sterk zijn en hoe deze te kiezen. Uiteindelijk passen we alle oplosmethoden toe op data van NS. De oplosmethoden ontwikkeld in de eerste drie hoofdstukken worden ieder toegepast op data verkregen van standplaats Utrecht, en de oplosmethode gepresenteerd in hoofdstuk 5 wordt geëvalueerd op basis van daadwerkelijke buitendienststellingen rondom station Leiden Centraal. Door de praktische data van NS te gebruiken om de oplosmethoden te evalueren kan een beter beeld verkregen worden van de meerwaarde van de verschillende methoden.

Het onderzoek in deze thesis heeft reeds geleid tot nieuwe beslissingsondersteuning bij NS. Begin 2018 leidde een vraag van standplaats Amersfoort ertoe dat NS begon met het ontwikkelen van roosteringssoftware. De standplaats kampte met een tekort aan ervaren roostermakers, waardoor het maken van de roosters niet alleen langer duurde maar ook roosters van mindere kwaliteit opleverde. Door middel van een goede en snelle samenwerking tussen centrale planners en de roostercommissie van standplaats Amersfoort, waarin de feedback van de planners van standplaats Amersfoort voortdurend gebruikt werd om de oplosmethode te verbeteren, werd er een tool ontwikkeld op basis van de oplosmethode gepresenteerd in hoofdstuk 2. Met

behulp van de uiteindelijke tool werd het roosterproces flink verkort, van drie weken naar één week, en werden kwalitatief hoogstaande roosters in de ogen van zowel planners als managers verkregen. De succesvolle ontwikkeling van de rooseringstool laat de potentie zien van verdere beslissingsondersteuning bij NS. Dit is echter niet altijd eenvoudig. Het ontwikkelen van een succesvolle tool vereist dat men het 'echte' onderliggende probleem goed begrijpt en ook een goede samenwerking met de eindgebruiker is onmisbaar om het eindproduct te laten slagen. Voor ons als onderzoekers betekent dit dat we verder moeten kijken dan enkel 'de wiskunde achter het probleem' en dit is zeker geen gemakkelijke klus.

About the author

As a researcher, Thomas aims at applying state-of-the-art Operations Research techniques to complex practical problems. His current work focuses on public transportation and humanitarian logistics. In both fields, a thorough understanding of the practical context is necessary to develop efficient decision support. In 2017, Thomas was awarded the INFORMS Railway Applications Section Best Student Paper Award for his work on integrating fairness into the crew rostering process at Netherlands Railways. Furthermore, he has published in scientific journals such as *Transportation Research Part B: Methodological* and *Computers and Operations Research*.

Thomas holds an MSc degree, summa cum laude, in Econometrics and Management Science, with a specialization in Operations Research and Quantitative Logistics from the Erasmus University Rotterdam. In 2015, Thomas started his PhD candidacy at the Erasmus Research Institute of Management at the Erasmus University Rotterdam. Thomas presently works as a Post-Doctoral Research Fellow at INSEAD's Technology and Operations Management department. Within this department, Thomas is affiliated with the Humanitarian Research Group, a group that focuses on impactful practice-based research in line with the UN Sustainable Development Goals.

Portfolio

Publications

International Journals

G.J. Polinder, T. Breugem, T. Dollevoet, G. Maróti (2019). "An adjustable robust optimization approach for periodic timetabling". *Transportation Research Part B: Methodological* 128, pp. 50-68.

T. Breugem, T. Dollevoet, and W. van den Heuvel (2017). "Analysis of FPTASes for the multi-objective shortest path problem". *Computers & Operations Research* 78, pp. 44-58.

Dutch Journals

T. Breugem (2018). "Is gelijkheid altijd gewenst? De afweging tussen gelijkheid en kwaliteit in personeelsroostering". *STAtOR* 19.2, pp. 30–33.

Technical Reports

T. Breugem, T. Dollevoet, D. Huisman (2019). *A Column Generation Approach for the Integrated Crew Re-Planning Problem.* Tech. rep. EI2019-31. Econometric Institute.

T. Breugem, R. Borndörfer, T. Schlechte and C. Schulz (2019). *A Three-Phase Heuristic for Cyclic Crew Rostering with Fairness Requirements.* Tech. rep. 19-43. ZIB.

T. Breugem, T. Dollevoet and D. Huisman (2018). *Analyzing a Family of Formulations for Cyclic Crew Rostering.* Tech. rep. EI2018-35. Econometric Institute.

T. Breugem, T. Dollevoet and D. Huisman (2017). *Is Equality always desirable? Analyzing the Trade-Off between Fairness and Attractiveness in Crew Rostering.* Tech. rep. EI2017-30. Econometric Institute.

Teaching

Introductory Seminar Case Studies Econometrics and Operations Research, Erasmus School of Economics, Econometrics and Operations Research. Supervision. 2018-2019.
Bachelor Thesis Econometrics and Operations Research, Erasmus School of Economics, Econometrics and Operations Research. Supervision. 2017-2019.
Analysis, Erasmus School of Economics, Econometrics and Operations Research. Tutorial lecturer and exercise lectures. 2016-2018.
Public Transport (Golden School Series), Beijing Jiaotong University. Preparation of course material and lectures. 2017.
Teaching Assistant, Erasmus School of Economics, Econometrics and Operations Research. Tutorial lecturer for a variety of courses. 2012-2015.

PhD Courses

Convex Analysis for Optimization
Networks and Polyhedra
Robust Optimization
Stochastic Programming
Integer Programming Methods
Algorithms and Complexity
Interior Point Methods
Public Transport
Constraint Programming
Publishing Strategy
Scientific Integrity
English (CPE certificate)

Conferences Attended

EURO 2019, Dublin, Ireland.
INFORMS 2018, Phoenix, USA.
EURO 2018, Valencia, Spain.
LNMB conference 2018, Lunteren, The Netherlands.
INFORMS 2017, Houston, USA.
IFORS 2017, Quebec, Canada.
EURO 2016, Poznan, Poland.
TRISTAN 2016, Palm Beach, Aruba.

The ERIM PhD Series

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://repub.eur.nl/pub. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam (EUR).

Dissertations in the last four years

Ahmadi, S., *A motivational perspective to decision-making and behavior in organizations*, Promotors: Prof. J.J.P. Jansen & Prof. T.J.M. Mom, EPS-2019-477-S&E, https://repub.eur.nl/pub/116727

Akemu, O., *Corporate Responses to Social Issues: Essays in Social Entrepreneurship and Corporate Social Responsibility*, Promotors: Prof. G.M. Whiteman & Dr. S.P. Kennedy, EPS-2017-392-ORG, https://repub.eur.nl/pub/95768

Albuquerque de Sousa, J.A., *International stock markets: Essays on the determinants and consequences of financial market development*, Promotors: Prof. M.A. van Dijk & Prof. P.A.G. van Bergeijk, EPS-2019-465-F&A, https://repub.eur.nl/pub/115988

Alexiou, A., *Management of Emerging Technologies and the Learning Organization: Lessons from the Cloud and Serious Games Technology*, Promotors: Prof. S.J. Magala, Prof. M.C. Schippers & Dr. I. Oshri, EPS-2016-404-ORG, https://repub.eur.nl/pub/93818

Alserda, G.A.G., *Choices in Pension Management*, Promotors: Prof. S.G. van der Lecq & Dr. O.W. Steenbeek, EPS-2017-432-F&A, https://repub.eur.nl/pub/103496

Arampatzi, E., *Subjective Well-Being in Times of Crises: Evidence on the Wider Impact of Economic Crises and Turmoil on Subjective Well-Being*, Promotors: Prof. H.R. Commandeur, Prof. F. van Oort & Dr. M.J. Burger, EPS-2018-459-S&E, https://repub.eur.nl/pub/111830

Avci, E., *Surveillance of Complex Auction Markets: a Market Policy Analytics Approach*, Promotors: Prof. W. Ketter, Prof. H.W.G.M. van Heck & Prof. D.W. Bunn, EPS-2018-426-LIS, https://repub.eur.nl/pub/106286

Balen, T.H. van, *Challenges of Early Stage Entrepreneurs : the Roles of Vision Communication and Team Membership Change*, Promotors: Prof. J.C.M van den Ende & Dr. M. Tarakci, EPS-2018-468-LIS, https://repub.eur.nl/pub/115654

Benschop, N, *Biases in Project Escalation: Names, frames & construal levels*, Promotors: Prof. K.I.M. Rhode, Prof. H.R. Commandeur, Prof. M. Keil & Dr. A.L.P. Nuijten, EPS-

2015-375-S&E, https://repub.eur.nl/pub/79408

Bernoster, I., *Essays at the Intersection of Psychology, Biology, and Entrepreneurship*, Promotors: Prof. A.R. Thurik, Prof. I.H.A. Franken & Prof. P.J.F Groenen, EPS-2018-463-S&E, https://repub.eur.nl/pub/113907

Beusichem, H.C. van, *Firms and Financial Markets: Empirical Studies on the Informational Value of Dividends, Governance and Financial Reporting*, Promotors: Prof. A. de Jong & Dr. G. Westerhuis, EPS-2016-378-F&A, https://repub.eur.nl/pub/93079

Bouman, P., *Passengers, Crowding and Complexity: Models for Passenger Oriented Public Transport*, Promotors: Prof. L.G. Kroon, Prof. A. Schöbel & Prof. P.H.M. Vervest, EPS-2017-420-LIS, https://repub.eur.nl/pub/100767

Bunderen, L. van, *Tug-of-War: Why and when teams get embroiled in power struggles*, Promotors: Prof. D.L. van Knippenberg & Dr. L. Greer, EPS-2018-446-ORG, https://repub.eur.nl/pub/105346

Burg, G.J.J. van den, *Algorithms for Multiclass Classification and Regularized Regression*, Promotors: Prof. P.J.F. Groenen & Dr. A. Alfons, EPS-2018-442-MKT, https://repub.eur.nl/pub/103929

Chammas, G., *Portfolio concentration*, Promotor: Prof. J. Spronk, EPS-2017-410-F&E, https://repub.eur.nl/pub/94975

Consiglio, I., *Others: Essays on Interpersonal and Consumer Behavior*, Promotor: Prof. S.M.J. van Osselaer, EPS-2016-366-MKT, https://repub.eur.nl/pub/79820

Cranenburgh, K.C. van, *Money or Ethics: Multinational corporations and religious organisations operating in an era of corporate responsibility*, Promotors: Prof. L.C.P.M. Meijs, Prof. R.J.M. van Tulder & Dr. D. Arenas, EPS-2016-385-ORG, https://repub.eur.nl/pub/93104

Darnihamedani, P., *Individual Characteristics, Contextual Factors and Entrepreneurial Behavior*, Promotors: Prof. A.R. Thurik & S.J.A. Hessels, EPS-2016-360-S&E, https://repub.eur.nl/pub/93280

Dennerlein, T., *Empowering Leadership and Employees' Achievement Motivations: the Role of Self-Efficacy and Goal Orientations in the Empowering Leadership Process*, Promotors: Prof. D.L. van Knippenberg & Dr. J. Dietz, EPS-2017-414-ORG, https://repub.eur.nl/pub/98438

Depecik, B.E., *Revitalizing brands and brand: Essays on Brand and Brand Portfolio Management Strategies*, Promotors: Prof. G.H. van Bruggen, Dr. Y.M. van Everdingen and Dr. M.B. Ataman, EPS-2016-406-MKT, https://repub.eur.nl/pub/93507

Duijzer, L.E., *Mathematical Optimization in Vaccine Allocation*, Promotors: Prof. R.

Dekker & Dr. W.L. van Jaarsveld, EPS-2017-430-LIS, https://repub.eur.nl/pub/101487

Duyvesteyn, J.G., *Empirical Studies on Sovereign Fixed Income Markets*, Promotors: Prof. P. Verwijmeren & Prof. M.P.E. Martens, EPS-2015-361-F&A, https://repub.eur.nl/pub/79033

El Nayal, O.S.A.N., *Firms and the State: An Examination of Corporate Political Activity and the Business-Government Interface*, Promotor: Prof. J. van Oosterhout & Dr. M. van Essen, EPS-2018-469-S&E, https://repub.eur.nl/pub/114683

Elemes, A., *Studies on Determinants and Consequences of Financial Reporting Quality*, Promotor: Prof. E. Peek, EPS-2015-354-F&A, https://repub.eur.nl/pub/79037

Erlemann, C., *Gender and Leadership Aspiration: The Impact of the Organizational Environment*, Promotor: Prof. D.L. van Knippenberg, EPS-2016-376-ORG, https://repub.eur.nl/pub/79409

Faber, N., *Structuring Warehouse Management*, Promotors: Prof. M.B.M. de Koster & Prof. A. Smidts, EPS-2015-336-LIS, https://repub.eur.nl/pub/78603

Feng, Y., *The Effectiveness of Corporate Governance Mechanisms and Leadership Structure: Impacts on strategic change and firm performance*, Promotors: Prof. F.A.J. van den Bosch, Prof. H.W. Volberda & Dr. J.S. Sidhu, EPS-2017-389-S&E, https://repub.eur.nl/pub/98470

Fernald, K., *The Waves of Biotechnological Innovation in Medicine: Interfirm Cooperation Effects and a Venture Capital Perspective*, Promotors: Prof. E. Claassen, Prof. H.P.G. Pennings & Prof. H.R. Commandeur, EPS-2015-371-S&E, https://repub.eur.nl/pub/79120

Fisch, C.O., *Patents and trademarks: Motivations, antecedents, and value in industrialized and emerging markets*, Promotors: Prof. J.H. Block, Prof. H.P.G. Pennings & Prof. A.R. Thurik, EPS-2016-397-S&E, https://repub.eur.nl/pub/94036

Fliers, P.T., *Essays on Financing and Performance: The role of firms, banks and board*, Promotors: Prof. A. de Jong & Prof. P.G.J. Roosenboom, EPS-2016-388-F&A, https://repub.eur.nl/pub/93019

Frick, T.W., *The Implications of Advertising Personalization for Firms, Consumer, and Ad Platfroms*, Promotors: Prof. T. Li & Prof. H.W.G.M. van Heck, EPS-2018-452-LIS, https://repub.eur.nl/pub/110314

Fytraki, A.T., *Behavioral Effects in Consumer Evaluations of Recommendation Systems*, Promotors: Prof. B.G.C. Dellaert & Prof. T. Li, EPS-2018-427-MKT, https://repub.eur.nl/pub/110457

Gaast, J.P. van der, *Stochastic Models for Order Picking Systems*, Promotors: Prof. M.B.M

de Koster & Prof. I.J.B.F. Adan, EPS-2016-398-LIS, https://repub.eur.nl/pub/93222

Ghazizadeh, P., *Empirical Studies on the Role of Financial Information in Asset and Capital Markets*, Promotors: Prof. A. de Jong & Prof. E. Peek. EPS-2019-470-F&A https://repub.eur.nl/pub/114023

Giurge, L., *A Test of Time; A temporal and dynamic approach to power and ethics*, Promotors: Prof. M.H. van Dijke & Prof. D. De Cremer, EPS-2017-412-ORG, https://repub.eur.nl/pub/98451

Gobena, L., *Towards Integrating Antecedents of Voluntary Tax Compliance*, Promotors: Prof. M.H. van Dijke & Dr. P. Verboon, EPS-2017-436-ORG, https://repub.eur.nl/pub/103276

Groot, W.A., *Assessing Asset Pricing Anomalies*, Promotors: Prof. M.J.C.M. Verbeek & Prof. J.H. van Binsbergen, EPS-2017-437-F&A, https://repub.eur.nl/pub/103490

Hanselaar, R.M., *Raising Capital: On pricing, liquidity and incentives*, Promotors: Prof. M.A. van Dijk & Prof. P.G.J. Roosenboom, EPS-2018-429-F&A-9789058925404, https://repub.eur.nl/pub/113274

Harms, J. A., *Essays on the Behavioral Economics of Social Preferences and Bounded Rationality*, Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas, EPS-2018-457-S&E, https://repub.eur.nl/pub/108831

Hekimoglu, M., *Spare Parts Management of Aging Capital Products*, Promotor: Prof. R. Dekker, EPS-2015-368-LIS, https://repub.eur.nl/pub/79092

Hendriks, G., *Multinational Enterprises and Limits to International Growth: Links between Domestic and Foreign Activities in a Firm's Portfolio*, Promotors: Prof. P.P.M.A.R. Heugens & Dr. A.H.L Slangen, EPS-2019-464-S&E, https://repub.eur.nl/pub/114981

Hengelaar, G.A., *The Proactive Incumbent: Holy grail or hidden gem? Investigating whether the Dutch electricity sector can overcome the incumbent's curse and lead the sustainability transition*, Promotors: Prof. R.J. M. van Tulder & Dr. K. Dittrich, EPS-2018-438-ORG, https://repub.eur.nl/pub/102953

Hogenboom, A.C., *Sentiment Analysis of Text Guided by Semantics and Structure*, Promotors: Prof. U. Kaymak & Prof. F.M.G. de Jong, EPS-2015-369-LIS, https://repub.eur.nl/pub/79034

Hollen, R.M.A., *Exploratory Studies into Strategies to Enhance Innovation-Driven International Competitiveness in a Port Context: Toward Ambidextrous Ports*, Promotors: Prof. F.A.J. Van Den Bosch & Prof. H.W. Volberda, EPS-2015-372-S&E, https://repub.eur.nl/pub/78881

Jacobs, B.J.D., *Marketing Analytics for High-Dimensional Assortments*, Promotors: Prof. A.C.D. Donkers & Prof. D. Fok, EPS-2017-445-MKT, https://repub.eur.nl/pub/103497

Jia, F., *The Value of Happiness in Entrepreneurship*, Promotors: Prof. D.L. van Knippenberg & Dr. Y. Zhang, EPS-2019-479-ORG, https://repub.eur.nl/pub/115990

Kahlen, M. T., *Virtual Power Plants of Electric Vehicles in Sustainable Smart Electricity Markets*, Promotors: Prof. W. Ketter & Prof. A. Gupta, EPS-2017-431-LIS, https://repub.eur.nl/pub/100844

Kampen, S. van, *The Cross-sectional and Time-series Dynamics of Corporate Finance: Empirical evidence from financially constrained firms*, Promotors: Prof. L. Norden & Prof. P.G.J. Roosenboom, EPS-2018-440-F&A, https://repub.eur.nl/pub/105245

Karali, E., *Investigating Routines and Dynamic Capabilities for Change and Innovation*, Promotors: Prof. H.W. Volberda, Prof. H.R. Commandeur & Dr. J.S. Sidhu, EPS-2018-454-S&E, https://repub.eur.nl/pub/106274

Keko, E., *Essays on Innovation Generation in Incumbent Firms*, Promotors: Prof. S. Stremersch & Dr. N.M.A. Camacho, EPS-2017-419-MKT, https://repub.eur.nl/pub/100841

Kerkkamp, R.B.O., *Optimisation Models for Supply Chain Coordination under Information Asymmetry*, Promotors: Prof. A.P.M. Wagelmans & Dr. W. van den Heuvel, EPS-2018-462-LIS, https://repub.eur.nl/pub/109770

Khattab, J., *Make Minorities Great Again: a contribution to workplace equity by identifying and addressing constraints and privileges*, Promotors: Prof. D.L. van Knippenberg & Dr. A. Nederveen Pieterse, EPS-2017-421-ORG, https://repub.eur.nl/pub/99311

Kim, T. Y., *Data-driven Warehouse Management in Global Supply Chains*, Promotors: Prof. R. Dekker & Dr. C. Heij, EPS-2018-449-LIS, https://repub.eur.nl/pub/109103

Klitsie, E.J., *Strategic Renewal in Institutional Contexts: The paradox of embedded agency*, Promotors: Prof. H.W. Volberda & Dr. S. Ansari, EPS-2018-444-S&E, https://repub.eur.nl/pub/106275

Kong, L., *Essays on Financial Coordination*, Promotors: Prof. M.J.C.M. Verbeek, Dr. D.G.J. Bongaerts & Dr. M.A. van Achter. EPS-2019-433-F&A, https://repub.eur.nl/pub/114516

Koolen, D., *Market Risks and Strategies in Power Systems Integrating Renewable Energy*, Promotors: Prof. W. Ketter & Dr. R. Huisman, EPS-2019-467-LIS, https://repub.eur.nl/pub/115655

Krämer, R., *A license to mine? Community organizing against multinational corporations*, Promotors: Prof. R.J.M. van Tulder & Prof. G.M. Whiteman, EPS-2016-383-ORG,

https://repub.eur.nl/pub/94072

Kyosev, G.S., *Essays on Factor Investing*, Promotors: Prof. M.J.C.M. Verbeek & Dr. J.J. Huij, EPS-2019-474-F&A, https://repub.eur.nl/pub/116463

Lamballais, T., *Optimizing the Performance of Robotic Mobile Fulfillment Systems*, Promotors: Prof. M.B.M de Koster & Prof. R. Dekker & Dr. D. Roy, EPS-2019-411-LIS, https://repub.eur.nl/pub/116477

Lee, C.I.S.G., *Big Data in Management Research: Exploring New Avenues*, Promotors: Prof. S.J. Magala & Dr. W.A. Felps, EPS-2016-365-ORG, https://repub.eur.nl/pub/79818

Legault-Tremblay, P.O., *Corporate Governance During Market Transition: Heterogeneous responses to Institution Tensions in China*, Promotor: Prof. B. Krug, EPS-2015-362-ORG, https://repub.eur.nl/pub/78649

Lenoir, A.S., *Are You Talking to Me? Addressing Consumers in a Globalised World*, Promotors: Prof. S. Puntoni & Prof. S.M.J. van Osselaer, EPS-2015-363-MKT, https://repub.eur.nl/pub/79036

Leung, W.L., *How Technology Shapes Consumption: Implications for Identity and Judgement*, Promotors: Prof. S. Puntoni & Dr. G. Paolacci, EPS-2019-485-MKT, https://repub.eur.nl/pub/117432

Li, D., *Supply Chain Contracting for After-sales Service and Product Support*, Promotor: Prof. M.B.M. de Koster, EPS-2015-347-LIS, https://repub.eur.nl/pub/78526

Li, X., *Dynamic Decision Making under Supply Chain Competition*, Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Prof. R. Zuidwijk. EPS-2018-466-LIS, https://repub.eur.nl/pub/114028

Liu, N., *Behavioral Biases in Interpersonal Contexts*, Promotors: Prof. A. Baillon & Prof. H. Bleichrodt, EPS-2017-408-MKT, https://repub.eur.nl/pub/95487

Ma, Y., *The Use of Advanced Transportation Monitoring Data for Official Statistics*, Promotors: Prof. L.G. Kroon & Dr. J. van Dalen, EPS-2016-391-LIS, https://repub.eur.nl/pub/80174

Maas, A.J.J., *Organizations and their external context: Impressions across time and space*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. T.H. Reus, EPS-2019-478-S&E, https://repub.eur.nl/pub/116480

Maira, E., *Consumers and Producers*, Promotors: Prof. S. Puntoni & Prof. C. Fuchs, EPS-2018-439-MKT, https://repub.eur.nl/pub/104387

Mell, J.N., *Connecting Minds: On The Role of Metaknowledge in Knowledge Coordination*, Promotor: Prof. D.L. van Knippenberg, EPS-2015-359-ORG,

https://repub.eur.nl/pub/78951

Meulen, D. van der, *The Distance Dilemma: the effect of flexible working practices on performance in the digital workplace*, Promotors: Prof. H.W.G.M. van Heck & Prof. P.J. van Baalen, EPS-2016-403-LIS, https://repub.eur.nl/pub/94033

Moniz, A, *Textual Analysis of Intangible Information*, Promotors: Prof. C.B.M. van Riel, Prof. F.M.G de Jong & Dr. G.A.J.M. Berens, EPS-2016-393-ORG, https://repub.eur.nl/pub/93001

Mulder, J., *Network design and robust scheduling in liner shipping*, Promotors: Prof. R. Dekker & Dr. W.L. van Jaarsveld, EPS-2016-384-LIS, https://repub.eur.nl/pub/80258

Neerijnen, P., *The Adaptive Organization: the socio-cognitive antecedents of ambidexterity and individual exploration*, Promotors: Prof. J.J.P. Jansen, P.P.M.A.R. Heugens & Dr. T.J.M. Mom, EPS-2016-358-S&E, https://repub.eur.nl/pub/93274

Okbay, A., *Essays on Genetics and the Social Sciences*, Promotors: Prof. A.R. Thurik, Prof. Ph.D. Koellinger & Prof. P.J.F. Groenen, EPS-2017-413-S&E, https://repub.eur.nl/pub/95489

Oord, J.A. van, *Essays on Momentum Strategies in Finance*, Promotor: Prof. H.K. van Dijk, EPS-2016-380-F&A, https://repub.eur.nl/pub/80036

Peng, X., *Innovation, Member Sorting, and Evaluation of Agricultural Cooperatives*, Promotor: Prof. G.W.J. Hendriks, EPS-2017-409-ORG, https://repub.eur.nl/pub/94976

Pennings, C.L.P., *Advancements in Demand Forecasting: Methods and Behavior*, Promotors: Prof. L.G. Kroon, Prof. H.W.G.M. van Heck & Dr. J. van Dalen, EPS-2016-400-LIS, https://repub.eur.nl/pub/94039

Petruchenya, A., *Essays on Cooperatives: Emergence, Retained Earnings, and Market Shares*, Promotors: Prof. G.W.J. Hendriks & Dr. Y. Zhang, EPS-2018-447-ORG, https://repub.eur.nl/pub/105243

Plessis, C. du, *Influencers: The Role of Social Influence in Marketing*, Promotors: Prof. S. Puntoni & Prof. S.T.L.R. Sweldens, EPS-2017-425-MKT, https://repub.eur.nl/pub/103265

Pocock, M., *Status Inequalities in Business Exchange Relations in Luxury Markets*, Promotors: Prof. C.B.M. van Riel & Dr. G.A.J.M. Berens, EPS-2017-346-ORG, https://repub.eur.nl/pub/98647

Pozharliev, R., *Social Neuromarketing: The role of social context in measuring advertising effectiveness*, Promotors: Prof. W.J.M.I. Verbeke & Prof. J.W. van Strien, EPS-2017-402-MKT, https://repub.eur.nl/pub/95528

Protzner, S., *Mind the gap between demand and supply: A behavioral perspective on*

demand forecasting, Promotors: Prof. S.L. van de Velde & Dr. L. Rook, EPS-2015-364-LIS, https://repub.eur.nl/pub/79355

Reh, S.G., *A Temporal Perspective on Social Comparisons in Organizations*, Promotors: Prof. S.R. Giessner, Prof. N. van Quaquebeke & Dr. C. Troster, EPS-2018-471-ORG, https://repub.eur.nl/pub/114522

Riessen, B. van, *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks*, Promotors: Prof. R. Dekker & Prof. R.R. Negenborn, EPS-2018-448-LIS, https://repub.eur.nl/pub/105248

Rietdijk, W.J.R., *The Use of Cognitive Factors for Explaining Entrepreneurship*, Promotors: Prof. A.R. Thurik & Prof. I.H.A. Franken, EPS-2015-356-S&E, https://repub.eur.nl/pub/79817

Roza, L., *Employee Engagement in Corporate Social Responsibility: A collection of essays*, Promotor: Prof. L.C.P.M. Meijs, EPS-2016-396-ORG, https://repub.eur.nl/pub/93254

Rösch, D., *Market Efficiency and Liquidity*, Promotor: Prof. M.A. van Dijk, EPS-2015-353-F&A, https://repub.eur.nl/pub/79121

Schie, R. J. G. van, *Planning for Retirement: Save More or Retire Later?*, Promotors: Prof. B. G. C. Dellaert & Prof. A.C.D. Donkers, EOS-2017-415-MKT, https://repub.eur.nl/pub/100846

Schoonees, P., *Methods for Modelling Response Styles*, Promotor: Prof. P.J.F. Groenen, EPS-2015-348-MKT, https://repub.eur.nl/pub/79327

Schouten, K.I.M., *Semantics-driven Aspect-based Sentiment Analysis*, Promotors: Prof. F.M.G. de Jong, Prof. R. Dekker & Dr. F. Frasincar, EPS-2018-453-LIS, https://repub.eur.nl/pub/112161

Schouten, M.E., *The Ups and Downs of Hierarchy: the causes and consequences of hierarchy struggles and positional loss*, Promotors; Prof. D.L. van Knippenberg & Dr. L.L. Greer, EPS-2016-386-ORG, https://repub.eur.nl/pub/80059

Sihag, V., *The Effectiveness of Organizational Controls: A meta-analytic review and an investigation in NPD outsourcing*, Promotors: Prof. J.C.M van den Ende & Dr. S.A. Rijsdijk, EPS-2019-476-LIS, https://repub.eur.nl/pub/115931

Smit, J., *Unlocking Business Model Innovation: A look through the keyhole at the inner workings of Business Model Innovation*, Promotor: Prof. H.G. Barkema, EPS-2016-399-S&E, https://repub.eur.nl/pub/93211

Straeter, L.M., *Interpersonal Consumer Decision Making*, Promotors: Prof. S.M.J. van Osselaer & Dr. I.E. de Hooge, EPS-2017-423-MKT, https://repub.eur.nl/pub/100819

Stuppy, A., *Essays on Product Quality*, Promotors: Prof. S.M.J. van Osselaer & Dr. N.L. Mead. EPS-2018-461-MKT, https://repub.eur.nl/pub/111375

Subaşi, B., *Demographic Dissimilarity, Information Access and Individual Performance*, Promotors: Prof. D.L. van Knippenberg & Dr. W.P. van Ginkel, EPS-2017-422-ORG, https://repub.eur.nl/pub/103495

Suurmond, R., *In Pursuit of Supplier Knowledge: Leveraging capabilities and dividing responsibilities in product and service contexts*, Promotors: Prof. J.Y.F Wynstra & Prof. J. Dul. EPS-2018-475-LIS, https://repub.eur.nl/pub/115138

Szatmari, B., *We are (all) the champions: The effect of status in the implementation of innovations*, Promotors: Prof. J.C.M van den Ende & Dr. D. Deichmann, EPS-2016-401-LIS, https://repub.eur.nl/pub/94633

Toxopeus, H.S., *Financing sustainable innovation: From a principal-agent to a collective action perspective*, Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas. EPS-2019-458-S&E, https://repub.eur.nl/pub/114018

Turturea, R., *Overcoming Resource Constraints: The Role of Creative Resourcing and Equity Crowdfunding in Financing Entrepreneurial Ventures*, Promotors: Prof. P.P.M.A.R Heugens, Prof. J.J.P. Jansen & Dr. I. Verheuil, EPS-2019-472-S&E, https://repub.eur.nl/pub/112859

Valogianni, K., *Sustainable Electric Vehicle Management using Coordinated Machine Learning*, Promotors: Prof. H.W.G.M. van Heck & Prof. W. Ketter, EPS-2016-387-LIS, https://repub.eur.nl/pub/93018

Vandic, D., *Intelligent Information Systems for Web Product Search*, Promotors: Prof. U. Kaymak & Dr. Frasincar, EPS-2017-405-LIS, https://repub.eur.nl/pub/95490

Verbeek, R.W.M., *Essays on Empirical Asset Pricing*, Promotors: Prof. M.A. van Dijk & Dr. M. Szymanowska, EPS-2017-441-F&A, https://repub.eur.nl/pub/102977

Vermeer, W., *Propagation in Networks:The impact of information processing at the actor level on system-wide propagation dynamics*, Promotor: Prof. P.H.M.Vervest, EPS-2015-373-LIS, https://repub.eur.nl/pub/79325

Versluis, I., *Prevention of the Portion Size Effect*, Promotors: Prof. Ph.H.B.F. Franses & Dr. E.K. Papies, EPS-2016-382-MKT, https://repub.eur.nl/pub/79880

Vishwanathan, P., *Governing for Stakeholders: How Organizations May Create or Destroy Value for their Stakeholders*, Promotors: Prof. J. van Oosterhout & Prof. L.C.P.M. Meijs, EPS-2016-377-ORG, https://repub.eur.nl/pub/93016

Vlaming, R. de, *Linear Mixed Models in Statistical Genetics*, Promotors: Prof. A.R.

Thurik, Prof. P.J.F. Groenen & Prof. Ph.D. Koellinger, EPS-2017-416-S&E, https://repub.eur.nl/pub/100428

Vries, H. de, *Evidence-Based Optimization in Humanitarian Logistics*, Promotors: Prof. A.P.M. Wagelmans & Prof. J.J. van de Klundert, EPS-2017-435-LIS, https://repub.eur.nl/pub/102771

Vries, J. de, *Behavioral Operations in Logistics*, Promotors: Prof. M.B.M de Koster & Prof. D.A. Stam, EPS-2015-374-LIS, https://repub.eur.nl/pub/79705

Wagenaar, J.C., *Practice Oriented Algorithmic Disruption Management in Passenger Railways*, Promotors: Prof. L.G. Kroon & Prof. A.P.M. Wagelmans, EPS-2016-390-LIS, https://repub.eur.nl/pub/93177

Wang, P., *Innovations, status, and networks*, Promotors: Prof. J.J.P. Jansen & Dr. V.J.A. van de Vrande, EPS-2016-381-S&E, https://repub.eur.nl/pub/93176

Wang, R., *Corporate Environmentalism in China*, Promotors: Prof. P.P.M.A.R Heugens & Dr. F. Wijen, EPS-2017-417-S&E, https://repub.eur.nl/pub/99987

Wasesa, M., *Agent-based inter-organizational systems in advanced logistics operations*, Promotors: Prof. H.W.G.M van Heck, Prof. R.A. Zuidwijk & Dr. A. W. Stam, EPS-2017-LIS-424, https://repub.eur.nl/pub/100527

Wessels, C., *Flexible Working Practices: How Employees Can Reap the Benefits for Engagement and Performance*, Promotors: Prof. H.W.G.M. van Heck, Prof. P.J. van Baalen & Prof. M.C. Schippers, EPS-2017-418-LIS, https://repub.eur.nl/pub/99312

Wiegmann, P.M., *Setting the Stage for Innovation: Balancing Diverse Interests through Standardisation*, Promotors: Prof. H.J. de Vries & Dr. K. Blind, EPS-2019-473-LIS, https://repub.eur.nl/pub/114519

Wijaya, H.R., *Praise the Lord! : Infusing Values and Emotions into Neo-Institutional Theory*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. J.P. Cornelissen, EPS-2019-450-S&E, https://repub.eur.nl/pub/115973

Williams, A.N., *Make Our Planet Great Again: A Systems Perspective of Corporate Sustainability*, Promotors: Prof. G.M. Whiteman & Dr. S. Kennedy, EPS-2018-456-ORG, https://repub.eur.nl/pub/111032

Witte, C.T., *Bloody Business: Multinational investment in an increasingly conflict-afflicted world*, Promotors: Prof. H.P.G. Pennings, Prof. H.R. Commandeur & Dr. M.J. Burger, EPS-2018-443-S&E, https://repub.eur.nl/pub/104027

Ye, Q.C., *Multi-objective Optimization Methods for Allocation and Prediction*, Promotors: Prof. R. Dekker & Dr. Y. Zhang, EPS-2019-460-LIS, https://repub.eur.nl/pub/116462

Ypsilantis, P., *The Design, Planning and Execution of Sustainable Intermodal Port-hinterland Transport Networks*, Promotors: Prof. R.A. Zuidwijk & Prof. L.G. Kroon, EPS-2016-395-LIS, https://repub.eur.nl/pub/94375

Yuan, Y., *The Emergence of Team Creativity: a social network perspective*, Promotors: Prof. D. L. van Knippenberg & Dr. D. A. Stam, EPS-2017-434-ORG, https://repub.eur.nl/pub/100847

Yuferova, D., *Price Discovery, Liquidity Provision, and Low-Latency Trading*, Promotors: Prof. M.A. van Dijk & Dr. D.G.J. Bongaerts, EPS-2016-379-F&A, https://repub.eur.nl/pub/93017

Zhang, Q., *Financing and Regulatory Frictions in Mergers and Acquisitions*, Promotors: Prof. P.G.J. Roosenboom & Prof. A. de Jong, EPS-2018-428-F&A, https://repub.eur.nl/pub/103871

Zuber, F.B., *Looking at the Others: Studies on (un)ethical behavior and social relationships in organizations*, Promotor: Prof. S.P. Kaptein, EPS-2016-394-ORG, https://repub.eur.nl/pub/94388