Integrated Duty Assignment and Crew Rostering

Master Thesis Operations Research

Thomas Breugem
(361074)

Supervisor: Dr. D. Huisman
Co-Supervisor: Dr. T.A.B. Dollevoet

August 13, 2015

Abstract

We propose an integrated model for the Duty Assignment and the Crew Rostering problem. Both problems are part of the crew planning process at Netherlands Railways; The Duty Assignment problem consists of finding a 'fair' allocation (according to some measure) of the duties among the roster groups. The Crew Rostering problem is well known in literature, and consists of finding good rosters given a set of duties.

Our model integrates the above two problems, and hence involves large scale optimization (since all duties have to be assigned simultaneously). Our research evaluates the effectiveness of an integrated approach compared to e.g., a sequential approach. We also propose a second model to counter the problem of weak lower bounds, a problem well known for rostering problems. We show that this new model leads to promising results.

Keywords: Duty Assignment, Crew Rostering, Netherlands Railways, Integrated Crew Planning

Preface

I would like to express my gratitude to those I met during the four years of my study, and, of course, to those I already met a long time ago.

First of all, I would like to thank Dennis and Twan, you were great supervisors. Not only during my thesis I enjoyed working with you, but also in the previous years. I am glad we will keep working together in the coming years.

Of course, I would like to thank all my teachers and all others I met at the university, you made the last four years a very pleasant experience. Many of you had a bigger influence on my development than you probably know.

I am thankful to my family and close friends, who were always there to listen to my stories; even though, as they would say 'did not understand anything of all those symbols'. The many cups of coffee and good meals, never underestimate the power of a good meal, were, I think, the most important factor in where I stand today.

Finally, I would like to thank everyone from π; they made writing my thesis a great experience, and provided me with indispensable knowledge for my further, non-academic, life. They taught me that love is expressed in the temperature of food, and introduced me to the great maritime experience of the Faroe Islands.

All in all, the last four years have been a great ride. I am looking forward what the next years as a PhD-candidate will bring.

Thomas Breugem
Zoetermeer, 2015

Contents

1 Introduction **1**
- 1.1 Netherlands Railways 1
- 1.2 Planning Problems at NS 1
- 1.3 An Integrated Approach to Crew Planning 3

2 Problem Description **5**
- 2.1 Crew Planning at NS 5
- 2.2 An Integrated Approach 10

3 Duty Assignment Problem **13**
- 3.1 Sequential DAP 13
- 3.2 DAP with Rota Schedule 14
- 3.3 Duty Attributes 15

4 Cyclic Crew Rostering Problem **19**
- 4.1 CCRP Single Roster Group 19
- 4.2 CCRP Multiple Roster Groups 20
- 4.3 Integrated CCRP 21
- 4.4 Roster Patterns 23

5 A Higher-Level Formulation for the CCRP **27**
- 5.1 Preliminaries 27
- 5.2 Modeling Roster Patterns 28
- 5.3 New CCRP Multiple Roster Groups 28
- 5.4 Integrated CCRP2 29
- 5.5 Theoretical Support ICCRP2 30

6 Solution Approaches **37**
- 6.1 Budget Constraints 37
- 6.2 Solution Approaches ICCRP 38
- 6.3 Empirical Trade-off Curves 40

7 Rostering Base Utrecht **43**
- 7.1 Data 43
- 7.2 Comparison CCRP and CCRP2 45
- 7.3 Results Integrated Crew Rostering 47

8 Conclusion and Further Research **51**

Bibliography **53**

Appendix A Distributions of Attribute Values **55**

Chapter 1

Introduction

1.1 Netherlands Railways

Netherlands Railways (abbreviated as NS, from the Dutch name Nederlandse Spoorwegen) is the major public transport operator in the Netherlands, with more than a million passengers on an average working day. Originally, NS was the only operator on the Dutch railway network. In 1995 NS was split up into a number of companies, among which ProRail (a state-owned company responsible for the railway network) and NS as it is currently known (that is, solely as a public transport operator). Although NS is not the only railway operator, it is by far the largest; approximately 90% of the passengers travel with NS.

NS Reizigers (NSR) is the daughter company of NS which operates the trains. They are concerned with the design of the timetable, the dispatching of rolling stock and the planning of the crew, among other things. These are all challenging problems, which are further complicated by the high density of demand in the Netherlands. It is therefore no surprise that Operations Research (OR) techniques are used intensively to support decision making at NS.

1.2 Planning Problems at NS

To give a general idea of the planning problems at NS, and their relation with OR, we will give a brief overview of the planing process. For a thorough discussion of the planning problems, we refer to e.g., Huisman et al. [2005], Kroon et al. [2009] and Abbink [2014]. We classify the problems into four different groups (see Abbink [2014]).

Strategic Planning Problems

The first type are *strategic* planning problems. These problems concern long term decision making (say numerous years), such as investing in new rolling stock. Another example of a strategic planning problem at NS is the creation of the line plan (i.e., where and with which frequency trains are operated). It is clear that this problem is not only difficult from an optimization perspective, but also heavily depends on good forecasts of passenger demand. Another example of a long term planning problem is whether or not to hire new train crew. Again, this process depends on forecasts (i.e., whether additional crew members are necessary) but also contains many practical considerations, since training new staff is expensive.

Tactical Planning Problems

The second group are the *tactical* planning problems. These problems often consider planning periods from a few months up to a year. Two important tactical planning problems are creating the timetable and scheduling the crews. The timetabling problem considers making a basic hour pattern for the timetable, which, because NS works with a cyclic timetable, can be used as a framework for the daily timetables. It is important to note that a timetable is not complete without a specification of how trains are routed through the station; due to the high density of the railway infrastructure in the stations such problems are far from trivial.

The crew scheduling problem refers to the process of creating a final set of rosters for the personnel given the set of *tasks* (i.e., the smallest amount of work that can be assigned to an employee). The schedules should satisfy numerous complex labor rules, as well as provide a 'fair' allocation of the workload over all employees. The crew scheduling problem is decomposed into duty scheduling (i.e., creating duties from the tasks) and crew rostering (i.e., assigning the duties to rosters). We will discuss the crew planning process in more detail in Chapter 2.

Operational Planning Problems

The third type of planning problems are operational planning problems. These problems already concern very detailed planning. The time span for such problems is roughly every month. One could think of these problems as finding adjustments to the somewhat general solutions obtained in the tactical phase. As an example, one would consider timetabling for specific days, instead of the more general weekly timetable. As a consequence, the timetable can be made more specific and take special events into account. Similarly, one considers crew scheduling on a more detailed level.

Operational Control Problems

The fourth and final category is called *operational* control problems, or short term planning problems, and is sometimes omitted in the classification. Such problems consist of reacting real-time to e.g., delays, thereby adjusting, say, the timetable that is operated. Another example would be updating the crew schedule due to a canceled train.

It is clear that every planning phase has unique characteristics and hence calls for different solution approaches. In the 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 planning phase, 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 planning problems are strongly represented in OR literature. The final planning phase, on the other hand, concerns real-time problem solving, and is therefore often approached using heuristic approaches or even simple 'rules of thumb'.

1.3 An Integrated Approach to Crew Planning

In this thesis we will develop an integrated approach for the Crew Rostering problem. Currently, duties are generated for a number of bases throughout the country (e.g., Utrecht, Amsterdam) which in turn divide their duties among a number of *roster groups*. A roster group is a group of employees with similar characteristics, i.e., they are able to perform the same set of duties. For each of these groups a roster is made. At this moment, both the allocation of duties to roster groups and the rostering per group are done manually; the allocation is done by an auction among representatives of the groups (often taking multiple days), while the latter is done by each of the roster groups manually, thereby using e.g., old rosters. It is clear that from an optimization perspective the above is far from optimal: In both phases possible good solutions are lost.

This motivated Abbink [2014] and Hartog et al. [2009] to develop mathematical models to support the rostering process at NS. Their results were promising, and showed that better rosters could be obtained. These rosters were not only better from a mathematical point of view, but were also perceived as more desirable by the employees. In their approach the duties are first allocated to the roster groups, then for each roster group a roster is created. In contrast, we will consider an integrated approach. That is, we develop a method to obtain good rosters, in which we directly take into account that the duties are fairly allocated among the different roster groups.

The lay-out of the thesis is as follows. In Chapter 2 we formalize our problem and embed it in the larger framework of crew rostering problems. We will discuss the relevant literature and give an overview of the labor restrictions that need to be taken into account. In Chapter 3 we consider a mathematical model for the Duty Assignment problem. This model is based on the work of Hartog et al. [2009] and Abbink [2014]. Then, in Chapter 4, we discuss the model for the cyclic crew rostering problem. We also show how the two problems (duty allocation and crew rostering) can be integrated. In Chapter 5 we propose an additional model, which assigns multiple duties at once. Then, in Chapter 6 we discuss our solution approaches. In Chapter 7 we apply our models to the rostering process at base Utrecht. We conclude with a general conclusion and our ideas for further research in Chapter 8.

Chapter 2

Problem Description

In this chapter we formalize the integrated duty assignment and crew rostering problem. We start with a more detailed discussion of the crew planning process at NS in Section 2.1. Thereafter, in Section 2.2, we motivate our integrated approach and show how this changes the structure of the planning process.

2.1 Crew Planning at NS

Recall that the rosters for NS employees are made for *roster groups*, which in turn have a *base* from which they operate. Given the set of *tasks*, i.e., indivisible building blocks of work, the goal of the crew planning phase is to create rosters for all employees that satisfy numerous rules. These rules concern not only single employees (e.g., a maximum workload), but also the division of, say, aggression work among the bases. The current overall planning process is schematically visualized in Figure 2.1.

From the set of tasks, which is regarded as input, numerous duties are constructed in the Crew Scheduling phase. These duties are then allocated to the different roster groups. Then, a roster is made for each group. Note that the approach is sequential and hence loss of optimality is unavoidable. There is, of course, still a certain level of interaction between the different phases (i.e., certain constraints are taken into account to assure feasible solutions and desirable properties in later phases). We will now discuss the separate parts of the scheme in more detail.

Crew Scheduling

The first step is *Crew Scheduling*, i.e., creating a set of duties that cover the set of tasks. Formally, a duty is a sequence of tasks that can be executed by an employee. Hence, a duty should satisfy many labor rules. Examples of such rules are a minimal meal break time, a maximum duty length or sufficient route knowledge.

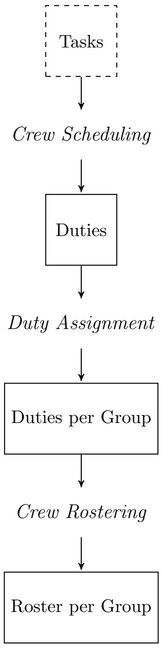

Figure 2.1: Schematic Overview Rostering process.

In Figure 2.2 we give a stylized example of a set of duties. Each duty is a sequence of trips, and can have a varying intensity, i.e., the first duty has a lot smaller workload than the second. For each of the duties a break is specified (indicated by a star). The braces show the minimum break length and maximum duty length, respectively. Note that each duty must end at the starting station.

A popular model for the Crew Scheduling problem is a *set covering* formulation which is often solved using e.g., Lagrangian Relaxation in combination with Column Generation (see e.g., Caprara et al. [1997], Huisman et al. [2005], Abbink et al. [2011], and references therein).

Duty Assignment

The next step in the Crew Planning process is the assignment of duties to the roster groups, i.e., the *Duty Assignment* problem. Note that a duty already has a specified base. As mentioned previously, the current approach to this problem is an auction among representatives. This can take multiple days and is thus rather inefficient. Furthermore, because not all representatives are equally skilled, it might also lead to 'unfair' allocations (i.e., a skilled representative claims the best duties for its group).

Therefore, a mathematical model for solving the Duty Assignment problem was proposed in Abbink [2014] and Hartog et al. [2009]. For each duty the following is assumed to be known:

- The start and end of the duty.

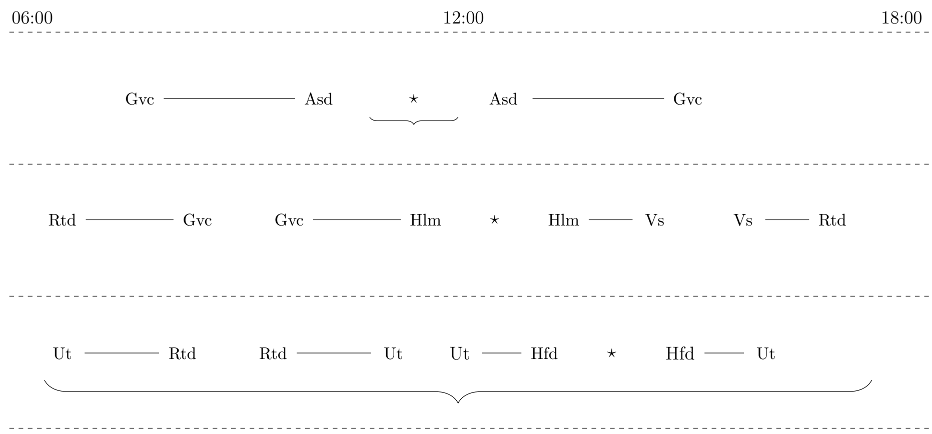

Figure 2.2: Example Duties.

- The length of the duty. Each duty has a length between 6 and 9.5 hours.
- The length of the break.
- Other properties (e.g., rolling stock types, possible aggression).

These properties allow NS to formally specify what a fair allocation looks like. Furthermore, it allows the specification of labor rules. For example, in our experiments, a 'fair' allocation assures that the sets of duties assigned to each roster group adhere to, among others, the rules:

- At least 35% of the work should be on A-trains (i.e., high quality work).
- The average duty length is not allowed to exceed 8 hours.

The goal of the assignment problem is to find a solution that not only satisfies all these constraints, but is also optimal in some sense. The model proposed in Abbink [2014] does this by minimizing the dis-balance of the solution, i.e., we try to find a solution where all groups have approximately the same percentages of certain attributes. We note that optimizing 'fairness' of duties and rosters is a relatively new research topic (see e.g., Zimmermann et al. [2012]).

Crew Rostering

When the duties are assigned to the different depots, the rosters for the different roster groups per depot are created. Recall that a roster group is a group of employees with similar characteristics, hence they are all able to execute similar duties. The problem of rostering the assigned duties is known as the *Crew Rostering* problem and is known to be NP-hard (see e.g., Mesquita et al. [2013], which prove NP-hardness for the *Driver Rostering* problem, a specific case). As with the Crew Scheduling problem, many complex labor rules are involved in the rostering of duties, such as a maximum total workload for each employee (i.e., the total amount of work in one week is not allowed to exceed a certain value).

At NS, and in general, many restrictions regarding rostering are specified using a higher level concept, which we will call the *type of a duty*. Main motivation for this is that many duties are 'almost identical', in the sense that they, say, all start early. Hence, it makes sense to say, for example, that we do not want an early duty after a late duty in our roster. It is clear that such requirements are best defined using duty types.

We consider a total of three different duty types: early duties, late duties and night duties. These types are denoted by V, L and N, respectively. The duty types are based on the start and end times of the duties. For a given duty d, we denote the type of the duty by $\tau(d)$.

In the roster there are also other types of days such as reserve days and rest days. These types of days are important due to the collective labor agreement (i.e., there should be a balance between those types of days). We note that, at NS, besides rest days, which are indicated by R, there are also WTV, RO and CO days. These types of days can be seen as 'compensation days', and in our approach are treated simply as rest days. For more details, see e.g., Hartog et al. [2009]. We will refer to the set of all types a day in the roster can have (i.e., V, L, N, reserve, rest, WTV, RO and CO days) as the set of *roster day types*.

In general, the Crew Rostering problem, abbreviated CRP, can be divided into two types, which are a *cyclic* and *non-cyclic* variant. We refer to these variants as CCRP and NCCRP, respectively. As the names suggest, the CCRP consists of finding one roster for multiple employees, where each employee cycles through the roster. An example of a cyclic roster is shown in Figure 2.3.

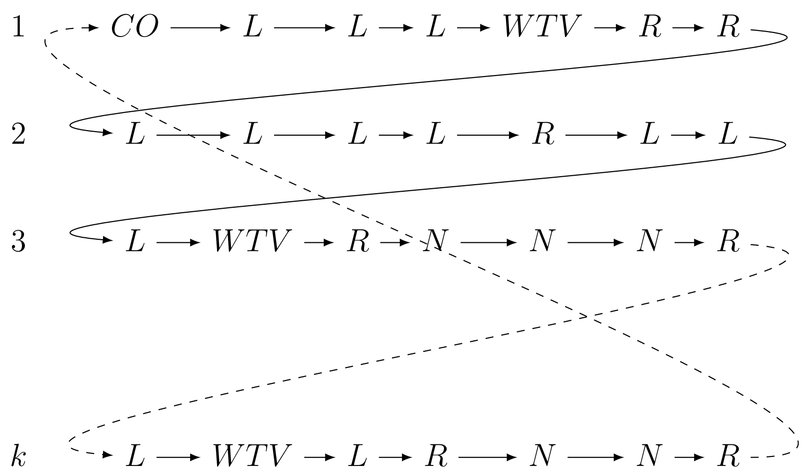

Figure 2.3: Example Cyclic Crew Rostering.

Here we see a cyclic roster for k employees, where only the roster day types are shown. Each of the employees executes a week of the roster, and thereafter continues to the next week (as indicated by the arrows). Note that in such a roster the number of weeks should equal the number of employees, since each employee cycles among the weeks. In the NCCRP, on the other hand, we try to find a separate roster for each employee. This would imply, for example, that the roster in Figure 2.3 would be for one single employee, and that the remaining $k-1$ employees have an entirely different roster.

The CCRP is a problem that often occurs in rostering for railway companies, while the NCCRP is a problem often arising in e.g., the airline industry. It is clear that both variants have their advantages. The CCRP, for example, allows us to reduce the number of rosters that need to be constructed, since each of the employees cycles through the same roster. For the NCCRP, on the other hand, a roster for all days has to be generated for all employees. This, however, also exposes the weakness of using a cyclic roster, since it is less flexible and we are not able to take employee specific preferences into account (at least not fully). From hereon we will focus on the CCRP, for solution approaches for the NCCRP see, for example, Xie and Suhl [2014] and references therein.

Formally, we are given a set of duties D, where each duty has a certain duty type, e.g. a late duty. Note that in the CCRP it is assumed that the duties are already assigned to the different roster groups (i.e., this problem is solved after the duty assignment problem). Hence, the duties need to be executed by groups of employees, where the set of all groups is denoted by G. Each of these groups can have different characteristics. For each group we determine a roster; let T be the set of days, and k the number of weeks in the roster for a given group (note that $|T| = 7k$). The goal is to assign the duties to the days (for all groups) such that all restrictions are satisfied or a certain objective is minimized.

The approach proposed by Hartog et al. [2009] splits the problem up in two phases. First, for each of the groups a roster is created which specifies for each day the roster day type, e.g., a late duty, instead of the exact duty that is executed. Similar to Emden-Weinert et al. [2001] and Xie and Suhl [2014] we call such a roster a *rota schedule*. Note that Figure 2.3 is an example of a rota schedule. Because we only consider roster day types, it is not as complex as assigning the duties *directly* to the days, but we can still take many constraints into account. Note that a rota schedule also specifies e.g., the rest days in the roster.

In the second step the actual duties are assigned to the days in the rota schedule. In this step the restrictions related to, for example, variations in work and exact work length are taken into account. This approach, i.e., splitting the problem up in these two phases, was first considered in Sodhi and Norris [2004].

Another possible way of formulating the problem is as a (multi-commodity) flow problem. One could construct a graph where the nodes represent combinations of duties and days, and hence a roster is represented by a path in this graph. Restrictions can now be modeled using the arc set or by defining certain subsets of nodes that can not be visited in the same path. Such a formulation is proposed both in Xie and Suhl [2014] and Mesquita et al. [2013]. The former considers both a two-step approach as in Sodhi and Norris [2004], as well as an integrated approach. They define sequences of nodes for which a penalty is given. In Mesquita et al. [2013], on the other hand, the more complex restrictions are taken care of by considering a multi-commodity flow formulation with a large number of commodities. The remaining restrictions can then be modeled using only the arc set.

2.2 An Integrated Approach

Our main contribution is to integrate the Duty Assignment and Crew Rostering problem on the *depot level*. That is, we want to take the balancing of the different duty properties into account simultaneously with the rostering of the different roster groups. Note that current models (Hartog et al. [2009], Abbink [2014], Xie and Suhl [2014], Mesquita et al. [2013]) assume that the duties are already assigned to the roster groups before the rostering starts or do not consider any restrictions on the allocation to the roster groups. We do assume that the duties are already assigned to the different depots. In Figure 2.4 we visualize this approach.

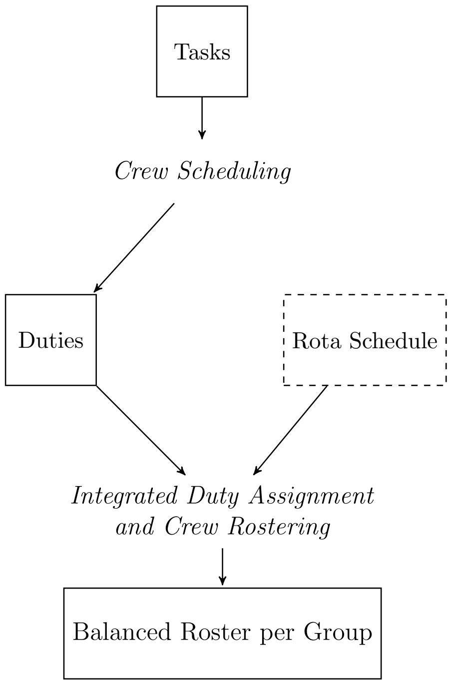

Figure 2.4: Schematic Overview Integrated Rostering approach.

As can be seen from the figure, the integrated approach merges two layers of the solution approach, thereby trying to obtain better rosters (which we simply call balanced rosters), in which the duties are balanced fairly among the employees. The dotted Rota Schedule block indicates that we will assume these to be fixed and given. Note that, otherwise, the rota schedules are determined based on the set of duties.

Because we assume that the rota schedules are given, our approach to the rostering problem is slightly different. This follows from the fact that we *preserve the structure* of the rota schedule, i.e., we assure that the roster days of a created roster match the rota schedule exactly. Because our input are rosters as they are currently operated, this implies that we only assign duties to days in which a duty was assigned in the given rosters. This approach is illustrated in Figure 2.5, where the first four weeks of a roster are shown. We obtain the rota schedule for this group by determining the duty type for all duties. The underlined days are the ones we consider variable, i.e., to which we need to assign a duty. We do not consider assigning duties to days to which no duty was assigned in the first place.

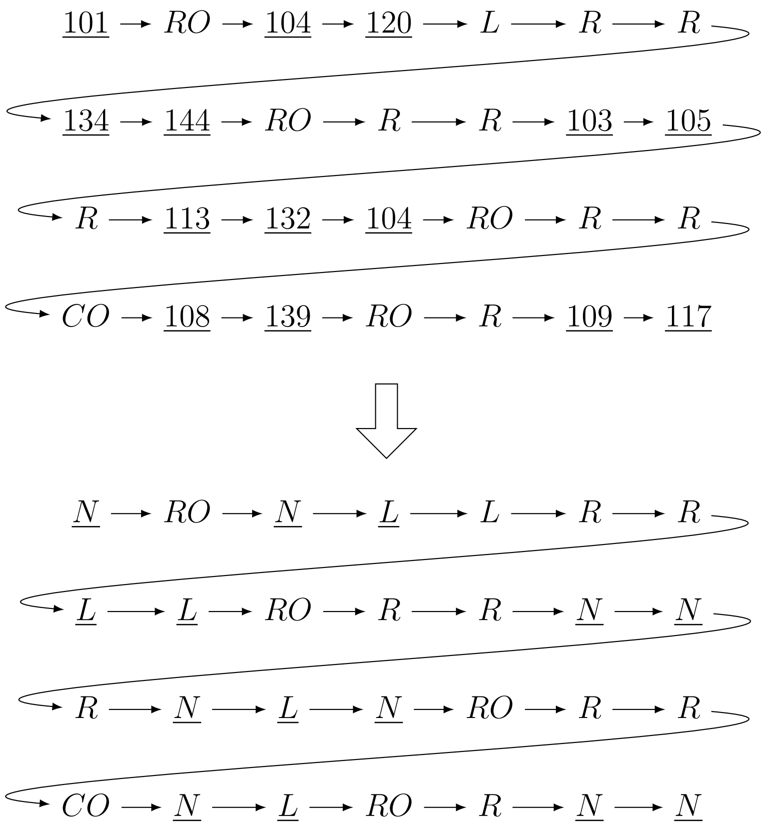

Figure 2.5: Example Rota Construction.

Motivation for our integrated approach can be best illustrated using an example. Consider a set of restrictions which impose that e.g., each driver should start his week on the same type of rolling stock. Clearly, such restrictions will lead to a skewed assignment of duties with respect to fairness, since some rolling stock types are simply more desirable than others. Hence, in a sequential approach it would often be difficult, if not impossible, to find a roster that satisfies the above constraints. This is because the different types of rolling stock will be assigned equally over the roster groups (since we minimize the dis-balance of the assignment) . Although we could take such restrictions (partially) into account when we assign the duties to the groups, it is clear that this would become very difficult when more restrictions of this type are considered.

Furthermore, we are able to obtain a better roster for the whole set of groups. Because we schedule all duties at once, we are able to obtain a better allocation of the duties to the groups. Note that the CCRP often considers finding good rosters for each *employee*, i.e., per week, while the duty assignment problem considers the group as a whole. Therefore, the non-integrated approach might lead to good distributions over the groups, but an integrated approach also assures a good distribution of the duties over the weeks. This is, however, also one of the main challenges of this type of solution method, since the number of duties that need to be scheduled simultaneously greatly increases.

Chapter 3

Duty Assignment Problem

In this chapter we discuss the mathematical formulation for the Duty Assignment problem, abbreviated as DAP, which was proposed in Abbink [2014]. In this model desirable attributes of the duties are to be allocated as fairly as possible over different roster groups. In Section 3.1 we formalize the model. In Section 3.2 we show how this model changes when rota schedules are assumed to be known. Finally, we conclude this chapter by briefly discussing the attributes in Section 3.3.

3.1 Sequential DAP

In Abbink [2014] the following model is considered for allocating the different duties to the groups (note that this model does *not* explicitly consider the rosters). This model assumes the set of duties to be known, but does not consider any detailed information about the rosters. We therefore say that this model is sequential; the rota schedules and rosters are made in a later phase.

The binary variable π_{dg} indicates whether duty $d \in D$ is assigned to group $g \in G$. The set M is the set of all combinations of duty types and weekdays (e.g., early and Monday). Parameters n_{mg} state the maximum number of duties of type m that can be allocated to group g. These parameters are used to assure a high probability for a feasible solution to the overall problem (i.e., obtaining rosters). In the DAP these parameters are assumed to be known (they are e.g., based on the old rosters). Let D_m be the set of duties that match the duty type and weekday of $m \in M$.

Every duty d has a certain score $a(d)$ for some attribute $a \in A$. Let η_g be the number of duties we need to roster for group g (we assume these parameters are known beforehand). Because the roster groups can have different sizes we compare the average score of the groups. Formally, we introduce the variables v_a and z_a indicating the minimum and maximum average score over all groups on attribute a, respectively. The goal of the allocation is to minimize a weighted sum of the 'spreads' between the scores on different attributes. We also introduce a lower bound l_{ag} and upper bound u_{ag} for the total score of group g on attribute a. Again, such restrictions can be used to 'guarantee' a feasible roster. In our case, we bound the averages of each of the groups, i.e., for every group g, the bounds l_{ag} and u_{ag} can be written as $\eta_g l_a$ and $\eta_g u_a$, for some l_a and u_a. Note that this implies that the lower and upper bounds l_a and u_a can be directly enforced upon the variables v_a and z_a (i.e., we can add $v_a \geq l_a$ and $z_a \leq u_a$). We will refer to this problem as DAP (Duty Assignment Problem).

$$
\begin{aligned}
\text{(DAP)} \quad \min \quad & \sum_{a \in A} b_a \left(z_a - v_a \right) && && (3.1) \\
\text{s.t.} \quad & \sum_{g \in G} \pi_{dg} = 1 && \forall d \in D && (3.2) \\
& \sum_{d \in D_m} \pi_{dg} \leq n_{mg} && \forall m \in M, g \in G && (3.3) \\
\eta_g v_a \leq & \sum_{d \in D} a(d) \pi_{dg} \leq \eta_g z_a && \forall a \in A, g \in G && (3.4) \\
& z_a \leq u_a && \forall a \in A && (3.5) \\
& v_a \geq l_a && \forall a \in A && (3.6) \\
& \pi_{dg} \in \mathbb{B} && \forall d \in D, g \in G && (3.7) \\
& v_a, z_a \in \mathbb{R}_+ && \forall a \in A. && (3.8)
\end{aligned}
$$

The objective (3.1) expresses that we minimize a weighted sum of the 'spreads', where b_a are the weights. Constraint (3.2) assures all duties are assigned to exactly one group. Furthermore, constraint (3.3) enforces that we do not exceed the upper bound n_{mg}. Constraints (3.4), (3.5) and (3.6) set the correct values of the spread variables and enforce the bounds, respectively. Finally, constraints (3.7) and (3.8) specify the domain of the decision variables.

We note that the DAP closely resembles the *Generalized Load Balancing* problem (GLBP), see e.g., Caragiannis [2008] and Zhu et al. [2014]. In this optimization problem, the goal is to divide a number of items over a set of machines such that the cost is divided as 'fair as possible', i.e., to minimize the maximum score over the machines. This problem is often approached in an online manner, which means items are to be assigned on the fly. In general, the GLBP is NP-hard (see Caragiannis [2008], and references therein). The DAP can be seen as an extension of the GLBP, because we also need to enforce certain lower and upper bounds as well as take the duty types into account.

3.2 DAP with Rota Schedule

If we assume a rota schedule to be given, we know in more detail how the duties should be assigned to the groups.

Formally, let a rota schedule δ for k employees (hence $T = 7k$ days) be a mapping of the set of days $\{1, 2, ..., T\}$ to set of roster day types (recall Figure 2.3). For this problem it is sufficient to represent a roster, and hence a rota schedule, by the set of values δ_m that specify how often a combination m occurs in the mapping δ. This representation is sufficient only because constraints (3.1) - (3.6) are invariant under any permutation of the duties in D_m in the roster, i.e., we are only interested in whether an assigned duty is in D_m, but it is of no importance to exactly which week it is assigned.

Let δ_{mg} denote the number of duties of type m in the rota schedule of group g. As explained above, instead of assigning a maximum number of duties n_{mg} for each $m \in M$ and $g \in G$, we now require that

$$\sum_{d \in D_m} \pi_{dg} = \delta_{mg} \qquad \forall m \in M, g \in G. \tag{3.9}$$

That is, we assure that our allocation exactly matches the rota schedule. The remainder of the model stays the same, hence by replacing constraints (3.3) with constraints (3.9), the complete model reads as follows.

$$\begin{aligned}
\text{(DAPR)} \quad \min \quad & \sum_{a \in A} b_a (z_a - v_a) & & (3.10)\\
\text{s.t.} \quad & \sum_{g \in G} \pi_{dg} = 1 & \forall d \in D \quad & (3.11)\\
& \sum_{d \in D_m} \pi_{dg} = \delta_{mg} & \forall m \in M, g \in G \quad & (3.12)\\
\eta_g v_a \leq & \sum_{d \in D} a(d) \pi_{dg} \leq \eta_g z_a & \forall a \in A, g \in G \quad & (3.13)\\
& v_a \geq l_a & \forall a \in A \quad & (3.14)\\
& z_a \leq u_a & \forall a \in A \quad & (3.15)\\
& \pi_{dg} \in \mathbb{B} & \forall d \in D, g \in G \quad & (3.16)\\
& v_a, z_a \in \mathbb{R}_+ & \forall a \in A. \quad & (3.17)
\end{aligned}$$

We will refer to this model as DAPR (Duty Assignment Problem with Rotas). In Chapter 4 we show how this model can be integrated into the crew rostering problem.

3.3 Duty Attributes

We will now discuss the set of attributes used in our experiments. Each of the attributes represents a certain (un)desirable property of a duty. We consider the following five attributes:

1. Duty length. For each of the duties the length is defined as the difference between the end and start time, minus the break length.

2. Percentage on type-A rolling stock. Trips on type-A rolling stock are desirable, and hence need to be fairly distributed. For each of the duties the percentage of work on type-A rolling stock is given.

3. Percentage Aggression work. Unfortunately, certain trips have a higher chance of passenger aggression. Clearly, such trips are undesirable and need to be fairly distributed. For each duty the percentage of aggression work is given.

4. Percentage on double decker trains. Similar to the type-A and Aggression percentages, we want to distribute the percentage of work on double decker trains as equally as possible.

5. RWD values. Finally, we consider the Repetition Within Duty (RWD) values, which 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, and hence we want to balance the RWD values as well.

The distribution of these values is often peaked around certain values, and skewed. For illustrative purposes, we will discuss two of the attributes in more detail, for an overview of all attributes we refer to Appendix A.

In Figure 3.1 the distribution of the duty lengths is shown. There is a large peak around nine hours, and then many duties spread between 6 and 8.5. The average duty length equals 7.97 hours. We require that the average duty length per group is at most 8 hours.

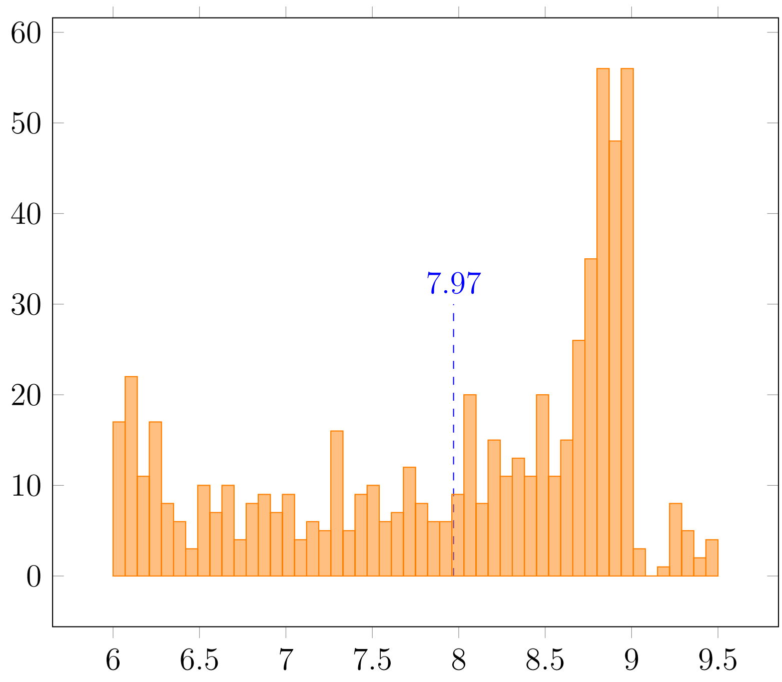

Figure 3.1: Histogram duty lengths.

In Figure 3.2 we show the different percentages of A-work in the duties. We see two clear peaks, one at 0% and the other at 100 %. In between the distribution is rather uniform. Based on the mean value of 41.6, we add the additional constraint that each roster group has at least 35% work on A-trains.

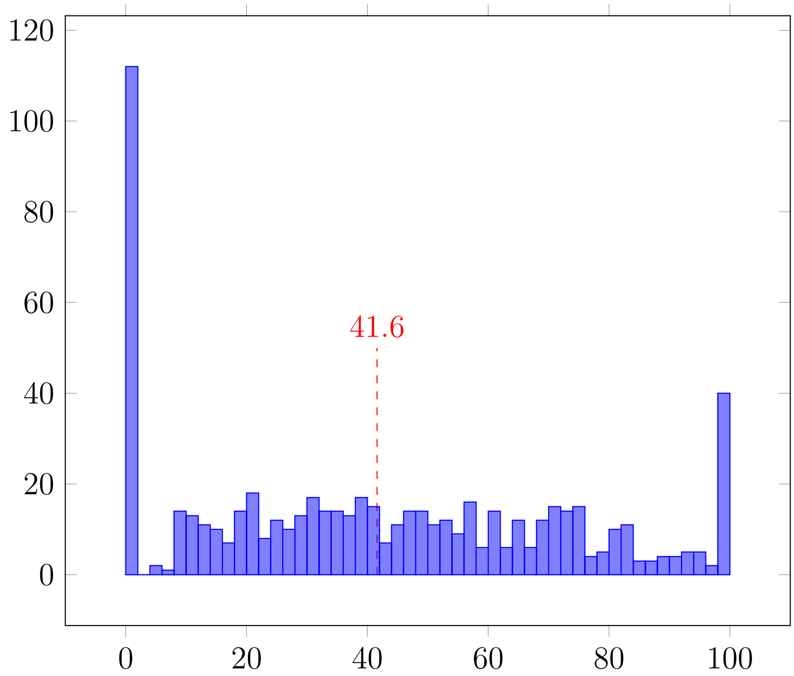

Figure 3.2: Histogram percentage work on type A-trains.

Chapter 4

Cyclic Crew Rostering Problem

In this chapter we propose a model for the CCRP. The model we propose is based on the work of Abbink [2014] and Hartog et al. [2009]. They consider a model in which the duties are assigned to the different days whilst taking into account the occurrence of certain patterns. A pattern is an undesirable property of a roster, e.g., an early duty after a late duty, and can be modeled as constraints in the mathematical formulation. We describe the model of Abbink [2014] and Hartog et al. [2009] in Section 4.1. In Section 4.2 we propose an extended model to allow for multiple roster groups. In Section 4.3 we show how the DAPR model of Chapter 3 can be integrated in this extended model. We conclude this chapter with a discussion of the roster patterns we considered in our experiments in Section 4.4.

4.1 CCRP Single Roster Group

We will first discuss the model for solely one roster group. As before, let T be the set of days. Furthermore, let $T' \subseteq T$ be the set of days to which a duty needs to be assigned (recall the example of Section 2.2). We will use the decision variable w_{dt} indicating whether duty d is assigned to day t. Clearly these variables are only relevant for feasible combinations of d and t, hence we define Ω as all pairs that represent a valid assignment of a duty and a day. $\Omega(\tau)$ denotes the subset of those pairs where $\tau(d) = \tau$ (recall that $\tau(d)$ denotes the duty type of duty d). Note that we assume that for each day $t \in T'$ a duty type has already been specified, which we will denote by δ_t.

The undesirable properties of the rosters are penalized using *patterns*, which can be expressed as linear restrictions. The set of patterns is denoted by P. The variables y_p indicate whether, or to what extent, pattern p is violated. For notational convenience, let W denote the vector of all decision variables w_{dt}. We denote with $f_p(W)$ the (linear) restriction corresponding to pattern p. Because all patterns are expressed using different constraints we will use this generic notation. For the same reason, we denote with Y_p the domain of the decision variable y_p. Throughout we assume that all Y_p consider only non-zero values (i.e., a violation is never negative) and that zero is included (i.e., it is possible to have 'no violation'). Clearly these assumptions are not restrictive from a modeling perspective. One can imagine that some patterns are expressed using binary variables (i.e., a solution has a certain property or not), while others are real numbers (i.e., the size of a certain violation). We discuss the patterns in detail at the end of this chapter, first we focus on the mathematical formulation.

Using the above definitions, the model now reads as follows.

$$
\begin{aligned}
\min \quad & \sum_{p \in P} c_p y_p && && (4.1)\\
\text{s.t.} \quad & \sum_{d:(d,t)\in\Omega(\delta_t)} w_{dt} = 1 && \forall t \in T' && (4.2)\\
& \sum_{t:(d,t)\in\Omega} w_{dt} = 1 && \forall d \in D && (4.3)\\
& y_p \geq f_p(W) && \forall p \in P && (4.4)\\
& w_{dt} \in \mathbb{B} && \forall (d,t) \in \Omega && (4.5)\\
& y_p \in Y_p && \forall p \in P. && (4.6)
\end{aligned}
$$

The objective (4.1) minimizes the penalties incurred from the patterns. Constraints (4.2) and (4.3) assure that every day is assigned a duty of the correct type and that every duty is assigned exactly once. Constraint (4.4) models the different patterns. Finally, constraints (4.5) and (4.6) specify the decision variables.

4.2 CCRP Multiple Roster Groups

We now show how the model first proposed by Abbink [2014] and Hartog et al. [2009] can be extended to multiple roster groups. As mentioned earlier, we will assign a large set of duties to multiple groups at once. This means that we assign duties to multiple rosters, and hence we not only have to decide to which day we assign a duty, but also to which group. We can model this as follows. We consider the decision variable w^g_{dt} indicating whether duty d is assigned to group g on day t. For each group the set T'_g represents the set of days for that group to which a duty needs to be assigned. Similarly, we group all patterns in $|G|$ groups, denoted by P_g for each $g \in G$. The set P_g contains all patterns relevant for only group g. Furthermore, the set P contains all patterns. We also extend the set Ω to contain triples (d, t, g) indicating that duty d can be assigned to group g at day t. $\Omega(\tau)$ is now the set of triples for which $\tau(d) = \tau$. Also the parameters δ_t are now per group, i.e., we now have parameters δ_{tg} specifying the duty type for each day and each group.

The above can be expressed by replacing constraints (4.4) by the following constraints (here W_g denote the subset of W related to group g)

$$y_p \geq f_p(W_g) \qquad \forall p \in P_g. \tag{4.7}$$

Note that the above restrictions are separable per group.

We also need to replace constrains (4.2) and (4.3) with

$$\sum_{d:(d,t,g)\in\Omega(\delta_{tg})} w^g_{dt} = 1 \qquad \forall g \in G, t \in T'_g \tag{4.8}$$

$$\sum_{(t,g):(d,t,g)\in\Omega} w^g_{dt} = 1 \qquad \forall d \in D, \tag{4.9}$$

that is, we assure that each group has for each day a duty of the correct type and that each duty is assigned exactly once.

For clarity we now give the complete model, to which we will refer to as CCRP (Cyclic Crew Rostering Problem).

$$\begin{aligned}
\text{(CCRP)} \quad \min \quad & \sum_{p\in P} c_p y_p && && (4.10)\\
\text{s.t.} \quad & \sum_{d:(d,t,g)\in\Omega(\delta_{tg})} w^g_{dt} = 1 && \forall g \in G, t \in T'_g && (4.11)\\
& \sum_{(t,g):(d,t,g)\in\Omega} w^g_{dt} = 1 && \forall d \in D && (4.12)\\
& y_p \geq f_p(W_g) && \forall g \in G, p \in P_g && (4.13)\\
& w^g_{dt} \in \mathbb{B} && \forall (d,t,g) \in \Omega && (4.14)\\
& y_p \in Y_p && \forall p \in P. && (4.15)
\end{aligned}$$

4.3 Integrated CCRP

Recall that in Chapter 3 we specified constraints which the allocation of duties must satisfy, and we also specified a cost for a given allocation of duties. Some of these restrictions are important because they guarantee (or give a very high probability) that a feasible roster exists for all groups. Clearly, such constraints are no longer relevant, since we assign the duties directly to the different days (and hence take care of feasibility). There are, however, also restrictions that deal with concepts such as “fairness” of the allocation. As an example, some duties are more desirable than others, and hence need to be balanced between the groups.

Many of such constraints are difficult, if not impossible, to express as patterns and it would often be very inefficient to do so. We therefore extend the CCRP model from the previous section with the, slightly modified, constraints of the DAPR model.

That is, we add the following restrictions. As before let π_{dg} indicate whether duty d is assigned to group g. Note that

$$\pi_{dg} = \sum_{t:(d,t,g)\in\Omega} w_{dt}^{g} \qquad \forall d \in D, g \in G. \tag{4.16}$$

First, we note that constraints (3.11) and (3.12) of the DAPR are already enforced by constraints (4.11) and (4.12) of the CCRP. We can use the same restrictions as earlier for the spread variables and bounds. Using (4.16) to eliminate the π_{dg} variables, these constraints read as

$$\eta_g v_a \leq \sum_{d\in D} a(d) \sum_{t:(d,t,g)\in\Omega} w_{dt}^{g} \leq \eta_g z_a \qquad \forall a \in A, g \in G \tag{4.17a}$$

$$v_a \geq l_a \qquad \forall a \in A \tag{4.17b}$$

$$z_a \leq u_a \qquad \forall a \in A. \tag{4.17c}$$

The integrated problem is a bi-objective optimization problem, since we want to find rosters with low cost, but we also want a ‘fair’ allocation of duties to the roster groups. We therefore state the problem as a bi-objective optimization problem.

The complete model now reads as follows:

$$\text{(ICCRP)} \quad \min \quad \sum_{p\in P} c_p y_p \tag{4.18}$$

$$\sum_{a\in A} b_a \left(z_a - v_a\right) \tag{4.19}$$

$$\text{s.t.} \quad \sum_{d:(d,t,g)\in\Omega(\delta_{tg})} w_{dt}^{g} = 1 \qquad \forall g \in G, t \in T_g' \tag{4.20}$$

$$\sum_{(t,g):(d,t,g)\in\Omega} w_{dt}^{g} = 1 \qquad \forall d \in D \tag{4.21}$$

$$y_p \geq f_p(W_g) \qquad \forall g \in G, p \in P_g \tag{4.22}$$

$$\eta_g v_a \leq \sum_{d\in D} a(d) \sum_{t:(d,t,g)\in\Omega} w_{dt}^{g} \leq \eta_g z_a \qquad \forall a \in A, g \in G \tag{4.23}$$

$$v_a \geq l_a \qquad \forall a \in A \tag{4.24}$$

$$z_a \leq u_a \qquad \forall a \in A \tag{4.25}$$

$$w_{dt}^{g} \in \mathbb{B} \qquad \forall (d,t,g) \in \Omega \tag{4.26}$$

$$y_p \in Y_p \qquad \forall p \in P \tag{4.27}$$

$$v_a, z_a \in \mathbb{R}_+ \qquad \forall a \in A. \tag{4.28}$$

We will refer to this model as ICCPR (Integrated Cyclic Crew Rostering Problem).

4.4 Roster Patterns

In the model above we represent patterns as generic linear restrictions. Because the patterns form the main body of the above formulation, it is important exactly which patterns are considered in our real-world instances. We therefore discuss all different patterns in this section. We state the patterns as if there is only one roster group; it is clear how to apply the patterns to each of the roster groups separately. We note that in general, a pattern corresponds to a certain day in the roster, e.g., a pattern considers the duty at a certain day, plus the duties on the two following days.

The patterns we consider range from 'hard patterns', in the sense that they are not allowed to occur, to 'soft patterns', thereby meaning that they are undesirable but not forbidden. For each of the patterns we briefly discuss how they can be modeled to fit the formulation, i.e., how they can be modeled as $y_p \geq f_p(W)$, with $y_p \in Y_p$.

Rest Patterns

The first type of patterns we consider model the rest times of the employees. After a duty we require that an employee has a certain minimum time to rest; we call the methods penalizing such violations *rest patterns*. We require that for a night duty the rest time is at least 14 hours, while for the other types of duties it is at least 12 hours. For the sake of illustration, consider the pattern p that models the rest time after duty $d \in D$ assigned to day $t \in T'_g$ for group $g \in G$. Suppose d is a night duty and let O_d be the set of duties for which the start time is less than 14 hours after the end time of duty d, provided it is rostered at day $t + 1$. The constraint that expresses whether or not rest pattern p is violated (i.e., we neglect the size of the violation) can be modeled as

$$y_p \geq w_{dt} + \sum_{d' \in O_d} w_{d',t+1} - 1. \tag{4.29}$$

because we want to enforce that such patterns *never* occur in a roster, we set $Y_p = \{0\}$. For non-night duties the patterns are defined similarly.

Although some combinations of duties do not violate the above rest patterns, they can still be undesirable. Therefore, we also add rest patterns that indicate if the rest time is less than 16 hours (by modifying the set O_d). Because such patterns are allowed, but not desired, we set $Y_p = \mathbb{B}$, i.e., we penalize if they occur.

Rest Day Patterns

Similar to rest patterns, we want to assure that when there is a rest day in the roster, the length of the rest period is sufficiently long. If two work days have one rest day in between we require that there is 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 is desired. For illustrative purposes, consider two work days with two rest days in between. This implies that the end and start times of the duties must differ by at least 30 + 24 = 54 hours. Given a day t and duty d, let R_d^2 be the set of duties that violate these 54 hours when rostered at day $t+3$ (note that $t+1$ and $t+2$ are the rest days). We can express the above as

$$y_p \geq w_{dt} + \sum_{d' \in R_d^2} w_{d',t+3} - 1. \tag{4.30}$$

because we want to enforce that such patterns *never* occur in a roster, we set $Y_p = \{0\}$. Work days with a different number of rest days in between are modeled similarly.

Workload Patterns

We also want to assure that no employee works too many hours in one week; we require that the total workload in one week should not exceed 45 hours. Here, and in the upcoming sections, a week is always to be interpreted as from Monday to Sunday (i.e., not a 'rolling week'). For a reserve duty a workload of 7.5 hours is taken into account (which is the average length minus the break). Similar to the previous patterns, we could model this as follows. Let $l(d)$ be the duty length of duty d, i.e., the difference between start and end time minus the break length. Let T_p be the set of days relevant for the workload pattern (i.e., the days in T_p form a week in the roster). A workload pattern p would be modeled as

$$y_p \geq \sum_{t \in T_p} l(d) w_{dt} - 45, \tag{4.31}$$

and, because a violation is not allowed, we have $Y_p = \{0\}$.

Variation Patterns

The final type of patterns we consider are *variation patterns*. These patterns are used to balance the duty attributes among the weeks, i.e., we want that an employee has approximately the same weeks in his or her roster. Note the similarity with the DAPR constraints discussed earlier. We consider the same attributes as for the DAPR. Furthermore, we also penalize if there is more than one duty of length at least 9 hours in a week. Finally, we also want to balance desirable and undesirable routes over the weeks.

All variation patterns are penalized in a linear way (i.e., the penalty equals the violation times a certain weight). This implies that for all variation patterns p we set the domain of y_p as $\mathbb{R}_+$. Let T_p be the set of days that are relevant for the variation pattern (in our instance this is the week to which the pattern corresponds). A variation pattern, concerning some attribute a, is expressed as

$$y_p \geq \sum_{t \in T_p} a(d) w_{dt} - \bar{a}, \tag{4.32}$$

and $Y_p = \mathbb{R}_+$. Here $\bar{a}$ is the target value of the pattern (i.e., the boundary value for which we start penalizing) and $a(d)$ the coefficient of duty d for this pattern. For the attributes we consider, $\bar{a}$ would be the average value, and $a(d)$ would be the score of the duty for this attribute divided by the total number of duties in the week.

Chapter 5

A Higher-Level Formulation for the CCRP

In this chapter we propose a second, more general, formulation for the CCRP. The main idea is that we model the problem on a 'higher-level', i.e., instead of assigning single duties to single days, we propose assigning multiple duties to multiple days at once. This implies that many pattern violations can be modeled implicitly; in Section 5.5 we prove that, under general conditions, this leads to a stronger formulation.

5.1 Preliminaries

We develop a model in which we directly assign a set of duties to numerous days. The general idea of this formulation is that we 'cut' a roster in a set of smaller 'pieces', which we will call *clusters*, and that each cluster is assigned a correct set of duties.

Formally, a cluster k is a subset of $T \times G$. We will denote the set of all clusters by K. In order for K to be correctly specified, it should be a partition of the set $T \times G$, that is $\bigcup_{k \in K} k = T \times G$ and $k \cap k' = \emptyset$, for every two clusters $k, k' \in K$. These requirements assure that we cover all duties, and also that no day in the roster is assigned multiple duties. Note that the model of Chapter 4 is simply a special case of this cluster formulation (i.e., the clusters are simply the days that need to be rostered). We assume that the set K can be partitioned in sets K_g, where each K_g contains clusters solely considering clusters for group $g \in G$.

We will call an assignment of a set of duties to a cluster a *duty sequence*. Intuitively, a duty sequence is a feasible assignment of the duties to the days in a cluster such that all duty types match the prespecified types. Formally a duty sequence, or simply a sequence, is a subset of Ω, i.e., the set of feasible pairs (d, t), that satisfy the predefined roster day types. Thus, for a sequence $s \subseteq \Omega$ it holds that for each $(d, t) \in s$ we have $(d, t) \in \Omega(\delta_t)$. Furthermore, no duty or day should occur twice in a sequence. It is clear how this concept generalizes to multiple roster groups. We define S_k as the set of sequences that can be assigned to cluster k and let $S = \bigcup_{k \in K} S_k$. Furthermore, for notational convenience, we define S_g as $\bigcup_{k \in K_g} S_k$.

The main motivation for our higher-level formulation is that we are able to remove certain patterns from the model. This follows from the fact that, if a pattern only considers the days of a certain cluster, the pattern violation can be modeled *a priori*, i.e., it only depends on whether or not we select a certain sequence. The generality of this model also allows for 'tailor-made' approaches, since the clusters can be chosen in such a way as to maximize the number of patterns we can remove (without making the clusters too large). As an example, the majority of our patterns, as discussed in Chapter 4, concern the weeks in the roster. Therefore, it makes sense to consider a cluster formulation where the clusters are weeks. In Chapter 7 we show that this indeed reduces the computation time.

5.2 Modeling Roster Patterns

Because clusters can concern multiple days we need to take a slightly more sophisticated approach in modeling the roster patterns. Intuitively, we try to remove as many patterns from the formulation as possible by modeling violations as pattern costs.

Formally, we proceed as follows. Define $P_k \subseteq P$ as the set of all patterns that depend solely on the days in cluster $k \in K$ and let $P_K = \bigcup_{k \in K} P_k$. It is clear that the cost of these patterns can be incorporated into the duty sequence costs and hence can be determined *a priori* solving the model. For each $p \in P_k$ we define for each sequence $s \in S_k$ the cost c_{ps} as the violation of pattern p in sequence s. Note that this cost is equal to $c_p y_p$, if we determine y_p based on the duties as assigned in s. By definition, we say $c_{ps} = 0$ if $s \notin S_k$. The total cost c_s of sequence s is now given by $\sum_{p \in P_K} c_{ps}$.

The remainder of the patterns (i.e., the set $P \backslash P_K$) still needs to be incorporated into the model. This is done by adding the constraints $y_p \geq h_p(S)$ for all $p \in P \backslash P_K$. The linear function h_p is directly obtained from f_p by substituting the assignment variables w^g_{dt} with $\sum_{s \in S_g} s(d,t) x_s$. Here x_s is the decision variable indicating whether sequence s is chosen and $s(d,t)$ equals one if duty d is assigned to day t in sequence s. For a single sequence s, we let $h_p(s)$ denote the value of $h_p(S)$ if we only set $x_s = 1$, i.e., $h_p(s)$ represents the function value for solely sequence s.

5.3 New CCRP Multiple Roster Groups

We are now able to formulate the new model. Let $S_d \subseteq S$ be the set of all sequences containing duty $d \in D$. The model, to which we will simply refer as CCRP2, reads as follows.

$$
\begin{aligned}
\text{(CCRP2)} \quad \min \quad & \sum_{s \in S} c_s x_s + \sum_{p \in P \setminus P_K} c_p y_p && && (5.1)\\
\text{s.t.} \quad & \sum_{s \in S_k} x_s = 1 && \forall k \in K && (5.2)\\
& \sum_{s \in S_d} x_s = 1 && \forall d \in D && (5.3)\\
& y_p \geq h_p(S) && \forall p \in P \setminus P_K && (5.4)\\
& x_s \in \mathbb{B} && \forall s \in S && (5.5)\\
& y_p \in Y_p && \forall p \in P \setminus P_K. && (5.6)
\end{aligned}
$$

The objective function (5.1) expresses that we minimize the total cost of penalties occurring in the clusters plus the penalties of the remaining patterns. Constraints (5.2) and (5.3) assure that we assign a sequence to each cluster and that all duties are assigned, respectively. Constraints (5.4) enforce the remainder of the patterns. Finally, Constraints (5.5) and (5.6) define the decision variables.

5.4 Integrated CCRP2

Similar to the ICCRP of Chapter 4, the restrictions from the DAPR can be added in a straightforward way.

Recall that π_{dg} indicates whether duty d is assigned to group g. Let $a(s) = \sum_{d \in D_s} a(d)$, where D_s are the duties assigned in cluster s, i.e., $a(s)$ is simply the total score of all duties in sequence s. Note that

$$\sum_{d \in D} a(d)\pi_{dg} = \sum_{s \in S_g} a(s) x_s \qquad \forall g \in G, \qquad (5.7)$$

hence, similar to the CCRP, we can eliminate the π_{dg} variables using (5.7). Adding the DAPR constraints to the CCRP2, we obtain the cluster version of the ICCRP, which we refer to as ICCRP2, as an alternative to the ICCRP of Chapter 4.

$$
\begin{aligned}
\text{(ICCRP2)} \quad \min \quad & \sum_{s \in S} c_s x_s + \sum_{p \in P \setminus P_K} c_p y_p && && (5.8)\\
& \sum_{a \in A} b_a \left(z_a - v_a\right) && && (5.9)\\
\text{s.t.} \quad & \sum_{s \in S_k} x_s = 1 && \forall k \in K && (5.10)\\
& \sum_{s \in S_d} x_s = 1 && \forall d \in D && (5.11)\\
& y_p \geq h_p(S) && \forall p \in P \setminus P_K && (5.12)\\
\eta_g v_a \leq & \sum_{s \in S_g} a(s) x_s \leq \eta_g z_a && \forall a \in A, g \in G && (5.13)\\
& v_a \geq l_a && \forall a \in A && (5.14)\\
& z_a \leq u_a && \forall a \in A && (5.15)\\
& x_s \in \mathbb{B} && \forall s \in S && (5.16)\\
& v_a, z_a \in \mathbb{R}_+ && \forall a \in A. && (5.17)
\end{aligned}
$$

5.5 Theoretical Support ICCRP2

Our main motivation for the ICCRP2 is that it allows us to incorporate pattern costs implicitly into the model. Intuitively, this will lead to a stronger formulation. In this section we prove that this is indeed the case; under very general conditions, going from a set of small clusters to a set of large clusters tightens the formulation. One of the interesting properties of this result is that it gives a good indication about which sets of clusters will lead to strong formulations, without needing very large clusters.

Intuition Behind the Proof

We illustrate the idea behind the proof by comparing a formulation where the clusters are the *days* of the roster, with another formulation where the clusters are the *weeks* of the roster. Let K be the the set of week clusters and K' the set of day clusters. Note that we compare the LP-relaxations of both formulations, i.e., a day can be assigned multiple duties with a fractional value.

We want to assure a certain 'gain' by assigning duties to weeks instead of assigning duties to days. Consider one of the variation patterns for, say, duty length. As discussed in Chapter 4, we incur a penalty from these patterns equal to the amount the average duty length for a week exceeds the average duty length over all duties. For notational convenience, let this average duty length over all duties be denoted by $\bar{a}$.

Let $k \in K$ be an arbitrary week in the roster. Suppose that we assign a set of duties to the days in k, such that their average duty length exactly equals $\bar{a}$. Here we assume some of the duties are assigned *fractionally*, i.e., multiple different duties are assigned to the same day with fractional values. It is clear that if we consider the set of clusters K', i.e., the days, we incur no penalty for this assignment. This is because the penalty is enforced by a constraint in the model which simply averages the length of all duties assigned to the week k (see Equation (4.32)).

Consider now, however, the model with the set of clusters K, i.e., the weeks. It is very likely that for every set of sequences we are able to construct with the assigned set of duties, at least one of the sequences has an average duty length above $\bar{a}$. This, in turn, would imply we incur a penalty, since the sequence costs are determined a priori, i.e., sequences with an average above $\bar{a}$ have a positive cost, while those with an average below $\bar{a}$ have zero cost.

The above implies that the same fractional solution, in terms of the actual roster, leads to a higher penalty for the set of clusters K, i.e., the weeks, compared to the set of clusters K', i.e., the days. Or, in other words, the penalty variable for this week and this variation pattern is enforced a higher value due to using the clusters K. This is equivalent to saying that certain solutions with low penalties are not feasible when we use the clusters K, or in other words that using the weeks as clusters leads to a stronger formulation than using the days as clusters.

Formal Proof

The above example illustrates a general relation between sets of clusters. We now formalize this example and prove that it holds for sets of clusters *in general*; in the remainder of this section we assume we are comparing two sets of clusters K and K'. We refer to their corresponding sets of duty sequences as S and S' and to their respective polytopes as $\mathcal{P}$ and $\mathcal{P}'$. Without loss of generality, we assume that all patterns have unit cost, i.e., $c_p = 1$, for all patterns p.

For notational convenience, we refer to the sets of decision variables as vectors x, x', y, v and z, i.e., $x = \{x_s : s \in S\}$ and $x' = \{x'_{s'} : s' \in S'\}$. In order to compare the two formulations we define a mapping between $\mathcal{P}$ and $\mathcal{P}'$. We will assume that K 'extends' K', which means that, among other things, every sequence $s \in S$ consists of numerous sequences $s' \in S'$ (e.g., a week consists of numerous days). We will formally define the idea of extension later on in this section.

First, we note that any solution $\bar{x}$ can be expressed as a solution $\bar{x}'$ by setting $\bar{x}'_{s'} = \sum_{s \in S} s(s')\bar{x}_s$ for all $s' \in S'$. Here $s(s')$ indicates if sequence s' is included in s. We will denote this solution $\bar{x}'$ by $\phi(\bar{x})$. It is important to note that for multiple $\bar{x}$ vectors we may have $\phi(\bar{x}) = \bar{x}'$, e.g., for a given assignment of duties to days we can construct multiple different sets of weeks.

We note that the formulation $\mathcal{P}$ contains less y variables, since the set of patterns $P_K \setminus P_{K'}$ is modeled implicitly in the sequence costs. Let S_p be the set of sequences relevant for pattern p, i.e, any sequence with non-zero coefficient in h_p. We add the additional, redundant, constraints

$$y_p \geq \sum_{s \in S_p} c_{ps} x_s \qquad P_K \setminus P_{K'} \tag{5.18}$$

to the formulation $\mathcal{P}$; this assures we are able to compare the two formulations. Note that the constraints (5.18) simply express the costs of the additional patterns we were able to model implicitly.

We are now able to prove that the ICCRP2 leads to a tighter formulation. We do this by showing that $\phi(\mathcal{P}) \subsetneq \mathcal{P}'$, where $\phi(\mathcal{P})$ is defined as

$$\phi(\mathcal{P}) \equiv \{(\phi(x), y, v, z) : (x, y, v, z) \in \mathcal{P}\}. \tag{5.19}$$

That is, the set $\phi(\mathcal{P})$ are the solutions in $\mathcal{P}$ expressed in the set of smaller clusters S'.

With the above definitions, we are able to translate the example of days and weeks into a set of sufficient conditions. We say that if the two sets of clusters K and K' satisfy these conditions, the set K *extends* the set K'.

Definition 5.1. *The set of clusters K extends K' if*

1. *For each $k \in K$ there are clusters $K^* \subseteq K'$ such that $k = \bigcup_{k^* \in K^*} k^*$.*
2. *There is at least one pattern p in $P_K \setminus P_{K'}$ and a feasible solution $(\bar{x}', \bar{y}, \bar{v}, \bar{z}) \in \mathcal{P}'$ with the following property; for every $\bar{x}$ that satisfies $\phi(\bar{x}) = \bar{x}'$, there are two sequences $s_1, s_2 \in S_p$ such that*

   I. *Both $\bar{x}_{s_1}$ and $\bar{x}_{s_2}$ are assigned non-zero values in $\bar{x}$.*

   II. *For the two sequences s_1 and s_2 it holds that $h_p(s_1) > 0$ and $h_p(s_2) < 0$.*

Property 1 assures that $P_K \supseteq P_{K'}$. Property 2 formalizes the idea expressed in the example; it assures that the penalty incurred from pattern p is strictly higher when using the set of clusters K (in terms of the example, note that $h_p(s_1) > 0$ corresponds to an average length higher than $\bar{a}$, and $h_p(s_2) < 0$ to an average length lower than $\bar{a}$). Put more general, property 2 assures that there is at least one pattern for which the y_p values are not trivially enforced by h_p.

We are now able to prove the main result.

Theorem 5.1. *Let K and K' by two sets of clusters. Furthermore, let $\mathcal{P}$ and $\mathcal{P}'$ be their respective polytopes. Given that K extends K', $\mathcal{P}$ is a stronger formulation than $\mathcal{P}'$. That is,*

$$\phi(\mathcal{P}) \subsetneq \mathcal{P}'.$$

Proof. Consider an arbitrary solution $(\bar{x}, \bar{y}, \bar{v}, \bar{z}) \in \mathcal{P}$. For notational convenience, let $\bar{x}' = \phi(\bar{x})$. We will first show that $(\bar{x}', \bar{y}, \bar{v}, \bar{z}) \in \mathcal{P}'$.

We claim that the above values of $\bar{x}'$, $\bar{v}$ and $\bar{z}$ are feasible with respect to $\mathcal{P}'$. To prove this it is sufficient to show that the DAPR constraints are equivalent. Note that $a(s) = \sum_{d \in D_s} a(d)$ and hence $a(s) = \sum_{s' \in s} a(s')$ for all $a \in A$. Here summing over all $s' \in s$ is shorthand notation for summing over all s' for which $s(s') = 1$, i.e., all sequences s' included in s. This implies we have

$$\sum_{s \in S_g} a(s)\bar{x}_s = \sum_{s \in S_g} \bar{x}_s \sum_{s' \in s} a(s') \tag{5.20a}$$

$$= \sum_{s \in S_g} \bar{x}_s \sum_{s' \in S'_g} s(s')a(s') \tag{5.20b}$$

$$= \sum_{s' \in S'_g} a(s') \sum_{s \in S_g} s(s')\bar{x}_s \tag{5.20c}$$

$$= \sum_{s' \in S'_g} a(s')\bar{x}'_{s'}, \tag{5.20d}$$

which justifies the above claim.

Next, we consider the variables $\bar{y}$. Recall that S_p is the set of all sequences relevant for pattern p. We can write the constraint $y_p \geq h_p(S')$ of $\mathcal{P}'$ in the extended form

$$y_p \geq \sum_{s' \in S'_p} \alpha_{ps'}\bar{x}'_{s'} - \beta_p., \tag{5.21}$$

where α and β are the coefficients of the linear function h_p. By substituting $\bar{x}'_{s'} = \sum_{s \in S} s(s')\bar{x}_s$ into (5.21), we obtain

$$y_p \geq \sum_{s' \in S'_p} \alpha_{ps'} \sum_{s \in S} s(s')\bar{x}_s - \beta_p. \tag{5.22}$$

Next, note that $s(s') = 0$ if $s \notin S_p$ and $s' \in S'_p$. Hence, combined with rewriting, we obtain

$$y_p \geq \sum_{s \in S_p} \bar{x}_s \sum_{s' \in S'_p} \alpha_{ps'} s(s') - \beta_p. \tag{5.23}$$

or, with minimal notation,

$$y_p \geq \sum_{s \in S_p} \bar{x}_s \sum_{s' \in s} \alpha_{ps'} - \beta_p. \tag{5.24}$$

Using this expression we analyze the possible values for the decision variables $\bar{y}$. Note that we only consider decision variables y for the set of patterns $P \setminus P_{K'}$, as all patterns $p \in P_{K'}$ are modeled implicitly in both models; it is clear we incur similar costs in both formulations for each of the patterns $p \in P_{K'}$.

Next, consider a pattern $p \in P \setminus P_K$. For these patterns, the value of $\bar{y}_p$ is enforced explicitly by a constraint in both models. One can imagine such a pattern as e.g., spanning multiple weeks, if we relate it to the example in the beginning of this section. Note that, by definition of h_p, the coefficient of x_s in $h_p(S)$ can be expressed as $\sum_{s' \in s} \alpha_{ps'}$. It follows from (5.24) that if $p \in P \setminus P_K$ the value of y_p is bounded by the exact same constraint in both models, hence $\bar{y}_p$ is feasible with respect to $\mathcal{P}'$.

Finally, consider a pattern $p \in P_K \setminus P_{K'}$. For such a pattern, the constraint in $\mathcal{P}$ is given by

$$y_p \geq \sum_{s \in S_p} c_{ps} \bar{x}_s. \tag{5.25}$$

In $\mathcal{P}'$, however, the pattern costs of p are not modeled implicitly. Thus in this case, the constraint is given by $y_p \geq h_p(S')$, which, as we derived earlier, can be written as

$$y_p \geq \sum_{s \in S_p} \bar{x}_s \sum_{s' \in s} \alpha_{ps'} - \beta_p. \tag{5.26}$$

Note that this situation was illustrated with the duty length example, i.e., constraint (5.25) penalizes every week that has an average above the target value, while constraint (5.26) only penalizes if the average over *all* duties assigned to the week is above the target value. Because $p \in P_K$ we know that $\sum_{s \in S_p} x_s = 1$ and hence we may write (5.26) as

$$y_p \geq \sum_{s \in S_p} \bar{x}_s \left(\sum_{s' \in s} \alpha_{ps'} - \beta_p \right). \tag{5.27}$$

Note that $c_{ps} \geq \max\{0, \sum_{s' \in s} \alpha_{ps'} - \beta_p\}$, due to the non-negativity of pattern costs. Hence, the value of y_p in $\mathcal{P}$ is *at least* as high as in $\mathcal{P}'$. This implies that $\bar{y}_p$ is feasible with respect to $\mathcal{P}'$. It follows that $\phi(\mathcal{P}) \subseteq \mathcal{P}'$.

To show that $\phi(\mathcal{P})$ is a strict subset of $\mathcal{P}'$ we use Property 2 of definition 5.1. We claim Property 2 implies there is a pattern $p \in P_K \setminus P_{K'}$ and a solution $(\bar{x}', \bar{y}, \bar{v}, \bar{z}) \in \mathcal{P}'$ such that there is no solution $(\bar{x}, \bar{y}, \bar{v}, \bar{z}) \in \mathcal{P}$ with $\phi(\bar{x}) = \bar{x}'$, i.e., every solution in $\mathcal{P}$, with $\phi(\bar{x}) = \bar{x}'$, incurs higher pattern costs (that is, the value of $\bar{y}_p$ is not feasible for any of these solutions). Here, we assume without loss of generality, that the $\bar{y}$ values are always set as low as possible (i.e., we do not penalize whenever this is not enforced by the constraints). Note that our claim is exactly what we illustrated in the example: we assigned a set of duties to days, such that any set of weeks constructed from these duties would incur higher pattern costs.

To prove our claim, we reason as follows. Consider any of the solutions in $\mathcal{P}$ with $\phi(\bar{x}) = \bar{x}'$. We know from Property 2 of Definition 5.1 there are two sequences s_1 and s_2 for which both $\bar{x}_{s_1}$ and $\bar{x}_{s_2}$ are assigned non-zero values. Furthermore, we know that $h_p(s_1) > 0$ and $h_p(s_2) < 0$. Because $\bar{x}_{s_1} > 0$ and $h_p(s_1) > 0$ we know that the penalty incurred for pattern p is strictly larger than zero. Furthermore, because $\bar{x}_{s_2} > 0$ and $h_p(s_2) < 0$ we have

$$\sum_{s \in S_p} c_{ps} \bar{x}_s > \sum_{s \in S_p} \bar{x}_s \left(\sum_{s' \in s} \alpha_{ps'} - \beta_p \right). \tag{5.28}$$

Combining these two observations it follows that the penalty for pattern p enforced in $\mathcal{P}'$ is strictly lower than the penalty enforced in $\mathcal{P}$ or, in other words, it follows that the solution $(\bar{x}', \bar{y}, \bar{v}, \bar{z}) \notin \phi(\mathcal{P})$. The main result follows. □

Chapter 6

Solution Approaches

In this chapter we discuss our solution approaches for the integrated crew rostering problem. In Section 6.1 we show how we deal with the bi-objective formulation of the ICCRP. In Section 6.2 we then discuss the different approaches we considered. In Section 6.3 we consider different trade-off curves to analyze the different approaches.

6.1 Budget Constraints

The first step in solving the ICCRP is deciding how to reduce the model from a bi-objective model to a single-objective model.

Recall that the ICCRP is formulated as a bi-objective optimization problem where we want to minimize the two objectives

$$\sum_{p \in P} c_p y_p \tag{6.1a}$$

$$\sum_{a \in A} b_a (z_a - v_a). \tag{6.1b}$$

Because multi-dimensional optimization is computationally difficult, we use the concept of *budget constraints* to reduce the problem to a one-dimensional optimization problem. A budget constraint eliminates one of the objective functions by adding it as a constraint to the model, i.e., we can bound the DAPR objective by a parameter γ by adding the budget constraint

$$\sum_{a \in A} b_a (z_a - v_a) \leq \gamma \tag{6.2}$$

to the model. Similarly, we can bound the CCRP objective by a parameter β by adding the constraint

$$\sum_{p \in P} c_p y_p \leq \beta \tag{6.3}$$

to the model. Using such constraints allows us to make a trade-off between the two objectives.

6.2 Solution Approaches ICCRP

When dealing with the integrated version of the crew planning problem, i.e., duty assignment and crew rostering, one ideally would like to solve the bi-objective version of the ICCRP. As mentioned above, this is computationally difficult and hence the concept of budget constraints was used. We note, however, that adding budget constraints to the model implicitly leads to a trade-off between the two objectives, i.e., once one of the budget constraints is added, the optimal value of the remaining objective can severely increase. It is therefore important to carefully apply such constraints, since they can have a large influence on the found solutions,. To analyze this effect, we consider multiple approaches to the ICCRP. Our main goal is to analyze the different solutions found with these approaches, and to analyze in detail the possible trade-off between the two objectives.

In our initial experiments we found that proving optimality, when minimizing either of the two objectives, was very difficult (it could take up to more than an hour for relatively small instances). We also found, on the other hand, that 'good' solutions were found quickly for most instances. We therefore decided to limit the solution time per optimization step. This leads to a different comparison of the approaches; although certain approaches might be inferior *in theory*, it is very well possible they perform best in reasonable time.

We now state our different approaches to the ICCRP. We group them based on which budget constraint was used, e.g., an approach that uses a γ-constraint first minimizes the DAPR objective, and thereafter minimizes the CCRP objective, whilst enforcing constraint (6.2).

Approaches with γ-constraint

We start with the approaches that first obtain a good solution in terms of the duty assignment objective, and then minimize the rostering objective.

1. We first minimize the DAPR objective to obtain a set of assignment variables π^* with objective value γ^*. Now, we fix π^* and solve the CCRP problem per group. Note that this is the sequential approach of Hartog et al. [2009] and Abbink [2014], while taking into account the existence of a feasible roster (since the ICCRP formulation is used). We bound the time for minimizing the DAPR objective by 15 minutes, while the subproblems per group are bounded by 3 minutes (almost all subproblems can be solved to optimality in this time). We denote this approach by DAPR + CCRP.
2. Similar to the DAPR + CCPR, we first minimize the DAPR objective to obtain a set of assignment variables π^* with objective value γ^*. We then,

however, solve the ICCRP whilst enforcing that $\gamma = \gamma^*$. Note that, in contrast to the first approach, the duty assignment variables π^* are not fixed after the first step. The solution time for the respective minimization problems is set to at most 15 and 30 minutes, respectively. We denote this approach by DAPR + ICCRP.

3. The third approach is very similar to the DAPR + ICCRP. In fact, we consider the exact same steps. We now also, however, *post-process* the solution by trying to improve the rostering objective for each of the groups separately. We refer to this approach as DAPR + ICCRP*.

4. The final approach we consider in this category is used only for the largest instance, i.e., when we solve the ICCRP for all roster groups simultaneously. Again, we first minimize the DAPR objective to obtain a set of duty assignment variables π^* with objective value γ^*. Now, however, we decompose the problem in smaller subproblems of *multiple groups*. For each of these sets of groups we then solve the ICCRP whilst enforcing a bound $\gamma = \gamma^*$ on the duty assignment objective. We note that this might lead to a higher γ measured over all groups, as the spread measured over all groups is in generally larger than the spread measured per subset of groups. We decomposed the large set of groups into groups of four, we therefore refer to this approach as DAPR + 4-ICCRP. The rostering time per subset of groups is limited to 7.5 minutes (to match the 30 minutes for the ICCRP).

Approaches with β-constraint

We now turn to the approaches that first obtain a good solution in terms of the rostering objective, and then try to minimize the duty assignment objective. That is, we *reverse* the approach. Note that these types of approaches are possible due to the ICCRP model, as in general the rostering objective can not be taken into account while solving the duty assignment problem.

5. We first minimize the CCRP objective to obtain a set of rosters with objective value, say, β^*. Thereafter, we minimize the DAPR objective while enforcing a bound of β^* on the rostering objective. As before, we limit the CCRP to 30 minutes and the DAPR to 15 minutes. Note that this approach can be seen as the 'reversed' version of approach 2. We denote this approach by CCRP + DAPR.

6. In this approach we again consider post-processing the found set of rosters. That is, we first minimize the CCRP objective. Therafter, we try to improve the solution by trying to reduce the rostering objective for each of the groups separately. For the solution obtained after this post-processing, we again minimize the DAPR objective while enforcing a bound of β^* on the rostering objective. The solution time per groups is again limited by 3 minutes. We refer to this approach as CCRP* +DAPR.

7. Finally, we consider the reversed variant of the DAPR + 4-ICCRP approach for large instances. That is, we first minimize the CCRP objective for each of the sets of groups (note that this implies the duties are also split into groups). Thereafter, we minimize the DAPR constraint considering *all* groups. Again, a time bound of 7.5 per subset of groups is used. For the DAPR we limit the solution time to 15 minutes. We refer to this approach as 4-CCRP + DAPR.

Motivation Approaches

Our motivation for the approaches presented above is twofold. First, we want to analyze in detail the impact of the sequence of optimization. That is, we want to analyze how much a difference it makes whether we first minimize the DAPR objective or whether we first minimize the CCRP objective. Secondly, we want to analyze the performance of 'large scale' optimization (i.e., solving the ICCRP for all groups at once) versus 'small scale' optimization (i.e., rostering per group or per multiple groups). Especially since we consider bounded running times, it is interesting to analyze which approaches perform best in which circumstances.

6.3 Empirical Trade-off Curves

We note that the above approaches do, in some sense, not fully exploit the 'integrated' part of the ICCRP model. By this we mean that they minimize both objectives in a sequential manner, and hence are destined to find solutions at the boundaries (in terms of the objective values). We therefore also consider a Pareto-alike analysis.

In this analysis we enforce a γ-bound on the solution, for varying γ, and use the different approaches to find solutions that satisfy this bound. To be more precise, we proceed as follows. Given γ, we minimize the DAPR objective until a solution is found with assignment variables π^* and objective value γ^* such that $\gamma^* \leq \gamma$. We stop the optimization process once such a solution is found, i.e., the found solution is considered 'good enough'.

Once this solution is found we minimize the CCRP objective. The first curve is based on the DAPR + CCRP approach; we fix the assignment variables π^* and minimize the rostering objective per group separately. Note that this approach depends heavily on the found solution, as the variables π^* are fixed in the rostering part. Hence, we expect that such a solution will not be able to benefit, in terms of the CCRP objective, from higher γ values, as the rostering objective is not taken into account in the first step. We note that this curve represents the solutions found if the sequential approach would be applied in practice.

The remaining curves are constructed in three different ways, similar to approaches 2 to 4. That is, we minimize the ICCRP while enforcing a γ-bound of γ^* (denoted by Curve ICCRP), we consider the post-processing solution approach (denoted by

Curve ICCRP*) and, finally, we consider for the large instance the 4-ICCRP approach, where we enforce a bound γ^* for each of the subsets of groups. These approaches solely use the solution found in the first step as an initial solution for the optimization process, and thereafter consider an integrated approach. These approaches, therefore, are expected to benefit, in terms of the CCRP objective, from higher γ values.

Using these empirical curves we are able to analyze the trade-off between 'good rosters' and 'fair allocations'. In particular, these curves will help us analyze what the true benefit is from the ICCRP model compared to e.g., the sequential approach proposed in Abbink [2014].

Chapter 7

Rostering Base Utrecht

In this chapter we show the results of our case study; we applied our mathematical models to the rostering process at base Utrecht. In Section 7.1 we give a general overview of the different roster groups at base Utrecht. We also state the parameters used in the optimization. In Section 7.2 we analyze the performance of the CCRP and CCRP2 model. In Section 7.3 we discuss the performance of the different optimization approaches discussed in Chapter 6.

7.1 Data

We apply the different approaches of Chapter 6 to the crew rostering process at base Utrecht. The roster groups we consider are groups of guards. In total, there are 714 duties to be rostered. We will use 625 of the duties in our optimization (we leave out duties that lack information, and only consider duties that are present in the given rota schedules). The duties are divided over 16 roster groups of varying size (i.e., number of employees). We have 4 groups of 6 employees, 2 groups of 8 employees, 9 groups of 12 employees and 1 group of 14 employees. On average, they work slightly less than 4 days a week. An overview of the groups is given in Table 7.1.

Table 7.1: Number of duties per group.

Group	Nr. Weeks	Nr. Duties
1	12	49
2	8	30
3	12	46
4	6	26
5	6	23
6	12	49
7	12	46
8	12	50

Group	Nr. Weeks	Nr. Duties
9	12	49
10	12	49
11	12	50
12	8	22
13	12	43
14	6	23
15	6	17
16	14	53

Attributes

Recall from Chapter 3 that the quality of the allocation of the duties to the roster groups is evaluated using a set of attributes, i.e., some aspects of the duties in a roster that are desirable or not. One of the goals of the optimization is to divide the duties over the group such that the obtained rosters differ as little as possible in these attributes. As discussed in Chapter 3 we consider the following attributes:

1. Average duty length.
2. Percentage on type-A rolling stock.
3. Percentage Aggression trains.
4. Percentage on double decker trains.
5. RWD values.

In Table 7.2 we show the bounds and weights for all considered attributes.

Table 7.2: Bounds and weights DAPR model.

Attribute	l_a	u_a	b_a
Duty length	0	8	30
Perc. Type-A	35	100	1
Perc. Aggression	0	18	1
Perc. Double Decker	0	40	1
RWD	0	2.5	25

We penalize all attributes considering percentages in a similar way, i.e., a spread of 2% in aggression work is equal to a spread of 2% of type A-train work. Furthermore, we give violations in duty length a penalty of 30, hence a ten minute violation equals one of 5%. For the RWD we set a weight of 25 (i.e., a 0.2 deviation equals a 5% deviation).

Patterns

As discussed in Chapter 4 the quality of the roster is expressed using patterns. We summarize the patterns and their respective costs in Table 7.3. For the variation patterns we consider the same attributes as for the DAPR model. Furthermore, we penalize the occurrence of certain routes, and duties longer than 9 hours. Since the number of different routes is large, we only consider a few; we want to divide the desirable routes Groningen-Leeuwarden and Maastricht- Sittard and the routes trajectories Hoofddorp - Weesp and Baarn - Utrecht evenly over all weeks.

We also stated the remaining patterns. Note that the 'forbidden' patterns (i.e., the strict rest, rest day and workload patterns) do not have a cost, since they are not allowed to be present in any roster. Here the strict rest pattern models the rest time that is not allowed to be violated, while the rest pattern penalizes the rest times less than 16 hours.

Table 7.3: Cost and Domain CCRP patterns.

Pattern	Y_p	c_p
Variation Patterns		
Duty length	$\mathbb{R}_+$	30
Perc. Type-A	$\mathbb{R}_+$	1
Perc. Aggression	$\mathbb{R}_+$	1
Perc. Double Decker	$\mathbb{R}_+$	1
RWD	$\mathbb{R}_+$	25
Duty Length > 9	$\mathbb{R}_+$	1
Routes	$\mathbb{R}_+$	1
Rest Pattern	$\mathbb{B}$	30
Strict Rest Pattern	$\{0\}$	-
Rest Day Pattern	$\{0\}$	-
Workload Pattern	$\{0\}$	-

We now turn to the results of our analysis. All results are obtained using CPLEX 12.5.1 on a computer with a Intel Xeon E5 3.10 GHz processor. We set the CPLEX emphasis to focus on finding good feasible solutions.

7.2 Comparison CCRP and CCRP2

In this section we compare the CCRP and CCRP2 formulations of Chapters 4 and 5. Recall that the main motivation for the CCRP2 model was to obtain a tighter formulation, and hence less difficulty in finding the optimal solution. We therefore analyzed the solution time and the relative gap (i.e., the relative difference between the upper and lower bound) for both models over time. For the CCRP2 model we used the weeks as clusters (recall the example of Chapter 5).

Because the number of possible duty sequences increases exponentially, we consider instances of reasonable size; we roster both groups 4, 5 and 6 simultaneously as well as groups 9 and 10. Both sets of groups contain a total of 98 duties that need to be assigned to 24 weeks. We note that for larger instances one could consider e.g., a Column Generation approach to circumvent this issue. The results are shown in Table 7.4.

We allowed CPLEX to run for at most 30 minutes on both instances. The Pre-CPLEX time indicates the time needed to generate all feasible duty sequences. The Final Gap is the gap between the best solution found as we terminate the algorithm, and the best found lower bound so far. As can be seen in Table 7.4 the CCRP2 is able to find an optimal solution very quickly for both instances. The CCRP, on the other hand, is not able to prove optimality for both instances within 30 minutes.

Table 7.4: Comparison CCRP and CCRP2

Groups	Model	Pre-CPLEX	CPLEX	Solution Value	Final Gap
4, 5 and 6	CCRP	-	30m	284.41	36.47%
4, 5 and 6	CCRP2	20m	138.5s	253.48	optimal
9 and 10	CCRP	-	30m	394.40	7.51%
9 and 10	CCRP2	11m	7.25s	394.40	optimal

Interestingly, the CCRP did find the optimal solution for groups 9 and 10, although there was still a relative gap of 7.51%.

In Figure 7.1 we plotted the relative gap between the best feasible solution and the lower bound for the CCRP over time. We see that for both sets of groups, there first is a sharp decrease of the gap, and thereafter it almost remains constant. This means that a 'good' solution is found quickly, but improving this solution, or the lower bound, turns out to be difficult. The implicit pattern costs of the CCRP2 seem to be a great advantage in this respect.

We conclude that the CCRP2 is a good addition to the CCRP. For instances of reasonable size it outperforms the CCRP, not only in solution time, but we are also able to prove optimality for both instances. It would therefore be interesting to consider more advanced solution methods (e.g., Column Generation) in order to apply the CCRP2 to larger instances.

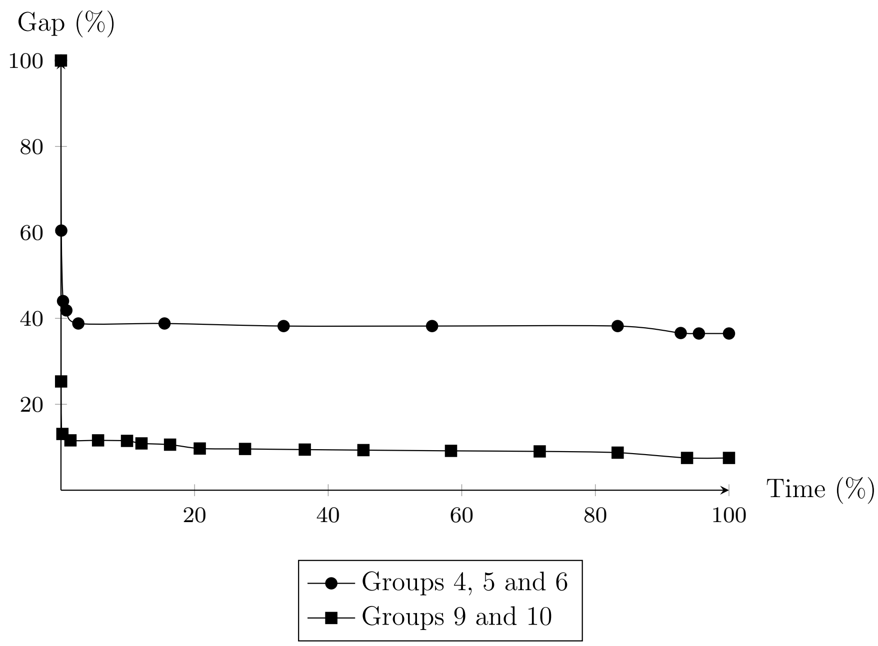

Figure 7.1: Relative Gap CCRP over Time

7.3 Results Integrated Crew Rostering

Next, we turn to the result of the overal crew rostering process. We consider each of the solution approaches as discussed in Chapter 6. We consider two instances; a medium sized instance involving only the first four roster groups and a large instance involving all roster groups.

Case 1: Roster Groups 1 to 4

For our first instance we solve the ICCRP for the first four roster groups. This gives us a total of 151 duties that need to be rostered. As mentioned in Chapter 6, we limit the solution times.

We consider all approaches of Chapter 6, except approaches 4 and 7, which we only consider for rostering all groups. Similarly, we construct all empirical curves discussed in Chapter 6, except the curve for the 4-ICCRP. Based on the results of the different approaches, we constructed the curves for $\gamma \in \{2, 4, 6, 8, 10\}$. The results are shown in Figure 7.2.

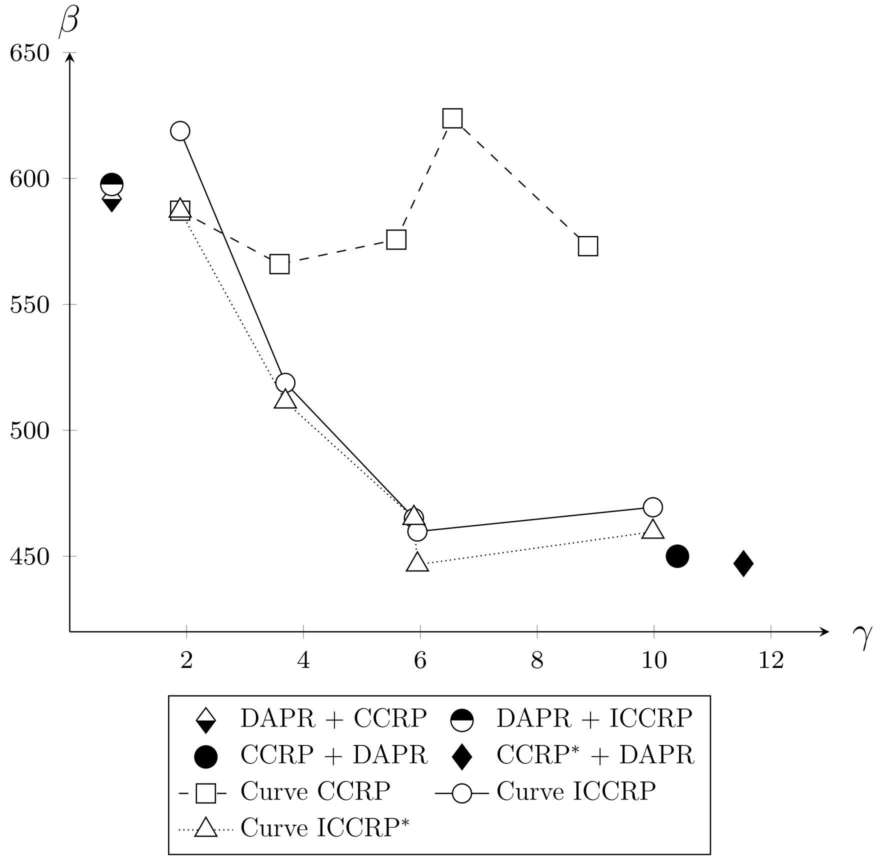

Figure 7.2: Results First Four Groups.

Let us first focus on the single points in Figure 7.2. As expected, we see there is a sharp distinction between the approaches with a γ-constraint (i.e., the black and white symbols) versus the approaches with a β-constraint (i.e., the solid symbols). (We omitted the DAPR+ICCRP* as it coincided with the DAPR+ICCRP, i.e., an additional optimization per group did not lead to a decrease in the objective value.) This sharp distinction is evidence that one should be careful in the type of approach considered, i.e., whether to first minimize the DAPR objective or the CCRP objective.

Interestingly, we see that the DAPR + CCRP outperforms the DAPR + ICCRP. We found that the second step, i.e., solving the ICCRP, did not lead to better rosters compared to the rosters found in the first optimization step. This is likely caused by the tight γ-bound we impose on the ICCRP in this approach. Similarly, the effect of a tight β-bound can be observed by considering the CCRP + DAPR and the CCRP* + DAPR; the small gain we obtain from post-processing the solution, in terms of the CCRP objective, implies that we are not able to reduce the DAPR objective in the second phase, as opposed to the CCRP + DAPR, where a small reduction was possible compared to the rosters found in the first step.

Next, we analyze the empirical curves. As can be seen from Figure 7.2 the Pareto-alike analysis shows the effectiveness of the integrated approach. (We remark that for $\gamma = 8$ all approaches found solutions around $\gamma = 6$.) Both the ICCRP and the ICCRP* approaches are able to exploit higher values of γ. Note that the curves are not necessarily decreasing because we do not solve the subproblems to optimality. The CCRP curve, on the other hand, seems not to be able to exploit these higher values at all. This is no surprise, as this curve represents the sequential approach, i.e., the allocation we find in the first step does not consider the rostering objective, and thereafter we assume the allocation as fixed. Comparing the ICCRP and ICCRP* curves we see that the benefit from post-processing is small for this instance, i.e., the solutions found by the ICCRP are not improved easily.

Summarizing, we observed that the use of the integrated approach was effective for this instance. The approaches with tight γ and β bounds found solution at the boundaries. The use of the ICCRP model, however, allowed us to find solutions in between to bridge this gap.

Case 2: All Roster Groups

For our second instance we solve the ICCRP for all 16 roster groups. This gives a total of 625 duties that need to be rostered. We now consider all approaches an curves as discussed in Chapter 6. For the 4-CCRP and 4-ICCRP The four groups are constructed simply on their index (e.g., groups 1 to 4, 5 to 9). The results are shown in Figure 7.3.

First, we look at the single points in Figure 7.3, i.e., the solutions founds by the approaches 1 to 7 discussed in Chapter 6. We see that the approaches that consider the large scale rostering problem perform poorly; both the DAPR + ICCRP and the CCRP + DAPR have a high rostering objective compared to the other solution approaches. In case of the DAPR + ICCRP we were not able to improve the

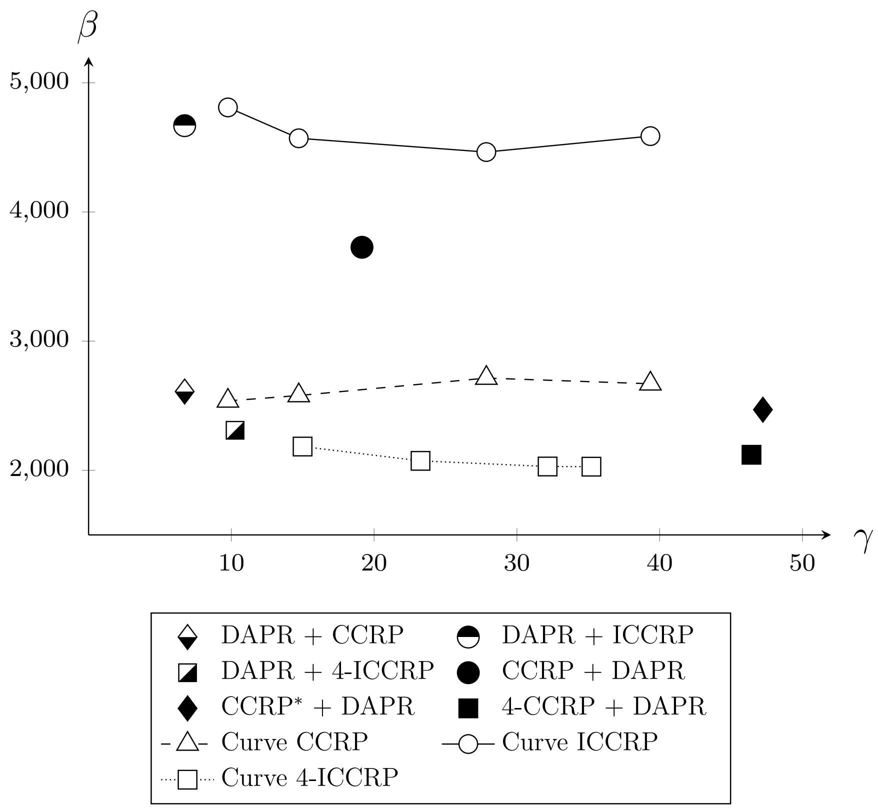

Figure 7.3: Results All Groups.

solution found in the first step (therefore the DAPR + ICCRP* is omitted in the figure). This indicates that solving the whole instance all at once is not efficient for the allowed time, as no good solutions are found. Again, we see that their is a clear distinction between the approaches with a γ-constraint (i.e., the black and white symbols) versus the approaches with a β-constraint (i.e., the solid symbols) in terms of the duty assignment objective. In terms of the rostering objective, however, this distinction is less clear. This can probably be attributed to the fact that the large scale rostering approaches had much trouble with finding good solutions.

Interesting to see is that the DAPR + 4-ICCRP and the 4-CCRP + DAPR outperform the DAPR + CCRP and the CCRP* + DAPR, in terms of the rostering objective. In this case, minimizing the rostering objective for the subsets of groups simultaneously is beneficial, compared to minimizing it for all groups separately. This was also observed for the first instance. We note that the DAPR + 4-ICCRP violates the γ-bound opposed (note that it is located right form the other two approaches with the γ-constraint). This is, as mentioned in Chapter 6, due to the fact we oppose the bound on each of the sets of four groups separately. We observe, however, that the violation is relatively small.

Next, we analyze the empirical curves. Based on the results of the singular approaches we considered $\gamma \in \{10, 15, 30, 40\}$. We omitted the curve for the ICCRP* as it coincided with the curve for the CCRP, i.e., the ICCRP was not able to improve any of the solutions when a γ-bound was enforced. We observe an interesting phenomenon in Figure 7.3, which was not visible in the results of the first instance. As mentioned, the ICCRP was not able to exploit the higher γ to improve the rostering objective. Similarly, we see that the curve for CCRP remains somewhat constant, as the sequential approach is not able to directly use the higher γ values in the solution process. Interesting is that the 4-ICCRP lies strictly below the CCRP curve; rostering multiple groups simultaneously allows us to obtain rosters with a lower objective value. We also see that this is the only curve that decreases in terms of γ, i.e., per set of four groups the higher γ values allow us to obtain better rosters. Again, we note that the desired γ is often not met, one could circumvent this issue by setting e.g., lower targets for each of the subsets of groups. Finally, we note that the curve contains points below the 4-CCRP + DAPR solution; this is possible because both solution approaches start with a certain fixed assignment of duties to the groups, hence a higher γ does not necessarily imply we find rosters with lower cost.

Summarizing, we found that for the large instance decomposing the problem into smaller problems greatly outperforms solving the problem as a whole. Due to the limited solution time, rostering many duties at once leads to very poor solutions. There is, however, a gain in considering subsets of groups instead of all groups separately. Such an approach still allows us to benefit from the ICCRP formulation, as was clearly visible in the first instance. We also note that the distinction between the approaches with a γ-constraint and a β-constraint is less clear in terms of the rostering objective. This, however, can be attributed to the difficulty in finding good rosters for all groups at once.

Chapter 8

Conclusion and Further Research

In this thesis we developed two mathematical models for the Integrated Duty Assignment and Crew Rostering problem. This problem consists of allocating duties to different roster groups in a 'fair' manner while minimizing the cost of the rosters. This problem is one of the many problems Netherlands Railways deals with in practice. We considered multiple solution approaches to this problem.

In Chapter 3 we discussed the model for the Duty Assignment problem as was proposed in Abbink [2014]. We showed how this model changes as the rota schedules are assumed to be known. This assumption is often satisfied in practice, as it is desirable for employees to have a certain level of predictability in their roster. In Chapter 4 we developed a mathematical model for the Crew Rostering problem, based on the work of Hartog et al. [2009] and Abbink [2014]. We extended this model to multiple roster groups and finally proposed the ICCRP model, in which we integrated the Duty Assignment and Crew Rostering problem. The model is formulated as a bi-objective optimization problem.

In Chapter 5 we proposed an alternative to the ICCRP model, which we referred to as the ICCRP2 model. This model is a 'higher-level' formulation of the problem, in the sense that large clusters of duties are assigned simultaneously. Our motivation for this model was that it gives a tighter formulation of the ICCRP; in Section 5.5 we proved this rigorously.

In Chapter 6 we proposed multiple approaches to solve the bi-objective version of the ICCRP. In our approaches we focused mainly on the trade-off between the two objectives, i.e., which objective we minimize first, as well as possible decompositions of the problem, e.g., solving the problem separately per group.

We applied our approaches to the rostering process at base Utrecht. Our results were promising, and showed possible avenues for further research. The CCRP2 was compared to the CCRP model on instances of approximately 100 duties. The results showed that the CCRP2 was able to find optimal solutions in reasonable time, a major improvement compared to the CCRP for these instances.

When applying our solution approaches to larger instances the results where somewhat divided. For four roster groups, i.e., roughly 150 duties, we obtained good results with our integrated approach. We saw a clear distinction between the solutions of approaches that either minimized the duty assignment objective first, or the rostering objective first. This was expected, and was, in fact, a motivation for our integrated approach. We observed that, using the ICCRP model, we were able to bridge the gap between such methods, i.e., we obtained good rosters, while still enforcing a certain level of the allocation.

When rostering all groups at once, the results were not as good. Especially the integrated approach failed to improve the found solutions. We did find evidence that decomposing the problem is beneficial in this case; especially our method that divided the set of 16 roster groups in sets of 4 groups was able to find good solutions, from a rostering perspective. We must note that we focused on finding good solutions in reasonable time.

In terms of further research, especially the ICCRP2 model seems promising. It would be interesting to combine this model with e.g., a Column Generation approach in order to apply it to larger instances. It is very likely that such an algorithm outperforms the current ICCRP model. Another interesting avenue for research would be the use of decomposition approaches. We note that the ICCRP is especially well suited for such approaches, as the patterns can be decomposed per group. In our research we only used this marginally, by solving the rostering problems for groups separately. We think, however, it would be fruitful to consider more advanced approaches.

Furthermore, as the different approaches showed, the ICCRP seems to be a good candidate for a more advanced neighborhood search algorithm. We note that our approaches can be seen as 'naive' versions of such algorithms, as we first optimize globally and thereafter try to improve locally (e.g., we first minimized the duty assignment objective, and then try to improve the roster per group). It would be interesting to consider such approaches in a larger framework of algorithms in which we fix e.g., only certain weeks in the roster and then re-optimize.

Finally, a thorough analysis of the relation between certain patterns would be an interesting new direction for research. In our approach, and many other approaches to rostering problems, the constraints are considered fixed and we try to find the best possible roster given those constraints. It could, however, be very fruitful to analyze how certain combinations of constraints influence the optimal solution, i.e., one can imagine that certain combinations can have a very big impact while others have very little impact. Such an analysis could have great practical value for e.g., Netherlands Railways.

Bibliography

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

E. Abbink, L. Albino, T. Dollevoet, D. Huisman, J. Roussado, and R. Saldanha. Solving large scale crew scheduling problems in practice. *Public Transport*, 3(2): 149–164, 2011.

A. Caprara, M. Fischetti, P. Toth, D. Vigo, and P. Guida. Algorithms for railway crew management. *Mathematical Programming*, 79(1-3):125–141, 1997.

I. Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 972–981. Society for Industrial and Applied Mathematics, 2008.

T. Emden-Weinert, H. Kotas, and U. Speer. Dissy - A driver scheduling system for public transport. *Lecture Notes in Economics and Mathematical Systems, Springer Publishers*, 2001.

A. Hartog, D. Huisman, E. Abbink, and L. Kroon. Decision support for crew rostering at NS. *Public Transport*, 1(2):121–133, 2009.

D. Huisman, L. G. Kroon, R. M. Lentink, and M. J. C. M. Vromans. Operations research in passenger railway transportation. *Statistica Neerlandica*, 59(4):467–497, 2005.

L. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, A. Schrijver, A.I. Steenbeek, and R. Ybema. The New Dutch Timetable: The OR Revolution . *Interfaces*, 39(1):6–17, 2009.

M. Mesquita, M. Moz, A. Paias, and M. Pato. A decomposition approach for the integrated vehicle-crew-roster problem with days-off pattern. *European Journal of Operational Research*, 229(2):318–331, 2013.

M. Sodhi and S. Norris. A flexible, fast, and optimal modeling approach applied to crew rostering at London Underground. *Annals of Operations Research*, 127 (1-4):259–281, 2004.

L. Xie and L. Suhl. Cyclic and non-cyclic crew rostering problems in public bus transit. *OR Spectrum*, 37(1):99–136, 2014.

X. Zhu, Q. Li, W. Mao, and G. Chen. Online vector scheduling and generalized load balancing. *Journal of Parallel and Distributed Computing*, 74(4):2304–2309, 2014.

B. Zimmermann, M. C. Schippers, and L. G. Kroon. Railway Crew Rostering and its Effects on the Planned Workforce. Technical report, 2012.

Appendix A

Distributions of Attribute Values

In this Appendix we show for all attributes the distribution of the values. In each of the figures the mean value is indicated.

In Figure A.1 the distribution of duty lengths is shown.

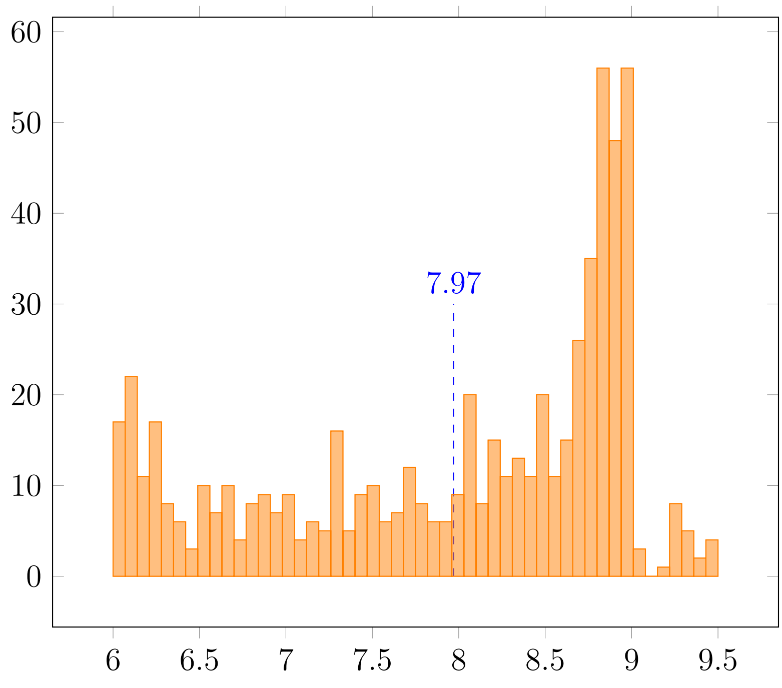

Figure A.1: Histogram duty lengths.

In Figure A.2 we show the distribution of the percentages of type-A rolling stock in the duties.

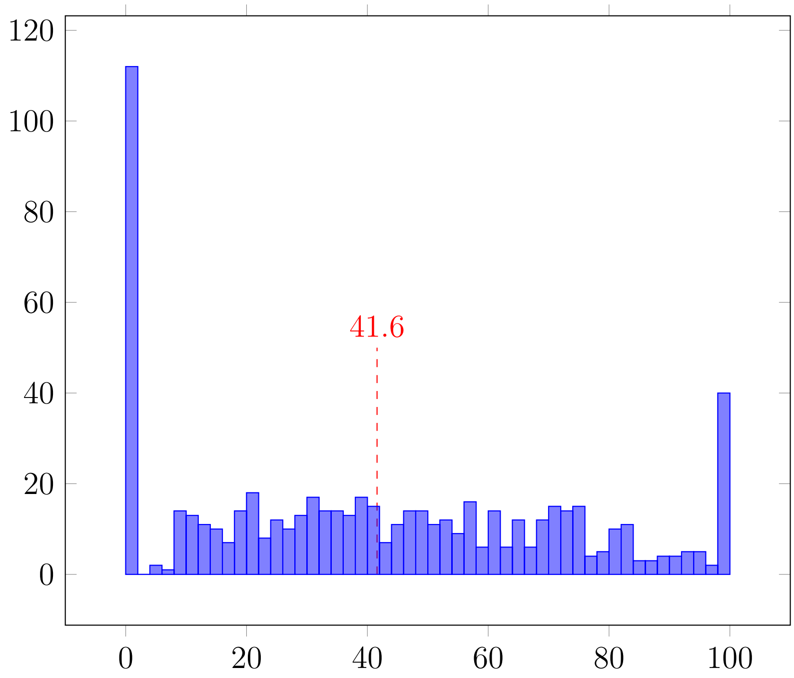

Figure A.2: Histogram percentage work on type A-trains.

In Figure A.3 we show the distribution of the percentages of aggression work in the duties.

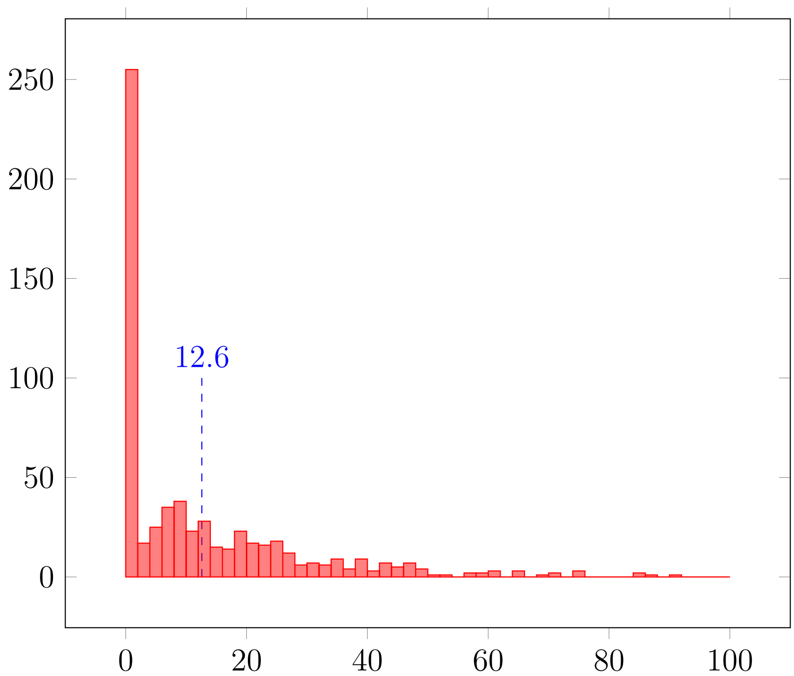

Figure A.3: Histogram percentage aggression work.

In Figure A.4 the distribution of work on double decker trains is shown.

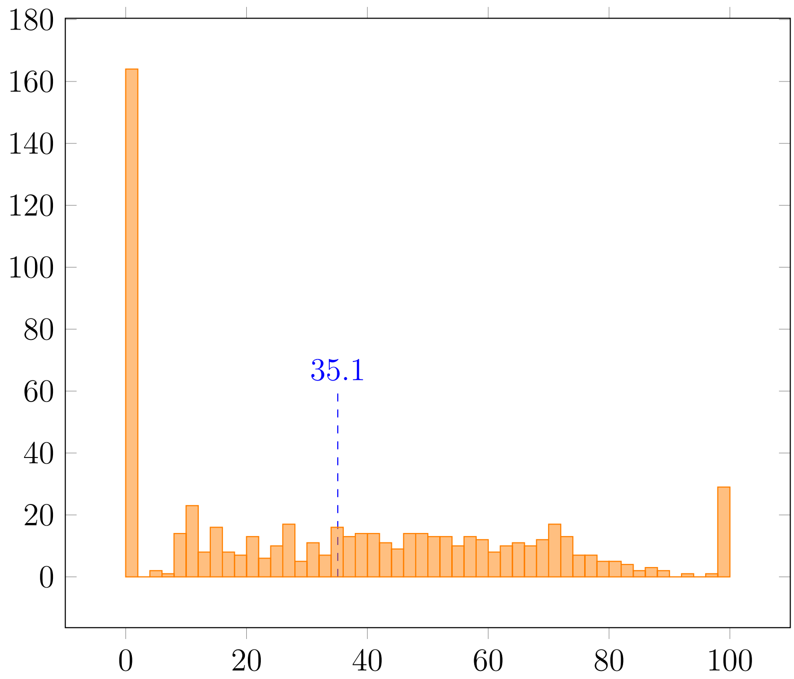

Figure A.4: Histogram percentage work on Double Decker trains.

Finally, we show in Figure A.5 the distribution of the values of the Repetition Within Duty (RWD) coefficient for the duties.

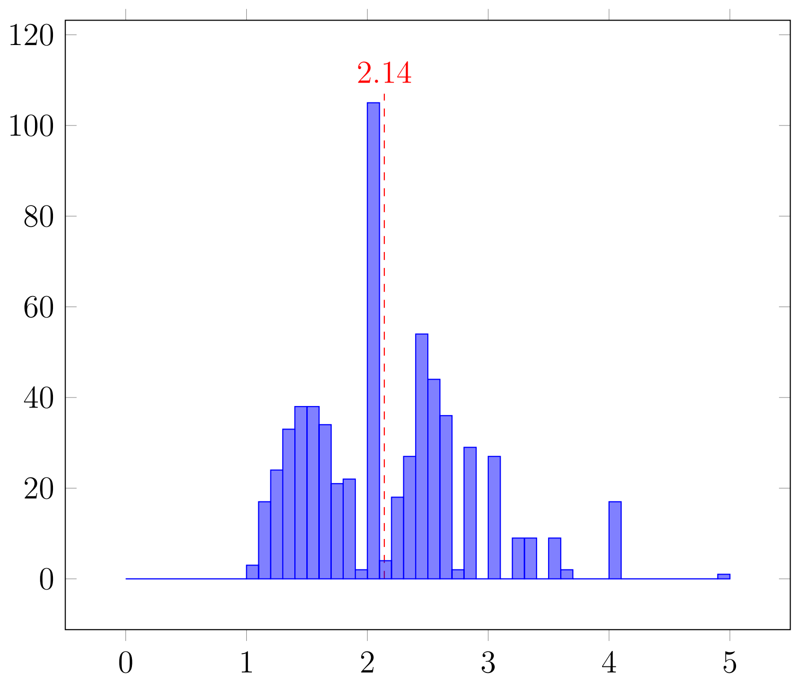

Figure A.5: Histogram RWD values.