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

Thomas Breugem[1], Twan Dollevoet[1], Dennis Huisman[1,2]

[1]Econometric Institute and Erasmus Center for Optimization in Public Transport
Erasmus University Rotterdam, The Netherlands

[2]Process quality and Innovation, Netherlands Railways
Utrecht, The Netherlands

breugem@ese.eur.nl, dollevoet@ese.eur.nl, huisman@ese.eur.nl

Econometric Institute Report Series - EI2017-30

Abstract

In this paper, we analyze the trade-off between perceived fairness and perceived attractiveness in crew rostering. First, we introduce the Fairness-oriented Crew Rostering Problem. In this problem, attractive cyclic rosters have to be constructed, while respecting a pre-specified fairness level. Then, we propose a flexible mathematical formulation, able to exploit problem specific knowledge, and develop an exact Branch-Price-and-Cut solution method. The solution method combines Branch-and-Bound with column generation, where profitable columns are separated by solving resource constrained shortest path problems with surplus variables. We also derive a set of valid inequalities to tighten the formulation. Finally, we demonstrate the benefit of our approach on practical instances from Netherlands Railways, the largest passenger railway operator in the Netherlands. We are able to construct the explicit trade-off curve between fairness and attractiveness and show that a sequential approach can lead to suboptimal results. In particular, we show that focusing solely on fairness leads to rosters that are disproportionally less attractive. Furthermore, this decrease in attractiveness is heavily skewed towards the most flexible employees. Thus, in order to generate truly fair rosters, the explicit trade-off between fairness and attractiveness should be considered.

Keywords: Fairness, Crew Rostering, Integrated Crew Planning, Column Generation, Branch-Price-and-Cut, Distributive Justice

1 Introduction

In recent years, the Netherlands has seen numerous strikes from personnel of Netherlands Railways (NS). Reasons for these strikes were, for example, little variation in work (adding the infamous ‘rondje om de kerk’[1] to the Dutch vocabulary), or demand for higher staffing levels for certain rolling stock. One of such conflicts led to the development of the ‘Sharing-Sweet-and-Sour’ rules (‘Lusten-en-Lasten-Delen’ in Dutch), a new set of scheduling rules aimed at increasing the quality of work. As mentioned in Abbink et al. [2005], one of the key success factors of the project was the openness and transparency during the development of the new rules. Using state-of-the-art Operations Research techniques, NS was able to generate schedules satisfying the new rules for all employees, leading to a new agreement between NS and the labor unions. These events highlight the importance of a participative approach to crew planning. In particular, they show the importance of incorporating the demands of personnel in the planning process.

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

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

[1] Translated as ‘circling the church’, a mocking reference to repetitive work.

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

The contribution of this paper is fourfold. First, we formalize the FCRP and propose a mathematical formulation to solve the problem. The formulation we propose is versatile, and can be easily adapted to different settings. Secondly, we develop an exact Branch-Price-and-Cut solution method for the FCRP. Thirdly, we apply the solution method to real life instances at NS, where we show the benefits of our integrated approach. In particular, we generate multiple rosters with a different trade-off between fairness and attractiveness, and show that the roster with the highest fairness might not be the most desirable one. Fourthly, we decompose the unattractiveness of the rosters over the different groups, and show that tight fairness levels lead to a decrease in the attractiveness that is unevenly distributed over the groups.

The remainder of the paper is organized as follows. In Section 2, we discuss crew rostering in detail and formalize the FCRP. In Section 3, we give an overview of related research. Our mathematical model is introduced in Section 4. In Section 5, we propose a Branch-Price-and-Cut approach to solve the FCRP. In Section 6, we show the benefits of our approach in a case study at NS, and discuss the acquired managerial insights. The paper is concluded in Section 7.

2 Problem Description

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

The input to the crew rostering phase, as considered in this paper, is a *basic schedule* for each roster group, and a set of *generic duties* (often simply referred to as duties). Each basic schedule specifies the key elements of the roster. This schedule, for example, could specify when employees have a day-off or what type of work will be scheduled on a certain day (the level of detail may vary according to the application). Figure 1 shows an example of a set of basic schedules. In this example, the foremost basic schedule specifies that the first week of the roster starts with a late duty, followed by two night duties, and then a day off (indicated by L, N and R in the basic schedule, respectively).

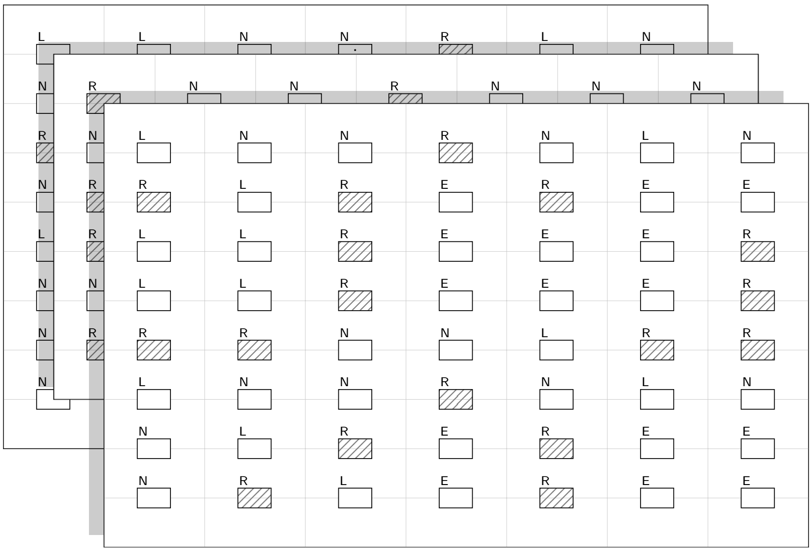

Figure 1: Example of basic schedules. The schedules specify the early (E), late (L), and night (N) duties; and rest days (R).

The use of generic duties is a consequence of the cyclic rosters. Recall that in a cyclic roster multiple employees cycle through the weeks of the same roster. Hence, scheduling a duty on a given day in the roster means that the duty is performed every week, only each week by a different employee of the roster group. The duties are therefore generic, which means that each duty belongs to, for example, Wednesday, and not to, say, Wednesday the 11th of October. A detailed example of a cyclic roster is shown in Figure 2. Each of the boxes represents a day of work, and each of the numbers indicate a different duty. The roster consists of four weeks, meaning that it belongs to a roster group of four employees. The cyclicity of the roster implies that the second employee starts in week 2, and, after completing this week, he or she carries out the duties of week 3. Similarly, the duty indicated by 118 on Monday, is first carried out by the third employee, and then a week later by the second employee. Note that the cyclicity implies that after four weeks each employee has carried out each duty shown in Figure 2 exactly once. Furthermore, note that some overlap between weeks can be present. For example, the last duty of week 3 starts on Sunday night and ends on Monday morning.

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

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

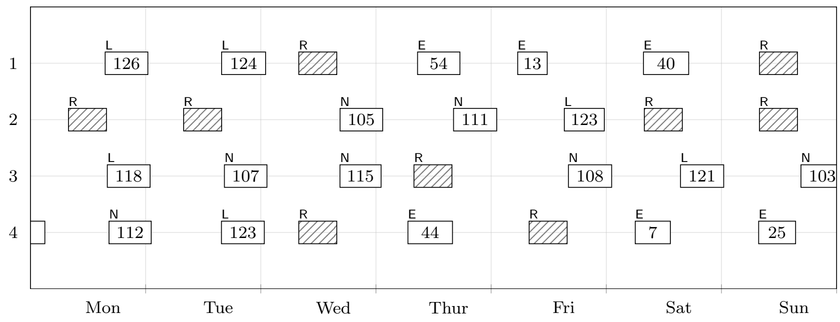

Figure 2: Example of a cyclic roster for four employees.

of multiple tasks, it has certain characteristics. Hence we can say that some duties are 'desirable' and some are not (e.g., a duty with many short trips is undesirable, while a duty with fewer but longer trips is considered desirable). We refer to these characteristics as duty attributes. Examples of duty attributes at NS are the duty length and the percentage of work on double decker trains. A fair allocation means that the spread (i.e., the difference between the maximum and minimum average value) over the groups is minimized. Furthermore, certain bounds on the allocation must be respected, following labor regulations.

The perceived attractiveness and the feasibility of a roster are expressed using *roster constraints*. These constraints impose restrictions on the assignment of the duties to the roster. Rest constraints, for example, are often present in crew rostering. Rest constraints enforce that an employee has a certain minimum time to rest after a duty (if it is a hard constraint), or penalize rest times shorter than a certain threshold (if it is a soft constraint). Another classical example are workload constraints, enforcing a maximal amount of work within a week. The perceived attractiveness is maximized by minimizing the penalty incurred from the soft roster constraints. As we will show in Section 4, the developed model allows for a broad range of roster constraints. In Section 6, we discuss how the duty attributes and roster constraints are applied at NS.

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

3 Literature Review

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

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

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

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

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

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

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

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

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

4 Mathematical Model

In this section we present a mathematical formulation of the FCRP. In Section 4.1, we introduce the concept of roster sequences, and in Section 4.2 we discuss roster constraints in detail. We introduce the necessary notation in Section 4.3, and we conclude by giving a mathematical formulation for the FCRP in Section 4.4.

4.1 Roster Sequences

We develop a mathematical model based on the *weeks* of each roster. That is, we simultaneously assign a number of duties to a week, instead of assigning a duty to each day separately. For the ease of explanation, we will assume that a week always starts on Monday and ends on Sunday. Constraints regarding the weeks occur naturally in rostering (e.g., workload constraints). Many roster constraints at NS, for example, consider the weeks of the roster (we will discuss these in detail in Section 6). Hence, modeling the rostering problem based on the weeks often leads to a strong formulation, as it implies that many of the roster constraints can be modeled implicitly. Initial experiments showed that modeling based on the weeks of the roster improved the performance of the algorithm substantially compared to modeling based on the days of the roster.[2]

The key concept of our modeling approach is the use of *roster sequences*. Modeling the problem based on the weeks of a roster implies we need to simultaneously assign multiple duties to all days in a week. We call such an assignment of duties a roster sequence (or

[2]We note that a week based model is not a prerequisite for our approach: the approach can be used to develop a model for every partition of the days in the basic schedule. For the clarity of the exposition, however, we will focus on weeks.

simply a *sequence*, if there is no ambiguity). Note that the roster sequences should always satisfy the basic schedule.

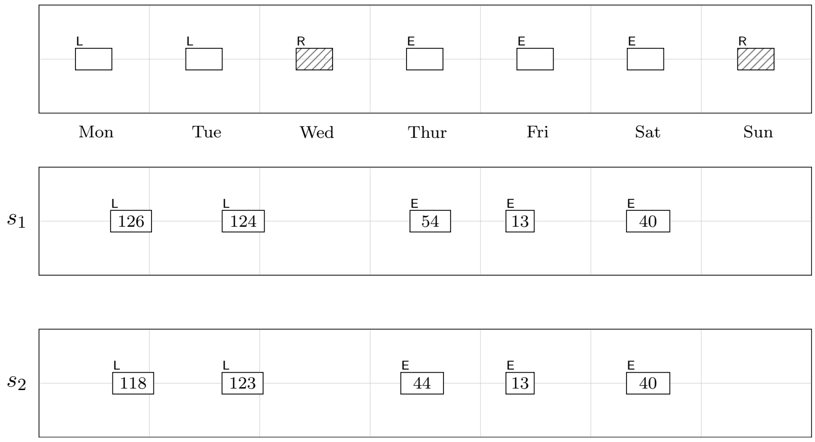

Figure 3: Example Sequences.

Figure 3 shows an example of possible roster sequences. The first row in Figure 3 shows the basic schedule of Week 1 (as shown in Figure 2), and the second and third row show two roster sequences s_1 and s_2 for this week. The first sequence is obtained from the assigned duties in Figure 2, while the second sequence is obtained by swapping some duties currently assigned to Week 1 with those assigned to Weeks 3 and 4. Note that both sequences satisfy the given basic schedule, and that the sequences only specify the assigned duties (i.e., the rest days are not specified in the sequences, as these are input to the problem).

4.2 Roster Constraints

Roster constraints impose restrictions on the assignment of duties in the roster, and are used to quantify the attractiveness of the roster. We consider both *hard* roster constraints (i.e., constraints that must always be satisfied) and *soft* roster constraints (i.e., constraints that may be violated against a certain penalty). Each roster constraint is fully specified by a coefficient for each possible assignment of a duty to a day in the basic schedule, a *threshold value*, and a *violation interval*. It is assumed that each violation interval is a closed interval on the real line (this assumption is important when developing the Branch-Price-and-Cut approach, as will be noted in Section 5.2).

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

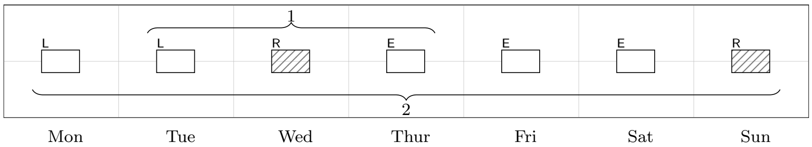

Figure 4: Example Roster Constraints.

A wide variety of practical constraints can be expressed in this general form (for a thorough discussion we refer to Hartog et al. [2009]). To illustrate this, consider the set of days depicted in Figure 4, which corresponds to the first week shown in Figure 2. Two different roster constraints are indicated.

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

Roster constraints concerning solely a single week of the basic schedule are modeled implicitly. That is, these roster constraints are incorporated in the penalties associated with the sequences, instead of explicitly modeled as a constraint in the model. Note that this holds for both roster constraints depicted in Figure 4.

4.3 Notation

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

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

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

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

4.4 Mathematical Formulation

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

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

fairness of the solution.

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

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

$$
\begin{aligned}
\min \quad & \sum_{k\in K}\sum_{s\in S_k} c_s^k x_s^k + \sum_{p\in P\setminus P_K} c_p\delta_p && && (1)\\
\text{s.t.} \quad & \sum_{a\in A} w_a\left(z_a - v_a\right) \le \zeta && && (2)\\
& \sum_{s\in S_k} x_s^k = 1 && \forall k \in K && (3)\\
& \sum_{k\in K}\sum_{s\in S_k^d} x_s^k = 1 && \forall d \in D && (4)\\
& \sum_{k\in K}\sum_{s\in S_k}\sum_{(t,d)\in s} f_{td}^p x_s^k \le b_p + \delta_p && \forall p \in P \setminus P_K && (5)\\
& \sum_{k\in K_r}\sum_{s\in S_k}\sum_{(t,d)\in s} g_{ad} x_s^k \le n_r z_a && \forall a \in A, r \in R && (6)\\
& \sum_{k\in K_r}\sum_{s\in S_k}\sum_{(t,d)\in s} g_{ad} x_s^k \ge n_r v_a && \forall a \in A, r \in R && (7)\\
& z_a \le u_a && \forall a \in A && (8)\\
& v_a \ge l_a && \forall a \in A && (9)\\
& x_s^k \in \mathbb{B} && \forall k \in K, s \in S_k && (10)\\
& \delta_p \in \Delta_p && \forall p \in P \setminus P_K && (11)\\
& v_a, z_a \in \mathbb{R} && \forall a \in A. && (12)
\end{aligned}
$$

The objective (1) expresses that we minimize the sum of the sequence penalties, together with the cost of all explicitly modeled roster constraints. Constraint (2) assures that the desired fairness level is achieved. Recall that the desired level ζ is assumed to be exogenous. As discussed in Section 2, the perceived fairness is calculated as a weighted sum of the spread (i.e., the difference between the maximum and minimum average values) of the duty attributes with respect to the roster groups. Note that (2) can be interpreted as an epsilon-constraint for the bi-objective problem (see Ehrgott [2000]). Constraints (3) and (4) assure that the duties are assigned correctly to the basic schedules. That is, each week is assigned exactly one roster sequence, and each duty is assigned exactly once. Constraints (5) consider the roster constraints. Note that (5) assures that δ_p takes the value of the constraint violation. Constraints (6) and (7) assure that the variables v_a and z_a are set to the minimum and maximum value, respectively, while (8) and (9) enforce the lower and upper bounds on the attribute values. Finally, Constraints (10)-(12) express the domains of the decision variables.

5 Solution Approach

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

The lay-out of this section is as follows. In Sections 5.1 and 5.2, we discuss the master problem and the resulting pricing problems, respectively. In Section 5.3 we discuss the branching strategy. We conclude in Section 5.4 with a discussion on how we obtain all efficient solutions of the FCRP.

5.1 Master Problem

In Branch-Price-and-Cut algorithms the problem is decomposed in a master problem and a set of pricing problems. The Master Problem is obtained from (1)–(12) by relaxing the integrality condition for the x_s^k variables. That is, Constraints (10) are replaced with $x_s^k \geq 0$, for all $k \in K$ and $s \in S_k$. Note that (3) assures that $x_s^k \leq 1$. Because the number of sequences can be very large, we use column generation to solve the Master Problem. This means that only a subset of the sequences is considered instead of the entire set of sequences. As this might lead to suboptimal solutions, profitable sequences are added based on their reduced cost. Deciding whether such sequences exist is done in the pricing problems, where sequences with negative reduced cost are separated. The linear relaxation of (1)–(12) is tightened by adding valid inequalities (see Appendix A).

5.2 Pricing Problem

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

An example of such a graph is shown in Figure 5. Here we show the pricing graph for the week depicted in Figure 4. Each vertex specifies a combination of a day and a duty. The duties are obtained from the roster in Figure 2. Note that certain days can be assigned multiple duties (e.g., Monday of type L can be assigned duties d_{126} and d_{118}). Clearly, each $s - t$ path in such a graph corresponds to a roster sequence.

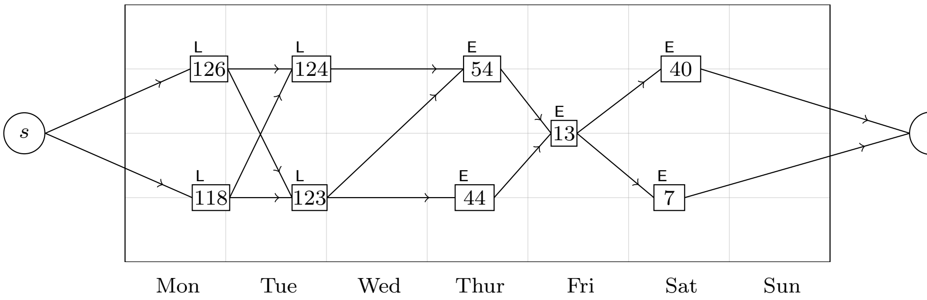

Figure 5: Example Pricing Graph.

The reduced cost of a sequence can be modeled using the arc set and the resource constraints. The dual multipliers are readily incorporated in the arc costs (see Appendix B). The difficulty lies in expressing the primal cost, that is, in expressing the cost related to the different roster constraints that are implicitly modeled. This is done using the resource constraints. Each roster constraint p relates to a resource with consumption f_{td}^p at node (t, d) and consumption limit b_p. The violation interval Δ_p is modeled using the domain of the surplus variables. Here we use the fact that each Δ_p represents a closed interval on the real line. We note that binary constraints (i.e., constraints involving two subsequent duties) are a special case, as they can be directly incorporated into the arcs. An example of this is the 'missing' arc between the duties d_{124} and d_{44} on Tuesday and Thursday, respectively. The time between duties d_{124} and d_{44} is insufficient for a proper rest day on Wednesday, hence no sequence can have these two duties scheduled sequentially.

We solve the RCSPP using an enumerative best first search approach, thereby using Lagrangian distance labels to prune provably suboptimal paths (cf. Beasley and Christofides [1989], Grötschel et al. [2003], Dumitrescu and Boland [2003]). The solution approach exploits the dual formulation of the RCSPP to obtain valid lower bounds on optimal path lengths (see Appendix C). This approach resembles the well-known A* algorithm, introduced by Hart et al. [1968].

5.3 Obtaining Integer Solutions

To obtain integer solutions we apply a bi-level branching strategy. In the first branching rule, we consider the assignment of duties to the basic schedules, i.e., pairs (r, d). We branch if a duty is only partially assigned to a basic schedule. That is, we construct a branch where the duty is assigned only to this schedule, and a branch where the duty is excluded from this schedule. Note that partially assigning a duty to a schedule is equivalent to assigning a duty to multiple schedules (a direct consequence of (3) and (4)).

If multiple branching possibilities are present, we branch on the most fractional value.

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

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

5.4 Finding Pareto-optimal Solutions

We want to identify all Pareto-optimal solutions for the FCRP. A solution is Pareto-optimal if there exists no solution that is better in all criteria. In the FCRP this means that there can be no solution with higher attractiveness *and* higher fairness level. We refer to Ehrgott [2000] for an introduction to multi-objective optimization.

All Pareto-optimal solutions to a bi-objective optimization problem can be found by imposing a bound on one objective, while optimizing the other. In the FCRP, this bound is the desired fairness level ζ in (2). By varying ζ iteratively, we obtain all Pareto-optimal solutions (up to a certain precision). That is, we solve the problem for a sequence of strictly decreasing fairness levels, until a feasible solution can no longer be found. In doing so, we are able to re-use the Branch-and-Bound tree of previous iterations, thereby avoiding the repeated exploration of the same part of the solution space.

6 Case Study at NS

To show the practical benefits of the FCRP we apply our solution method to a set of rostering instances of NS. The instances are based on roster groups of guards at depot Utrecht, one of the major crew depots in the Netherlands. In Section 6.1, we discuss our experimental set-up, and all attributes and roster constraints that NS takes into account. In Section 6.2, we present the result of our experiments. We conclude in Section 6.3 with the practical insights obtained from the case study.

6.1 Rostering at Base Utrecht

Each rostering instance specifies a set of basic schedules and a set of duties. The instances in our experiments are based on a set of roster groups and their corresponding basic schedules. The duties are obtained from the original rosters, to assure compatibility with the basic schedules. We consider three different types of duties: early, late and night. These are denoted by E, L and N, respectively. Furthermore, a day in the roster can be an official rest day (denoted by R), or an unofficial rest day. For the latter, no constraints have to be taken into account. The basic schedules specify one of these types for each day in the roster. Figure 1 gives an example of a basic schedule.

The roster groups are divided into three categories, based on their basic schedules. The first category consists of groups performing solely early duties, the second category of groups performing solely late and night duties, and the third category of groups performing all three types of duties. The first two instances consist of two large groups of the first two categories, and a relatively small group of the third category. The other two instances consist of one group of each category, each representing roughly the same number of employees. The details are shown in Table 1. The group size refers to the number of employees in the group (i.e., the number of weeks that need to be rostered).

Table 1: Rostering Instances Base Utrecht.

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

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

- *Duty length.* For each of the duties, the length is defined as the difference between the start and end time of a duty.
- *Percentage of type-A work.* Trips with type-A work are desirable, and hence this work needs to be fairly distributed.
- *Percentage of aggression work.* Certain trips have a higher chance of passenger aggression (due to e.g., passengers not having a ticket). Trips where such situations frequently occur are undesirable and therefore need to be fairly distributed.
- *Percentage of work on double decker trains.* Work on double decker trains is considered undesirable, as it is physically more demanding (due to the high amount of stairs an employee has to climb during work on such a train). We therefore want to distribute this type of work equally.

- *RWD values.* The Repetition Within Duty (RWD) values are defined as the total number of routes divided by the total number of *distinct* routes in the duty. From an employee point of view, variation is desirable (reflected by a low RWD value), and hence we want to balance the RWD values among the groups.

Table 2: Specification Duty Attributes.

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

Each attribute has a pre-specified lower or upper bound which should be respected for each group (see Table 2). Furthermore, each attribute is given a weight to model the relative contribution to the fairness measure. For example, a spread of 5% in aggression work and a spread of 10 minutes in duty length are penalized equally. Table 2 also shows the average value for each of the instances, displaying the tightness of the imposed bounds.

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

- *Rest Constraints.* After completing a duty it is required that an employee has a certain minimum time to rest. A rest time of 14 hours is required after a night duty. For the other types of duties the required rest time is 12 hours. Rest times shorter than 16 hours are undesirable, and are therefore penalized with a penalty of 30. Note that this penalty is independent of the shortage in rest time.
- *Rest Day Constraints.* If two work days have one rest day in between, there should be at least 30 hours between the end of the first duty and the start of the second. Furthermore, for each additional rest day that is between the duties, another 24 hours are required.
- *Workload Constraints.* We require that the total workload in one week should not exceed 45 hours. Here, workload is measured as the difference between start and end time (i.e., including the meal break). The workload is always measured from Monday to Sunday (i.e., a 'regular week').
- *Variation Constraints.* It is considered desirable if e.g., aggression work is distributed evenly over the weeks. Therefore, variation constraints penalize positive deviations from the average for each of the duty attributes. Furthermore, we take duties with a duration longer than 9 hours and a set of (un)desirable routes into account. This gives a total of 10 variation constraints per week. Similar to the workload constraints, the variation is always measured from Monday to Sunday.

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

6.2 Computational Results

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

6.2.1 Computation Times

All experiments were done on a computer with a 1.6 GHz Intel Core i5 processor. We used the LP solver embedded in CPLEX 12.7.1 to solve the master problems. Recall from Section 5.4 that we re-use the Branch-and-Bound tree, hence all solutions are obtained in one execution of the algorithm. The time to find a Pareto-optimal point is therefore cumulative, i.e., depends on the solutions previously found. Table 3 shows for each instance the number of Pareto-optimal points, and the total computation time of the algorithm (in minutes). Furthermore, Table 3 shows for each instance the time until the point with minimum attractiveness penalty (i.e., the rightmost point of the curve) is obtained. Note that this is always the first point found by the algorithm. We used a step-size of 0.1 for iteratively updating the fairness level.

Table 3: Computation times per instance.

Instance	Nr. Points	Min Attr. Pen. (min)	Total Time (min)
U1	23	7.9	287.9
U2	12	37.0	236.8
U3	17	0.1	132.2
U4	22	3.9	470.8

Each of the Pareto-optimal curves could be determined in reasonable time. The Pareto-optimal curve for instance U4 took most time to compute (about 8 hours). The other three Pareto-optimal curves were found in substantially less time (at most 5 hours). Furthermore, the solution with minimal attractiveness penalty was found quickly for all instances. For three out of four instances the corresponding Pareto-optimal point was found in only a few minutes.

6.2.2 Pareto-optimal Curves

The Pareto-optimal curves for the four instances are depicted in Figure 6. From hereon we refer to the objective value (i.e., the measure of perceived attractiveness) as the *attractiveness penalty* and to ζ as the *fairness penalty*.

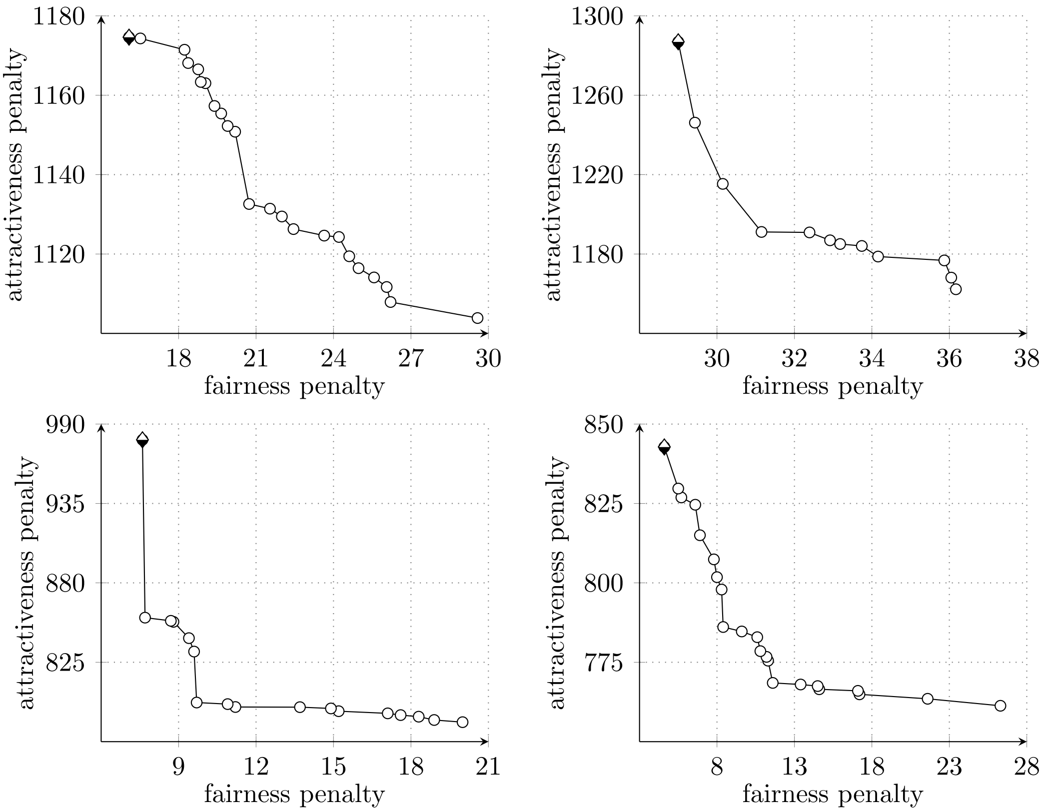

Figure 6: Pareto curve for instances U1 (top left), U2 (top right), U3 (bottom left) and U4 (bottom right).

Figure 6 shows that many Pareto-optimal solutions are lost in a sequential approach. Note that a sequential approach, where perceived fairness is optimized first, results (at best) in the solution represented by the leftmost point in the curve (those highlighted in Figure 6). The set of Pareto-optimal points, however, covers a broad spectrum of solutions, each of which might be optimal depending on the given preferences. The curve of instance U1, for example, ranges from 1104 to 1175 (a relative spread of about 6%) with respect to the attractiveness penalty and from 16 to 30 (a relative spread of about 85%) with respect to the fairness penalty.

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

6.2.3 Trade-Off Analysis

The Pareto curves in Figure 6 exhibit a clear 'tail-off' effect. That is, decreasing the fairness penalty close to the minimum leads to a disproportionally large increase in the attractiveness penalty, and vice versa. To highlight this behavior, we standardize each solution based on the attained minimum and maximum penalties. For example, if some solution attains a fairness penalty of ζ, we compute the *relative decrease* of ζ. That is, we compute the quantity

$$\frac{\zeta^{\max} - \zeta}{\zeta^{\max} - \zeta^{\min}} \in [0, 1], \tag{13}$$

where $\zeta^{\min}$ and $\zeta^{\max}$ denote the lowest and highest observed fairness penalties, respectively. Note that $\zeta^{\min}$ and $\zeta^{\max}$ correspond to the left- and rightmost points in the curve. We transform the observed attractiveness penalties similarly. The result is shown in Figure 7. The horizontal axis shows the relative decrease of the fairness penalty and the vertical axis shows the relative increase of the attractiveness penalty.

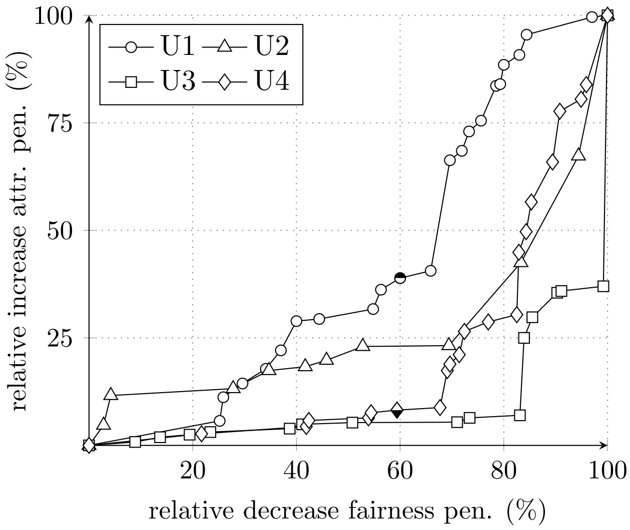

Figure 7: Relative Trade-Off Fairness and Attractiveness.

The transformed curves clearly show an increasing rate. We see, for example, that 60% of the fairness gap, i.e., the difference between the minimum and maximum, can be closed against a relative small increase of the attractiveness penalty. This 60% decrease leads to a relative increase of the attractiveness penalty of at most 38% (for instance U1) and sometimes even of as little as 8% (for instance U4). These points are highlighted in Figure

7. Similarly, we see that for almost all instances the first 25% can be closed at almost no additional increase in attractiveness penalty. On the other hand, Figure 7 also shows that the final improvements of the fairness penalty often come at a disproportionally large increase of the attractiveness penalty.

6.2.4 Attractiveness Per Group

Using an integrated approach allows to analyze the change in attractiveness per group for the different levels of fairness. This implies we can analyze the trade-off between fairness and attractiveness on a *local* level, i.e., for each group separately, instead of only on a *global* level, i.e., seeing the three groups as a whole. We compute for each group the *average* incurred attractiveness penalty per week. The groups are classified based on the types of duties they contain (recall the classification discussed in Section 6.1). The result is shown in Figure 8.

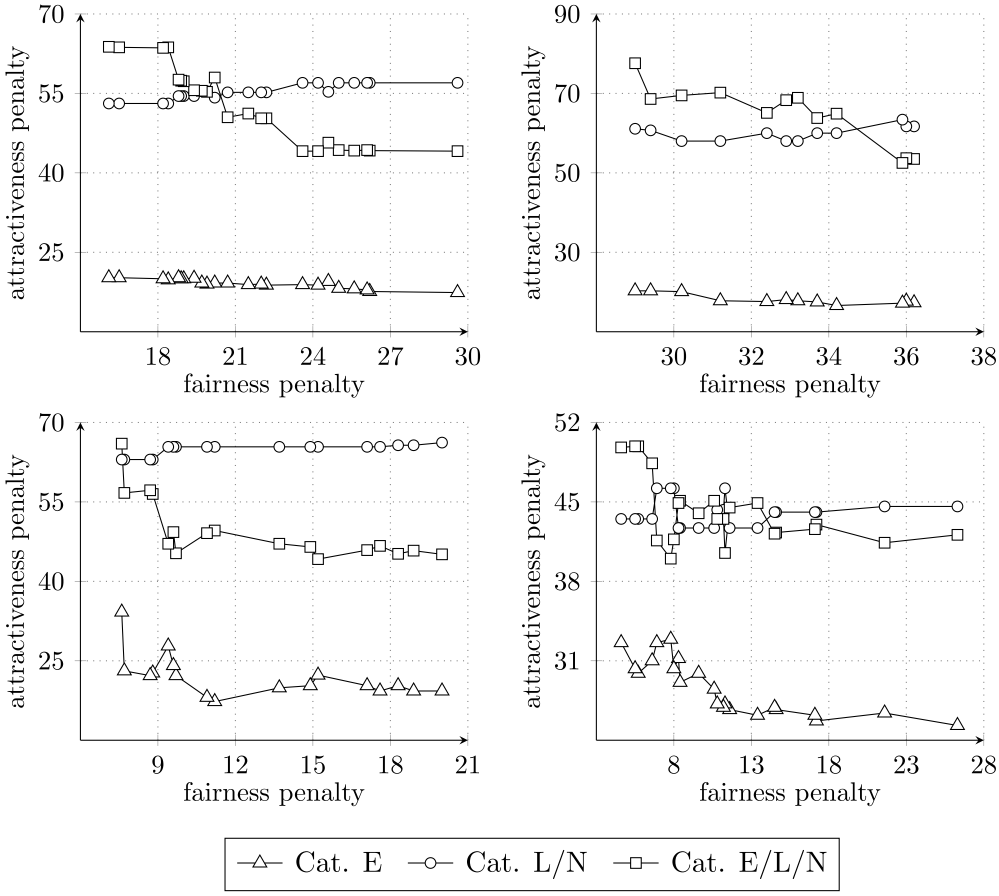

Figure 8: Average attractiveness penalty per group for instances U1 (top left), U2 (top right), U3 (bottom left) and U4 (bottom right).

Figure 8 shows that solutions with a low fairness penalty might be perceived as 'skewed' with respect to the attractiveness penalty. When the attractiveness penalty is averaged and decomposed per group, we see that the different types of group have, in general, a different attractiveness. This can be attributed to the different duty types defining each group (e.g., rosters alternating between late and night duties often have more rest time violations). When considering solutions with a low fairness penalty, we see a clear increase in the average attractiveness penalty for groups of the third category, and, for the equally sized instances, also for the groups of the first category. On the other hand, we see that for the groups of the second category, the average attractiveness penalty hardly increases, or even decreases. Especially for groups of the third category, those most flexible with respect to the duty allocation, we observe a disproportionally large increase.

6.3 Managerial Insights

Our experiments indicate the necessity of a simultaneous optimization of fairness and attractiveness. The sequential approach is inadequate for analyzing the trade-off between fairness and attractiveness. By using an integrated approach, we could analyze this trade-off in detail. The analysis in Section 6.2 led to the following important insights.

Firstly, decision makers should be careful not to 'over-optimize' fairness. The trade-off analysis of Section 6.2.3 showed that tight fairness levels lead to a rapid decrease of the attractiveness of the rosters. In our experiments, we observed cases where an almost negligible increase in fairness led to a major decrease in attractiveness. Hence, fully prioritizing fairness over attractiveness should be considered only when fairness is extremely highly valued.

Secondly, it is questionable whether a set of rosters with high perceived fairness will be perceived as desirable by all employees. Our analysis in Section 6.2.4 showed that the decrease in attractiveness is unevenly distributed over the different roster groups. That is, for some groups the attractiveness of the roster deteriorates rapidly, while for other groups the attractiveness of the roster might actually improve. This skewed distribution of attractiveness is an implicit consequence of minimizing perceived fairness. The decrease of attractiveness occurs most rapidly for those roster groups that are willing to accept an irregular roster (i.e., those groups flexible with respect to the work that can be assigned). A rapid decrease in attractiveness for these roster groups is highly undesirable from a practical point of view, as the employees in these groups already 'accept' a loss of attractiveness by working an irregular roster.

Our experiments show that the current practice should be revised. In particular, the trade-off between fairness and attractiveness should be incorporated in the construction of the rosters. Furthermore, the current metric for perceived fairness, as discussed in Section 2, should be reconsidered. For example, the skewed distribution of attractiveness could be incorporated in a new fairness metric. Such adjustments would have a great impact in practice, as fairness plays a major role in the negotiations with the labor unions.

7 Conclusion

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

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

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

Based on our findings we recommend to revise the current approach. The trade-off between fairness and attractiveness should be incorporated in the construction of the rosters. This will give a more comprehensive image of the problem. Furthermore, additional properties (e.g., the distribution of attractiveness over the groups) should be taken into account when determining the fairness level of a solution.

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

A Valid Inequalities

The linear relaxation of (1)–(12) can be tightened by adding valid inequalities. We consider the set of roster constraints $P \setminus P_K$. Recall from Section 6 that in our experiments the set $P \setminus P_K$ contains the rest constraints (both the strict version regarding the 12/14 hour rest period and the penalized version regarding the 16 hour rest period) and rest day constraints involving more than one week. We derive a set of valid inequalities to bound the penalty incurred from the rest constraints regarding the 16 hour rest period. For each $Q \subseteq R$ let $P_Q \subseteq P \setminus P_K$ be the set of 16 hour rest constraints related to the set of basic schedules Q. We derive a lower bound for the incurred penalty by minimizing

$$\sum_{p \in P_Q} c_p \delta_p, \tag{14}$$

subject to (3)–(4) and

$$\sum_{k \in K} \sum_{s \in S_k} \sum_{(t,d) \in s} f_{td}^p x_s^k \leq b_p + \delta_p \tag{15}$$

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

The above optimization problem can be solved 'efficiently' for practical instances using a commercial solver. In our experiments, for example, the valid inequalities could be obtained in only a fraction of the time needed to obtain the Pareto-optimal curves, and they strengthened the formulation greatly: For all but one instance the penalty incurred in the root node solution was equal to the actual penalty incurred in the found optimal integer solution. We note that the above procedure leads to $2^{|R|} - 1$ valid inequalities, and hence should only be applied when the number of basic schedules is relatively small. In case of a large set of basic schedules, a more economical alternative is to derive the above bound only for all basic schedules simultaneously (i.e., $Q = R$). Although this can lead to a slightly weaker formulation, we observed that this valid inequality still strengthened the formulation greatly.

B Modeling Reduced Cost

In this section we show how to model the reduced cost of a sequence. We first introduce the necessary notation. Let μ_k denote the dual variables corresponding to (3), η_d those corresponding to (4) and σ_p those corresponding to (5). Furthermore, let γ_{ar} and θ_{ar} denote the dual variables corresponding to (6) and (7), respectively. The reduced cost $\bar{c}_s^k$

of sequence $s \in S_k$, with $k \in K_r$, is given by

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

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

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

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

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

C Lagrangian Distance Labels

As discussed in Section 5.2, the pricing problem can be modeled as a RCSPP with surplus variables. Consider a RCSPP with surplus variables on the graph $G = (V, E)$. Let R denote the set of resources and let w_{ij}^r be the consumption of resource r over arc ij. Each resource has a threshold b_r, and surplus variable δ_r restricted to a closed interval Δ_r. The cost of exceeding b_r is given by q_r. Let V_i^+ denote all vertices j for which $ij \in E$, and define V_i^- analogously. Let y_{ij} denote the flow over arc ij. The problem can be formulated as follows.

$$\begin{aligned}
\min \quad & \sum_{ij\in E} c_{ij} y_{ij} + \sum_{r\in R} q_r \delta_r && && (16)\\
\text{s.t.} \quad & \sum_{ij\in E} w_{ij}^r y_{ij} \le b_r + \delta_r && \forall r \in R && (17)\\
& \sum_{j\in V_i^+} y_{ij} - \sum_{j\in V_i^-} y_{ji} = 0 && \forall i \in V \setminus \{s,t\} && (18)\\
& \sum_{j\in V_s^+} y_{sj} = 1 && && (19)\\
& \sum_{j\in V_t^-} y_{jt} = 1 && && (20)\\
& y_{ij} \in \mathbb{B} && \forall ij \in E && (21)\\
& \delta_r \in \Delta_r && \forall r \in R. && (22)
\end{aligned}$$

The objective (16) is to minimize the path cost plus the cost of the surplus variables. Constraints (17) model the resource consumption. Constraints (18)-(20) describe the shortest path polytope (see Ahuja et al. [1993]). Constraints (21) and (22) specify the domain of the decision variables.

Constraints (17) are relaxed in a Lagrangian fashion, giving rise to the Lagrangian dual problem $\max_{\lambda \geq 0} \Theta(\lambda)$, where

$$\begin{aligned}\Theta(\lambda) = \min \quad & \sum_{ij \in E} \left(c_{ij} + \sum_{r \in R} \lambda_r w_{ij}^r \right) y_{ij} + \Omega(\lambda) \\ \text{s.t.} \quad & (18) - (21),\end{aligned}$$

and

$$\Omega(\lambda) = \min \left\{ \sum_{r \in R} (q_r - \lambda_r)\, \delta_r - \sum_{r \in R} \lambda_r b_r : \delta_r \in \Delta_r \right\}.$$

Note that the Lagrangian subproblem has the integrality property. Let $\theta_i(\lambda)$ denote the shortest $i - t$ path length with respect to the cost coefficients $c_{ij} + \sum_{r \in R} \lambda_r w_{ij}^r$. The value $\theta_i(\lambda)$ can be obtained by solving an all-to-one shortest path problem. The following lemma (see e.g., Borndörfer et al. [2015]) shows the main idea behind the approach.

Lemma C.1. *Let $\lambda \geq 0$ and let p be an $s - i$ path in G. The cost of any $s - t$ path containing p is at least*

$$\sum_{jk \in p} \left(c_{jk} + \sum_{r \in R} \lambda_r w_{jk}^r \right) + \theta_i(\lambda) + \Omega(\lambda). \tag{23}$$

Proof. Note that (23) expresses exactly the objective value for the Lagrangian subproblem, conditional on the solution containing p. As this is a lower bound on the actual path cost for any set of admissible dual multipliers, the result follows. □

The values $\theta_i(\lambda) + \Omega(\lambda)$ are also known as *Lagrangian distance labels*, and can be interpreted as conservative cost estimates of the remaining path cost. To obtain the tightest bound on the $s - t$ path length the optimal dual multipliers of (16)–(22) are used to compute the distance labels.

References

E. Abbink, M. Fischetti, L. Kroon, G. Timmer, and M. Vromans. Reinventing crew scheduling at Netherlands Railways. *Interfaces*, 35(5):393–401, 2005.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

J. E. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

R. Borndörfer, M. Reuther, T. Schlechte, C. Schulz, E. Swarat, and S. Weider. Duty rostering in public transport-facing preferences, fairness, and fatigue. In *CASPT*, 2015.

E. K. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Van Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7(6):441–499, 2004.

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

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

J. A. Colquitt, D. E. Conlon, M. J. Wesson, C. O. Porter, and K. Yee Ng. Justice at the millennium: A meta-analytic review of 25 years of organizational justice research. *Journal of Applied Psychology*, 86(3):425–445, 2001.

J. A. Colquitt, B. A. Scott, J. B. Rodell, D. M. Long, C. P. Zapata, D. E. Conlon, and M. J. Wesson. Justice at the millennium, a decade later: a meta-analytic test of social exchange and affect-based perspectives. *Journal of Applied Psychology*, 98(2):199–236, 2013.

G. Desaulniers, J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Daily aircraft routing and scheduling. *Management Science*, 43(6):841–855, 1997.

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

J. Desrosiers and M. E. Lübbecke. Branch-price-and-cut algorithms. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

I. Dumitrescu and N. Boland. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42(3):135–153, 2003.

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

A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004.

R. Freling, R. M. Lentink, and A. P. M. Wagelmans. A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm. *Annals of Operations Research*, 127(1-4):203–222, 2004.

J. Greenberg. Organizational justice: Yesterday, today, and tomorrow. *Journal of Management*, 16(2):399–432, 1990.

M. Grötschel, R. Borndörfer, and A. Löbel. Duty scheduling in public transit. In *Mathematics-Key Technology for the Future*, pages 653–674. Springer, 2003.

P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2): 100–107, 1968.

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.

S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In *Column Generation*, pages 33–65. Springer, 2005.

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

S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

A. Löbel. Vehicle scheduling in public transit and lagrangean pricing. *Management Science*, 44(12-part-1):1637–1649, 1998.

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

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

B. Maenhout and M. Vanhoucke. A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research*, 206(1): 155–167, 2010.

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.

T. Nishi, T. Sugiyama, and M. Inuiguchi. Two-level decomposition algorithm for crew rostering problems with fair working condition. *European Journal of Operational Research*, 237(2):465–473, 2014.

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

L. Rhoades and R. Eisenberger. Perceived organizational support: A review of the literature. *Journal of Applied Psychology*, 87(4):698–714, 2002.

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.

M. Stojković and F. Soumis. An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, 47(9):1290–1305, 2001.

J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

L. Xie and L. Suhl. Cyclic and non-cyclic crew rostering problems in public bus transit. *OR Spectrum*, 37(1):99–136, 2015.