Computers & Operations Research 154 (2023) 106186

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

A three-phase heuristic for the Fairness-Oriented Crew Rostering Problem

Thomas Breugem [a,*], Thomas Schlechte [b], Christof Schulz [b], Ralf Borndörfer [c,d]

[a] School of Economics and Management, Tilburg University, Warandelaan 2, 5037AB, Tilburg, The Netherlands
[b] LBW Optimization GmbH, Englerallee 19, 14195, Berlin, Germany
[c] Zuse Institute Berlin, Takustr. 7, 14195, Berlin, Germany
[d] Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany

ARTICLE INFO

Keywords:
Crew planning
Fairness
Column generation
Variable-depth neighborhood search

ABSTRACT

The Fairness-Oriented Crew Rostering Problem (FCRP) considers the joint optimization of attractiveness and fairness in cyclic crew rostering. Like many problems in scheduling and logistics, the combinatorial complexity of cyclic rostering causes exact methods to fail for large-scale practical instances. In case of the FCRP, this is accentuated by the additionally imposed fairness requirements. Hence, heuristic methods are necessary. We present a three-phase heuristic for the FCRP combining column generation techniques with variable-depth neighborhood search. The heuristic exploits different mathematical formulations to find feasible solutions and to search for improvements. We apply our methodology to practical instances from Netherlands Railways (NS), the main passenger railway operator in the Netherlands Our results show the three-phase heuristic finds good solutions for most instances and outperforms a state-of-the-art commercial solver.

1. Introduction

Scheduling of personnel is a challenging problem in many sectors, especially when employees work irregular rosters (e.g., in aviation, health care, public transportation, security, see Ernst et al. (2004) and Van den Bergh et al. (2013) for overviews). It is the employer's responsibility to balance multiple objectives (Ernst et al., 2004), among which the work-life balance (e.g., sufficient rest times and days off, limited workload) of each employee's roster, also defined as the attractiveness of the roster (Abbink, 2014). Depending on the application, additional aspects can be part of attractiveness. For example, at Netherlands Railways (NS), the main passenger railway operator in the Netherlands, it is important to ensure sufficient variation (type of work, location, etc.) in the assigned work (see, e.g., Abbink et al., 2005). Besides attractiveness, one should ensure 'undesirable work' (e.g., more demanding shifts or tasks) is balanced equally over the employees. This is known as the fairness of the rosters. We refer to Abbink (2014) for a detailed discussion on attractiveness and fairness in crew planning at NS.

Attractiveness and fairness are not mere 'nice-to-haves'. Both can have a real impact on operations. For example, when NS presented their new work schedules to their employees in 2001 "the drivers and conductors were quite unhappy [... and] went on strike for several days in an attempt to prevent their introduction". The introduction of the 'Sharing-Sweet-and-Sour' rules ('Lusten-en-Lasten-Delen' in Dutch), scheduling rules for increased attractiveness and fairness of work, was instrumental in resolving this conflict (see Abbink et al., 2005). Another example is given in Borndörfer et al. (2017b). The authors argue the importance of attractive work at BVG (Berlin's public transport company). Bus drivers at BVG have an average age of around 50 years. Attractiveness is a key instrument for recruiting new (younger) drivers, as the possibility for higher salaries is generally limited.

The Fairness-Oriented Crew Rostering Problem (FCRP) considers the joint optimization of attractiveness and fairness in cyclic crew rostering. The problem, introduced in Breugem et al. (2021a), is motivated by crew rostering practices at NS. Like many problems in scheduling and logistics, the combinatorial complexity of cyclic rostering causes exact methods to fail for large-scale practical instances. In case of the FCRP, this is worsened by the additionally imposed fairness requirements. Short computation times are desirable, because in practice crew rosters are often constructed iteratively, with decision makers tweaking parameter settings in each iteration. Hence, heuristic methods for the FCRP are necessary to deal with large-scale practical instances.

The main contributions of this article are twofold. Firstly, we present a scalable method for the FCRP. Our proposed three-phase heuristic combines column generation techniques with variable-depth neighborhood search (VDNS), and exploits different formulations for the FCRP to effectively find feasible solutions and search for improvements. This method bridges the gap between the exact method

* Corresponding author.
E-mail addresses: t.breugem@tilburguniversity.edu (T. Breugem), Schlechte@lbw-optimization.de (T. Schlechte), Schulz@lbw-optimization.de (C. Schulz), Borndoerfer@zib.de (R. Borndörfer).

https://doi.org/10.1016/j.cor.2023.106186
Received 3 November 2021; Received in revised form 1 February 2023; Accepted 6 February 2023
Available online 24 February 2023
0305-0548/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

of Breugem et al. (2021a) and commercial solvers, which are computationally not feasible for large practical instances, and the current (suboptimal) sequential practice for large instances, where first duties are assigned to roster groups, and then rosters are optimized per group. Our method fundamentally differs from the one presented in Breugem et al. (2021a) by focusing on the development of new heuristic methods, based on column generation and VDNS, to iteratively improve allocations across subsets of roster groups, rather than finding a globally optimal solution. Secondly, we apply our methodology to practical instances from NS, the main passenger railway operator in the Netherlands, and show that the three-phase heuristic finds good solutions for most instances and outperforms a state-of-the-art commercial solver. Our computational experiments consider instances up to six groups and almost 250 duties. This is approximately the size of the crew base where NS conducted its latest pilot study regarding decision support for crew rostering (early 2018).

The remainder of this article is organized as follows. In Section 2 we provide the necessary background on cyclic crew rostering and present the FCRP, and in Section 3, we give an overview of related work. In Section 4 we discuss the core mathematical formulation for the heuristics, followed by a detailed description of the three-phase heuristic in Section 5. Section 6 evaluates the performance of the heuristic on practical instances from NS, and the article is concluded in Section 7.

2. Problem description

This section introduces the different components underlying the FCRP in more detail. In Section 2.1, we present the necessary background on cyclic crew rostering. Section 2.2 discusses attractiveness and fairness.

2.1. Cyclic crew rostering

Cyclic rosters specify an allocation of duties (days of work) and days off to the employees. The rosters are constructed for a fixed time period, and the work for this period is assumed to be cyclic. For example, the work scheduled on Monday is the same for every week in the year (note that other cycle periods are also possible, see, e.g., Mesquita et al., 2013). This type of rosters is typical for public transportation, where the timetable typically does not change over the year (excluding special occasions such as national holidays etc.).

A cyclic roster consists of cells (containing duties or days-off), grouped into rows (weeks of work) and columns (generic weekdays). Hence, every cell represents a unique row and column combination to which a duty or day-off must be assigned. Fig. 1 shows an example of two rosters, one for group A, consisting of three rows, and one for group B, consisting of four rows. The first three cells in the first row of the roster for group B are assigned duty 112, a day-off (indicated by no number), and duty 142 (the number being the unique identifier of the duty).

Each roster has an underlying structure, known as the basic schedule. The basic schedule specifies the type of work of each cell (e.g., a late duty or day-off). The basic schedule for both rosters in Fig. 1 is shown at the top left of the cells, and the numbers indicate the assigned duties. For example, the basic schedule of group B specifies that the first row starts with a late duty (L), followed by a day-off (R), and then again a late duty. In Fig. 1, the assigned duties 112 and 142 are both late duties.

The roster is executed by a group of employees (roster group), equal in size to the number of rows in the roster. Every week, each employee of the roster group executes one row of the roster, and continues with the next row the following week (e.g., first row three, then row four, and so forth). In other words, they 'cycle' through the roster, with the length of one cycle being proportional to the number of rows. This way, all rows are executed each week and every employee in the group does the exact same work, only at different moments in time (provided the time period is a multitude of the cycle length). Fig. 2 illustrates this by showing the schedule for the first two employees in group A over a two week period, assuming the first employee starts the first week with the first row.

2.2. Fairness and attractiveness

Fairness relates to the distribution of work among the roster groups. Since every duty represents a work day, some duties can be considered more desirable than others. This is captured by duty attributes. In this research, we consider five duty attributes when measuring fairness: the percentage of *high quality work* (e.g., Intercity work), the percentage of *aggression work* (e.g., trips where passengers are less likely to have a ticket, and hence might become aggressive), the percentage of *double decker work* (work on double decker trains is physically more demanding), *duty length*, and *repetition within duty* (duties with many of the same trips are repetitive, which is considered undesirable). Fairness is expressed as a weighted sum of the maximum and minimum average values of these attributes among the roster groups. Furthermore, there is a hard lower and upper bound on the average for each group. This is in line with the 'Sharing-Sweet-and-Sour' rules used by NS. In Section 4.2, we provide a mathematical definition of fairness.

Attractiveness relates to the structure of the rosters. It is modeled via roster constraints. Each roster constraint penalizes, or forbids, certain combinations of assignments of duties to the cells of the basic schedules. Attractiveness is determined by four types of roster constraints. These types can be divided into two classes.

1. The first class contains 'binary' roster constraints, linking exactly two cells in the basic schedule. *Rest time* constraints require that an employee has a minimum rest time after each duty. After a night duty this rest time should be 14 h and after any other type of duty it should be 12 h. Rest times shorter than 16 h are penalized but allowed. *Rest day* constraints ensure the length of each scheduled rest period is sufficient. Between any two duties separated by a series of rest days, there must be 6 h plus 24 h for each rest day in between.
2. The second class of roster constraints consists of 'row-based' roster constraints. This class contains *workload* constraints, i.e., the total workload in a row is not allowed to exceed 45 h, and a large collection of *variation* constraints. This latter type of constraint aims at balancing different duty attributes (e.g., duty length, percentage double decker work) equally over the rows, which is achieved by penalizing positive deviations from the average (measured over all duties) for each row in the roster.

In Section 4.3, we provide a mathematical framework for modeling roster constraints. Summarizing, fairness is modeled via hard constraints (lower and upper bounds per group) and soft constraints captured by a suitable metric (the weighted sum of maximum and minimum average values). Similarly, attractiveness is modeled via hard constraints (rest time, rest day, and workload constraints) and soft constraints expressed via penalties (variation constraints and rest times shorter than 16 h). Hence, there is a trade-off between attractiveness and fairness in terms of soft constraints. The FCRP can now be stated as follows: Given as input the basic schedules and duties, determine rosters that are both attractive and fair. The solution to the FCRP is a collection of sets of rosters that represent a trade-off curve between attractiveness and fairness.

3. Related work

Crew planning is widely studied in literature, dating back as far as Dantzig (1954). Applications range from health care (e.g., nurse rostering) to transportation (e.g., airline, railway, and bus planning).

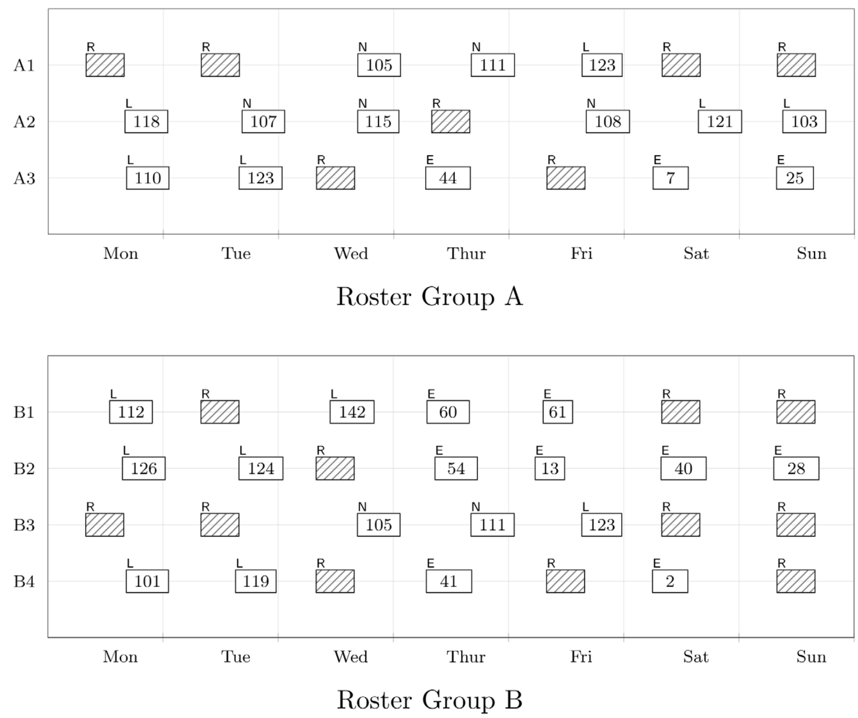


Fig. 1. Example of two rosters for roster group A, consisting of three employees, and B, consisting of four employees. The basic schedule for each group is indicated by the types Early (E), Late (L), Night (N), and Rest (R) above the cells, and the numbers indicate the assigned duties. The numbers are the unique identifiers of scheduled duties. The horizontal axis shows the generic weekdays (columns) and the vertical axis the rows of the employees. Note that the positioning of a duty cell within a day depends on the start time of the duty.

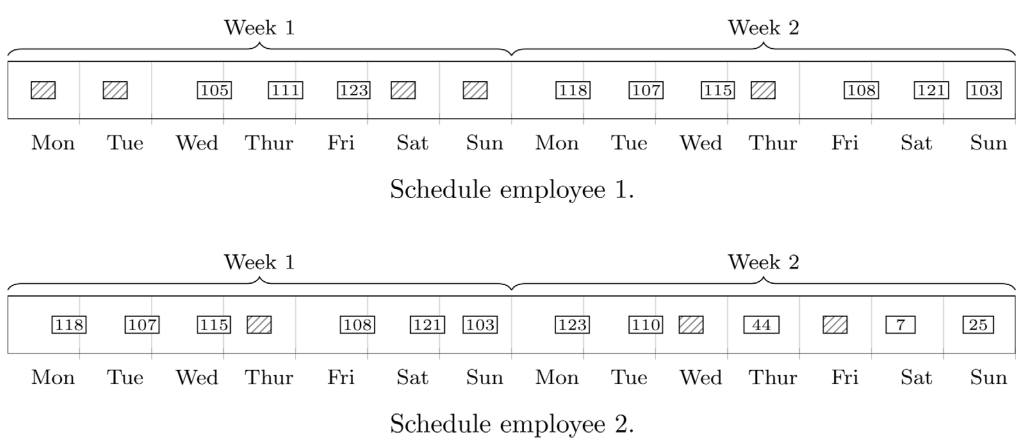


Fig. 2. Roll-out of the cyclic roster shown in Fig. 1 for the first two employees of group A over a two week period, assuming the first employee starts with the first row. The numbers are the unique identifiers of scheduled duties. The horizontal axis shows the weekdays for each week.

We refer to Ernst et al. (2004) and Van den Bergh et al. (2013) for detailed reviews of crew planning literature. In this section, we focus on crew planning within the transportation sector. We refer to Borndörfer et al. (2018) for an overview of optimization techniques in rail transportation.

Crew planning is commonly decomposed into two sequential planning phases: scheduling/pairing and rostering. In crew scheduling and crew pairing, days of work (i.e., duties or pairings) are constructed. This phase determines operational cost (i.e., the number of necessary crew members), and incorporates key factors, such as fairness and rules specified in the collective labor agreement. Crew rostering consists of combining duties (pairings) into rosters, which are sequences of duties (pairings) satisfying numerous labor constraints. Typical constraints consider days off, rest times, variation of work, or personal preferences.

The crew scheduling problem is well-studied (see, for example, Desrochers and Soumis, 1989; Hoffman and Padberg, 1993; Grötschel et al., 2003; Abbink et al., 2005, among others), and has been considered in numerous variants, such as rescheduling whenever the underlying tasks are modified (e.g., Lettovskỳ et al., 2000; Huisman, 2007; Potthoff et al., 2010; Breugem et al., 2021b) and in conjunction with other planning problems, such as aircraft routing and vehicle scheduling (e.g., Cordeau et al., 2001; Huisman et al., 2005a). Perumal et al. (2019, 2021b) consider the simultaneous scheduling of drivers and staff cars, proposing solution methods based on adaptive large neighborhood search (ALNS). Quesnel et al. (2020) propose a new variant of the airline crew planning problem, the crew planning problem with complex features, that considers crew preferences in order to create pairings that are better suited for crew rostering in the later stage. Jütte et al. (2017) consider the integration of fairness aspects in crew scheduling in railway, taking equal distribution of unpopular duties over employees into account. Perumal et al. (2022) provide an overview of problems and methodology in electric bus planning and scheduling, including approaches to crew planning (both scheduling and rostering) in combination with vehicle scheduling. Perumal et al. (2021a) provide a solution methodology for the latter based on ALNS combining break and repair operations with branch-and-price.

In crew rostering, rosters are generally classified as cyclic, i.e., multiple employees working the same roster, and acyclic, i.e., each individual employee working his or her own roster. The latter type is common in the healthcare sector and airline industry (see, for example, Kohl and Karisch (2004) and De Causmaecker and Vanden Berghe (2011), and references therein), whereas the former type is often used in railway operations and mass transit (see Huisman et al., 2005b; Caprara et al., 2007; Abbink et al., 2018; Heil et al., 2020).

Cyclic crew rostering is well-studied in the literature and a variety of formulations have been proposed to model the problem, including a generalized assignment formulation (e.g., Hartog et al., 2009), a multi-commodity flow formulation (e.g., Caprara et al., 1997; Xie and Suhl, 2015; Borndörfer et al., 2015), and a set covering or set partitioning formulation (e.g., Caprara et al., 1997; Freling et al., 2004; Borndörfer et al., 2013, 2015). Er-Rbib et al. (2021) consider the duty assignment problem with group-based driver preferences aimed at building rosters that cover all the duties over a predetermined cyclic horizon while respecting a set of rules (hard constraints), balancing the workload between the drivers and satisfying as much as possible the driver preferences (soft constraints). The authors present different solution methods, including first assigning duties based on the duty assignment problem (where workload is balanced) and then optimizing the rosters, and a partitioning approach where duties are allocated to subsets of the rosters.

Different metaheuristic approaches have been successfully applied to crew rostering, especially when multiple objectives need to be considered. For example, Lourenço et al. (2001) propose tabu search and genetic algorithms for bus driver scheduling to cope with the multiple objectives that need to be considered in practice. The authors discuss how the solutions found by the developed algorithms were preferred by practitioners over the baseline algorithm based on linear programming techniques. Recent work especially relevant to fairness include (Zhou et al., 2020), who develop an ant colony system algorithm to solve the airline crew rostering problem taking both fairness and satisfaction into account, and Chutima and Arayikanon (2020), who develop a hybrid evolutionary algorithm for airline cockpit crew rostering while accounting for four different objectives, some closely related to fairness. Lučić and Teodorović (2007) and Xie et al. (2017) discuss and compare different metaheuristic approaches (i.e., simulated annealing, genetic algorithms, tabu search, and ant colony optimization) for crew rostering in the airline and public transport context, respectively.

The integration of crew scheduling and rostering has been considered in Mesquita et al. (2013) and Borndörfer et al. (2017b), which both propose a solution method based on Benders decomposition. Doi et al. (2018) propose a decomposition method for the airline crew rostering problem with the objective to minimize the deviation in working time, while a number of hard constraints have to be taken into account. The authors propose a two-level decomposition approach where the master problem considers assigning pairings and rest days, and the subproblem ensures the feasibility of the original problem. Recently, Ge et al. (2022) revisited the model of Mesquita et al. (2013) to analyze whether state-of-the-art commercial solvers are nowadays able to solve complex instances for which previously heuristic methods were necessary. The authors show solvability has indeed increased and more complex features can now be added.

The incorporation of fairness measures in combinatorial optimization problems has received considerable attention in recent years (see Karsu and Morton (2015) for an overview). Focus is typically on the trade-off between efficiency (e.g., minimizing cost) and fairness (Karsu and Morton, 2015). Recent work includes, among others, work in public health and humanitarian operations (e.g., Hooker and Williams, 2012; McCoy and Lee, 2014; Eisenhandler and Tzur, 2019), air traffic flow management (e.g., Bertsimas et al., 2012; Bertsimas and Gupta, 2015), organ allocation (Bertsimas et al., 2013; Dickerson et al., 2019). Closely related work includes work in crew scheduling and crew rostering (Nishi et al., 2014; Borndörfer et al., 2015, 2017a; Jütte et al., 2017; Breugem et al., 2021a). Wolbeck (2019) provides a comprehensive overview of fairness in personnel scheduling.

Summarizing, there exists substantial literature on heuristics for crew rostering and on fairness in optimization. This article adds to both streams of literature by presenting a novel method for jointly optimizing fairness and attractiveness in crew rostering. Extending the work in Breugem et al. (2021a) to develop new heuristic methods, we present a solution method for the FCRP that can be easily scaled and is able to find high-quality solutions to large practical instances of size not yet considered in the literature.

4. Mathematical formulation

In this section we present a mathematical formulation for the FCRP based on the family of formulations introduced in Breugem (2020). The formulation first partitions each basic schedule into disjoint subsets called *clusters*. This partition is called the *clustering* for the basic schedule. Natural choices for clusters are cells (the assignment model of Hartog et al. (2009)) or rows (the row-based formulation of Breugem et al. (2021a)). Each cluster is assigned a number of duties simultaneously. This assignment is called a *roster sequence*.

The clustering allows to trade-off the number of variables (roster sequences) and the strength of the formulation. For example, a clustering based on cells has relatively few variables but also a relatively weak linear relaxation, whereas a clustering based on rows has many variables but also a stronger linear relaxation (because all weekly roster constraints, such as weekly variation and maximum workload constraints, can be modeled implicitly in the cost of the variables). For a detailed analysis of this, we refer to Breugem (2020). The three-phase heuristic relies on both the cell and row clusterings.

The remainder of this section is organized as follows. In Section 4.1, we formalize notation and terminology. Sections 4.2 and 4.3 discuss how fairness, respectively attractiveness, are modeled. We conclude in Section 4.4, by presenting the mathematical formulation.

4.1. Notation and terminology

Let D denote the set of duties, and let R denote the set of basic schedules. Each basic schedule r is defined by a set of cells T_r, and the set of all cells is denoted by T. An assignment of a duty d to a cell t in a basic schedule is represented by the pair (t, d). We define n_r as the total number of duties to be assigned to basic schedule r.

Let K denote the set of all clusters, and K_r denote the set of clusters for basic schedule $r \in R$. We define S_k as the set of all roster sequences for cluster $k \in K$, where each roster sequence is formally defined as a sequence of assignments (t, d) for the cells in k. For example, a roster sequence for row A1 in Fig. 1 specifies a duty for all three cells (recall that rest days are assumed fixed). In this specific case, the selected roster sequence consists of the assignment of duty 105 to Wednesday, 111 to Thursday, and 123 to Friday. The parameter h^k_{ds} indicates whether roster sequence $s \in S_k$ contains duty d.

The assignment of duties to the cells is modeled using binary decision variables x^k_s, for all $k \in K$ and $s \in S_k$, indicating whether roster sequence $s \in S_k$ is assigned to cluster k. Fairness and attractiveness are modeled as follows.

4.2. Modeling fairness

Let A denote the set of duty attributes, and let g_{ad} denote the value of attribute $a \in A$ for duty $d \in D$. Each attribute has a specified lower bound ℓ_a and upper bound u_a, representing the minimum and maximum allowed average values for a basic schedule. In case of duty length, for example, one could enforce that no roster group works more than 8 h on average. These bounds are based on labor agreements and considered input to the problem. Besides these bounds, each duty attribute has an associated weight w_a, representing the relative importance of the different duty attributes when calculating fairness.

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

$$\sum_{k\in K_r}\sum_{s\in S_k}\sum_{(t,d)\in s} g_{ad}x_s^k \le n_r z_a \qquad \forall a \in A, r \in R \tag{1a}$$

$$\sum_{k\in K_r}\sum_{s\in S_k}\sum_{(t,d)\in s} g_{ad}x_s^k \ge n_r v_a \qquad \forall a \in A, r \in R, \tag{1b}$$

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

$$\sum_{a\in A} w_a(z_a - v_a), \tag{2}$$

and a fairness budget ζ is enforced by assuring that (2) does not exceed ζ.

4.3. Modeling attractiveness

Let Q denote the set of roster constraints. Each roster constraint q is modeled using a set of linear constraints $p \in P_q$. Let f_{td}^{pq} denote the coefficient for assigning duty d to cell t for roster constraint $p \in P_q$.

For any given assignment of duties to the cells, the violation of the roster constraint is given by the sum of these coefficients minus some allowed threshold value b_{pq}, and this violation is restricted to the interval $\Delta_q = [0, m_q]$. The interval $\Delta_q = [0, 1]$, for example, would allow a violation of at most one. Each roster constraint $p \in P_q$ can be written as

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{(t,d)\in s} f_{td}^{pq}x_s^k \le b_{pq} + \delta_q, \tag{3}$$

where $\delta_q \in \Delta_q$ represents the violation. The attractiveness is maximized by minimizing the sum of roster constraint violations $c_q\delta_q$, where the cost coefficients c_q regulate the relative importance of the different roster constraints. The formulation allows every roster constraint that is *contained* in a cluster, i.e., it only has non-zero coefficients f_{td}^{pq} for the cells in one cluster, to be modeled implicitly. Let $Q_K \subseteq Q$ denote the set of roster constraints that are contained in the clusters $k \in K$ (and can therefore be modeled implicitly), and let c_s^k denote the cost associated with sequence $s \in S_k$. That is, c_s^k is the sum of all roster constraint violations in the sequence s. All roster constraints not contained in a cluster are modeled explicitly via (3).

4.4. Cluster formulation

The concepts introduced in Sections 4.2 and 4.3 can be integrated to obtain the final mathematical formulation. For a given fairness budget ζ, the model reads as follows:

$$\min \quad \sum_{k\in K}\sum_{s\in S_k} c_s^k x_s^k + \sum_{q\in Q\setminus Q_K} c_q\delta_q \tag{4a}$$

$$\text{s.t.} \quad \sum_{a\in A} w_a\left(z_a - v_a\right) \le \zeta \tag{4b}$$

$$\sum_{s\in S_k} x_s^k = 1 \qquad \forall k \in K \tag{4c}$$

$$\sum_{k\in K}\sum_{s\in S_k} h_{ds}^k x_s^k = 1 \qquad \forall d \in D \tag{4d}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{(t,d)\in s} f_{td}^{pq}x_s^k \le b_{pq} + \delta_q \qquad \forall q \in Q \setminus Q_K, p \in P_q \tag{4e}$$

$$\sum_{k\in K_r}\sum_{s\in S_k}\sum_{(t,d)\in s} g_{ad}x_s^k \le n_r z_a \qquad \forall a \in A, r \in R \tag{4f}$$

$$\sum_{k\in K_r}\sum_{s\in S_k}\sum_{(t,d)\in s} g_{ad}x_s^k \ge n_r v_a \qquad \forall a \in A, r \in R \tag{4g}$$

$$z_a \le u_a \qquad \forall a \in A \tag{4h}$$

$$v_a \ge \ell_a \qquad \forall a \in A \tag{4i}$$

$$x_s^k \in \mathbb{B} \qquad \forall k \in K, s \in S_k \tag{4j}$$

$$\delta_q \in \Delta_q \qquad \forall q \in Q \setminus Q_K \tag{4k}$$

$$v_a, z_a \in \mathbb{R}_+ \qquad \forall a \in A. \tag{4l}$$

The objective (4a) expresses that we minimize the penalties incurred from the roster constraints, partly expressed by the roster sequence costs and partly expressed by the cost of the explicitly modeled roster constraints. The fairness budget is enforced by (4b). Constraints (4c) and (4d) assure that the duties are assigned correctly to the basic schedules: each cluster is assigned exactly one roster sequence, and each duty appears in exactly one roster sequence. Constraints (4e) model the roster constraint violations, as discussed in Section 4.3. Furthermore, Constraints (4f)–(4i) are those discussed in Section 4.2: Constraints (4f) and (4g) assure that the variables v_a and z_a are bounded by the minimum and maximum value, respectively, while (4h) and (4i) enforce the lower and upper bounds on the attribute values. Finally, Constraints (4j)–(4l) express the domains of the decision variables.

5. Methodology

In this section, we present the three-phase heuristic for the FCRP. The three phases of the heuristic are:

1. finding a fair allocation (Phase 1),
2. improving the solution via column generation (Phase 2),
3. improving the solution via VDNS (Phase 3).

Section 5.1 discusses the general framework for constructing the trade-off curve for the FCRP. We then explain the improvement phases separately: In Section 5.2 we discuss pairwise improvement based on column generation, and in Section 5.3, we discuss the VDNS approach.

5.1. General framework

Constructing a trade-off curve implies (i) exploring the range of possible fairness budgets, and (ii) determining attractive rosters for each fairness budget. We do this via the framework presented in Algorithm 1. Let an FCRP instance and ascending percentages p_i, for $i = 1, \dots, n$, be given. The percentages determine which fairness levels are part of the trade-off curve.

First, a fair solution is determined (line 1). This is Phase 1 of the heuristic. The fair solution is determined by solving (4) using the cell clustering while minimizing ζ rather than (4a). The cell clustering has only a limited number of variables and can therefore be input into a commercial solver. The solution provides an allocation of duties to roster groups. The attractiveness of each roster is then maximized separately. For practical roster group sizes, this can be done efficiently using the branch-price-and-cut approach of Breugem et al. (2021a). The fairness level of this solution is used to set the minimum fairness budget explored, denoted by $\underline{\zeta}$ (line 2).

Next, the improvements steps pairwise improvement (Phase 2) and VDNS (Phase 3), discussed in more detail in Sections 5.2 and 5.3, are

Algorithm 1 General framework

Input: FCRP instance I and ascending percentages $p_1, \dots, p_n$

Output: heuristic trade-off curve for instance I

1: compute $S_{\mathrm{F}} \leftarrow \mathrm{FAIR}(I)$ ▷ *compute fair solution (Phase 1)*

2: set $\underline{\zeta} \leftarrow \mathrm{FAIRNESS}(S_{\mathrm{F}})$

3: compute $S_{\mathrm{A}} \leftarrow \mathrm{IMPROVE}(I, S_{\mathrm{F}}, \infty)$ ▷ *compute most attractive solution*

4: set $\bar{\zeta} \leftarrow \mathrm{FAIRNESS}(S_{\mathrm{A}})$

5: set $S_0 \leftarrow S_{\mathrm{F}}$

6: **for** $i = 1, \dots, n$ **do** ▷ *iteratively construct curve (Phase 2 & 3)*

7: $\quad \zeta_i \leftarrow \underline{\zeta} + p_i \left(\bar{\zeta} - \underline{\zeta}\right)$ ▷ *set next fairness level*

8: $\quad S_i \leftarrow \mathrm{IMPROVE}(I, S_{i-1}, \zeta_i)$ ▷ *re-use previous solution*

9: **end for**

10: return solutions $S_1, \dots, S_n$

invoked on the fair solution without any fairness requirements (line 3). This provides an upper bound $\bar{\zeta}$ on the fairness levels that will be explored (line 4). The interval is then partitioned based on input percentages p_i and the curve is constructed iteratively (lines 6–9). In each iteration, the improvement steps are invoked, using the previously found solution as starting solution. Note that the heuristic determines both the range of fairness levels explored (via lines 3 and 4) and the quality (attractiveness) of the found solutions (via line 8).

5.2. Pairwise improvement

The exact approach of Breugem et al. (2021a) performs well for reasonably sized instances, but fails to find good solutions for larger instances (see also (Breugem, 2020)). The strength of the approach lies in the stronger linear relaxation obtained from using a row clustering (as discussed in Section 4). The pairwise improvement phase uses this formulation to search for improvements, while limiting the search space to subsets of groups to remain computationally tractable.

The pairwise improvement phase searches for profitable re-allocations of duties among pairs of roster groups using column generation. The pseudo-code is given in Algorithm 2. The pairs of roster groups are ordered such that the small groups are considered first (lines 1 and 2). For example, consider four roster groups A, B, C, and D, of 8, 9, 10, and 12 employees, respectively. The heuristic first looks for a profitable re-allocation between A and B, then between A and C, then A and D, then between B and C, then B and D, and finally between C and D, each time updating the rosters when a profitable re-allocation has been found (lines 3 to 14). Hence, given k groups, we check $k(k-1)/2$ pairs of roster groups.

Given a pair of roster groups, a profitable re-allocation of duties is determined between the two groups, keeping all other assignments fixed (line 5). To ensure the re-allocation is feasible with respect to the fairness budget for the global problem we set upper and lower bounds on v_a and z_a based on the minimum and maximum over *all* roster groups, excluding those considered in the re-allocation (lines 6 to 9).

The search for profitable re-allocations is done by solving the row-based formulation (formulation (4) for the row clustering) for *only* the two roster groups using a reduced set of roster sequences (lines 10 to 12). The reduction is based on the solution to the linear relaxation (line 10). Consider a given feasible integer solution $\hat{x}$ and an optimal (fractional) solution of the linear relaxation $\bar{x}$. For each cell $t \in T$, we allow only a subset $\bar{D}_t \subseteq D_t$ of duties to be assigned. The set $\bar{D}_t$ is initialized by the duty assigned to t in the solution $\hat{x}$, to assure the integer solution remains feasible, and is then enriched based on $\bar{x}$. For

Algorithm 2 Pairwise improvement

Input: FCRP instance I, initial solution S, and fairness level ζ

Output: solution for I feasible with respect to ζ

1: set $G \leftarrow \mathrm{GROUPS}(I)$ ▷ *order groups*

2: order $G \leftarrow \mathrm{ORDER}(G)$

3: **for** $i = 1, \dots, k-1$ **do** ▷ *iteratively improve allocation*

4: $\quad$ **for** $j = i+1, \dots, k$ **do**

5: $\quad\quad$ construct $I' \leftarrow \mathrm{REDUCED}(I, S, G_i, G_j)$ ▷ *reduce instance*

6: $\quad\quad$ **for** balance constraint $a \in A$ **do** ▷ *ensure feasibility*

7: $\quad\quad\quad$ Compute $\hat{v}_a \leftarrow \mathrm{MIN}(a, S, G \setminus \{G_i, G_j\})$

8: $\quad\quad\quad$ Compute $\hat{z}_a \leftarrow \mathrm{MAX}(a, S, G \setminus \{G_i, G_j\})$

9: $\quad\quad$ **end for**

10: $\quad\quad$ compute $x \leftarrow \mathrm{LP}(I', S, \hat{v}, \hat{z}, \zeta)$ ▷ *solve LP*

11: $\quad\quad$ compute $S' \leftarrow \mathrm{IMPROVE}(I', S, x, \hat{v}, \hat{z}, \zeta)$ ▷ *search for improved allocation*

12: $\quad\quad$ Update $S \leftarrow \mathrm{UPDATE}(S, S')$ ▷ *update solution*

13: $\quad$ **end for**

14: **end for**

15: return solution S

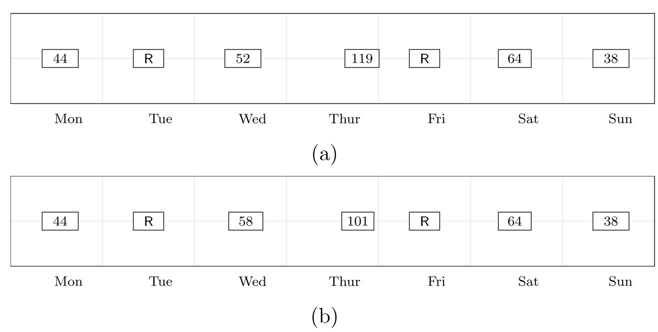


Fig. 3. An example of the problem reduction. Suppose the shown roster sequences are assigned positive values in $\bar{x}$ for some given row. The allowed duties obtained from $\bar{x}$ are given by $\bar{D}_{\mathrm{Mon}} = \{44\}$, $\bar{D}_{\mathrm{Wed}} = \{52, 58\}$, $\bar{D}_{\mathrm{Thur}} = \{101, 119\}$, $\bar{D}_{\mathrm{Sat}} = \{64\}$, and $\bar{D}_{\mathrm{Sun}} = \{38\}$. In this example, two additional roster sequences will be generated leading to a total of four roster sequences.

each $k \in K$ and $t \in k$, we enlarge $\bar{D}_t$ by adding the duties $d \in D_t$ for which

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

i.e., we add those duties to $\bar{D}_t$ that have a non-zero coefficient for t in $\bar{x}$. Fig. 3 illustrates this reduction. Generally, not too many duties are selected this way.

The reduced set of roster sequences $\bar{S}_k \subseteq S_k$ for each row $k \in K$, consists of exactly those roster sequences assigning only duties in $\bar{D}_t$ to each $t \in k$. This set is determined by complete enumeration. The resulting reduced model is solved using a commercial solver to obtain a possible improved allocation of duties (line 11). The solution is then updated (line 12) and the next pair of roster groups is considered. Note that the improvement step (line 11) is a neighborhood search method, with the neighborhood defined based on the solution to the linear relaxation. That is, we search the neighborhood of re-allocations between roster groups G_i and G_j, only allowing allocating duties $\bar{D}_t$ to cell t.

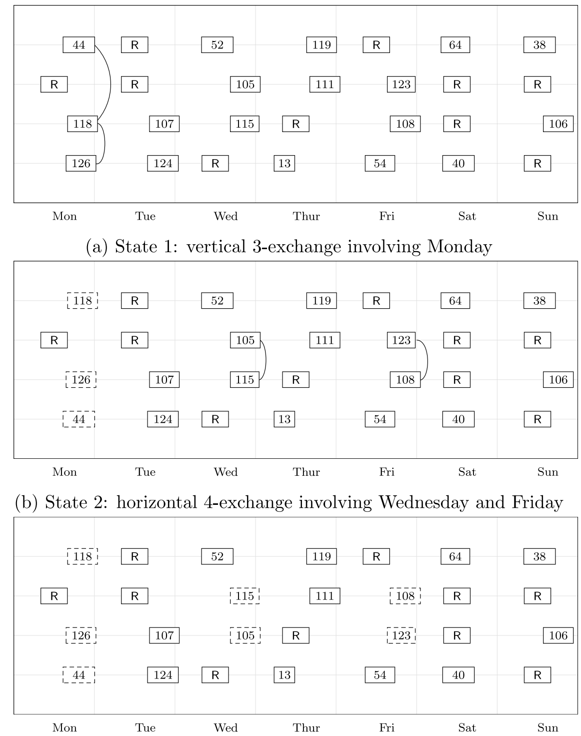
5.3. Variable-depth neighborhood search

The VDNS heuristic aims to find good solutions using local search. The idea behind VDNS is to define a neighborhood (e.g., swapping two duties), and use this neighborhood to heuristically search (much) larger neighborhoods by constructing a *chain* of moves within this neighborhood (e.g., three duty swaps leading to swapping six duties in total). In this process some deterioration of the solution quality is allowed to construct larger chains, hence searching larger neighborhoods. This way, a large part of the solution space can be searched, including moves involving a large number of elements, whilst avoiding excessive computation times. This idea was first proposed in Lin and Kernighan (1973). We refer to Ahuja et al. (2002) and Pisinger and Ropke (2010), and references therein, for discussions on VDNS and its applications.

To apply VDNS one first needs to define a parametrized search neighborhood. The neighborhood should (i) be able to escape local optima, (ii) be searchable in reasonable time, and (iii) guarantee that the solution remains feasible, or at least can be easily made feasible. The latter depends on the structure of the underlying problem. For example, the basic schedule implies duties cannot be freely exchanged (e.g., a late duty on Monday cannot be exchanged with an early duty on Monday nor a late duty on Tuesday). We consider two neighborhoods for the FCRP, based on two different duty exchange operations: *vertical* and *horizontal* duty exchanges.

Vertical k-exchanges are exchanges between k duties of the same type and in the same column, i.e., they are vertically aligned. This assures that the involved duties can always be exchanged without violating the basic schedule. Note that the feasibility with respect to the roster constraints can be readily checked when performing an exchange, hence the feasibility of the solution can always be assured. The vertical k-exchange neighborhood can be searched in $\mathcal{O}\left(|D|^k\right)$ time. Fig. 4(a) gives an example of a vertical 3-exchange. We denote the vertical k-exchange neighborhood by V_k.

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



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

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

(c) State 3: final roster

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

Algorithm 3 VDNS method

Input: FCRP instance I, initial solution S, fairness level ζ, and deterioration function ε

Output: solution for I feasible with respect to ζ

1: set $v \leftarrow v(S)$
2: set solution $T \leftarrow S$, assignments $A \leftarrow \emptyset$, improvement $\leftarrow$ false
3: **while** $T' \leftarrow$ MOVE(T, A, H_4, V_3) with $v(T') < \varepsilon(v)$ **do** ▷ $H_4 + V_3$ *move*
4: update $T \leftarrow T'$, $A \leftarrow A \cup$ FIX(T, T') ▷ *fix assignments*
5: **if** $v(T) < v(S)$ **then**
6: update $S \leftarrow T$ ▷ *update incumbent*
7: improvement $\leftarrow$ true
8: **end if**
9: **end while**
10: **if** improvement **then**
11: goto step 1
12: **end if**
13: compute $T \leftarrow$ MOVE$(S, \emptyset, H_6, V_4)$ ▷ $H_6 + V_4$ *move*
14: **if** $v(T) < v(S)$ **then**
15: update $S \leftarrow T$ ▷ *update incumbent*
16: goto step 1
17: **end if**
18: return S

The proposed VDNS algorithm combines horizontal 4- and vertical 3-exchanges with horizontal 6- and vertical 4-exchanges. The algorithm and this selection of neighborhood size is based on the Dynamic Depth-EXchange (DEX) algorithm, introduced in Borndörfer et al. (2015). A pseudo-code formulation is given in Algorithm 3. Starting from an initial solution, we construct improving $H_4 + V_3$ chains (lines 3–9). Each chain consists of a sequence of moves where after each move the involved assignments (set A, i.e., duties to cells) are fixed (line 4), meaning they are no longer allowed to be part of moves. Each time, the move that leads to the solution with lowest objective value is selected. We allow for some small deterioration in the chaining process (line 3), to add more flexibility to the search. We use a similar control function ε as proposed in Kanellakis and Papadimitriou (1980) which allows a high relative deterioration in the first iteration and which then monotonously becomes more strict until no deterioration is accepted at all. Once the chaining procedure finishes (line 9), i.e., no profitable $H_4 + V_3$ move among the non-fixed duties A exists, the solution

Table 1
Characteristics of the instances. For each instance, the number of duties of type early (E), late (L), and night (N) is shown, together with the total number of duties, and the number of employees of each category early (E), late-night (LN), and mixed (M). Finally, the number of roster groups per instance is shown.

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

is updated and all duties are removed from the set of fixed duties. If an improving chain was found, we again search for an improving chain for the updated solution (line 11). Otherwise, we try to escape the current local optimum by using one (strictly profitable) $H_6 + V_4$ move (line 13). If this succeeds, we repeat the chaining procedure for the updated solution (lines 15 and 16). Otherwise the algorithm terminates and the final solution is found.

6. Computational experiments

To evaluate the performance of the heuristic, we apply our solution approach to different instances based on data from NS. In Section 6.1, we discuss the experimental set-up and in Section 6.2, we show the computational results.

6.1. Experimental set-up

We consider a total of 10 instances. The first four instances are those considered in Breugem et al. (2021a), and can be used to validate the performance of the heuristic, as the optimal solutions are known for these instances. For the remaining six (larger) instances, no optimal solutions are known.

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

For each instance, Table 1 shows the number of duties of each type, and the size of the groups per category (i.e., the entry 12/8 for LN means the instance contains one LN group of 12 employees and one LN group of 8 employees). The first four instances all consist of three groups, one of each type, of varying sizes. The second four instances all consist of four groups of varying type and varying size. The fifth instance is the largest among these four, with approximately 190 duties, whereas the eighth instance is the smallest, with roughly 120 duties. The final two instances both consist of six groups, with two groups of each type. The total number of duties for both instances is roughly 250.

As discussed in Section 2.2, fairness is based on five different duty attributes and attractiveness is based on rest time, rest day, workload, and variation constraints. We model binary roster constraints using clique constraints (see Breugem (2020)). We consider nine variation constraints, one for each attribute used to compute fairness and four additional constraints concerning the appearance of specific trajectories in duties.

Similar to Breugem et al. (2021a), each instance enforces an upper bound of 8 h on duty length, 2.5 on the reputation within duty coefficient, the average working time per group, and upper bound of 2.5 on the repetition within duty coefficient, an upper bound of 18% and 40% on aggression and double decker work, respectively, and a lower bound of 35% on high quality work. All weights are set to one, except for the duty length (0.5) and repetition within duty (25). The same weights are used for the variation constraints. Rest times shorter than 16 h, but feasible, are penalized with weight 30. Hence, avoiding a shorter than 16 h rest time is equivalent to reducing the average duty length in a row by one hour.

6.2. Computational results

We apply our methodology to each of the ten instances. The cell-based formulation used to compute the fair solution is solved using CPLEX 22.1.0 with a deterministic time limit.[1] of 15 min. Similarly, we set a time limit of 15 min for the VDNS method. For the pairwise improvement using column generation (in this section referred to as CG method), we set a deterministic time limit of two minutes per pair of roster groups. The linear relaxation of the row-based formulation is solved using column generation, and the reduced problem is again solved using CPLEX 22.1.0. We use the methodology of Breugem et al. (2021a) for solving the linear relaxation. The CPLEX implementation and column generation are coded in Java. The VDNS heuristic is coded in C. For completeness, we also compute the results when allowing a running time of one hour for CPLEX. The results in this case, as shown in Appendix A.2, are similar.

We compare our solutions with those obtained from solving the problem directly using CPLEX for the cell-based formulation (from hereon we simply refer to this approach as CPLEX). Here we also use a deterministic time limit of 15 min. We also analyze the performance of the CG and VDNS method separately.

For each instance, we consider the fairness levels based on the percentages 0%, 25%, 50%, and 75%. This way, the fairness levels cover the entire spectrum of possible solutions.

We use instances 1 to 4 to validate the performance of the different components of the heuristic. The results are presented in Section 6.2.1. We then analyze the performance of CPLEX, the CG method, the VDNS method, and finally the three-phase heuristic (combining CG and VDNS) on the larger instances in Section 6.2.2.

6.2.1. Results instances 1 to 4

The results for the first four instances are shown in Table 2. For each instance and percentage, we show the fairness and attractiveness of the solution obtained from CPLEX, CG, and VDNS. Table 2 also shows the optimality gap (in percentages), computed based using the known optimal solution obtained from Breugem et al. (2021a), and the running time (in seconds).

Table 2 shows that the CPLEX and CG method find high-quality, close to optimal, solutions for almost all instances. CPLEX finds optimal solutions for all but the first instance, often using the entire allotted computation time (except for instances 3 and 4), while the CG method is able to find (close to) optimal solutions, finishing consistently in a few seconds. The VDNS method is able to find good solutions for all instances, but shows a larger gap than the other two methods. Running times are comparable to the CG method. Note that all heuristics find similar solutions in terms of range of fairness levels and solution quality. Overall, Table 2 shows the improvement methods in the three-phase heuristic perform well, achieving an optimality gap of at most a few percentage points in all cases.

6.2.2. Results instances 5 to 10

Table 3 shows the results for instances five to ten for CPLEX, CG, and VDNS separately. We observe that the CG method often outperforms

[1] To avoid randomness in found solutions, see: https://www.ibm.com/docs/en/icos/22.1.0?topic=parameters-deterministic-time-limit.

Table 2
Results instances one to four for the different methods. The table shows the fairness level, the attractiveness (objective), the gap (in percentages) and the run time (in seconds). Each method is given a deterministic maximum time of 15 min. The gap given the fairness level is computed using the known optimal solution obtained from Breugem et al. (2021a).

	CPLEX				CG				VDNS			
	Fairness	Attractiveness	Gap	Time	Fairness	Attractiveness	Gap	Time	Fairness	Attractiveness	Gap	Time
1	16.1	1196.0	1.7	865	16.1	1196.0	1.7	5	16.1	1196.0	1.7	38
	19.0	1173.0	0.7	912	19.0	1188.0	2.0	10	16.1	1196.0	1.7	10
	21.5	1148.1	1.4	814	21.1	1161.0	2.4	8	20.4	1167.3	1.3	11
	22.4	1135.5	0.7	784	22.4	1127.5	0.0	6	21.4	1164.8	2.8	11
2	29.0	1288.0	0.0	764	29.0	1288.0	0.0	3	29.0	1288.0	0.0	23
	30.2	1222.2	0.5	797	30.5	1257.3	3.2	2	30.7	1226.3	0.8	9
	31.2	1192.5	0.0	789	30.5	1257.3	3.2	3	33.6	1218.2	2.6	10
	33.1	1188.3	0.0	780	34.3	1204.5	2.0	3	34.7	1226.1	3.3	10
3	7.6	979.3	0.0	803	7.6	979.3	0.0	2	7.6	979.3	0.0	3
	9.7	797.3	0.0	10	10.8	840.4	5.3	1	13.4	834.1	4.8	1
	13.7	794.0	0.0	11	14.2	798.6	0.6	1	18.8	806.9	2.4	1
	15.2	791.3	0.0	13	15.2	791.3	0.0	1	18.8	806.9	2.4	1
4	4.5	910.5	0.0	954	4.5	910.5	0.0	6	4.5	910.5	0.0	4
	9.6	785.1	0.0	76	5.5	851.4	2.5	1	6.5	877.2	5.7	1
	15.3	766.9	0.0	48	9.0	808.4	2.5	1	9.8	817.3	3.4	2
	15.3	766.9	0.0	42	12.1	774.2	0.0	1	11.0	822.5	4.8	1

Table 3
Results for instances five to ten for the different methods. The table shows the fairness level, the attractiveness (objective), the gap (in percentages) and the run time (in seconds). Each method is given a deterministic maximum time of 15 min. The gap given the fairness level is computed using the optimal value of the linear relaxation for (4) for the row clustering.

	CPLEX				CG				VDNS			
	Fairness	Attractiveness	Gap	Time	Fairness	Attractiveness	Gap	Time	Fairness	Attractiveness	Gap	Time
5	1.4	2127.2	26.3	850	1.4	1988.0	21.1	225	1.4	2147.2	27.0	172
	5.8	2065.4	26.3	903	6.5	1675.4	9.4	36	5.8	1939.6	21.5	70
	9.6	1665.8	9.6	932	11.2	1595.3	5.9	23	10.7	1800.8	16.6	69
	9.6	1660.8	9.4	1060	12.4	1587.3	5.6	17	15.0	1789.1	16.4	67
6	1.3	1909.1	31.1	964	1.3	1909.1	31.1	229	1.1	1880.8	30.0	125
	4.5	1698.1	23.6	1003	5.9	1436.0	10.0	12	6.5	1582.6	18.5	48
	9.9	1420.4	9.7	864	11.2	1345.4	4.8	7	12.2	1538.5	16.8	49
	12.9	1417.0	9.7	808	17.1	1318.1	3.1	6	15.4	1471.5	13.1	50
7	1.0	1831.3	22.9	909	1.0	1831.3	22.9	193	1.0	1831.3	22.9	122
	7.2	1510.6	10.9	860	6.6	1423.7	5.1	12	8.2	1625.4	17.7	49
	13.8	1416.3	8.1	913	12.2	1390.9	5.8	7	15.5	1492.2	13.3	48
	20.1	1368.5	6.7	913	18.0	1377.3	6.8	8	22.6	1405.8	9.7	50
8	4.6	1460.8	22.6	767	4.6	1428.5	20.9	247	4.6	1395.6	19.0	53
	6.6	1414.5	20.9	869	8.1	1270.3	12.4	4	4.6	1395.6	19.0	21
	8.4	1277.1	13.0	762	12.8	1229.0	10.4	2	11.2	1255.6	12.2	21
	10.5	1216.0	9.2	767	17.3	1149.4	4.2	2	14.8	1211.2	9.1	20
9	2.7	2921.1	37.1	996	2.6	2336.2	21.3	200	2.6	2821.2	34.9	750
	6.0	2890.7	36.9	996	7.2	2103.0	13.4	22	6.5	2303.3	20.9	295
	10.3	2651.5	31.4	983	12.4	2033.3	10.6	17	6.5	2303.3	20.9	297
	14.3	2615.1	30.5	1002	17.8	1980.8	8.2	18	14.3	2191.8	17.0	309
10	1.9	2828.7	43.8	999	1.9	2241.9	29.1	433	1.9	2824.1	43.8	926
	9.0	2286.4	33.4	983	9.2	1850.1	17.8	52	9.0	2138.0	28.8	424
	15.8	2177.4	31.7	1054	16.2	1653.3	10.2	34	15.8	2054.2	27.6	417
	23.9	1900.9	23.1	1065	24.1	1589.1	8.0	30	23.3	1894.8	22.8	430

CPLEX, especially for the larger instances. Furthermore, the computation times for the CG method are often below one minute. The VDNS method is outperformed by CG. For many instances, the method is competitive to CPLEX, especially for larger instances the VDNS method performs better.

In Appendix A.1, we analyze the performance of the CPLEX and CG method when considering only the soft rest time constraints or the variation constraints. The results show the former can in many cases be solved efficiently by CPLEX, wheres for the later the CG method is by superior. This indicates the strength of the CG method is better capturing the penalty incurred from variation constraints. The good performance of CPLEX in some of the cases is in line with the observations in Ge et al. (2022).

For all instances, we observe a substantial increase in attractiveness can be achieved for higher fairness levels. The gap with the root bound indicates that the found solutions are of high quality, although in some cases a substantial gap is still present. We note, however, that a substantial integrality gap can be expected for the FCRP, especially for tight fairness budgets (see Breugem (2020)).

Finally, we analyze the performance of the three-phase heuristic, sequentially applying the CG and VDNS method. To assess the performance, in particular the gain from Phase 3, we also analyze the solution

Table 4
Results instances five to ten for the three-phase heuristic and CPLEX with the Phase 2 solution as starting solution. Entries in bold indicate Phase 3 improved on the solution found in Phase 2. For each solution, we show the fairness and attractiveness level, the optimality gap (in percentages, computed using the optimal value of the linear relaxation), and the gain (percentage reduction in gap) achieved in Phase 3 (i.e., compared to the solution found in Phase 2).

	CG + CPLEX					CG + VDNS				
	Fairness	Attractiveness	Gap	Gain	Time	Fairness	Attractiveness	Gap	Gain	Time
5	1.4	1988.0	21.1	0.0	918	1.4	**1986.8**	21.1	0.2	74
	6.5	1675.4	9.4	0.0	967	6.1	**1672.5**	9.2	1.7	73
	11.2	1595.3	5.9	0.0	909	11.6	**1593.8**	5.8	1.5	71
	12.4	1587.3	5.6	0.0	928	12.4	1587.3	5.6	0.0	71
6	1.3	1909.1	31.1	0.0	873	1.3	1909.1	31.1	0.0	52
	5.9	1436.0	10.0	0.0	1023	6.5	**1433.4**	9.8	1.6	49
	11.2	1345.4	4.8	0.0	1042	11.9	**1336.3**	4.1	13.6	49
	17.1	1318.1	3.1	0.0	1020	16.8	**1315.3**	2.9	6.7	51
7	1.0	1831.3	22.9	0.0	868	1.0	1831.3	22.9	0.0	51
	6.6	**1422.7**	5.0	1.3	1107	6.6	**1423.2**	5.1	0.7	50
	12.2	1390.9	5.8	0.0	1055	12.2	**1390.8**	5.8	0.1	49
	17.2	**1359.0**	5.6	18.4	1017	17.8	**1358.3**	5.5	19.1	52
8	4.6	**1424.7**	20.7	1.0	767	4.6	**1410.2**	19.8	4.9	22
	7.0	**1261.3**	11.8	5.0	812	8.4	**1254.1**	11.3	9.1	21
	12.7	**1187.6**	7.2	30.1	804	12.8	1229.0	10.4	0.0	22
	17.2	**1144.3**	3.8	10.1	838	17.3	1149.4	4.2	0.0	23
9	2.6	2336.2	21.3	0.0	1148	2.6	**2320.8**	20.8	2.4	309
	7.2	2103.0	13.4	0.0	1213	7.7	**2096.9**	13.1	1.9	310
	12.4	**2031.4**	10.5	0.8	1145	12.7	**2010.1**	9.5	9.8	297
	16.9	**1978.9**	8.1	1.1	1229	14.6	**1971.5**	7.8	5.3	313
10	1.9	2241.9	29.1	0.0	1185	1.8	**2228.3**	28.7	1.5	420
	9.2	1850.1	17.8	0.0	1136	9.3	**1837.5**	17.2	3.2	440
	15.9	**1649.5**	10.0	2.0	1263	16.7	**1618.1**	8.3	19.1	429
	23.8	**1584.9**	7.8	3.0	1118	23.7	**1578.8**	7.4	7.5	444

found if CPLEX would be used to search for further improvements in Phase 3. That is, we consider the solution found in Phase 2 as starting solution for CPLEX. Table 4 shows the results of this analysis for instances five to ten. Entries in bold indicate Phase 3 improved on the solution found in Phase 2. For each solution, we show the fairness and attractiveness level, the optimality gap (in percentages, computed using the optimal value of the linear relaxation), and the gain (percentage reduction in gap) achieved in Phase 3 (i.e., compared to the solution found in Phase 2).

Table 4 shows the third phase of the heuristic can substantially improve the solution. The improvement varies per instance and is typically smaller compared to the second phase. Nevertheless, reductions of the gap of about 10% are not uncommon. The results show that the VDNS method outperforms CPLEX for all but one instance (instance 8). For the large instances this is most clear. The VDNS method also performs better in terms of running time.

Summarizing, our results show the three-phase heuristic performs well for practically sized instances. The heuristic outperforms CPLEX for the larger instances, and the different improvement steps find close-to-optimal solutions for the first four solutions used to validate the performance. The largest improvement in solution quality is due to the pairwise improvement method in the second phase of the algorithm. Depending on the instance, the VDNS method in the third phase can realize a substantial improvement, in almost all cases outperforming CPLEX in terms of solution quality and time.

7. Conclusion

We considered the joint optimization of attractiveness and fairness in cyclic crew rostering. Cyclic rosters, common in public transportation, are long-term rosters executed by groups of employees (roster groups). Fairness relates to differences in work between roster groups, and attractiveness concerns properties (e.g., work-life balance) of each individual cyclic roster. The resulting problem was introduced in Breugem et al. (2021a) as the Fairness-Oriented Crew Rostering Problem (FCRP), and is motivated by crew rostering practices at NS.

The combinatorial complexity of cyclic rostering combined with fairness requirements makes solving the FCRP challenging for large-scale practical instances. Short computation times are desirable, because in practice crew rosters are often constructed iteratively, with decision makers tweaking parameter settings in each iteration. We therefore presented a three-phase heuristic for the FCRP, combining column generation with variable-depth neighborhood search. The three phases of the heuristic are as follows: finding a fair allocation (Phase 1), and improving the solution via column generation (Phase 2) and variable-depth neighborhood search (Phase 3). The heuristic exploits two mathematical formulations for the problem, one to effectively find feasible solutions and one to search for improvements.

We evaluated the heuristic on practical data from Netherlands Railways (NS), and showed that the heuristic finds good solutions for most instances. Furthermore, the heuristic outperforms a state-of-the-art commercial solver. The largest improvement in solution quality is due to the pairwise improvement method in the second phase of the algorithm. Depending on the instance, the variable-depth neighborhood search method in the third phase can also realize a substantial improvement.

We see most promising avenues for further research in the extension of our methodology to different practical contexts. For example, to non-cyclic crew planning in airline applications, or to scheduling of security personnel. Each application comes with unique characteristics and challenges, necessitating different adaptions of the heuristic. However, applications are in essence similar regarding attractiveness (avoiding irregular rosters) and fairness (balancing work among employees). As such, we believe similar heuristics can be developed for these contexts. On the methodological level, interesting opportunities exist in the development of novel hybrid methods combining our proposed methodology with metaheuristic approaches for contexts where the latter are known to perform especially well.

Table 5
Results for instances five to ten for the CPLEX and CG method when excluding variation constraints. The table shows the fairness level, the attractiveness (objective), and the run time (in seconds). Each method is given a deterministic maximum time of 15 min.

	CPLEX			CG		
	Fairness	Attractiveness	Time	Fairness	Attractiveness	Time
5	1.4	1380.0	917	1.4	1380.0	277
	10.2	930.0	7	7.9	990.0	46
	10.2	930.0	1	15.1	960.0	12
	10.2	930.0	1	15.0	960.0	16
6	1.2	750.0	822	1.3	1050.0	299
	8.9	720.0	1	7.1	780.0	8
	8.9	720.0	1	9.7	750.0	5
	8.9	720.0	1	9.7	750.0	5
7	1.0	810.0	527	1.0	1080.0	225
	14.4	750.0	1	7.4	810.0	9
	14.4	750.0	1	11.0	780.0	7
	14.4	750.0	1	11.0	780.0	7
8	4.6	840.0	814	4.6	840.0	291
	10.5	600.0	1	6.9	690.0	4
	10.5	600.0	1	10.6	600.0	2
	10.5	600.0	1	10.6	600.0	2
9	2.7	1890.0	781	2.7	1560.0	229
	13.9	1170.0	2	10.7	1260.0	17
	14.6	1170.0	1	16.8	1230.0	19
	14.6	1170.0	1	16.8	1230.0	17
10	1.9	1560.0	851	1.8	900.0	413
	12.4	570.0	7	9.8	720.0	72
	12.4	570.0	4	14.4	690.0	54
	12.4	570.0	3	14.4	690.0	49

Table 6
Results for instances five to ten for the CPLEX and CG method when excluding soft rest time constraints. The table shows the fairness level, the attractiveness (objective), and the run time (in seconds). Each method is given a deterministic maximum time of 15 min.

	CPLEX			CG		
	Fairness	Attractiveness	Time	Fairness	Attractiveness	Time
5	1.4	718.2	746	1.4	632.8	193
	5.8	709.9	766	5.2	484.9	68
	8.9	667.8	748	10.3	463.8	28
	8.9	664.7	769	11.9	460.0	29
6	1.3	797.8	716	1.2	682.4	308
	6.8	705.3	729	3.3	565.2	68
	10.3	547.8	743	5.8	519.3	25
	17.6	519.1	762	8.0	487.1	14
7	1.0	740.2	684	1.0	695.3	229
	10.8	750.0	752	6.4	535.2	41
	19.7	465.1	761	12.8	479.5	12
	23.6	453.1	820	18.3	471.8	17
8	4.6	554.2	661	4.6	494.3	266
	8.5	491.5	712	6.6	451.5	8
	10.7	453.6	731	8.7	436.6	4
	14.9	424.4	710	11.6	422.0	2
9	2.7	988.4	776	2.7	648.2	327
	10.0	901.9	809	5.9	617.3	34
	12.6	875.6	814	9.1	596.4	30
	15.3	855.6	774	11.7	575.1	21
10	1.9	1268.7	765	1.9	944.5	542
	9.3	1108.0	758	8.7	807.4	233
	15.6	1026.6	761	15.5	757.0	219
	18.5	1009.5	826	21.5	734.9	285

Data availability

The data that has been used is confidential.

Acknowledgments

Parts of this work have been developed within the Research Campus MODAL (Mathematical Optimization and Data Analysis Laboratories) funded by the German Ministry of Education and Research (BMBF). The majority of this work has been done while at Erasmus University, Rotterdam, the Netherlands and while visiting the Zuse Institute, Berlin, Germany.

Appendix. Additional computational experiments

A.1. Sensitivity analysis

we analyze the performance of the CPLEX and CG method when considering instances five to ten considering only the soft rest time constraints and only the variation constraints. The results are shown in Tables 5 and 6, respectively. The results show the former can in many cases be solved efficiently by CPLEX, wheres in the later case the CG method is superior. This shows the strength of the CG method is better capturing the penalty incurred from variation constraints.

A.2. Longer computation times

we analyze the performance of the CPLEX and CG+CPLEX method when considering instances five to ten and a maximum deterministic running time of one hour for CPLEX. For ease of comparison, we take the fairness level found using the CG approach as basis. The results are shown in Table 7. The results show that in almost all cases there is limited improvement from the longer computation time for the CG+CPLEX method compared to the results in Table 4, hence the method is still outperformed by CG+VDNS in most cases, especially for the larger instances. Running stand-alone CPLEX for an hour, instead of 15 min, does lead to improvements for the medium instances, while the attractiveness stays well above the level achieved by the CG method for the largest instances, as can be seen when comparing with Table 3.

Table 7
Results instances five to ten for CPLEX and CG + CPLEX when allowing a deterministic running time of one hour for CPLEX. For each solution, we show the fairness and attractiveness level, and the computation time.

	CPLEX*			CG + CPLEX*		
	Fairness	Attractiveness	Time	Fairness	Attractiveness	Time
5	1.4	2127.2	3515	1.4	1988.0	3596
	6.4	2000.4	3903	6.5	1661.6	4171
	10.6	1649.7	3944	11.1	1590.5	4105
	12.4	1620.9	4053	12.4	1587.3	4031
6	1.3	1909.1	3327	1.3	1909.1	3464
	5.8	1458.7	3498	5.9	1436.0	3675
	10.2	1357.2	3671	11.2	1345.4	3759
	16.6	1339.6	3910	17.1	1318.1	3886
7	1.0	1831.3	3179	1.0	1831.3	3250
	6.2	1504.8	3834	6.6	1422.7	4150
	11.5	1451.3	3634	12.2	1390.9	3566
	17.9	1375.0	3723	17.2	1359.0	3604
8	4.6	1460.8	3130	4.6	1424.7	3183
	7.8	1279.7	3398	7.0	1258.8	3290
	11.5	1152.6	3381	12.7	1164.7	3282
	17.3	1149.4	3441	17.2	1144.3	3240
9	2.7	2921.1	4021	2.6	2335.0	3077
	6.6	2669.4	4115	7.2	2103.0	4121
	11.6	2503.9	4340	12.4	2031.4	4279
	17.5	2185.5	4381	16.9	1978.9	4424
10	1.9	2828.7	4206	1.9	2241.9	3856
	8.8	2585.2	4400	9.2	1850.1	4387
	15.6	2161.7	4378	15.9	1649.5	4478
	21.7	1838.9	4447	23.8	1584.9	4329

References

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

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

Abbink, E., Huisman, D., Kroon, L., 2018. Railway crew management. In: Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M., Schlechte, T. (Eds.), Handbook of Optimization in the Railway Industry. Springer International Publishing, pp. 243–264.

Ahuja, R.K., Ergun, Ö., Orlin, J.B., Punnen, A.P., 2002. A survey of very large-scale neighborhood search techniques. Discrete Appl. Math. 123 (1–3), 75–102.

Bertsimas, D., Farias, V.F., Trichakis, N., 2012. On the efficiency-fairness trade-off. Manage. Sci. 58 (12), 2234–2250.

Bertsimas, D., Farias, V.F., Trichakis, N., 2013. Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. Oper. Res. 61 (1), 73–87.

Bertsimas, D., Gupta, S., 2015. Fairness and collaboration in network air traffic flow management: an optimization approach. Transp. Sci. 50 (1), 57–76.

Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M., Schlechte, T., 2018. Handbook of Optimization in the Railway Industry, Vol. 268. Springer.

Borndörfer, R., Langenhan, A., Löbel, A., Schulz, C., Weider, S., 2013. Duty scheduling templates. Public Transp. 5 (1–2), 41–51.

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

Borndörfer, R., Sagnol, G., Schlechte, T., Swarat, E., 2017a. Optimal duty rostering for toll enforcement inspectors. Ann. Oper. Res. 252 (2), 383–406.

Borndörfer, R., Schulz, C., Seidl, S., Weider, S., 2017b. Integration of duty scheduling and rostering to increase driver satisfaction. Public Transp. 9 (1–2), 177–191.

Breugem, T., 2020. Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency (Ph.D. thesis). Erasmus Research Institute of Management (ERIM).

Breugem, T., Dollevoet, T., Huisman, D., 2021a. Is equality always desirable? Analyzing the trade-off between fairness and attractiveness in crew rostering. Manage. Sci..

Breugem, T., van Rossum, B., Dollevoet, T., Huisman, D., 2021b. A column generation approach for the integrated crew re-planning problem. Omega 102555.

Caprara, A., Fischetti, M., Toth, P., Vigo, D., Guida, P.L., 1997. Algorithms for railway crew management. Math. Program. 79 (1–3), 125–141.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 2007. Passenger railway optimization. Handbooks Oper. Res. Management Sci. 14, 129–187.

Chutima, P., Arayikanon, K., 2020. Many-objective low-cost airline cockpit crew rostering optimisation. Comput. Ind. Eng. 150, 106844.

Cordeau, J.F., Stojković, G., Soumis, F., Desrosiers, J., 2001. Benders decomposition for simultaneous aircraft routing and crew scheduling. Transp. Sci. 35 (4), 375–388.

Dantzig, G.B., 1954. Letter to the editor-A comment on Edie's "Traffic delays at toll booths". J. Oper. Res. Soc. Am. 2 (3), 339–341.

De Causmaecker, P., Vanden Berghe, G., 2011. A categorisation of nurse rostering problems. J. Sched. 14 (1), 3–16.

Desrochers, M., Soumis, F., 1989. A column generation approach to the urban transit crew scheduling problem. Transp. Sci. 23 (1), 1–13.

Dickerson, J.P., Procaccia, A.D., Sandholm, T., 2019. Failure-aware kidney exchange. Manage. Sci. 65 (4), 1768–1791.

Doi, T., Nishi, T., Voß, S., 2018. Two-level decomposition-based matheuristic for airline crew rostering problems with fair working time. European J. Oper. Res. 267 (2), 428–438.

Eisenhandler, O., Tzur, M., 2019. The humanitarian pickup and distribution problem. Oper. Res. 67 (1), 10–32.

Er-Rbib, S., Desaulniers, G., Elhallaoui, I., Munroe, P., 2021. Preference-based and cyclic bus driver rostering problem with fixed days off. Public Transp. 13 (2), 251–286.

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

Freling, R., Lentink, R.M., Wagelmans, A.P.M., 2004. A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm. Ann. Oper. Res. 127 (1–4), 203–222.

Ge, L., Kliewer, N., Nourmohammadzadeh, A., Voß, S., Xie, L., 2022. Revisiting the richness of integrated vehicle and crew scheduling. Public Transp. 1–27.

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

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

Heil, J., Hoffmann, K., Buscher, U., 2020. Railway crew scheduling: Models, methods and applications. European J. Oper. Res. 283 (2), 405–425.

Hoffman, K.L., Padberg, M., 1993. Solving airline crew scheduling problems by branch-and-cut. Manage. Sci. 39 (6), 657–682.

Hooker, J.N., Williams, H.P., 2012. Combining equity and utilitarianism in a mathematical programming model. Manage. Sci. 58 (9), 1682–1693.

Huisman, D., 2007. A column generation approach for the rail crew re-scheduling problem. European J. Oper. Res. 180 (1), 163–173.

Huisman, D., Freling, R., Wagelmans, A.P.M., 2005a. Multiple-depot integrated vehicle and crew scheduling. Transp. Sci. 39 (4), 491–502.

Huisman, D., Kroon, L.G., Lentink, R.M., Vromans, M.J.C.M., 2005b. Operations research in passenger railway transportation. Stat. Neerl. 59 (4), 467–497.

Jütte, S., Müller, D., Thonemann, U., 2017. Optimizing railway crew schedules with fairness preferences. J. Sched. 20 (1), 43–55.

Kanellakis, P.C., Papadimitriou, C.H., 1980. Local search for the asymmetric traveling salesman problem. Oper. Res. 28 (5), 1086–1099.

Karsu, Ö., Morton, A., 2015. Inequity averse optimization in operational research. European J. Oper. Res. 245 (2), 343–359.

Kohl, N., Karisch, S.E., 2004. Airline crew rostering: Problem types, modeling, and optimization. Ann. Oper. Res. 127 (1–4), 223–257.

Lettovskỳ, L., Johnson, E.L., Nemhauser, G.L., 2000. Airline crew recovery. Transp. Sci. 34 (4), 337–348.

Lin, S., Kernighan, B.W., 1973. An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. 21 (2), 498–516.

Lourenço, H.R., Paixão, J.P., Portugal, R., 2001. Multiobjective metaheuristics for the bus driver scheduling problem. Transp. Sci. 35 (3), 331–343.

Lučić, P., Teodorović, D., 2007. Metaheuristics approach to the aircrew rostering problem. Ann. Oper. Res. 155 (1), 311–338.

McCoy, J., Lee, H., 2014. Using fairness models to improve equity in health delivery fleet management. Prod. Oper. Manage. 23 (6), 965–977.

Mesquita, M., Moz, M., Paias, A., Pato, M., 2013. A decomposition approach for the integrated vehicle-crew-roster problem with days-off pattern. European J. Oper. Res. 229 (2), 318–331.

Nishi, T., Sugiyama, T., Inuiguchi, M., 2014. Two-level decomposition algorithm for crew rostering problems with fair working condition. European J. Oper. Res. 237 (2), 465–473.

Perumal, S., Dollevoet, T., Huisman, D., Lusby, R., Larsen, J., Riis, M., 2021a. Solution approaches for integrated vehicle and crew scheduling with electric buses. Comput. Oper. Res. 132, 105268.

Perumal, S., Larsen, J., Lusby, R., Riis, M., Christensen, T., 2021b. A column generation approach for the driver scheduling problem with staff cars. Public Transp. 1–34.

Perumal, S., Larsen, J., Lusby, R., Riis, M., Sørensen, K., 2019. A matheuristic for the driver scheduling problem with staff cars. European J. Oper. Res. 275 (1), 280–294.

Perumal, S., Lusby, R., Larsen, J., 2022. Electric bus planning & scheduling: A review of related problems and methodologies. European J. Oper. Res. 301 (2), 395–413.

Pisinger, D., Ropke, S., 2010. Large neighborhood search. In: Handbook of Metaheuristics. Springer, pp. 399–419.

Potthoff, D., Huisman, D., Desaulniers, G., 2010. Column generation with dynamic duty selection for railway crew rescheduling. Transp. Sci. 44 (4), 493–505.

Quesnel, F., Desaulniers, G., Soumis, F., 2020. Improving air crew rostering by considering crew preferences in the crew pairing problem. Transp. Sci. 54 (1), 97–114.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L., 2013. Personnel scheduling: A literature review. European J. Oper. Res. 226 (3), 367–385.

Wolbeck, L., 2019. Fairness Aspects in Personnel Scheduling. Diskussionsbeiträge 2019/16, Freie Universität Berlin, http://dx.doi.org/10.17169/refubium-26050.

Xie, L., Merschformann, M., Kliewer, N., Suhl, L., 2017. Metaheuristics approach for solving personalized crew rostering problem in public bus transit. J. Heuristics 23 (5), 321–347.

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

Zhou, S.Z., Zhan, Z.H., Chen, Z.G., Kwong, S., Zhang, J., 2020. A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction. IEEE Trans. Intell. Transp. Syst. 22 (11), 6784–6798.