10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

A two-phase heuristic and a lexicographic rule for improving fairness in personnel rostering

Komarudin · Marie-Anne Guerry · Pieter Smet ·
Tim De Feyter · Greet Vanden Berghe

Abstract A fair allocation of workload to people is crucial for securing job satisfaction. Researchers have introduced numerous objectives and algorithms to represent and improve fairness in personnel rostering problems. These approaches should not ignore the roster quality that is influenced by personnel rostering constraints, such as maximum working times, minimum rest times, etc. The present paper proposes a new fairness objective and an effective two-phase heuristic for optimizing rosters, taking into consideration the established personnel rostering constraints and the fairness. The new fairness objective is based on a lexicographic rule that offers a beneficial trade-off between roster quality and fairness. The new heuristic is tested on real world data and the results show that fair rosters can be obtained without significantly decreasing the roster quality.

Keywords Personnel rostering · roster quality · fairness · two-phase heuristic · lexicographic evaluation

1 Introduction

Personnel rostering aims to produce a timetable for personnel that satisfies the coverage requirements in a predefined time period. Furthermore, the generated timetable should meet a variety of contractual and personal constraints. Generally, each constraint is specified with a certain weight, denoting its penalty value when violated. The *individual penalty* refers to the sum of penalties that can be associated with one member of personnel. The objective of the problem is to improve the

Komarudin · Marie-Anne Guerry
Vrije Universiteit Brussel, Department of Business Technology and Operations
E-mail: {komarudin, maguerry}@vub.ac.be

Tim De Feyter
KU Leuven, Department of Business Management Research
E-mail: tim.defeyter@kuleuven.be

Pieter Smet · Greet Vanden Berghe
KU Leuven, Department of Computer Science, CODeS
E-mail: {pieter.smet,greet.vandenberghe}@cs.kuleuven.be

roster quality by minimizing the total penalty (see e.g. Smet et al (2013)). This kind of personnel rostering problem arises in several domains, including health care, transportation, and security (Ernst et al, 2004).

In addition to the roster quality, Stolletz and Brunner (2012) advocate to also consider fairness of work assignment. Fairness can be seen as the degree to which individual penalties are balanced over all employees. An unfair roster is likely to result in larger differences in e.g. workload among employees, which could induce job dissatisfaction (Larrabee et al, 2003). It is thus important that a solution has both a high roster quality and a high fairness.

Several approaches have been proposed to improve fairness of personnel rosters. Bard and Purnomo (2005) consider fairness by specifying an upper limit on the individual penalty for each employee, determined by the individual penalty that he/she collected in the previous planning period. This approach does not actually balance work assignments over all employees in a certain period. Rather it allows to compensate for a low quality roster (i.e. low individual penalty) from the previous planning period.

Chiaramonte and Chiaramonte (2008) represent fairness by a ratio that is based on the standard deviation of the individual penalties. Similarly, Stolletz and Brunner (2012) use the range (the difference between the maximum and the minimum) of the individual penalties to balance the number of working times and work assignments over all employees. Several other objectives have been defined for improving both the roster quality and the fairness. Smet et al (2012b) compare three objectives: (1) the maximum individual penalty, (2) the absolute deviation of the individual penalties, and (3) the range of the individual penalties. Martin et al (2013) in addition introduce the sum of squared penalties.

The objectives defined by Smet et al (2012b) are generalizations of Chiaramonte and Chiaramonte (2008)'s and Stolletz and Brunner (2012)'s objectives, which can be used to improve fairness over different rostering periods. They are not restricted to specific constraints only, but they can be used to balance penalties for all types of constraints associated with an individual employee. They provide a general way to extend the established personnel rostering models with a fairness aspect. Furthermore they have the advantage that they preserve the value of weights that represent the degree of importance of the rostering constraints.

All previous studies consider fairness by aggregating the individual penalties. In contrast, the current paper introduces a new objective that employs a different methodology. The new objective is represented by a vector of all individual penalties. It facilitates high fairness through beneficial penalty trade-off between employees. Moreover, the new objective does not strive to level the individual penalties over all employees, since this is not always appropriate due to the heterogeneous nature of a workforce in terms of contractual and personal constraints (Komarudin et al, 2013).

In Section 2, we describe the personnel rostering problem in general and discuss examination criteria of the roster quality and the fairness. Martin et al (2013) compared different objectives in an optimization algorithm using Jain's fairness index (Jain et al, 1984). This index may not suitable since it depends on the magnitude of individual penalties. We show that if the difference between two individual penalties is kept the same, the index is likely to improve as the individual penalties increase. A good Jain's index value is thus not always the result of a fair roster, but it can also be caused by high individual penalties.

Therefore, similar to Bard and Purnomo (2005), we suggest to use three examination criteria that provide quantitative information for comparing rosters: (1) the total penalty, (2) the standard deviation of the individual penalties, and (3) the maximum individual penalty. These criteria provide a comprehensive assessment of a roster since they examine the roster quality, the fairness, and the most unattractive individual rosters. We will use these examination criteria to compare the different objectives.

In Section 3, we discuss several local search moves that are used for solving a personnel rostering problem with a fairness objective. These local search moves were introduced by Smet et al (2012b) and used by Martin et al (2013). When optimizing fairness, such moves may hinder the exploration for finding high quality rosters, since the use of a fairness objective may ignore a local search move that improves the roster quality but does not improve the fairness. For example, when the maximum individual penalty is used as the objective, a local search move that does not decrease the maximum individual penalty is likely to be rejected.

In order to deal with the contradictory objectives, we propose a two-phase heuristic. The first phase ignores the fairness and uses only the total penalty as the objective of the heuristic. In the second phase, one of the fairness objectives is used in order to obtain balanced work assignments. This approach allows the heuristic to reach high quality rosters in the search space before considering fairness.

In Section 4, we describe and analyze a series of computational experiments in order to investigate the effectiveness of the two-phase heuristic compared to the one-phase heuristic. Moreover, the roster quality and the fairness resulting from the experiments with different objectives are compared. The heuristics are applied to the personnel rostering model of Bilgin et al (2012) and they are evaluated on data from six hospital wards (Smet et al, 2012a).

Finally, the conclusion and suggestions for further research are discussed in Section 5.

2 The personnel rostering model and three examination criteria

The personnel rostering model is based on the work by Smet et al (2012a). A number of objectives that can be used for improving fairness are presented. We also discuss the limitations of the objectives and then propose a new objective that has several advantages. Furthermore, we explain three examination criteria for assessing the roster quality and the fairness.

2.1 The personnel rostering model

The personnel rostering problem has been formulated in several models (Ernst et al, 2004), which vary in formulation of the objective and the rostering constraints. This paper considers the model of Smet et al (2012a) which is formulated in a general way such that it can take into account a large set of rostering constraints. Due to this flexibility in modeling, it has been implemented in several Belgian hospitals.

The personnel rostering problem is defined as the problem of assigning personnel to shifts, subject to hard and soft constraints. The hard constraints define the feasibility of a solution while the soft constraints determine the quality of a solution. The hard and soft constraints of Smet et al (2012a)'s model are summarized in Table 1. A weight is associated with each soft constraint, denoting its penalty value for a violation. The objective of the problem is to satisfy all hard constraints and to minimize the total penalty.

Table 1 Hard and soft constraints of Smet et al (2012a)'s personnel rostering model

Hard constraints	Soft constraints
Single assignment per day	Coverage requirements
Assignment of defined requirements only	Training requirements
No overlapping assignments	Collaboration restriction
Assignment requires specific skill type	Rest time between two consecutive assignments
Fixed assignments	Skill type priorties
	Absence requests
	Counter restrictions on assignments
	Specific series pattern assignment

Smet et al (2012a) divide rostering constraints in two categories: the horizontal constraints and the vertical constraints. A horizontal constraint corresponds to a specific employee. Satisfaction of a horizontal constraint solely depends on the employee's roster and not on other employees. Meanwhile, a vertical constraint is defined according to general characteristics that apply to a group of employees. Satisfaction of a vertical constraint is subject to the shift assignments of the employees in the group. For example, satisfaction of a training requirement constraint depends on the shift assignments of the trainer and the trainee. In Table 1, the first three soft constraints (coverage requirements, training requirements and collaboration restriction) are vertical constraints, while the other soft constraints are horizontal constraints.

Assume the personnel is indexed $i = 1 \ldots n$, the individual penalty $P_{H,i}$ is the sum of penalties from the horizontal constraints of employee i. The vertical penalty P_V is the sum of penalties from the vertical constraints. The calculation of P_V and $P_{H,i}$ follows the formulation of Smet et al (2012a). When we only consider the roster quality, the objective of the personnel rostering problem is to minimize the total penalty P^{WS} (defined in Eq. 1) while obeying all hard constraints.

$$\begin{aligned} &\text{minimize } P^{WS}; \\ &\text{with } P^{WS} = P_V + \textstyle\sum_{i=1}^{n} P_{H,i}; \end{aligned} \tag{1}$$

2.2 The fairness objectives

Eq. 1 only optimizes the roster quality but does not pay attention to improving fairness. In the literature, four alternative objectives have been presented to take into account the roster quality as well as the fairness (Smet et al, 2012b; Martin et al, 2013). They are:

1. The vertical and the maximum individual penalty P^{Max} (Eq. 2),

2. The total penalty and the absolute deviation of the individual penalties P^{Dev} (Eq. 3),
3. The total penalty and the range of individual penalties P^{Error} (Eq. 4), and
4. The sum of squared penalties P^{SS} (Eq. 5).

$$P^{Max} = P_V + n. \max_{i \in \{1..n\}} P_{H,i}; \tag{2}$$

$$P^{Dev} = P^{WS} + \sum_{i=1}^{n} |P_{H,i} - \frac{1}{n} \sum_{i=1}^{n} P_{H,i}|; \tag{3}$$

$$P^{Error} = P^{WS} + n.(\max_{i \in \{1..n\}} P_{H,i} - \min_{i \in \{1..n\}} P_{H,i}); \tag{4}$$

$$P^{SS} = \sqrt{(P_V)^2 + \sum_{i=1}^{n} (P_{H,i})^2}; \tag{5}$$

Objective functions 2-5 target improvements both in terms of roster quality and fairness. Minimizing P^{Max} expresses minimization of the worst individual penalty. In this way, the other individual penalties cannot be higher than the worst one. Minimizing P^{Dev} implies improving the roster quality and minimizing the individual penalty differences among employees. On the other hand, P^{Error} improves fairness by decreasing the difference between individual penalties among employees. The quadratic expression in P^{SS} prevents one individual penalty to be too high compared to other.

It should be noted that objective P^{SS} modifies the surface of the solution space of the original objective P^{WS} to a large extent, due to the quadratic operation. This effect is noteworthy since the relative importance of the constraints changes. Martin et al (2013), therefore, introduced weights for P_V and $P_{H,i}$ to reformulate P^{SS}. Many trial-and-error experiments may be needed to find appropriate weights that can restore the importance degrees of the constraints. The surface of the solution space also changes for P^{Max}, P^{Dev} and P^{Error}. However, the effect will not be as large as it is for P^{SS}, since in the former objectives, the weights of the constraints are preserved. That is, P_v and $P_{H,i}$ are included instead of $(P_v)^2$ and $(P_{H,i})^2$.

In order to provide a more accurate representation of the fairness, we introduce a new objective: P^{Lexi}. This objective is similar to the decomposition fairness model for course timetabling problems proposed by Mühlenthaler and Wanka (2013). P^{Lexi} is defined as a permutation of all individual penalties, sorted in a non-increasing order (represented in Eq. 6).

$$P^{Lexi} = (P'_{H,1}, P'_{H,2}, .., P'_{H,n}) \; s.t. \; P'_{H,1} \geq P'_{H,2} \geq ... \geq P'_{H,n} \tag{6}$$

For P^{Lexi}, fairness can then be defined as follows.

Definition 1 Roster p is considered to be more fair than roster q if $P^{Lexi}(p) <^L P^{Lexi}(q)$, i.e. $P^{Lexi}(p)$ is lexicographically smaller than $P^{Lexi}(q)$.

In other words, we say that roster p is more fair than roster q if the first non-zero component of the vector $P^{Lexi}(p) - P^{Lexi}(q)$ is negative.

The objective P^{Lexi} offers several advantages. Similar to the use of P^{Max}, the use of P^{Lexi} minimizes the maximum individual penalty. In fact, the individual penalty of each employee is minimized when P^{Lexi} is used since all individual penalties are represented. This is realized by minimizing the next maximum individual penalty in stages over all employees. The individual penalty for all employee is minimized step by step, starting with the employee having the maximum penalty up to the employee having the minimum penalty. This behavior differs from the behavior of P^{Dev} that tries to minimize the individual penalty differences.

Minimizing P^{Lexi} involves a trade-off mechanism that has an intuitive meaning. A trade-off between two employees is said to be beneficial if it results in a situation that is better for both employees. For example, moving workload from employee i to employee j is beneficial if $max\{P_{H,i}, P_{H,j}\}$ becomes smaller. That is, two employees can help to alleviate of each other's workload as long as the maximum of their individual penalties decreases. In addition, the objective P^{Lexi} does not favor negative trade-off. This is in contrast with P^{Dev} or P^{Error} that can favor to increase an individual penalty without decreasing other individual penalties.

The main disadvantage of P^{Lexi} is that it is only suitable for rostering problems without vertical soft constraints, since P_V is not considered in Eq. 6. If there are vertical soft constraints, P^{Lexi} could be used by adding a new hard constraint to the model which specifies an upper bound on the vertical penalty P_V. A less restrictive approach would be to incorporate P_V in P^{Lexi}, as is shown in Eqs. 7-8. Eq. 7 combines the objectives P^{WS} and P^{Lexi} and prioritizes the former against the latter. Eq. 8 combines the objectives P^{WS} and P^{Lexi} and tries to find a good balance between the roster quality and the fairness.

$$P^{ModLexi1} = (P^{WS}, P'_{H,1}, P'_{H,2}, .., P'_{H,n}) \; s.t. \; P'_{H,1} \geq P'_{H,2} \geq ... \geq P'_{H,n} \tag{7}$$

$$\begin{aligned} P^{ModLexi2} = (P'_{H,1} + P^{WS}, P'_{H,2} + P^{WS}, .., P'_{H,n} + P^{WS}) \\ s.t. \; P'_{H,1} \geq P'_{H,2} \geq ... \geq P'_{H,n} \end{aligned} \tag{8}$$

P^{Lexi} should be applied with care. Objective P^{Lexi} in a rostering problem with vertical hard constraints can produce a higher total penalty than the objective P^{WS}. When optimizing P^{Lexi}, a beneficial trade-off may increase the total penalty P^{WS}.

2.3 Criteria to compare the roster quality and the fairness

The objectives P^{Max}, P^{Dev}, P^{Error}, and P^{SS} integrate the roster quality and the fairness in their respective way. Comparing the qualities of several rosters based on these objectives can produce biased results. Furthermore, the objectives P^{Lexi}, $P^{ModLexi1}$, $P^{ModLexi2}$ can be used to obtain the order of the fairness of two or more rosters. However, it cannot provide quantitative differences in roster quality. Similar to Bard and Purnomo (2005), we suggest to calculate three examination criteria that can be used for comparing the quality of two or more rosters:

1. The total penalty P^{WS}
2. The standard deviation of the individual penalties $\sigma(P_H)$
3. The maximum individual penalty $P_{H,max} = \max_{i \in \{1..n\}} P_{H,i}$

with $P_H = \{P_{H,1}, ..., P_{H,n}\}$.

The first examination criterion corresponds to the roster quality, the second one represents the fairness, and the third relates to the most unattractive individual rosters.

Contrary to Martin et al (2013), we do not recommend Jain's index to compare the quality of rosters based on the following consideration. Let $J(p)$ be Jain's index of roster p, and let the individual roster quality of employee i in roster p be $P_{H,i}(p)$. $J(p)$ can be calculated as shown in Eq. 9. When the difference between individual penalties decreases, Jain's index increases. Therefore, a high value of $J(p)$ indicates high fairness. The maximum value of $J(p)$ is one, which indicates that all employees receive the same individual penalty.

$$J(p) = \frac{(\sum_{i=1}^{n} P_{H,i}(p))^2}{n.\sum_{i=1}^{n}(P_{H,i}(p))^2} \tag{9}$$

Unfortunately, Jain's index can also be improved by increasing the individual penalties while keeping the differences fixed, as stated in Proposition 1. The proposition compares two rosters that have individual penalties differing by the same amount $d > 0$. In this situation, both rosters have the same value of $\sigma(P_H)$. However, the roster with the higher total penalty has a better Jain's index. Therefore, a high value of Jain's index may not always refer to high fairness since it may also be the result of an increase of the individual penalties. Jain's index is still suitable when the total horizontal penalty is fixed, or when the objective is to be maximized.

Proposition 1 *Suppose there are two rosters p and q. If $J(p) < 1$ and $P_{H,i}(q) = P_{H,i}(p) + d$, $\forall i \in \{1, .., n\}, d > 0$, then $J(q) > J(p)$.*

Proof Substituting $P_{H,i}(q) = P_{H,i}(p) + d$ into Eq. 9, results in

$$J(q) = \frac{(\sum_{i=1}^{n} P_{H,i}(p))^2 + M}{n.\sum_{i=1}^{n}(P_{H,i}(p))^2 + M}$$

with $M = 2nd\sum_{i=1}^{n} P_{H,i}(p) + n^2d^2$. Since $d > 0$ and $J(p) < 1$, then $J(q) > J(p)$. $\square$

3 The algorithm

We implemented a tabu search algorithm for solving the rostering problems. Tabu search is a well known local search heuristic that has a mechanism to prevent the search from returning to already visited solutions. Algorithm 1 outlines the procedure. The algorithm starts from a random feasible solution, which is improved by exploring neighborhoods through local search moves. The local search moves used for the current personnel rostering problem are Bilgin et al (2012):

- Make or delete an assignment of an employee at a specific day

- Change a shift assignment of an employee at a specific day
- Change a skill assignment of an employee at a specific day
- Swap shift assignments of two employees at a specific day

The algorithm accepts the best neighboring solution that improves upon the best solution obtained so far. If there is no such solution, it accepts the best neighboring solution that results from a non-tabu move. A local search move is considered non-tabu if it does not match with any element in the tabu list. The tabu list keeps record of the characteristics of the recently executed moves. Specifically, it keeps track of the task assignments that have been changed by performing a move. In each record, four variables are maintained: the day index, the employee index, the skill assignment, and the shift assignment. In case a local search move involves two employees, two records are saved in the tabu list.

A greedy local search is applied to every new best solution found in order to intensify the search.

```
Input: Tabu search parameters, initial solution n^0;
Best solution := initial solution;
Current solution := initial solution;
while stopping condition not met do
    Generate k neighboring solutions from the current solution;
    if A new best solution found then
        Improve the new best solution through greedy local search;
        Update best solution and current solution;
        Update tabu list;
    end
    else if Non-tabu solution(s) found then
        Update current solution with best non-tabu solution;
        Update tabu list;
    end
end
Output: best solution;
```

Algorithm 1: The tabu search algorithm

The local search moves from Bilgin et al (2012) may not be suitable for minimizing P^{Max}. The value of the objective P^{Max} can be decreased by decreasing P_V and/or $P_{H,max}$. In this way, local search moves that do not decrease P_V or $P_{H,max}$ are not useful in the search. Meanwhile, improving individual rosters that do not correspond to $P_{H,max}$ can be beneficial at later iterations when these individual rosters become the worst ones.

For minimizing P^{Dev}, the local search moves from Bilgin et al (2012) are also not always effective. Consider a situation where $P_{H,i} = P_{H,j}, \forall i, j \in \{1..n\}$. In this situation, the roster can be improved by decreasing P_V. Decreasing one or two individual penalties (assume that $n > 4$) can only make P^{Dev} increase since the absolute deviation increases more than the gained improvement. This situation can indicate the search is trapped in a local optimum. In other words, a local search move that makes an individual penalty to deviate largely from the average is usually not beneficial for improving P^{Dev}. Similarly, a heuristic minimizing P^{Error} can face the same issue.

Considering that the local search moves may not be effective to minimize P^{Max}, P^{Dev} and P^{Error}, we propose a two-phase approach. The first phase optimizes objective P^{WS} while the second phase uses one of the fairness objectives. The first phase is intended to obtain a roster with few constraint violations. The second phase aims to improve the fairness. In order to show the advantage of a two-phase heuristic, we tested the different algorithmic configurations shown in Table 2.

Table 2 Overview of algorithms

Heuristic method	Type	Objective function
MinWS (base method)	one-phase	P^{WS}
OP-MinMax	one-phase	P^{Max}
OP-MinDev	one-phase	P^{Dev}
OP-MinError	one-phase	P^{Error}
OP-MinSS	one-phase	P^{SS}
OP-MinModLexi1	one-phase	$P^{ModLexi1}$
OP-MinModLexi2	one-phase	$P^{ModLexi2}$
TP-MinMax	two-phase	first phase P^{WS}, second phase P^{Max}
TP-MinDev	two-phase	first phase P^{WS}, second phase P^{Dev}
TP-MinError	two-phase	first phase P^{WS}, second phase P^{Error}
TP-MinSS	two-phase	first phase P^{WS}, second phase P^{SS}
TP-MinModLexi1	two-phase	first phase P^{WS}, second phase $P^{ModLexi1}$
TP-MinModLexi2	two-phase	first phase P^{WS}, second phase $P^{ModLexi2}$

4 Experimental results

4.1 Experimental setup

The algorithms are evaluated using instances based on data from six wards (Smet et al, 2012a). Table 3 provides an overview of the instance characteristics. Note that we simplified the problem by omitting the continuity constraints that consider assignments before and after the rostering period.

Table 3 Instance characteristics of Smet et al (2012a)

No	Ward	Abbr.	No of skill types	No of shift types	No of employees	No of days
1	Emergency	Em	4	27	27	28
2	Geriatrics	Gr	2	9	21	28
3	Meal Preparations	MP	2	9	32	31
4	Psychiatry	Ps	3	14	19	31
5	Reception	Re	4	19	19	42
6	Palliative Care	PC	4	23	27	91

The tabu search algorithm was parametrized with a tabu list length of 1000 and a maximum neighborhood size of 1000. If a new best solution is found, 1000 greedy local search iterations are executed. A time limit of 600 seconds is imposed. It should be noted that the computation time for each phase in the two-phase heuristic, is half of the maximum computation time. For each problem instance, five repeated runs were executed. The algorithm was coded in C++ and the experiments were performed on a PC with a 2.7 GHz Intel processor operating on Windows 7.

In the following sections, two experiments are analyzed. Section 4.2 discusses the results for the base experiment, while Section 4.3 presents results for the slack experiment.

4.2 Base experiment

The first experiment evaluates the effectiveness of the two-phase approach and the one-phase approach. The instances are solved using one of the heuristic methods from Table 2. First, we compare the resulting objective values obtained with the two approaches. Then, the resulting rosters are compared in terms of the three examination criteria discussed in Section 2.3.

The objective values obtained with the one-phase approach (OP) and two-phase approach (TP) are shown in Figures 1-2. The horizontal axis corresponds to different heuristic methods. The objective values are obtained by evaluating the final rosters produced by each heuristic method according to the respective fairness objective. The vertical axis in Figure 1 is a ratio obtained by dividing the objective value by the minimum one for each problem instance. This allows to collect several problem instances that have different magnitudes of the weight values into one figure. In Figure 2, the vertical axis represents an index that is obtained as follows. For each problem instance, the vectors P^{Lexi} of the rosters are sorted using $<^L$ (lexicographic sorting). Then, the index represents the position of a roster in the sorted list.

When comparing MinWS with the heuristic methods with a fairness objective, the latter generally perform better. Figure 1 shows that the medians of the ratios obtained with MinWS are always higher than those obtained with the heuristic fairness methods. Figure 2 shows that the median of the index obtained using TP-MinModLexi1 is slightly higher than the one obtained with MinWS. Both methods try to optimize the objective P^{WS}, which is the first criterion when lexicographically sorting the results. TP-MinModLexi1 has the advantage that the range of the results is significantly smaller than the range obtained with MinWS.

The results show that the two-phase heuristics always perform better than the one-phase heuristics, when optimizing P^{Max}, P^{Dev}, P^{Error}, $P^{ModLexi1}$ and $P^{ModLexi2}$. There is no difference between the one-phase and two-phase heuristics when applying P^{SS}.

Now, we compare the performance of the heuristics from Table 2 using the three examination criteria discussed in Section 2.3. Computational results are presented in Tables 4-6. The values of P^{WS}, $P_{H,max}$ and $\sigma(P_H)$ are the averages over five runs. The last two rows of each table summarize the performance of each heuristic. *# with 5% gap* denotes the number of instances for which the heuristic obtained results within 5% from the best result. Results that differ no more than 5% from the best of all heuristics are indicated in bold. The *average ratio* represents the average difference between the best obtained P^{WS}, $P_{H,max}$, $\sigma(P_H)$ and the results obtained with heuristics with a fairness objective.

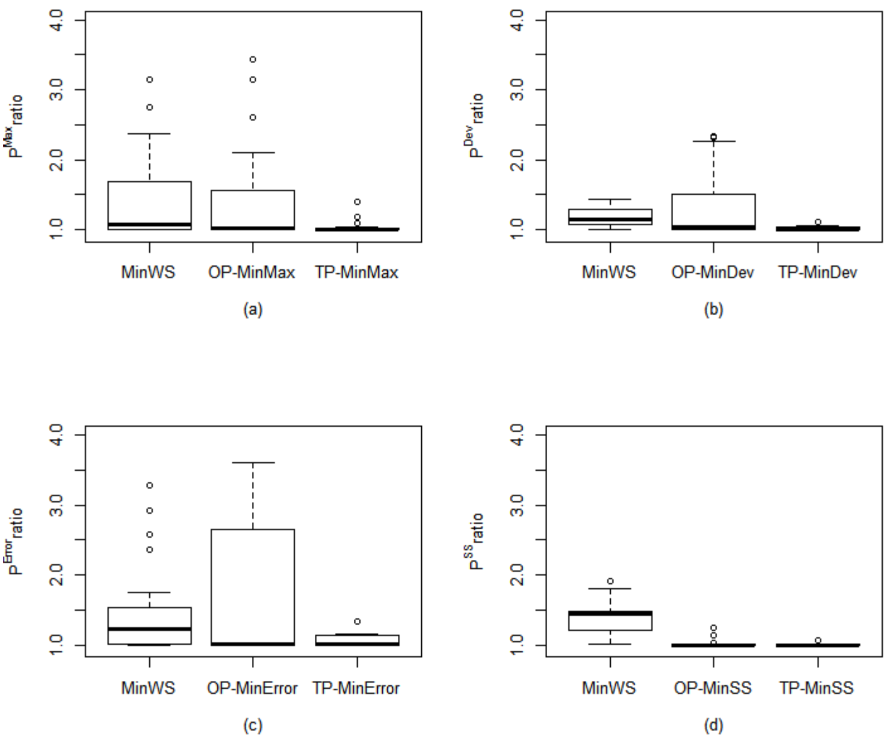

Fig. 1 Comparison of heuristics (a) MinWS, OP-MinMax and TP-MinMax, (b) MinWS, OP-MinDev and TP-MinDev, (c) MinWS, OP-MinError and TP-MinError, (d) MinWS, OP-MinSS and TP-MinSS

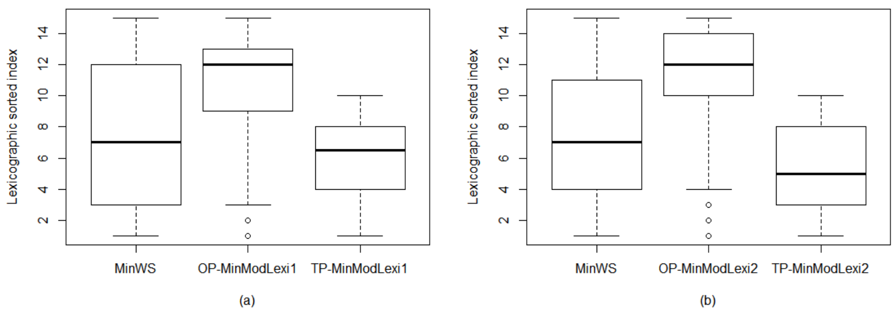

Fig. 2 Comparison of heuristics (a) MinWS, OP-MinModLexi1 and TP-MinModLexi1, (b) MinWS, OP-MinModLexi2 and TP-MinModLexi2

With respect to P^{WS}, the heuristics MinWS, TP-MinDev, TP-MinError, TP-MinModLexi1 and TP-MinModLexi2 all produce good results; i.e. four out of six instances within 5% of the best. This is to be expected, since their objectives all contain P^{WS}. P^{Max} only considers the total vertical penalty and the worst individual penalty. Improving P^{Max} can be achieved by moving (if possible) the vertical penalty to the non-worst individual penalty. This explains the results of TP-MinMax, which deviate largely from the results of MinWS. The objective used in TP-MinSS is different from P^{WS} because of the quadratic operation. Regarding $P_{H,max}$ and $\sigma(P_H)$, most two-phase heuristics with fairness objectives achieve better results than MinWS.

In general, the one-phase heuristics with fairness objectives perform worse than MinWS for all three examination criteria. In Section 3, we argued that the use of fairness objectives can result in ineffective search. P^{Max} and P^{Error} guide the search mainly based on the local search moves that reduce the maximum individual penalty. P^{Dev} has the disadvantage that it does not accept an improvement of one or two individual penalties that largely deviate from the average of all individual penalties.

Except for TP-MinSS, the two-phase heuristic generally performs better than the one-phase heuristic on the three examination criteria, as can be noted by their lower average ratio values. Among the two-phase methods, TP-MinSS produces the worst results for all three examination criteria, because, as mentioned in Section 2.2, the objective surface of the solution space can be quite different.

Table 4 Comparison of P^{WS} results obtained by different heuristics

Instance	MinWS	MinMax Heu.		MinDev Heu.		MinError Heu.		MinSS Heu.		MinModLexi1 Heu.		MinModLexi2 Heu.	
		OP	TP	OP	TP	OP	TP	OP	TP	OP	TP	OP	TP
Em	11,683	306%	112%	130%	**99%**	112%	**101%**	220%	229%	151%	**100%**	138%	**100%**
Gr	13,051	145%	146%	119%	115%	128%	124%	111%	108%	107%	**102%**	110%	**105%**
MP	22,724	163%	171%	**100%**	**100%**	**102%**	**102%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Ps	9,432	239%	101%	177%	97%	168%	142%	202%	204%	**91%**	**91%**	109%	**94%**
Re	28,238	156%	132%	108%	**100%**	**100%**	**100%**	110%	**102%**	108%	**101%**	108%	**101%**
PC	147,710	243%	123%	**100%**	**100%**	**100%**	**100%**	**101%**	**102%**	107%	**100%**	**104%**	**100%**
# with 5% gap	6	0	0	2	4	3	4	2	3	2	6	2	6
Average ratio	100%	209%	131%	122%	**102%**	118%	111%	141%	141%	111%	**99%**	112%	**100%**

Table 5 Comparison of $P_{H,max}$ results obtained by different heuristics

Instance	MinWS	MinMax Heu.		MinDev Heu.		MinError Heu.		MinSS Heu.		MinModLexi1 Heu.		MinModLexi2 Heu.	
		OP	TP	OP	TP	OP	TP	OP	TP	OP	TP	OP	TP
Em	326	414%	**68%**	247%	**68%**	363%	**71%**	362%	370%	255%	**69%**	222%	79%
Gr	2,429	31%	32%	47%	48%	34%	**26%**	61%	55%	85%	86%	55%	61%
MP	3,827	**57%**	**57%**	**58%**	**58%**	**57%**	**57%**	**58%**	**58%**	73%	66%	**58%**	**58%**
Ps	400	260%	**100%**	370%	**100%**	410%	**100%**	285%	281%	**100%**	**100%**	282%	**100%**
Re	1,520	**104%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
PC	24,320	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
# with 5% gap	6	3	5	3	5	3	6	3	3	3	4	3	4
Average ratio	100%	161%	76%	154%	79%	177%	76%	161%	161%	119%	87%	136%	83%

Table 6 Comparison of $\sigma(P_H)$ results obtained by different heuristics

Instance	MinWS	MinMax Heu.		MinDev Heu.		MinError Heu.		MinSS Heu.		MinModLexi1 Heu.		MinModLexi2 Heu.	
		OP	TP	OP	TP	OP	TP	OP	TP	OP	TP	OP	TP
Em	72	448%	83%	264%	**74%**	329%	90%	296%	321%	336%	83%	266%	83%
Gr	720	36%	37%	34%	35%	42%	**31%**	51%	51%	79%	76%	59%	61%
MP	825	69%	68%	**38%**	**38%**	65%	68%	**38%**	**38%**	51%	45%	**38%**	**38%**
Ps	140	149%	105%	269%	99%	217%	**54%**	142%	150%	92%	92%	217%	98%
Re	496	**92%**	102%	98%	100%	100%	100%	103%	109%	97%	100%	**95%**	99%
PC	5,283	127%	107%	**83%**	**83%**	102%	102%	**82%**	**82%**	88%	88%	88%	88%
# with 5% gap	6	1	0	2	3	0	2	2	2	0	0	2	1
Average ratio	100%	153%	84%	131%	72%	143%	74%	119%	125%	124%	81%	127%	78%

Overall, TP-MinDev and TP-MinError are shown to perform best as they can balance P^{WS}, $P_{H,max}$ and $\sigma(P_H)$. TP-MinDev results in an increase of 2 percentage points over MinWS for P^{WS}, but it manages to reduce $P_{H,max}$ and $\sigma(P_H)$ with 21 and 28 percentage points, respectively. TP-MinError results in an increase of 11 percentage points over MinWS for P^{WS}, while $P_{H,max}$ and $\sigma(P_H)$ improve with 24 and 26 percentage points, respectively. However, it should be noted that for some instances, the results of TP-MinDev and TP-MinError differ largely from the results of MinWS. TP-MinDev produces a P^{WS} value 15 percentage points higher than MinWS for the Geriatrics ward, and TP-MinError produces P^{WS} 42 percentage points higher than MinWS for the Psychiatry ward.

TP-MinModLexi1 and TP-MinModLexi2 present a more balanced result, i.e. these methods are capable of producing both rosters with few violations and high fairness. For P^{WS}, the values are comparable to those produced by MinWS, however, for $P_{H,max}$ and $\sigma(P_H)$, the lexicographic heuristics produce significantly lower values. TP-MinModLexi1 improves 13 and 19 percentage points on $P_{H,max}$ and $\sigma(P_H)$, while TP-MinModLexi2 improves 17 and 22 percentage points on $P_{H,max}$ and $\sigma(P_H)$.

4.3 Slack experiment

The second experiment compares the fairness resulting from the two-phase heuristic methods with different objectives. First, the algorithms are modified such that they behave similar to goal programming, i.e. the value of the objective P^{WS} obtained in the first phase is added as a hard constraint to the model for the second phase. This new constraint ensures that new solutions in the second phase are only accepted if their total penalty P^{WS} is less than or equal to the value obtained in the first phase, allowing some level of slack β. Three values for β are considered: 0%, 5%, and 10%. In the second phase, one of the following objectives is optimized: P^{Max}, P^{Dev}, P^{Error}, P^{SS} or P^{Lexi}. Algorithm 1 is still used in both phases. Note that in the experiments, the first phase is only run once (for each problem instance, each replication) such that the second phase always starts from the same initial solution.

Table 7 shows that the final total penalty P^{WS} for the two-phase heuristic is indeed always within $(100 + \beta)$% of MinWS. TP-MinDev and TP-Minerror seemingly do not make much use of the allowed slack as their P^{WS} values are close to 100%. TP-MinLexi on the other hand, takes much more advantage of the allowed slack.

In general, the results show that the additional slack can help the heuristics to improve fairness. An increase of P^{WS} is usually accompanied by a decrease of $P_{H,max}$ and $\sigma(P_H)$. This behavior is true for all the two-phase heuristics except for the heuristics with objective P^{SS}.

Figures 3-5 show the heuristics ordered by fairness. Similar with Figure 2, the order index in Figures 3-5 is obtained by lexicographically sorting ($<^L$) the vectors P^{Lexi}. The results show that TP-MinLexi generally produces vectors P^{Lexi} that are lexicographically smaller than the ones produced by other heuristics. This shows that TP-MinLexi minimizes the penalty of each employee by allowing a positive trade-off among them, which is a significant advantage and makes this heuristic suitable for practical applications.

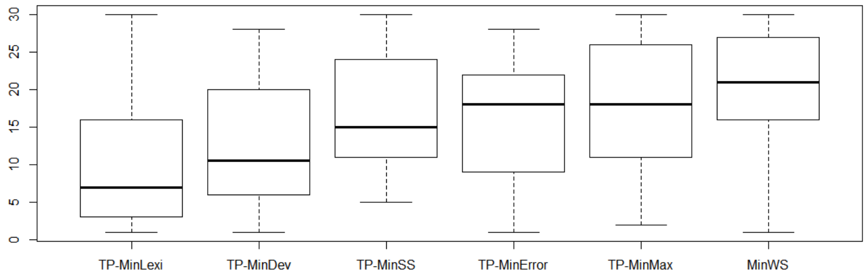

Fig. 3 Comparison of two-phase heuristics with 0% slack

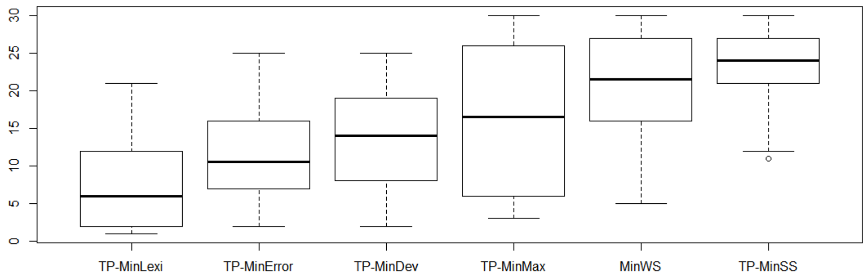

Fig. 4 Comparison of two-phase heuristics with 5% slack

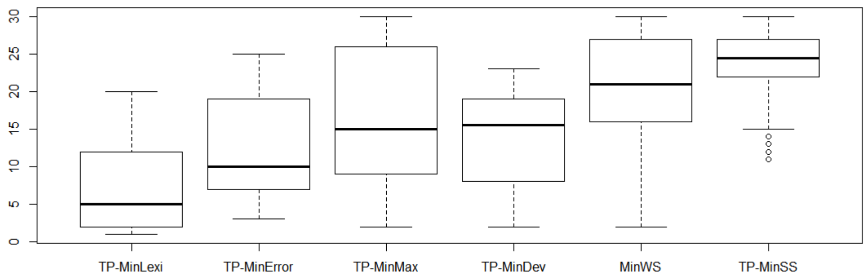

Fig. 5 Comparison of two-phase heuristics with 10% slack

Table 7 Comparison of P^{WS}, $P_{H,max}$ and $\sigma(P_H)$ results obtained by two-phase heuristics and the slack approach

Index	Slack	MinWS	TP-MinMax	TP-MinDev	TP-MinError	TP-MinSS	TP-MinLexi
P^{WS}	0%	100.0%	99.9%	99.0%	99.3%	99.6%	99.7%
	5%	100.0%	102.8%	99.3%	100.2%	102.4%	104.4%
	10%	100.0%	105.4%	100.9%	100.9%	103.5%	108.6%
$P_{H,max}$	0%	100.0%	87.5%	87.3%	85.0%	98.6%	85.1%
	5%	100.0%	83.5%	86.2%	81.0%	118.7%	83.5%
	10%	100.0%	78.4%	82.8%	77.5%	121.8%	76.3%
$\sigma(P_H)$	0%	100.0%	92.8%	79.4%	89.3%	88.4%	80.2%
	5%	100.0%	91.7%	75.9%	86.5%	107.6%	77.5%
	10%	100.0%	88.5%	74.2%	86.0%	117.6%	74.8%

5 Conclusion and future research

The present paper introduced methodologies to improve fairness in personnel rostering. First, a new lexicographic objective was described. Second, a two-phase heuristic approach, which makes use of the lexicographic evaluation, was presented. Finally, an extension of the two-phase approach was introduced which allows for some slack on the total penalty in order to enable fairness improvements in the second phase.

Computational experiments have been analyzed to identify the effectiveness of the new contributions. Three examination criteria from the academic literature were used to asses the roster quality and the fairness. The computational results showed that fair rosters can be produced, without significantly decreasing the roster quality.

This research can be extended in several ways. We argued that the existing local search moves may not be effective for optimizing the objectives representing fairness. A chain of local search moves as described by Burke et al (2013), may be beneficial. An algorithm that directly optimizes the three examination criteria in an aggregated manner can be investigated. Possible directions may be to consider a multi-objective approach or several iterative phases.

References

Bard JF, Purnomo HW (2005) Preference scheduling for nurses using column generation. European Journal of Operational Research 164(2):510–534

Bilgin B, De Causmaecker P, Rossie B, Vanden Berghe G (2012) Local search neighbourhoods for dealing with a novel nurse rostering model. Annals of Operations Research 194(1):33–57

Burke EK, Curtois T, Qu R, Vanden Berghe G (2013) A time predefined variable depth search for nurse rostering. INFORMS Journal on Computing 25(3):411–419

Chiaramonte MV, Chiaramonte LM (2008) An agent-based nurse rostering system under minimal staffing conditions. International Journal of Production Economics 114(2):697–713

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

Jain R, Chiu D, Hawe W (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Tech. Rep. DEC-TR-301, Digital Equipment Corporation

Komarudin, Guerry M, De Feyter T, Vanden Berghe G (2013) The roster quality staffing problem - a methodology for improving the roster quality by modifying the personnel structure. European Journal Of Operational Research 230:551–562

Larrabee JH, Janney MA, Ostrow CL, Withrow ML, Hobbs Jr GR, Burant C (2003) Predicting registered nurse job satisfaction and intent to leave. Journal of Nursing Administration 33(5):271–283

Martin S, Ouelhadj D, Smet P, Vanden Berghe G, Özcan E (2013) Cooperative search for fair nurse rosters. Expert Systems with Applications 40:6674–6683

Mühlenthaler M, Wanka R (2013) A decomposition of the max-min fair curriculum-based course timetabling problem : The impact of solving subproblems to optimality. In: Multidisciplinary International Sceduling Conference (MISTA) 2013

Smet P, Bilgin B, De Causmaecker P, Vanden Berghe G (2012a) Modelling and evaluation issues in nurse rostering. Annals of Operations Research pp 1–24

Smet P, Martin S, Ouelhadj D, Özcan E, Vanden Berghe G (2012b) Investigation of fairness measures for nurse rostering. In: the International Conference on the Practice and Theory of Timetabling (PATAT 2012)

Smet P, De Causmaecker P, Bilgin B, Vanden Berghe G (2013) Nurse Rostering: A Complex Example of Personnel Scheduling with Perspectives. Studies in Computational Intelligence 505:129–153

Stolletz R, Brunner JO (2012) Fair optimization of fortnightly physician schedules with flexible shifts. European Journal of Operational Research 219(3):622–629