Annals of Operations Research (2021) 302:363–390
https://doi.org/10.1007/s10479-020-03552-5

S.I.: PATAT 2018

An integer programming approach for the physician rostering problem

Toni I. Wickert[1,2] · **Alberto F. Kummer Neto**[1] · **Márcio M. Boniatti**[3] · **Luciana S. Buriol**[1]

Published online: 28 February 2020

Abstract

This paper presents an integer programming model and a matheuristic for the physician rostering problem. Constraints related to physician's preferences, legal restrictions, hospital requirements and workload balance are investigated. Moreover, a comparison of physician and nurse rostering is analyzed to identify common constraints present in both problems. The lack of a mathematical model that could be applied in the studied problem motivated the development of a new formulation. Data and constraints of the proposed physician rostering problem were provided by Hospital de Clínicas de Porto Alegre (HCPA), Brazil. Due to the difficulty of solving large instances, a matheuristic which combines integer programming and heuristic techniques was proposed to approach the problem. Computational experiments were conducted employing commercial and open-source solvers. Results revealed that open-source solvers are a good alternative to solve small problems, while the proposed matheuristic generated near-optimal results within acceptable computation time employing larger instances.

Keywords Physician rostering · Integer programming · Matheuristic

✉ Toni I. Wickert
tiwickert@inf.ufrgs.br

Alberto F. Kummer Neto
afkneto@inf.ufrgs.br

Márcio M. Boniatti
mboniatti@hcpa.edu.br

Luciana S. Buriol
buriol@inf.ufrgs.br

1 Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil

2 Department of Computer Science, CODeS & imec, KU Leuven, Gebroeders De Smetstraat 1, 9000 Gent, Belgium

3 Hospital de Clínicas de Porto Alegre (HCPA), Porto Alegre, Brazil

1 Introduction

Personnel rostering has received significant attention in academic literature due to its benefits and practical applicability (Ernst et al. 2004). However, various institutions still organize their rosters manually, demanding a considerable effort of planners and in many cases generating poor results. In addition, a large set of input data is required to organize a roster, such as legal labor requirements, patients attendance, employee's preferences and constraints related to contractual restrictions. The automation of this problem using computational techniques brings a series of benefits such as better employee distribution across the rosters, less time preparing the rosters and consequently an overall cost reduction.

The physician rostering problem (PRP), which aims to schedule physicians qualified in one or more specialties, has obtained increased academic attention. Erhard et al. (2018) provide a thorough overview of the existing literature. Problems were classified as staffing problems, focusing on how the required size of the workforce is defined; rostering problems, where the main objective is the generation of the rosters; and re-planning problems, addressing short-term adjustments to already scheduled physicians.

Similarly to the PRP the nurse rostering problem (NRP) is among the most common problems found related to personnel scheduling. There is a large number of publications approaching problems related to specific hospitals such as Burke et al. (2010), Petrovic and Vanden Berghe (2012) and Burke et al. (2006). Supplementary to such publications, two competitions, the first and second nurse rostering competition (INRC-I) (Haspeslagh et al. 2014) and (INRC-II) (Ceschia et al. 2019), have been organized. For these competitions, a common set of instances was proposed, making it easier to compare solving techniques and stimulating the research on algorithms for solving the NRP. The INRC-I addressed a single-skilled static problem, where the entire planning horizon must be solved at once. By contrast, the INRC-II approached a multi-skilled problem and each week of the planning horizon must be solved separately. Meaning that the final roster is a combination of separate solutions satisfying global constraints. Each solving method was free to decide how to deal with historical data and the uncertainty of future data.

Clearly, the INRC-II has made significant progress compared to the INRC-I by trying to specify a problem-solving environment that is closer to real-world conditions by way of including multi-skilled employees and considering the history of the previous roster. Nevertheless, these models still lack generality. For example, there is no possibility to specify the balance of working hours between the scheduled employees or a preferred skill if an employee has more than one. In addition, only a fixed problem was proposed without the option of turning off certain constraints or changing them from hard to soft (or vice versa). Such changes would render solving methods more flexible and, therefore, heighten the chances of them actually being used in real-world scenarios.

This work approaches the PRP based on the data provided by Hospital de Clínicas de Porto Alegre (HCPA), Brazil. The objective is to answer the following research question: Is recent academic progress relevant such that there is a possibility to generalize an IP formulation to solve a hospital's specific problem? The methodology used to answer this question includes a basic integer programming model developed based on initial requirements provided by the hospital. Afterward, managers provided feedback and report back with improvements that should be implemented to obtain a better balance of working hours between the scheduled physicians. The combination of both basic and extended models resulted in a general model that attended all the hospital requirements. Another contribution lies in the fix-and-optimize (F&O) matheuristic that generates good results in acceptable computation time limits. The

F&O matheuristic was necessary to address the most challenging instances when standalone solvers were incapable of generating good results within the imposed time limit.

Computational experiments are conducted using real-world instances provided by HCPA and generated instances. Detailed insights are presented regarding the constraints violation that improved the balance of working hours between the physicians. In addition, to enable an academic validation, both the IP formulation and the F&O matheuristic were compared to the heuristic Late Acceptance Hill Climbing (LAHC) developed by Sanchotene et al. (2018). Logs of the experiments, instances and results are available on-line.[1]

The remainder of the paper is organized as follows. Section 2 presents the literature review. Section 3 details the definition of the studied PRP and compares its constraints against those in the existing literature. Section 4 provides the basic and extended integer programming formulation for the PRP. Section 5 introduces the F&O matheuristic employed to solve the PRP instances. Section 6 details the computational results, while Sect. 7 discusses the conclusions.

2 Literature review

Due to similarities between the PRP and the NRP, this section details related literature regarding both problems. Moreover, methods that were successfully applied to solve instances proposed for the INRC-I and INRC-II are also detailed. Since there is a large body of publications related to the NRP, we limited the scope of the literature to only those solving methods which address the instances from INRC-I and INRC-II.

Sanchotene et al. (2018) proposed a heuristic for the physician rostering problem. The method is divided into two phases. In the first phase, a constructive heuristic is employed to generate a feasible solution. Afterward, the LAHC heuristic is executed to improve the solution. Computational experiments were conducted utilizing the same instances used in this research. Results demonstrated that the LAHC heuristic obtained slightly better results for instances with 150 physicians, while the present research achieved better results for instances with up to 100 physicians.

An integer programming (IP) model was developed by Bruni and Detti (2014) to address a real-world physician rostering problem of an Italian university hospital. The IP model satisfies all service requirements and contractual agreements (including rest periods and annual leave) while trying to respect, as much as possible, employee preferences. Particular attention is paid to workload balancing. Stolletz and Brunner (2012) addressed a PRP with flexible shifts, which have a minimum duration and can begin at any time during the day. Also, the fair distribution of working hours is addressed. The problem was solved using a decomposition heuristic, where the entire problem is broken down into weekly subproblems. Computational experiments demonstrated improved results when compared to previous research (Brunner et al. 2009). Brunner and Edenharter (2011) developed a column generation method to tackle the PRP of an anesthesia department in an 1100-bed hospital. This procedure was necessary because a standalone MIP solver was incapable of solving weekly subproblems to optimality within several hours. A real-world problem in the surgery department of a large government hospital in Singapore was approached by Gunawan and Lau (2013) using a heuristic. Instead of assigning physicians to shifts, they are assigned to a set of tasks incorporating a large number of constraints and complex physician preferences. Constraint programming combined with local search and genetic algorithms was proposed by Rousseau et al. (2002)

[1] http://www.inf.ufrgs.br/~tiwickert/download/2017/physician.

using data provided by two Canadian hospitals. A genetic algorithm to schedule physicians for emergency rooms was proposed by Puente et al. (2009), while a combined emergency and surgery scheduling problem was addressed by Van Huele and Vanhoucke (2014).

Compared to existing literature, the primary differences of this study are the shift regime, which does not follow the same pattern on weekends. Furthermore, a larger number of workload balancing constraints are required to equilibrate overtime, worked day and night shifts as well as worked hours on non-business days.

The following literature addresses relevant solving methods applied to the NRP. The INRC-I winner method proposed by Valouxis et al. (2012) used a two-stage approach to decompose the problem in manageable parts, which can be solved to optimality using a mathematical solver. The second-placed solving method (Burke and Curtois 2011) and third-placed (Bilgin et al. 2010) developed methods based on branch-and-price and hyperheuristics, respectively. After the end of the competition, Santos et al. (2016) developed a MIP model and employed heuristic techniques to decompose the problem. These subproblems are generated by fixing a subset of days or shifts and the resulting subproblem is solved using a MIP solver. Computational results demonstrated that several best-known solutions were improved.

The INRC-II winner method formulated the problem as a network flow model (Römer and Mellouli 2016). The second-placed team modeled the problem utilizing integer programming. Each column of the IP corresponds to a rotation, that is, a sequence of consecutive worked days for a nurse and not a complete individual roster. This procedure is called rotation-based branch-and-price (Legrain et al. 2019).

In both competitions, the best-ranked methods include at least one component based on mathematical models outperforming approaches based only on metaheuristics. After the end of the competition, many methods were applied to the static version of the INRC-II instances. This means that the entire planning horizon (4 or 8 weeks depending on the instance) is solved at once. During the competition it was mandatory to solve each week separately, that is, the solving method must deal with the uncertainty of future data solving sequential weeks and having the complete result only after the last week had been solved. These methods include a Variable Neighborhood Search (VNS) to accelerate the column generation procedure proposed by Gomes et al. (2017), and the same procedure developed by the second-placed (Legrain et al. 2019) but now applied to solve the entire planning horizon at once.

Nurse rostering problems usually include a combination of constraints concerning the minimum and maximum number of consecutive working days, days off and working days on the same shift. These sets of consecutiveness constraints render the problem particularly challenging to solve (Smet 2018). This is not the case for the studied PRP, which only has consecutiveness constraints regarding the maximum number of consecutive working days and working days on the same shift. Despite these restricted number of consecutiveness constraints, the problem is still challenging as detailed in the computational experiments in Sect. 6.

In terms of solving technique, the present fix-and-optimize matheuristic is similar to the method proposed by Santos et al. (2016) and Della Croce and Salassa (2014). The main difference with respect to Santos et al. (2016) is that we decompose the problem into subsets of physicians, days and shifts similar to Della Croce and Salassa (2014) which in our preliminary experiments generated better results compared against using only days and shifts decompositions. The reason for choosing this solving technique in contrast to those proposed by Gomes et al. (2017) and Legrain et al. (2019) is due to good results on the instances proposed for the INRC-I and the natural way to decompose the problem in subsets of physicians, days and shifts. Moreover, as the problem size will increase in a near-future, heuristic methods are more appropriate for these large instances.

3 Problem definition

The general physician rostering problem aims to assign physicians to shifts for each day during a scheduling horizon. The objective is to minimize the cost associated with the violation of the soft constraints such as the maximum number of consecutive working days, overtime and physician's preferences. The case-specific PRP addressed in the present study also includes the concept of locations which means that physicians may be allowed to work at specific locations and not at others within the hospital unit. In addition to the minimization of the overall cost and violation of physician preferences, this study also introduces constraints to generate a fair distribution of working hours between the physicians.

3.1 Basic model

Throughout the model definition, non-business days refer to Saturdays, Sundays and holidays. Working days on weekdays, meanwhile, refer to weekdays which are not holidays. Constraints are either hard (H) or soft (S):

- **H1** A physician can be assigned to at most one shift per day during weekdays;
- **H2** Minimum number of physicians per day/shift/location;
- **H3** Maximum number of physicians per day/shift/location;
- **H4** A physician must be assigned to both Early and Late shifts, or a Night shift, or have a day off on non-business days;
- **H5** Invalid shift succession;
- **H6** A physician can be unavailable for some shifts or days;
- **H7** When working both Early and Late shifts, they must be worked at the same location;
- **H8** A physician must be qualified to work at specific locations;
- **S1** Maximum number of consecutive assignments to the same shift;
- **S2** Maximum number of consecutive assignments;
- **S3** Undesired working day or shift;
- **S4** Complete weekend, that is, a physician ideally works both Saturday and Sunday or none;
- **S5** Minimum number of assignments over the planning horizon (according to the working contract);
- **S6** Maximum number of assignments over the planning horizon (according to the working contract);
- **S7** Maximum number of working weekends.

Constraints included in the basic model are commonly found in the existing literature. For example, Constraints H2, H5 and S1–S7 are present in the instances proposed for the INRC-I and INRC-II. Constraint H8 is present in the INRC-II but a slightly different manner. The difference is that for the INRC-II nurses have specific skills, for example, a nurse can have skills caretaker and trainee, while here physicians are qualified to be assigned to particular locations within a working unit. Another difference is that in the case of the NRP a head nurse can assume the work of less qualified nurses, while this is not the case in the PRP. Constraint H1 is commonly found in the literature for all business and non-business days (Beaulieu et al. 2000; Haspeslagh et al. 2014; Ceschia et al. 2019). However, the present PRP has an exception because this constraint is only valid for business days. During weekends and holidays, working both Early and Late shifts is mandatory. Moreover, these shifts must be worked at the same location.

In comparison with existing PRP literature, Bruni and Detti (2014) described heterogeneous working contracts as well as similar 12-h working shifts. However, these 12-h shifts are valid for the entire week and not only for non-business days. Brunner et al. (2009) and Brunner and Edenharter (2011) contrasting the case of this work where each shift has a fixed start and end times, provided models where shifts can start at any time during a working day and have an arbitrary duration. Another observation is the restricted number of consecutiveness constraints of this model compared for example, to those proposed for the INRC-I and INRC-II. While this model has only two consecutiveness constraints (maximum number of consecutive working days and worked days on the same shift), INRC-II problem has six consecutiveness constraints.

3.2 Extended model

The primary objective of adding this set of constraints is to improve the balance of working hours between physicians. For example, an equal distribution of overtime and working hours on non-business days is desired. The following constraints were incorporated into the basic model:

H9 Minimum number of assignments on non-business days;
H10 Maximum number of assignments on non-business days;
H11 Maximum number of monthly assignments on weekdays;
S8 Preferred number of assignments on non-business days;
S9 Preference for a location;
S10 Equal day and night working hours during weekends;
S11 Maximum weekly assignments during day (Early and Late) or Night shifts;
S12 Assign physicians to the minimum possible number of locations.

Compared to the literature, Salassa and Vanden Berghe (2012) proposed a solving method in which the basic idea is to have a long term balance in terms of workload between employees. Similarly to constraints H9–H11, S8 and S10, Stolletz and Brunner (2012) addressed fairness concerning the distribution of working hours, both in terms of work less-than-contracted hours and overtime by penalizing deviation from a pre-established minimum and maximum working times. In the same way, the model introduced by Stolletz and Brunner (2012) has the possibility of modeling employee-specific preferences or restrictions. This is possible in the presented model through constraints S9 and S3 from the basic model. Constraint S11 concerns a particular case where some physicians have a small number of contracted hours per month. This constraint ensures that such physicians do not work all their contracted hours during the two first weeks of the month.

3.3 Shifts organization and example of a roster

Table 1 provides a simple roster presented in the physician-day view, where rows represent the physicians and columns the days. The example has three physicians (Physician 1, Physician 2 and Physician 3), three shifts (Early [E], Late [L] and Night [N]), and three locations (Inpatient Units [1], [2] and [3]). Day shifts (Early and Late) are 6 h, while Night shifts are 12 h long. The shifts are organized as follows:

Table 1 Example of a roster with 7 days and three physicians

Physician	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Physician 1	L[1]	L[2]	**N[1]**	–	–	**E/L[1]**	**E/L[1]**
Physician 2	**N[2]**	**N[3]**	–	–	–	**N[3]**	**N[2]**
Physician 3	–	E[1]	–	L[2]	**N[1]**	–	–

- Early (6 h): 08–14 h;
- Late (6 h): 14–20 h;
- Night (12 h): 20–08 h.

When a physician works a Night shift, it is considered as two worked shifts. This procedure is necessary to calculate the total number of working shifts during the planning horizon. As an example, Physician 1 works on Monday the Late shift at In-patient Unit 1, on Tuesday on Late shift at In-patient Unit 2, and on Saturday/Sunday on both day shifts (Early and Late) at In-patient Unit 1. Dashes represent days off. Day shifts (Early and Late) on non-business days that must be worked together (enforced by H4) and Nights shifts with 12 working hours, which need to be considered twice in order to calculate the total number of worked shifts per roster, are highlighted in bold.

4 Integer programming formulation

This section introduces the integer programming formulation considering both hard and soft constraints for the PRP. Table 2 presents the indices (first column) used to identify the variables associated with their respective constraints. The second column describes the constraints, the third column (Id) is the identifier, while the last column presents the weight per unit of violation of each constraint. For example, in the objective function in Eq. 1 the first term (considering index 3) $c^3_{nd}\omega^3$ refers to the variables related to the maximum number of consecutive assignments to the same shift (identified by index 3 in Table 2) multiplied by its respective weight ω^3 (15).

Table 3 presents the sets, decision and auxiliary variables employed in the formulation.

Although rosters are typically organized with a planning horizon of 1 month, data from the previous month is essential to avoid infeasible solutions. For example, if physicians work night shifts on the last day of the previous month, they cannot work early or late shifts on the first day of the current month. To avoid such situations, before the solving method starts, the border data from the previous month is read, that is, the total number of assignments, last assigned shift type, number of consecutive assignments of the last shift type, and number of consecutive worked days. This data is necessary correctly evaluate the constraints: invalid shift type succession (H5), maximum number of consecutive assignments to the same shift (S1), maximum number of consecutive working days (S2) and complete weekend (S4).

4.1 Basic model

In this section, we describe the basic model, that is, those constraints that are most likely to be found in many applications.

Table 2 Indices used to associate each soft or hard constraint with their respective variable or input data in the formulation

Index	Constraint description	Id	Weight (ω^i)
Basic model			
1	Minimum number of physician per day/shift/location	H2	–
2	Maximum number of physician per day/shift/location	H3	–
3	Maximum number of consecutive assignments to the same shift	S1	15
4	Maximum number of consecutive assignments (worked days)	S2	30
5	Physician undesired working day/shift	S3	10
6	Complete weekend	S4	30
7	Minimum number of assignments over the planning horizon	S5	20
8	Maximum number of assignments over the planning horizon	S6	20
9	Maximum number of working weekends	S7	30
Extended model			
10	Minimum number of assignments on non-business days	H9	–
11	Maximum number of assignments on non-business days	H10	–
12	Maximum number of assignments on working days	H11	–
13	Number of assignments below the ideal on non-business days	S8	100
14	Number of assignments above the ideal on non-business days	S8	100
15	Priority per location (physicians may express priority for one or more location)	S9	15
16, 17	Equilibrium between day and night working hours during weekends	S10	10
18	Maximum weekly working assignments day (Early and Late) or Night shifts	S11	10
19	Assign physicians to the minimum possible number of locations	S12	50

$$\textbf{Minimize:} \quad \sum_{n\in N}\sum_{d\in D}\sum_{i\in\{3,4\}} c^i_{nd}\omega^i + \sum_{n\in N}\sum_{d\in D}\sum_{s\in S} g^5_{nds}\omega^5 + \sum_{n\in N}\sum_{w\in W} h^6_{nw}\omega^6 + \sum_{n\in N}\sum_{i\in\{7,8,9\}} j^i_n\omega^i \tag{1}$$

Subject to:

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} \le 1 \qquad \forall n \in N, d \in D \setminus \tilde{D} \tag{2}$$

$$\sum_{k\in K}(x_{nds_ek} + x_{nds_lk}) = 2z_{nd} \qquad \forall n \in N, d \in \tilde{D} \tag{3}$$

$$\sum_{k\in K} x_{nds_nk} + z_{nd} \le 1 \qquad \forall n \in N, d \in \tilde{D} \tag{4}$$

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} \le 2o_{nd} \qquad \forall n \in N, d \in D \tag{5}$$

$$\sum_{k\in K}(x_{nds'k} + x_{n(d+1)s''k}) \le 1 \qquad \forall n \in N, d \in \{1,\ldots,|D|-1\}, (s',s'') \in \hat{S} \tag{6}$$

Table 3 Indices, sets and variables used in the mathematical formulation for both the basic and extended models

Symbol	Definition
Input data	
$n \in N$	n is the index of the physician and N is the set of all physicians indices
$d \in D$	d is the index of the day and D is the set of all days indices
$d \in \tilde{D}$	d is the index of the non-business day and $\tilde{D}$ is the set of all non-business days indices
$s \in S$	s is the index of the shift and S is the set of all shifts indices
$k \in K$	k is the index of the location and K is the set of all locations indices
$(n, k) \in L$	Set containing the pairs of forbidden locations of physician n at location k
$(n, k) \in P$	Set containing the pairs where the physician n has a preference not to work at location k
$(n, d, s) \in R$	Set containing triples with unavailable physician n on day d and shift s
$(n, d, s) \in U$	Set containing triples with the undesired working day d and shift s of physician n
$(s', s'') \in \hat{S}$	Set containing the pairs of invalid shift successions
$w \in W$	w is a Saturday index and W the set of all Saturdays indices not including the last Saturday if it is the last day of the month
$w \in \tilde{W}$	w is the week index and $\tilde{W}$ the set of all week indices
α^i_{dsk}	$i \in \{1, 2\}$, that is, the minimum and maximum number of physicians per day d, shift s and location k
β^i_n	Limits of constraints with indices 3, 4, 7, ..., 13 in Table 2. That is, the maximum number of consecutive assignments to the same shift (3), maximum number of consecutive working days (4), minimum/maximum number of assignments over the planning horizon (7, 8), maximum number of working weekends (9), minimum/maximum number of assignments on non-business days (10, 11), maximum number of assignments on working days (12) and ideal number of assignments on non-business days (13)
ω^i_n	Weight for violating the lower and upper bounds of soft constraints i for physician n
s_e	Early shift index
s_l	Late shift index
s_n	Night shift index
Decision variables	
$x_{ndsk} \in \{0, 1\}$	1 if physician n is allocated to shift s, day d, location k and 0 otherwise
$y_{nw} \in \{0, 1\}$	1 if physician n works weekend w and 0 otherwise

Table 3 continued

Symbol	Definition
$z_{nd} \in \{0, 1\}$	1 if physician n works both the Early and Late shifts on day d and 0 otherwise
$o_{nd} \in \{0, 1\}$	1 if physician n is allocated to work on day d and 0 otherwise
$q_{nk} \in \{0, 1\}$	1 if physician n works at location k and 0 otherwise
Auxiliary variables	
$c^i_{nd} \in \mathbb{N}$	Number of violations of the soft constraints with indices $i \in \{3, 4\}$ in Table 2, for physician n on day d
$g^5_{nds} \in \mathbb{N}$	Number of violations of the soft constraints with index 5 in Table 2, for physician n on day d, shift s
$h^6_{nw} \in \mathbb{N}$	Number of violations of the soft constraints with index 6 in Table 2, for physician n on weekend w
$j^i_n \in \mathbb{N}$	Number of violations of the soft constraints with indices $i \in \{7, \ldots, 9, 13, 14, 16, 17, 19\}$ in Table 2, for physician n
$m^{15}_{nk} \in \mathbb{N}$	Number of violations of the soft constraints with indices 15 in Table 2, for physician n at location k
$l^{18}_{nws} \in \mathbb{N}$	Number of violations of the soft constraints with index 18 in Table 2, for physician n on week w

$$\sum_{d\in D}\sum_{s\in S} x_{ndsk} = 0 \qquad \forall (n,k) \in L \tag{7}$$

$$\sum_{k\in K} x_{ndsk} = 0 \qquad \forall (n,d,s) \in R \tag{8}$$

$$x_{nds_ek} - x_{nds_lk} = 0 \qquad \forall n \in N, d \in \tilde{D}, k \in K \tag{9}$$

$$\sum_{n\in N} x_{ndsk} \geq \alpha^1_{dsk} \qquad \forall d \in D, s \in S, k \in K \tag{10}$$

$$\sum_{n\in N} x_{ndsk} \leq \alpha^2_{dsk} \qquad \forall d \in D, s \in S, k \in K \tag{11}$$

$$\sum_{d'=d}^{\beta^3_n+d}\sum_{k\in K} x_{nd's_nk} - c^3_{nd} \leq \beta^3_n \qquad \forall n \in N, d \in \{1, \ldots, |D| - \beta^3_n\} \tag{12}$$

$$\sum_{d'=d}^{\beta^4_n+d} o_{nd'} - c^4_{nd} \leq \beta^4_n \qquad \forall n \in N, d \in \{1, \ldots, |D| - \beta^4_n\} \tag{13}$$

$$\sum_{k\in K} x_{ndsk} \leq g^5_{nds} \qquad \forall (n,d,s) \in U \tag{14}$$

$$o_{nw} + o_{n(w+1)} + h^6_{nw} = 2y_{nw} \qquad \forall n \in N, w \in W \tag{15}$$

$$\sum_{d\in D}\sum_{s\in\{s_e,s_l\}}\sum_{k\in K} x_{ndsk} + \sum_{d\in D}\sum_{k\in K} 2x_{nds_nk} + j_n^7 \geq \beta_n^7 \qquad \forall n \in N \tag{16}$$

$$\sum_{d\in D}\sum_{s\in\{s_e,s_l\}}\sum_{k\in K} x_{ndsk} + \sum_{d\in D}\sum_{k\in K} 2x_{nds_nk} - j_n^8 \leq \beta_n^8 \qquad \forall n \in N \tag{17}$$

$$\sum_{w\in W} y_{nw} - j_n^9 \leq \beta_n^9 \qquad \forall n \in N \tag{18}$$

Constraints (2) ensure a physician is assigned to at most one shift per day on business days. Constraints (3) and (4) ensure that a physician must be assigned to both day shifts, or one Night shift, or no shift on non-business days. Constraints (5) set auxiliary variable o_{nd} to one if physician n works on day d, and zero otherwise. Constraints (6) ensure a shift type succession must be valid (for example, if physicians work Night shifts they cannot be followed by Early or Late shifts on the next day). Constraints (7) ensure a physician is assigned to a location only if allowed. Constraints (8) ensure a physician is scheduled only if he/she is available. Constraints (9) ensure a physician works both shifts (Early and Late) at the same location if they are worked on non-business days. This constraint is to avoid that a physician splits a 12-h working shift across two different locations. Constraints (10) and (11) ensure the minimum and maximum number of physicians per day/shift/location, respectively. Constraints (12) calculate the maximum number of consecutive assignments to Night shifts violations. Constraints (13) calculate the maximum number of consecutive assignments (worked days) violations. Constraints (14) calculate the undesired worked day or shift violations. Constraints (15) calculate the complete weekend violations. Constraints (16) and (17) calculate the minimum and maximum number of worked shifts violations over the scheduling period, respectively. Constraints (18) calculate the maximum number of working weekends violations.

4.2 Extended model

In this section the extended model is described, that is, constraints that aim a fair distribution of the working hours between the physicians.

$$\textbf{Minimize:} \quad \sum_{n\in N}\sum_{w\in\tilde{W}}\sum_{s\in S} l_{nws}^{18}\omega^{18} + \sum_{n\in N}\sum_{k\in K} m_{nk}^{15} + \sum_{n\in N}\sum_{i\in\{13,14,16,17,19\}} j_n^i\omega^i + (1) \tag{19}$$

Subject to:

$$\beta_n^{10} \leq \sum_{d\in\tilde{D}} o_{nd} \leq \beta_n^{11} \qquad \forall n \in N \tag{20}$$

$$\sum_{d\in D\setminus\tilde{D}} o_{nd} \leq \beta_n^{12} \qquad \forall n \in N \tag{21}$$

$$\sum_{d\in\tilde{D}} o_{nd} + j_n^{13} - j_n^{14} = \beta_n^{13} \qquad \forall n \in N \tag{22}$$

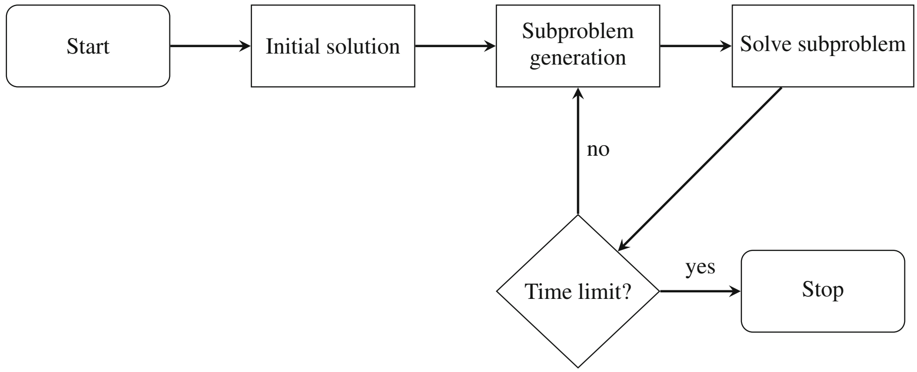

Fig. 1 Algorithm execution flowchart

$$\sum_{d\in D}\sum_{s\in S} x_{ndsk} - m^{15}_{nk} = 0 \qquad \forall (n,k) \in P \tag{23}$$

$$\sum_{d\in \tilde{D}}\sum_{k\in K} \left(x_{nds_ek} + x_{n(d+1)s_ek}\right) - \sum_{d\in \tilde{D}}\sum_{k\in K} \left(x_{nds_nk} + x_{n(d+1)s_nk}\right) - j^{16}_n + j^{17}_n = 0 \qquad \forall n \in N \tag{24}$$

$$\sum_{d\in\{1,\ldots,7\}}\sum_{k\in K} x_{n(d\times w)sk} - l^{18}_{nws} \le \gamma^{18}_{nws} \qquad \forall n \in N, w \in \tilde{W}, s \in S \tag{25}$$

$$x_{ndsk} \le q_{nk} \qquad \forall n \in N, d \in D, s \in S, k \in K \tag{26}$$

$$\sum_{k\in K} q_{nk} - j^{19}_n \le 1 \qquad \forall n \in N \tag{27}$$

Constraints (20) ensure the minimum and maximum number of assignments on non-business days for each physician. Constraints (21) ensure the maximum number of assignments on business days for each physician. Constraints (22) penalize the difference between the ideal and actual number of assignments on non-business days. Constraints (23) penalize physicians working out of the preferred location. Constraints (24) penalize the difference between the number of assignments of day and night shifts on non-business days. Constraints (25) penalize weekly allocations in excess of the maximum. Constraints (26) and (27) calculate the number of distinct locations that a physician works and penalize if this value is greater than one.

5 Fix-and-optimize matheuristic

This section provides the F&O matheuristic developed to approach the PRP. The proposed algorithm was adapted from a previous version to address the NRP (Wickert et al. 2016). Figure 1 provides an overview of the algorithm execution flow. The algorithm begins generating a feasible solution using a MIP solver only considering the hard constraints. Afterward, a subset of variables is iteratively fixed to their current values, decomposing the problem into subproblems, which are then successively solved using a MIP solver until the computation time limit is reached. All hard and soft constraints are considered when the subproblem is solved. The algorithm returns the best solution found.

Before fully explaining the algorithm, the following general terminology is introduced. Variables or physicians are denoted as *free* when the associated decision variable has the lower bound set to zero and the upper bound to one. This consequently implies that the solver can decide on setting the value either to zero or to one. On the other hand, *fixing* a day, physician or shift indicates that the decision variable is set to the corresponding value in the incumbent solution. In this case, the solver cannot change the variable's value. *PhysicianFreeCombinationSet* is a set of combinations without repetition.

For example, when there are $n = 5$ physicians and the parameter *kPhysician = 2*, the combinations without repetitions is 10 unique possibilities, resulting in the set *PhysicianFreeCombinationSet([1,2], [1,3], [1,4], [1,5], [2,3], [2,4], [2,5], [3,4], [3,5], [4,5])*. If *kLimitPhysician* = 5, only 5 out of 10 items will be randomly added to the set *PhysicianFreeCombinationSet*. If the *kLimitPhysician* $\geq$ *10* all possible combinations will be added to the set *PhysicianFreeCombinationSet*. During the algorithm's execution, the decision variable will have physicians 1 and 2 with lower bound zero and upper bound one, while physicians 3–10 will have their lower and upper bounds fixed to their current values. As such, the solver can only change the value of the decision variables of physicians 1 and 2.

Algorithm 1 provides the pseudo-code. The input parameters *kMaxDay*, *kMaxPhysician*, *kMaxWeek* and *kMaxShift* represent the maximum number of free variables of each type. Meanwhile, *kLimitDay*, *kLimitPhysician*, *kLimitWeek* and *kLimitShift* are the limits of combinations generated for each type of neighborhood. Algorithm 1 begins by generating an initial feasible solution *x* (line 2) considering only the hard constraints using a MIP solver. The number of free variables to optimize is initialized with one (line 3), and the loop (lines 4 to 17) is iterated until the time limit (TL) is reached.

Algorithm 1: Fix-and-optimize matheuristic algorithm.

```
1  FixAndOptimize(kMaxWeek, kLimitWeek, kMaxShift, kLimitShift, kMaxDay, kLimitDay, kMaxPhysician,
   kLimitPhysician, TL, STL)
2  x = generateInitialSolution()
3  kWeek = kShift = kDay = kPhysician = 1
4  do
5      x = FixPerDay(x, kDay, kLimitDay, STL)
6      kDay = kDay+1
7      x = FixPerPhysician(x, kPhysician, kLimitPhysician, STL)
8      kPhysician=kPhysician+1
9      x = FixPerWeek(x, kWeek, kLimitWeek, STL)
10     kWeek=kWeek+1
11     x = FixPerShift(x, kShift, kLimitShift, STL)
12     kShift=kShift+1
13     if (kDay > kMaxDay) kDay = 1
14     if (kPhysician > kMaxPhysician) kPhysician = 1
15     if (kWeek > kMaxWeek) kWeek = 1
16     if (kShift > kMaxShift) kShift = 1
17 while TL not reached;
18 return x
```

Inside the loop (lines 5 to 16), the algorithm executes different methods representing the neighborhoods. Each neighborhood is explored until either a local minimum is found or it reaches the STL (subproblem time limit). For each step, the value of *kDay*, *kPhysician*, *kWeek* and *kShift* is incremented. If the limit of each type is exceeded, variables are reset to 1 (lines 13 to 16).

Algorithm 2 begins generating the combination of physicians that will be free to be optimized until the *kLimitPhysician* is reached (the *combination* function at line 2). The loop (lines 4 to 16) is iterated while the best neighbor solution is at least 20% better than the current solution. The loop begins by storing the current solution value and the best neighbor value (function *OFV* lines 5 and 6). The nested loop (lines 7 to 14) explores the neighborhood by fixing the entire problem (line 8), and unfixing only the free variables that will be optimized (line 9). The MIP solver is executed until either the optimal solution is found or STL is reached (line 10). If the Objective Function Value (OFV) of subproblem x is lower than the OFV of the best neighbor (line 11), then the *bestNeighborValue* variable is updated accordingly (line 12).

Algorithm 2: Fix per physician.

```
1  FixPerPhysician(x, kPhysician, kLimitPhysician, STL)
2  physicianFreeCombinationSet = combination(kPhysician, kLimitPhysician)
3  improved = false
4  do
5      currentSolutionValue = OFV(x)
6      bestNeighborValue = OFV(x)
7      foreach Integer free : physicianFreeCombinationSet do
8          fixAll(x)
9          unFix(free, x)
10         solve(x, STL)
11         if OFV(x) < bestNeighborValue then
12             bestNeighborValue = OFV(x)
13         end
14     end
15     improved = bestNeighborValue*1.2 < currentSolutionValue
16 while improved;
17 return x
```

Figures 2 and 3 detail an iteration of a *fix per physician* neighborhood with *kPhysician* = 1 and *kPhysician* = 2, respectively. Rows with a gray background are available to be optimized by the solver, meaning that the decision variables have lower bounds set to zero and upper bounds of one. By contrast, rows with a white background have the associated decision variable bounds fixed to the current incumbent value, forbidden the solver to change these values. Observe that, when a decision variable has both upper and lower bounds set to the same value, they are ignored by the MIP solver. Since the *fix per week*, *fix per shift* and *fix per day* decompositions follow the same idea, the pseudo-code of these algorithms has been omitted for textual economy.

6 Computational experiments

This section analyzes a series of computational experiments to investigate whether the proposed IP formulation can be solved using commercial and open-source MIP solvers for both small and large instances.

6.1 Data sets and experimental setup

The source code was written in Java and compiled with OpenJDK 1.8. The experiments were conducted on an Intel Core i5-2410M CPU 2.30 GHz (2 cores) with 6GB of RAM memory

Fig. 2 Fix per physician (kPhysician = 1)

Physician	Mon	Tue	Wed
P1	L[1]	L[2]	N[2]
P2	N[1]	N[1]	N[3]
P3	–	E[3]	–

↓

P1	L[1]	N[2]	N[2]
P2	N[1]	N[1]	N[3]
P3	–	E[3]	–

↓

P1	L[1]	N[2]	N[2]
P2	N[2]	N[1]	N[3]
P3	–	E[3]	–

Fig. 3 Fix per physician (kPhysician = 2)

Physician	Mon	Tue	Wed
P1	L[1]	N[2]	N[2]
P2	N[2]	N[1]	N[3]
P3	–	E[3]	–

↓

P1	L[1]	N[1]	N[2]
P2	N[2]	L[1]	N[3]
P3	–	E[3]	–

↓

P1	L[1]	N[1]	N[2]
P2	E[2]	L[1]	N[3]
P3	E[3]	E[3]	–

running Linux Mint 17.2 64-bits. The solvers employed were CPLEX version 12.6.2 and Coin-OR CBC version 2.9.9. Both solvers were run with default parameters. The *gap* is calculated using the equation $gap = 100 \times \frac{OFV-LB}{LB}$, where OFV is the objective function value and LB is the lower bound. All parameters of the F&O matheuristic were tuned using irace (López-Ibáñez et al. 2016).

Table 4 presents the parameter names, tested ranges and the chosen parameter values computed by irace, respectively. Irace reported the values *STL = 8s, kMaxWeek = 2, kLimitWeek = 4, kMaxShift = 3, kLimitShift = 3, kMaxPhysician = 20, kLimitPhysician = 30, kMaxDay = 8, and kLimitDay = 8* as the best parameter values to be used by the proposed F&O matheuristic. The default irace parameters were used for the experiments, that is, the confidence level is 0.95.

The dataset employed in the experiments was generated based on the information provided by HCPA. The algorithm was tested in 30 generated instances. Currently, the real number of physicians to schedule is 50. However, the number of physicians will increase in a near future, and so this is the reasoning behind generating larger instances. The objective is to analyze whether these larger and more demanding instances can still be solved using the proposed methods. The following instances were generated:

Table 4 Tested parameter ranges

Name	Range of values	Irace
STL	[5, 8, 12]	8
kMaxWeek	[1, …, 4]	2
kLimitWeek	[1, …, 8]	4
kMaxShift	[1, …, 3]	3
kLimitShift	[1, …, 7]	3
kMaxPhysician	[1, 5, 10, 15, 20, 25, 30]	20
kLimitPhysician	[5, 10, 15, 30, 35, 40, 45, 50]	30
kMaxDay	[1, 8, 16]	8
kLimitDay	[4, 8, 16, 32]	8

- 10 instances with 50 physicians and four weeks;
- 10 instances with 100 physicians and four weeks;
- 10 instances with 150 physicians and four weeks.

The computation time limit for each experiment was fixed according to the instance size and algorithm. Note that the F&O matheuristic has half of the computational time limit compared against the MIP solvers. This solving method is used when results are required in reduced computational time. The following experimental setup was proposed:

- One single execution using CPLEX solver and a computation time limit of 12 h;
- One single execution using CPLEX and Coin-OR CBC with computation time limits of 20, 40 and 60 min for the instances with 50, 100 and 150 physicians, respectively;
- 10 executions using the F&O matheuristic with computation time limits of 10, 20 and 30 min for the instances with 50, 100 and 150 physicians, respectively.

In addition, two instances using data from April and May of 2019 provided by HCPA were employed for the computational experiments using the extended model. The reported results and statistics constitute the real roster used in the hospital.

6.2 MIP solver results

Table 5 provides the results when the basic IP formulation is solved using the standalone MIP solvers CPLEX and Coin-OR CBC. The column labels LB, OFV and Gap correspond to the lower bound (provided by CPLEX), objective function value and the gap calculated according to the equation introduced in the previous section. Experiments are split into three blocks. The first block provides the results when CPLEX was executed for a long computation time (12 h). The objective of this experiment was to generate good LBs to use them for comparing the other solving methods. In practice, this time limit is not considered acceptable. Therefore, other experiments have shorter computation time limits.

The second and third blocks present the results when CPLEX and Coin-OR CBC were employed to solve the IP formulation using computation time limits of 20, 40 and 60 min for instances with 50, 100 and 150 physicians, respectively. Instances containing 50 physicians were solved to optimality employing CPLEX and near-optimality using Coin-OR CBC. For larger instances, with 100 and 150 physicians, Coin-OR was not capable of generating feasible solutions within the time limit, while CPLEX generated 6 out of 10 feasible solutions when instances with 100 physicians are tackled. Both solvers could not find feasible solutions for instances with 150 physicians within 1h.

Table 5 MIP solver results

Instance	CPLEX						Coin-OR CBC		
	Time limit—12 h			Time limit—20, 40, 60 min[a]			Time limit—20, 40, 60 min[a]		
	LB	OFV	Gap (%)	OFV	Gap (%)	Time (s)	OFV	Gap (%)	Time (s)
p050_inst_01	30, 305	30, 305	0.00	30,305	0.00	438	30,305	0.00	754
p050_inst_02	30, 460	30, 460	0.00	30,460	0.00	352	30,460	0.00	520
p050_inst_03	30, 505	30, 505	0.00	30,505	0.00	428	30,505	0.00	1185
p050_inst_04	30, 965	30, 965	0.00	30,965	0.00	511	30,965	0.00	283
p050_inst_05	30, 685	30, 685	0.00	30,685	0.00	374	30,685	0.00	435
p050_inst_06	31, 705	31, 705	0.00	31,705	0.00	415	31,705	0.00	301
p050_inst_07	30, 015	30, 015	0.00	30,015	0.00	401	30,015	0.00	315
p050_inst_08	30, 215	30, 215	0.00	30,215	0.00	403	30,225	0.03	1200
p050_inst_09	31, 670	31, 670	0.00	31,670	0.00	447	31,670	0.00	484
p050_inst_10	30, 765	30, 765	0.00	30,765	0.00	408	30,765	0.00	1182
Average			0.00		0.00			0.0003	
p100_inst_01	24, 429	25, 525	4.49	26,320	7.74	2400	–	–	2400
p100_inst_02	26, 720	27, 945	4.58	29,940	12.05	2400	–	–	2400
p100_inst_03	25, 082	26, 300	4.86	–	–	2400	–	–	2400
p100_inst_04	24, 280	25, 285	4.14	–	–	2400	–	–	2400
p100_inst_05	24, 633	25, 775	4.64	28,615	16.17	2400	–	–	2400
p100_inst_06	25, 660	26, 920	4.91	29,490	14.93	2400	–	–	2400
p100_inst_07	23, 205	24, 505	5.60	26,180	12.82	2400	–	–	2400
p100_inst_08	25, 282	26, 445	4.60	–	–	2400	–	–	2400
p100_inst_09	25, 946	27, 130	4.56	–	–	2400	–	–	2400
p100_inst_10	23, 775	25, 030	5.28	28,185	18.55	2400	–	–	2400
Average			4.77		–				

Table 5 continued

Instance	CPLEX						Coin-OR CBC		
	Time limit—12 h			Time limit—20, 40, 60 min[a]			Time limit—20, 40, 60 min[a]		
	LB	OFV	Gap (%)	OFV	Gap (%)	Time (s)	OFV	Gap (%)	Time (s)
p150_inst_01	55, 742	60, 030	7.69	–	–	3600	–	–	3600
p150_inst_02	54, 011	58, 320	7.98	–	–	3600	–	–	3600
p150_inst_03	54, 135	58, 070	7.27	–	–	3600	–	–	3600
p150_inst_04	53, 457	57, 595	7.74	–	–	3600	–	–	3600
p150_inst_05	54, 786	59, 740	9.04	–	–	3600	–	–	3600
p150_inst_06	54, 170	58, 720	8.40	–	–	3600	–	–	3600
p150_inst_07	53, 959	57, 570	6.69	–	–	3600	–	–	3600
p150_inst_08	53, 199	56, 750	6.67	–	–	3600	–	–	3600
p150_inst_09	53, 396	57, 580	7.84	–	–	3600	–	–	3600
p150_inst_10	51, 782	55, 810	7.78	–	–	3600	–	–	3600
Average			7.71		–				

[a]Computation time limits for instances with 50, 100 and 150 physicians, respectively

These results show that MIP solvers are a good option for solving instances of up to 50 physicians, which is the hospital's current situation. Both CPLEX and the open-source solver Coin-OR CBC generated good results. Optimal and near-optimal results were obtained within acceptable computation time limits. However, for large instances with 100 and 150 physicians, improvements in the IP formulation or other solving methods are necessary to generate better results.

6.3 Fix-and-optimize results

Table 6 presents the results using the fix-and-optimize (F&O) matheuristic. The three last columns provide CPLEX results. However, a direct comparison with heuristics is not possible since MIP solvers address problems improving both upper and lower bounds aiming to prove optimality. The gap is calculated relative to the LB (second column), obtained by CPLEX when executed with a time limit of 12h. Values in bold represent the best results.

As explained in Sect. 5, the F&O matheuristic uses CPLEX to solve the subproblems and results are compared to the Late Acceptance Hill Climbing (LAHC) developed by Sanchotene et al. (2018). Computational results demonstrated that for instances with 50 physicians, results were very close to optimality with an average relative gap of 0.03%. Instances with 100 and 150 physicians have average relative gaps of 6.00% and 8.75%, respectively.

In general, the F&O matheuristic generated similar results as the LAHC heuristic developed by Sanchotene et al. (2018). These computational experiments show that both the LAHC heuristic and the F&O matheuristic are good alternatives to address large problems with 100 and 150 physicians. Both methods generated good results within short computation times. By contrast, when solving problems with up to 50 physicians, the standalone MIP solvers are the most effective alternative. Note that the proposed F&O matheuristic is solver-independent and general, being capable of addressing both the basic and extended models without compromising the quality of the results compared to the case-specific LAHC heuristic.

6.4 Objective function analysis

Figure 4 provides an analysis of the OFV when varying the number of physicians and removing different subsets of constraints. The objective is to evaluate the influence of the different sets of constraints upon the OFV. Black bars indicate the results when all constraints are considered when varying the number of physicians from 50 to 90. With 60 physicians, the OFV reduced approximately one third and eventually reached an ideal of zero when the number of available physicians is 90. Moreover, results indicate a notable reduction in the OFV when S1 and S2 are removed considering 50 physicians. Experiments show that these two constraints represent a minor influence on the OFV when the number of physicians is 60 or more. Such results indicate that the majority of the violations concern overtime constraints.

6.5 Extended model results

This section provides the results when using the extended model for the generated roster of April and May of 2019. Since a CPLEX license is required for commercial use, only Coin-OR CBC was used for the computational experiments in this section. Table 7 provides the results employing Coin-OR CBC standalone solver (second column) and using the

Table 6 Heuristic LAHC and F&O results

Instance	LB	LAHC—10, 20, 30 min[a]		F&O—10, 20, 30 min[a]			CPLEX—20, 40, 60 min[a]		
		OFV	Gap (%)	OFV	SD	Gap (%)	OFV	Gap (%)	Time (s)
p050_inst_01	30,305	30,329	0.08	30,318	± 7	0.04	30,305	**0.00**	438
p050_inst_02	30,460	30,480	0.07	30,468	± 8	0.03	30,460	**0.00**	352
p050_inst_03	30,505	30,517	0.04	30,516	± 12	0.04	30,505	**0.00**	428
p050_inst_04	30,965	30,977	0.04	30,975	± 8	0.03	30,965	**0.00**	511
p050_inst_05	30,685	30,695	0.03	30,692	± 11	0.02	30,685	**0.00**	374
p050_inst_06	31,705	31,737	0.10	31,719	± 7	0.04	31,705	**0.00**	415
p050_inst_07	30,015	30,051	0.12	30,022	± 7	0.02	30,015	**0.00**	401
p050_inst_08	30,215	30,236	0.07	30,226	± 10	0.04	30,215	**0.00**	403
p050_inst_09	31,670	31,714	0.14	31,685	± 12	0.05	31,670	**0.00**	447
p050_inst_10	30,765	30,783	0.06	30,772	± 8	0.02	30,765	**0.00**	408
Average			0.07			0.03		**0.00**	
p100_inst_01	24,429	25,979	6.34	25,835	± 66	**5.76**	26,320	7.74	2400
p100_inst_02	26,720	28,479	6.58	28,324	± 55	**6.00**	29,940	12.05	2400
p100_inst_03	25,082	26,659	6.29	26,608	± 95	**6.08**	–	–	2400
p100_inst_04	24,280	25,764	6.11	25,558	± 78	**5.26**	–	–	2400
p100_inst_05	24,633	26,144	6.13	26,034	± 50	**5.69**	28,615	16.17	2400
p100_inst_06	25,660	27,391	6.75	27,253	± 45	**6.21**	29,490	14.93	2400
p100_inst_07	23,205	25,008	7.77	24,801	± 66	**6.88**	26,180	12.82	2400
p100_inst_08	25,282	26,867	6.27	26,754	± 89	**5.82**	–	–	2400
p100_inst_09	25,946	27,592	6.34	27,333	± 58	**5.35**	–	–	2400
p100_inst_10	23,775	25,534	7.40	25,438	± 89	**6.99**	28,185	18.55	2400
Average			6.60			**6.00**		–	

Table 6 continued

Instance	LB	LAHC—10, 20, 30 min[a]		F&O—10, 20, 30 min[a]			CPLEX—20, 40, 60 min[a]		
		OFV	Gap (%)	OFV	SD	Gap (%)	OFV	Gap (%)	Time (s)
p150_inst_01	55,742	60,207	**8.01**	60,517	± 123	8.57	–	–	3600
p150_inst_02	54,011	58,691	**8.66**	58,950	± 154	9.14	–	–	3600
p150_inst_03	54,135	58,539	**8.14**	58,875	± 95	8.75	–	–	3600
p150_inst_04	53,457	57,842	**8.20**	58,133	± 139	8.75	–	–	3600
p150_inst_05	54,786	59,375	**8.38**	59,640	± 122	8.86	–	–	3600
p150_inst_06	54,170	58,973	**8.87**	59,141	± 56	9.18	–	–	3600
p150_inst_07	53,959	58,119	**7.71**	58,443	± 115	8.31	–	–	3600
p150_inst_08	53,199	57,378	**7.86**	57,733	± 199	8.52	–	–	3600
p150_inst_09	53,396	57,673	**8.01**	58,029	± 126	8.68	–	–	3600
p150_inst_10	51,782	55,990	**8.13**	56,311	± 100	8.75	–	–	3600
Average			**8.20**			8.75		–	

[a]Computation time limits for instances with 50, 100 and 150 physicians, respectively

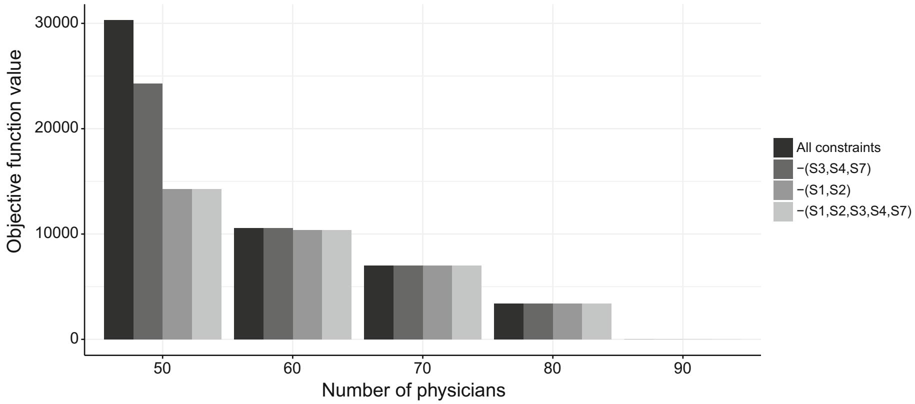

Fig. 4 OFV impact when varying the number of physicians and removing constraints

Table 7 Results using Coin-OR CBC standalone solver and the F&O matheuristic

	20 min Coin-OR CBC Gap (%)	10 min F&O Gap (%)
April 2019	0.05	1.83
May 2019	0.09	2.46

F&O matheuristic (third column). The objective of this experiment is not a direct comparison between the exact and heuristic method. Instead, they are complementary, being the F&O matheuristic employed mainly when fast results must be available in reduced time. Results show that the standalone solver was capable of reaching near-optimum results with a relative gap of less than 0.1% within 20 min. Another experiment, using the F&O matheuristic with a 10 min time limit, demonstrated that the algorithm is a good alternative when results are needed quickly. The gaps (last column), calculated relative to the lower bound provided by the MIP solver, were 1.83% and 2.46% for the months of April and May, respectively.

Note that both the F&O matheuristic and the standalone MIP solver are used in practice at HCPA. The F&O matheuristic method is employed to generate fast solutions when the roster of a new month is going to be organized. During this process, managers may change the input data, including days off requests, vacations, constraints violation weight, and recompute the roster several times. When this process is more stable, the exact method is executed for longer runtimes to generate near-optimum final rosters.

Tables 8 and 9 provide the most relevant constraint violation analysis concerning the rosters generated for April and May of 2019. The first column represents the physician identification, while the S1 column provides the maximum of two consecutive night shift violations. Column S4 details the number of incomplete worked weekends, column S5/S6 provides the contracted hours (in parentheses), where positive and negative values indicate whether the respective physician worked more or less than their contracted hours. Observe that Early and Late shifts have 6 h and Night shifts 12 h. However, the majority of the physicians' contracts are not a multiple of six, and therefore it is technically impossible for most physicians to work their precise contractual hours. Column S7 presents the maximum number of worked weekends in parentheses and the number effectively worked. Column

Table 8 April 2019—Roster analysis

Name	S1	S4	S5/S6	S7	S8	S10	S12
P1	0	0	(100) + 8	(2) 1	(24) 0	0	[0, 10, 0, 0, 0, 0]
P2	0	1	(100) + 2	(2) 2	(36) 0	12	[12, 0, 0, 0, 0, 0]
P3	0	2	(100) + 2	(2) 2	(36) 0	0	[15, 0, 0, 0, 0, 0]
P4	0	1	(100) + 2	(2) 2	(36) 0	12	[12, 0, 0, 0, 0, 0]
P5	0	2	(100) + 2	(2) 2	(36) 0	0	[0, 12, 0, 0, 0, 0]
P6	0	1	(115) + 5	(2) 2	(36) 0	− 12	[0, 14, 0, 0, 0, 0]
P7	0	0	(72) 0	(2) 1	(24) 0	0	[0, 8, 0, 0, 0, 0]
P8	0	0	(60) 0	(2) 1	(24) 0	0	[0, 0, 0, 0, 6, 0]
P9	0	0	(83) + 1	(2) 1	(24) 0	0	[0, 10, 0, 0, 0, 0]
P10	0	2	(100) + 8	(2) 2	(36) 0	0	[0, 11, 0, 0, 0, 0]
P11	0	1	(100) + 2	(2) 2	(36) 0	− 12	[11, 0, 0, 0, 0, 0]
P12	0	1	(100) + 2	(2) 2	(36) 0	12	[12, 0, 0, 0, 0, 0]
P13	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 12, 0, 0, 0, 0]
P14	0	0	(48) 0	(2) 1	(24) 0	0	[0, 0, 0, 0, 5, 0]
P15	0	2	(100) + 2	(2) 2	(36) 0	0	[12, 0, 0, 0, 0, 0]
P16	0	0	(125) + 1	(2) 2	(48) 0	0	[0, 0, 14, 0, 0, 0]
P17	1	1	(68) + 4	(2) 2	(36) 0	− 12	[7, 1, 0, 0, 0, 0]
P18	0	2	(100) + 2	(2) 2	(36) 0	0	[11, 0, 0, 0, 0, 0]
P19	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 0, 12, 0, 0, 0]
P20	0	2	(100) + 8	(2) 2	(36) 0	0	[0, 0, 13, 0, 0, 0]
P21	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 0, 0, 13, 0, 0]
P22	0	2	(150) 0	(2) 2	(36) 0	0	[7, 0, 0, 13, 0, 0]
P23	0	1	(100) + 2	(2) 2	(36) 0	− 12	[0, 0, 0, 12, 0, 0]
P24	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 0, 12, 0, 0, 0]
P25	0	0	(125) − 11	(2) 2	(48) 0	0	[13, 0, 0, 0, 0, 0]
P26	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 0, 0, 0, 12, 0]
P27	0	2	(100) + 2	(2) 2	(36) 0	0	[0, 0, 0, 12, 0, 0]
P28	0	2	(100) + 2	(2) 2	(36) − 12	0	[0, 0, 0, 14, 0, 0]
P29	0	1	(100) + 8	(2) 2	(36) 0	− 12	[10, 0, 0, 0, 0, 0]
P30	0	2	(100) − 4	(2) 2	(36) 0	0	[0, 11, 0, 0, 0, 0]
P31	0	1	(100) + 2	(2) 2	(36) 0	− 12	[0, 7, 5, 0, 0, 0]
P32	0	2	(100) − 4	(2) 2	(36) 0	0	[0, 0, 11, 0, 0, 0]
P33	0	1	(65) + 1	(2) 1	(24) 0	12	[0, 0, 1, 0, 7, 0]
P34	0	2	(100) − 4	(2) 2	(36) 0	0	[0, 0, 0, 0, 12, 0]
P35	0	2	(125) − 5	(2) 2	(36) 0	0	[0, 0, 0, 0, 0, 16]
P36	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 0, 0, 12, 0, 0]
P37	0	1	(125) + 7	(2) 2	(36) 0	− 12	[0, 0, 0, 0, 11, 5]
P38	1	1	(100) − 4	(2) 2	(36) 0	− 12	[0, 0, 0, 0, 11, 0]
P39	0	2	(125) − 11	(3) 2	(36) 0	0	[0, 0, 0, 0, 0, 12]
P40	0	0	(100) + 2	(2) 1	(36) − 12	0	[0, 0, 0, 14, 0, 0]

Table 8 continued

Name	S1	S4	S5/S6	S7	S8	S10	S12
P41	0	1	(100) + 2	(2) 2	(36) 0	– 12	[0, 7, 0, 0, 5, 0]
P42	0	1	(100) + 2	(2) 2	(36) 0	12	[3, 10, 0, 0, 0, 0]
P43	0	1	(100) + 2	(2) 2	(36) 0	– 12	[0, 5, 7, 0, 0, 0]
P44	0	1	(100) + 2	(2) 2	(36) 0	12	[0, 0, 0, 0, 0, 13]
P45	0	1	(100) + 2	(2) 2	(36) 0	– 12	[3, 9, 0, 0, 0, 0]
P46	0	1	(125) – 5	(2) 2	(36) 0	– 12	[0, 0, 0, 0, 0, 15]
P47	0	1	(125) – 5	(2) 2	(36) 0	– 12	[0, 0, 0, 0, 0, 15]
P48	0	1	(125) – 11	(2) 2	(36) 0	12	[0, 0, 0, 0, 0, 14]

Overtime: 99 h; debt: 64 h; difference: + 35 h

S8 provides the ideal number of worked hours on non-business days in parentheses, where positive and negative values indicate if the physician worked more or less than the ideal. These ideal hours vary from one physician to another, depending on the total number of working hours and the seniority of the contract. Column S10 presents the difference between worked day and night shifts (day–night) for which the result should, ideally, be zero. Finally, column S12 details the number of times a physician was allocated for each area. For example, Physician 1 was scheduled to work ten times at Area2 and zero times in other areas (in the example there are six areas).

From Table 8, it can be observed that there are a low number of violations concerning constraint maximum number of consecutive worked night shifts (S1). The maximum number of worked weekends (S7), which for the majority of physicians was limited to two, had no violations. The same tendency is observed for ideal working hours on non-business days (S8), which was always equal or below the stipulated value. Although incomplete worked weekends (S4) presented a few violations, they were still within an acceptable range. Constraint S10 only presented multiples of 12 violations because this is the minimum number of working hours on non-business days. This constraint presented more violations, but it is acceptable given the reduced number of assigned areas (S12), which is desirable.

Table 8, which represents April of 2019, was a month where almost all physicians were available the entire month contributing to less overtime. Because of that, some physicians worked fewer hours than their contract stipulates. Before the solver was implemented, manual rosters were deployed to the physicians with an average of 400 overtime hours.

Table 9, which corresponds to May of 2019, had more physicians on vacation compared to Table 8 (April of 2019). Overtime was also more prevalent (+ 231), however, still far below the average (+ 400) when the roster was organized manually. Another consequence is the number of working hours on non-business days (S8) and the day and night shift balance (S10), which presented more violations compared to the previous month.

Managers considered the presented results much better than those generated manually. The primary reasons being the reduction of overtime, better distribution of overtime and working hours on non-business days between physicians. Another important remark is the reduction of mistakes in the roster. For example, scheduling a physician for a Night shift on the last day of the previous month and an Early or Late shift the first day of the next month, was a common error when the rosters were organized manually. Moreover, the development of the overview tables improved the transparency of how the rosters are prepared and organized.

Table 9 May 2019—Roster analysis

Name	S1	S4	S5/S6	S7	S8	S10	S12
P1	0	1	(104) + 4	(2) 2	(36) 0	− 12	[0, 10, 0, 0, 0, 0]
P2	0	2	(52) + 8	(2) 2	(24) 0	0	[6, 0, 0, 0, 0, 0]
P3	0	2	(104) + 4	(2) 2	(36) 0	0	[7, 0, 0, 0, 9, 0]
P4	0	1	(72) 0	(2) 2	(24) + 12	− 12	[7, 0, 0, 0, 0, 0]
P5	0	1	(52) + 2	(2) 1	(24) 0	12	[0, 7, 0, 0, 0, 0]
P6	0	0	(130) + 8	(2) 1	(36) 0	24	[0, 13, 0, 6, 0, 0]
P7	0	2	(104) + 4	(2) 2	(36) 0	0	[0, 10, 0, 2, 0, 0]
P8	0	0	(60) 0	(2) 1	(24) 0	0	[0, 0, 0, 0, 6, 0]
P9	0	0	(86) + 10	(2) 1	(24) 0	0	[0, 10, 0, 0, 0, 0]
P10	0	1	(104) + 4	(2) 2	(36) 0	12	[0, 11, 0, 0, 0, 0]
P11	0	0	(52) + 2	(2) 1	(24) 0	0	[6, 0, 0, 0, 0, 0]
P12	0	2	(104) + 4	(2) 2	(36) 0	0	[9, 0, 4, 0, 0, 0]
P13	0	1	(104) + 4	(2) 2	(36) 0	− 12	[0, 9, 0, 3, 0, 0]
P14	0	1	(30) + 6	(2) 1	(12) 0	12	[0, 0, 0, 0, 4, 0]
P15	0	0	(104) + 4	(2) 1	(36) 0	0	[10, 3, 0, 0, 0, 0]
P16	0	0	(130) + 2	(2) 2	(48) 0	0	[0, 0, 16, 0, 0, 0]
P17	0	1	(104) + 4	(2) 2	(36) 0	− 12	[10, 0, 0, 0, 0, 0]
P18	0	1	(104) + 10	(2) 2	(36) 0	− 12	[9, 0, 0, 0, 3, 0]
P19	0	2	(116) + 10	(2) 2	(36) 0	0	[0, 0, 14, 0, 0, 0]
P20	0	0	(104) + 4	(2) 2	(36) + 12	0	[0, 0, 12, 0, 0, 0]
P21	0	0	(104) + 4	(2) 1	(36) 0	0	[0, 0, 0, 12, 0, 0]
P22	0	2	(156) + 6	(2) 2	(36) 0	0	[5, 0, 0, 13, 3, 0]
P23	0	2	(88) + 2	(2) 2	(24) 0	0	[0, 0, 0, 10, 0, 0]
P24	0	0	(104) + 4	(2) 2	(36) + 12	0	[0, 0, 12, 0, 0, 0]
P25	0	1	(95) + 1	(2) 2	(36) 0	12	[8, 0, 0, 0, 4, 0]
P26	0	2	(104) + 4	(2) 2	(36) 0	0	[0, 0, 0, 0, 12, 0]
P27	0	2	(104) + 4	(2) 2	(36) 0	0	[0, 0, 0, 14, 0, 0]
P28	0	2	(52) + 2	(2) 2	(24) 0	0	[0, 0, 0, 6, 0, 0]
P29	0	0	(104) + 4	(2) 2	(36) + 12	0	[11, 0, 0, 0, 0, 0]
P30	0	2	(104) + 4	(2) 2	(36) 0	0	[0, 11, 0, 0, 0, 0]
P31	0	0	(104) + 4	(2) 2	(36) + 12	0	[7, 4, 0, 0, 0, 0]
P32	0	2	(104) + 10	(2) 2	(36) 0	0	[0, 0, 13, 0, 0, 0]
P33	0	1	(104) + 10	(2) 2	(36) 0	− 12	[0, 0, 0, 0, 14, 0]
P34	0	2	(52) + 2	(2) 2	(24) 0	0	[0, 0, 0, 0, 6, 0]
P35	0	1	(130) + 8	(2) 2	(36) 0	− 12	[0, 0, 0, 0, 0, 18]
P36	0	0	(104) + 4	(2) 2	(36) + 12	0	[0, 0, 0, 13, 0, 0]
P37	0	0	(130) + 8	(2) 2	(36) + 12	0	[0, 0, 0, 0, 11, 6]
P38	0	2	(104) + 4	(2) 2	(36) 0	0	[0, 0, 0, 0, 13, 0]
P39	0	2	(130) + 2	(3) 3	(36) + 12	0	[0, 3, 0, 0, 0, 12]

Table 9 continued

Name	S1	S4	S5/S6	S7	S8	S10	S12
P40	0	0	(104) + 4	(2) 2	(36) + 12	0	[0, 0, 0, 14, 0, 0]
P41	0	0	(104) + 4	(2) 2	(36) + 12	0	[0, 11, 0, 0, 1, 0]
P42	0	0	(104) + 10	(2) 2	(36) + 12	0	[2, 10, 0, 0, 0, 0]
P43	0	1	(104) + 4	(2) 2	(36) + 12	12	[0, 0, 0, 0, 0, 13]
P44	0	0	(104) + 10	(2) 2	(36) + 12	0	[7, 3, 0, 0, 0, 4]
P45	0	2	(52) + 8	(2) 2	(24) 0	– 24	[0, 0, 0, 0, 0, 6]
P46	0	1	(130) + 8	(2) 2	(36) 0	– 12	[0, 0, 0, 0, 0, 18]
P47	1	1	(130) + 2	(2) 2	(36) + 12	36	[0, 0, 0, 0, 0, 16]

Overtime: 231 h; debt: 0 h; difference: + 231 h

7 Conclusions

The present paper proposed an integer programming formulation and a fix-and-optimize matheuristic for the PRP. Moreover, a comparison between constraints present in the NRP and the studied PRP demonstrate similarities and differences between these problems. A basic model was developed, addressing the most common constraints to generate a physician roster. The extended model aims to improve the balance of overtime, working hours during non-business days and working hours during day and night shifts. Such constraints are important to be considered when real-world solving methods are developed because the resulting roster is fair between the scheduled physicians.

Computational experiments indicate that both MIP solvers, CPLEX and Coin-OR CBC, were capable of generating optimal and near-optimal results, respectively, when solving small instances with up to 50 physicians. For these small instances, the computation times of Coin-OR CBC were similar to CPLEX. Open-source solver such as Coin-OR CBC, therefore, is a suitable alternative to commercial solvers when the number of physicians to schedule is limited. For larger instances, with 100 and 150 physicians, both solvers were unable to find feasible solutions in most of the instances within an acceptable computation time limit.

A fix-and-optimize matheuristic was proposed to improve these results employing the standalone solvers. Near-optimum results were generated for instances with 50 physicians in 10 min. Moreover, utilizing instances with 100 and 150 physicians, even with short computation time limits (20 and 30 min), the F&O matheuristic generated good results. For the larger instances (150 physicians) the LAHC proposed by Sanchotene et al. (2018) obtained slightly better results when compared to the F&O matheuristic.

Manual rosters had an average of 400 h of overtime. This number was reduced to 35 and 231 using the proposed IP formulation in April and May of 2019. The primary reason for the significant reduction is due to the capability of physician reallocation in different locations. It was not possible with manual scheduling because the number of possibilities for a human to organize the roster rendered the problem very difficult to solve manually. Optimality for instances with 100 and 150 physicians was not proved.

The development of a general personnel rostering model covering constraints present in both nurse and physician rostering problems is a perspective for future research. Such a model would have the advantage of covering several constraints present in real-world scenarios, maximizing the possibility of its applicability in practice.

Acknowledgements The authors would like to thank staff members of Hospital de Clínicas de Porto Alegre (HCPA) that collaborate with this work. Alberto F. Kummer Neto would like to thank the CAPES–Coordination for the Improvement of Higher Education Personnel–for his doctoral scholarship. This research has the support of FAPERGS, Project PqG 17/2551-0001201-1.

References

Beaulieu, H., Ferland, J. A., Gendron, B., & Michelon, P. (2000). A mathematical programming approach for scheduling physicians in the emergency room. *Health Care Management Science*, *3*(3), 193–200.

Bilgin, B., Demeester, P., Mısır, M., Vancroonenburg, W., Vanden Berghe, G., & Wauters, T. (2010). A hyper-heuristic combined with a greedy shuffle approach to the nurse rostering competition. In *Proceedings of the 8th international conference on the practice and theory of automated Timetabling (PATAT'10)*.

Bruni, R., & Detti, P. (2014). A flexible discrete optimization approach to the physician scheduling problem. *Operations Research for Health Care*, *3*(4), 191–199.

Brunner, J. O., Bard, J. F., & Kolisch, R. (2009). Flexible shift scheduling of physicians. *Health Care Management Science*, *12*(3), 285–305.

Brunner, J. O., & Edenharter, G. M. (2011). Long term staff scheduling of physicians with different experience levels in hospitals using column generation. *Health Care Management Science*, *14*(2), 189–202.

Burke, E. K., & Curtois, T. (2011). New computational results for nurse rostering benchmark instances. Technical report. Retrieved September 30, 2019, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.227.6178&rep=rep1&type=pdf.

Burke, E. K., De Causmaecker, P., Petrovic, S., & Vanden Berghe, G. (2006). Metaheuristics for handling time interval coverage constraints in nurse scheduling. *Applied Artificial Intelligence*, *20*(9), 743–766.

Burke, E. K., Li, J., & Qu, R. (2010). A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research*, *203*(2), 484–493.

Ceschia, S., Dang, N., De Causmaecker, P., Haspeslagh, S., & Schaerf, A. (2019). The second international nurse rostering competition. *Annals of Operations Research*, *274*(1), 171–186.

Della Croce, F., & Salassa, F. (2014). A variable neighborhood search based matheuristic for nurse rostering problems. *Annals of Operations Research*, *218*(1), 185–199.

Erhard, M., Schoenfelder, J., Fügener, A., & Brunner, J. O. (2018). State of the art in physician scheduling. *European Journal of Operational Research*, *265*(1), 1–18.

Ernst, A., 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.

Gomes, R. A. M., Toffolo, T. A. M., & Santos, H. G. (2017). Variable neighborhood search accelerated column generation for the nurse rostering problem. *Electronic Notes in Discrete Mathematics*, *58*, 31–38.

Gunawan, A., & Lau, H. C. (2013). Master physician scheduling problem. *Journal of the Operational Research Society*, *64*(3), 410–425.

Haspeslagh, S., De Causmaecker, P., Schaerf, A., & Stølevik, M. (2014). The first international nurse rostering competition 2010. *Annals of Operations Research*, *218*(1), 221–236.

Legrain, A., Omer, J., & Rosat, S. (2019). A rotation-based branch-and-price approach for the nurse scheduling problem. *Mathematical Programming Computation*,. https://doi.org/10.1007/s12532-019-00172-4.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Stützle, T., & Birattari, M. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, *3*, 43–58. https://doi.org/10.1016/j.orp.2016.09.002.

Petrovic, S., & Vanden Berghe, G. (2012). A comparison of two approaches to nurse rostering problems. *Annals of Operations Research*, *194*(1), 365–384.

Puente, J., Gómez, A., Fernández, I., & Priore, P. (2009). Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms. *Computers & Industrial Engineering*, *56*(4), 1232–1242.

Römer, M., & Mellouli, T. (2016). A direct MILP approach based on state-expanded network flows and anticipation for multi-stage nurse rostering under uncertainty. In *Proceedings of the 11th international confenference on practice and theory of automated timetabling (PATAT-2016)* (pp. 549–551).

Rousseau, L. M., Pesant, G., & Gendreau, M. (2002). A general approach to the physician rostering problem. *Annals of Operations Research*, *115*(1), 193–205.

Salassa, F., & Vanden Berghe, G. (2012). A stepping horizon view on nurse rostering. In *Proceedings of the 9th international confenference on practice and theory of automated timetabling (PATAT-2012)* (pp. 161–174).

Sanchotene, T. C., Buriol, L. S., & Kummer Neto, A. F. (2018). Abordagem Heurística para Solução do Problema de Alocação de Médicos do HCPA. Retrieved September 23, 2019, from https://lume.ufrgs.br/handle/10183/185051.

Santos, H. G., Toffolo, T. A. M., Gomes, R. A. M., & Ribas, S. (2016). Integer programming techniques for the nurse rostering problem. *Annals of Operations Research*, *239*(1), 225–251. https://doi.org/10.1007/s10479-014-1594-6.

Smet, P. (2018). Constraint reformulation for nurse rostering problems. In *Proceedings of the 12th international conference of the practice and theory of automated timetabling* (pp 69–80).

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

Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., & Housos, E. (2012). A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, *219*(2), 425–433.

Van Huele, C., & Vanhoucke, M. (2014). Analysis of the integration of the physician rostering problem and the surgery scheduling problem. *Journal of Medical Systems*, *38*(6), 43.

Wickert, T. I., Sartori, C., & Buriol, L. S. (2016). A fix-and-optimize VNS algorithm applied to the nurse rostering problem. In *Proceedings of the 6th international workshop on model-based meta-heuristics (Matheuristics-2016)* (pp. 1–12). Retrieved September 30, 2019, from http://iridia.ulb.ac.be/IridiaTrSeries/link/IridiaTr2016-007.pdf.