Applied Soft Computing 103 (2021) 107151

Contents lists available at ScienceDirect

Applied Soft Computing

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

A GRASP-based algorithm for solving the emergency room physician scheduling problem

M. Cildoz [a], F. Mallor [a,*], P.M. Mateo [b]

[a] ISC, Public University of Navarre, Campus Arrosadia, Pamplona 31006, Spain
[b] University of Zaragoza, Pedro Cerbuna, Zaragoza 50009, Spain

A R T I C L E I N F O

Article history:
Received 9 June 2020
Received in revised form 5 October 2020
Accepted 24 January 2021
Available online 30 January 2021

Keywords:
OR in health services
GRASP
Network flow optimization
Physician scheduling
Emergency room

A B S T R A C T

This paper addresses a physician scheduling problem in an Emergency Room (ER) requiring a long-term work calendar to allocate work days and types of shift among all the doctors. The mathematical model is created without simplifications, using the real calendar, including holidays. This precludes the possibility of cyclic-type solutions, and involves numerous and varied constraints (demand, workload, ergonomics, fairness, etc.). An effective solution to this very difficult practical problem cannot be obtained, for large instances, with exact solution methods. We formulate a mathematical representation of a real-world ER physician scheduling problem featuring a hybrid algorithm combining continuous linear programming with a greedy randomized adaptive search procedure (GRASP). Linear programming is used to model a general physician-demand covering problem, where the solution is used to guide the construction phase of the GRASP, to obtain initial full schedules for subsequent improvement by iterative application of Variable Neighborhood Descent Search (VNDS) and Network Flow Optimization (NFO). A computational study shows the superiority of our approach over the Integer Linear Programming method in a set of instances of varying size and difficulty inspired by a real setting. The methodology is embedded in a software tool for generating one-year-ahead physician schedules for a local ER. These solutions, which are now in use, outperform the manually-created schedules used previously.

1. Introduction

The Emergency Room (ER) of a hospital is where medical and/or surgical care is given to patients arriving in need of immediate attention. An ER is therefore a 24/7 service. Physicians are required to work night, day and weekend shifts, and to take on different ER assignments. Complex constraints add to the difficulty of finding good and equitable schedules for the physicians. Examples of ergonomic constraints are described in Knauth [1], while Gendreau et al. [2] offer an overview of other typical constraints to classifying them into four categories: (1) supply and demand, (2) workload, (3) fairness and (4) ergonomics, based on five case studies performed in Canadian hospitals. This paper addresses a real physician scheduling problem in which constraints of all four categories are considered.

Although the physician scheduling problem shares many characteristics with the nurse scheduling problem (and other workforce planning problems, see, for example, De Bruecker et al. [3] and Van den Bergh et al. [4]), it has received much less attention in the literature. A review of the nurse rostering problem can be found in Burke et al. [5] and Cheang et al. [6]. One can, of course, expect the type of techniques that work well in one problem to do just as well in another, but, despite their basic similarity, they also have differences that can condition the solution. A thorough analysis of such differences is provided in Erhard et al. [7], which highlights the importance of modeling preferences and fairness, among other issues. Their conclusion is that its combined characteristics make the physician scheduling problem highly unique, and thus distinct from general personnel scheduling problems. A similar line of reasoning is given in Damcı-Kurt et al. [8], where it is also pointed out that, in physician scheduling, the issue of staffing costs is not as relevant as that of minimizing deviations from the soft scheduling requirements. The same paper also reports on an analysis of over 5500 department schedules involving a total of 57 medical specialties, concluding that the most complex physician scheduling problems arise in cases where patient care coverage is provided 24/7 in variable settings such as Emergency Medicine departments.

The physician scheduling problem addressed in this paper is complex because it addresses each and every detail of the real-life situation, including the real work calendar and a one-year planning horizon. Managing public holiday shifts remains

* Corresponding author.
E-mail addresses: marta.cildoz@unavarra.es (M. Cildoz), mallor@unavarra.es (F. Mallor), mateo@unizar.es (P.M. Mateo).

https://doi.org/10.1016/j.asoc.2021.107151

a major problem because they preclude the possibility of using a cyclic schedule [9]. In addition to demand constraints, the model considers all mandatory constraints, as well as staff heterogeneity, and personnel preferences. The objective function also pays attention to the fairness of the schedules among physicians, which entails balancing the distribution of different types of shifts among physicians under a range of often conflicting criteria [2]. Moreover, as stated in Bruni and Detti [10], a perfect workload balance would only be obtainable when considering long planning horizons (with the above-mentioned public holiday shifts fairly distributed, or irregular daily demand for physicians).

Our study addresses a one-year planning horizon because twelve-month work calendars are a legal requirement in some countries, including Spain, where they are drawn up annually by the company (after consultation and a subsequent report to the workers' representatives) and posted in a visible area of the workplace (Article 36 of Workers' Statute, BOE-A-2015-11430, [11]). This calendar must contain both the work schedule and annual distribution of working days and holidays. It may undergo modification throughout the year due to changes affecting the staff, family care leave, sickness, etc. In such cases, the manager has to meet staffing demand with minimum change to the original calendar. However, the operational management of the work calendar is a different problem and lies beyond the scope of this article.

Because of the many factors taken into account when planning schedules for ER physicians, it is not easy to ensure equity or fairness in shift distribution. Some shifts are less desirable than others, such as those worked on holidays, on weekends and at nights. An unbalanced distribution of these shifts can affect the dynamics of the physicians' group, their job satisfaction, and the effectiveness of the healthcare received by patients [12]. Schedules failing at achieving a fair distribution of shifts can create feelings of injustice and the existence of favoritism in the work assignment. This can happen, for example, when some physicians have to work more weekends, and in a consecutive way, than other colleagues do. This lack of equity can create dissatisfaction with life-work balance and, ultimately, push physicians to burn-out [13].

Manually created schedules at Hospital Compound of Navarre (HCN) in Spain, failed at getting a fair distribution of these less desired shifts, not because of favoritism but because of the complexity of the task and the difficulty to obtain a balanced distribution. For example, at HCN physicians should work an average of 25 weekends per year, the best manual solution gets ranges from 21 to 28 working weekends with many consecutive worked weekends. Similar inequities were found in the distribution of nights and holidays. This situation was perceived as unfair and gave the scheduler a hard time to justify the distribution choices. Computer-implemented algorithms, as the one proposed in this paper, can obtain better solutions (in the previous example, solutions with no physicians working two consecutive weekends). These schedules improve the equity in the distribution of shifts, and physicians perceive them as fairer and unbiased, improving the group dynamics, the quality of physician life and the healthcare provided to patients.

The physician scheduling problem is a combinatorial optimization problem that falls into the category of NP-hard problems [14], which are intractable for large instances. Metaheuristics are powerful algorithmic approaches, which have been applied with great success to many difficult combinatorial optimization problems [15]. Good solutions can be obtained by designing heuristic algorithms, usually guided by metaheuristics, or by a combination of heuristics and exact methods (see Karp [16]). The last type of algorithm is known as matheuristics [17], which integrates (meta)heuristics and Mathematical Programming (MP) strategies. The hybridization benefits the performance of the algorithm by exploiting the structure of the optimization problem to get better solutions (contribution of MP) while keeping a reasonable computation time to reach the solution (contribution of heuristics).

We initially modeled the physician scheduling problem as an Integer Linear Programming (ILP) problem, but, after a real instance of this problem remains unsolved by CPLEX in one week, using a powerful computer, the need to solve the problem by using a different type of algorithms arises. We design a hybrid algorithm that combines the metaheuristic Greedy Randomized Adaptive Search Procedure (GRASP), Variable Neighborhood Descent (VNDS), and MP. The resulting algorithm falls in the category of matheuristic algorithms. In general, hybrid algorithms presents a "master–slave" structure, with one of the techniques guiding the other. In our case, the heuristic is the master and the MP is the slave. The GRASP construction phase provides full schedules, which are subsequently improved through a VNDS type algorithm, in combination with Network Flow Optimization (NFO) models. Besides, the fitness function used in the GRASP algorithm depends on the result of a Linear Programming (LP) problem, which solves a general physicians' demand-covering problem. The contribution of MP is double: on the one hand, the solution of a linear programming problem guides the constructive phase to promising solutions and, on the other hand, solutions of a series of small network flow problems build up the local search.

The main practical contributions of this paper are, firstly, to present a mathematical model accounting for all types of constraints and objectives considered in practice by a manager when creating a hospital ER physicians' schedule for a one-year planning horizon, and secondly, to provide a hybrid algorithm with the capacity to obtain near optimal solutions to large instances of a real physician scheduling problem within minutes. The main methodological contributions of this paper are the design of a greedy constructive method with a randomized component dependent upon the exact solution to a general covering problem which is solved by LP. This hybridization provides high quality solutions, in terms both of feasibility and of Objective Function Value (OFV). The proposed VNDS method, in combination with NFO, is applied to repair feasibility when it is necessary. Once feasibility is achieved, NFO is used alone to explore large neighborhoods to improve the OFV.

The proposed methodology is tested on a real problem by solving the physician scheduling problem in a hospital ER with 42 physicians and a one-year planning horizon. From 2018 to 2020, the solution was directly used in practice, being deemed by the scheduler as sufficiently superior to replace the manually-created schedule, which was not able even to fulfill all hard ergonomic constraints.

The paper is organized as follows. Next section provides a revision of the related literature. In Section 3, the physician scheduling problem is defined and modeled as an ILP problem. Section 4 presents and explains the hybrid methodology with its four components: (1) the covering problem solved by continuous LP, (2) the construction of a full solution by a greedy random algorithm, and (3–4) the two local search procedures (VNDS and NFO). A computational study is carried out in Section 5, which also includes the case study, a sensitivity analysis of the algorithm parameters and an analysis of the contribution of each component of the algorithm in obtaining good solutions. The paper ends with some conclusions. All the notation is summarized in Appendix B.

2. Related work

The major solution approaches for solving the physician scheduling problem involve MP, metaheuristics, constraint programming, and column generation (reviewed in Gendreau et al. [2]). Similar results are presented by Carter and Lapierre [18], who analyze the characteristics of the problem and scheduling techniques based on Linear Programming (LP) and metaheuristics (mainly Tabu Search). See Erhard et al. [7] for a recent review of 68 relevant papers addressing different types of physician scheduling problem in hospitals. They are classified diversely as staffing, rostering, or re-planning problems. The majority, 61 papers, use MP models. They can be exactly solved for small instances or for problems that are not highly constrained [10,19,20]. In Topaloglu [21], resident physicians in a hospital's pulmonary unit are scheduled for a 6-month period. The author considers 29 instances with the number of variables ranging from 486 to 1995 and the number of constraints ranging from 552 to 2907. Most of the instances are exactly solved within seconds by commercial solvers, but not in all cases, even when the problems are small in size. In other cases, as in Beaulieu et al. [22], where ILP model could not be solved by a modified version of the branch and bound method, a heuristic approach based on a partial branch and bound was used. In fact, when the problem at hand is large (a large number of physicians to be scheduled over a long planning horizon) and very detailed models are formulated, exact solution approaches are usually impractical, being necessary to apply heuristic or heuristic-based hybrid algorithms. For example, Puente et al. [23] solved the physician rostering problem by using a genetic algorithm for a one-month planning horizon and a small/medium size ER with 16 physicians. In Carrasco [24], a simple heuristic is used to assign guard shifts over a one-year horizon. The problem is not highly constrained and the number of shifts assigned per day is small: two or three depending on the day type.

The table in Appendix A compares several characteristics of physician rostering problems addressed by different studies. Their basic goal is to assign employees to work shifts, taking into consideration organizational and regulatory rules, employee skills and preferences, staffing requirements, and other problem-specific issues. Staffing and re-planning problems are not taken into account.

The planning horizon considered in most published studies tends to be small, ranging from two to four weeks. In Brunner, Bard, and Kolisch [25], for example, the physicians in an anesthesia department are scheduled to cover a two-week planning horizon, later extended to six weeks in Brunner, Bard, and Kolisch [26]. The need for long-term schedules is not exclusive to Spain (as reported in Carrasco [24]); it also occurs in countries such as Germany [27], Canada [18], and the United States [8,28,29]. When a long-term schedule is obtained by means of exact solution methods, it is often necessary to partition the full problem into a sequence of interlinked medium-term scheduling problems to be solved by ILP, as in Beaulieu et al. [22] and Topaloglu [21]. This approach is criticized by some authors, who claim that a good solution cannot be obtained by combining partial solutions [24]. One-year-ahead planning is also considered for a variety of staffing problems, as in Brunner, Bard, and Kolisch [25,26] and Brunner and Edenharter [30]; and 39-week schedules are obtained in Green et al. [31].

Generally, the number of different shifts considered in the literature is small (<10). Problems involving a one-year planning horizon, in particular, consider few shift types. Carrasco [24] assigns only on-call shifts, Bruni and Detti [10] two different shift types, and Cohn et al. [28] five. Schoenfelder and Pfefferlen's [32] is the only study that schedules a large number of shift types. Since few shift-assignment studies consider the real calendar, few direct implementations of the solution in real settings are reported. Sometimes, considerable additional manual scheduling is required to enable the use of the model solution. In Ferrand et al. [33] for example, after obtaining a yearly calendar by rolling out an 8-week cycle, public holiday shifts are manually assigned independently by the scheduler, which also requires the manual adjustment of assigned shifts adjacent to the holidays to overcome incompatibilities.

In addition to the size of the problem — in terms of the number of physicians, the planning horizon, variability and heterogeneity in the number of shifts to be assigned each day, types of days (workdays, weekends, and holidays), shifts lengths, and other characteristics all add to the difficulty of obtaining balanced schedules. When shifts vary greatly in length, the equitable distribution of the annual working hours among physicians becomes in itself a difficult task. This problem is related to the optimal multi-way partitioning problem, which is one of the original 21 problems that Richard Karp proved NP-complete [34]. Until now, only Schoenfelder and Pfefferlen's [32] process plans monthly schedules taking into account the hours worked in the previous 22 weeks.

The GRASP metaheuristic, introduced by Feo and Resende [35] and formally presented by Feo and Resende [36], is a multi-start method, with each iteration of the algorithm comprising a construction phase and a local search phase. The first phase leads to a complete solution, and the second is the improvement phase, which continues until a locally optimal solution is reached. After several iterations of the construction phase and the local search procedure, the best overall solution is kept as the result. The construction phase is guided by a greedy function that measures the benefit of including each new element. The benefit of selecting each element changes at each step of the construction. The method is randomized by randomly choosing the next element from a list of candidates. GRASP can be easily hybridized with other approaches and optimization strategies, such as Tabu Search, Simulated Annealing, Variable Neighborhood Search (VNS), and population-based heuristics [37].

The VNS metaheuristic method, introduced by Mladenović and Hansen [38], is based on performing systematic changes of neighborhoods during the search space exploration. The application of VNS is quite simple, requiring only the choice of a metric to measure the distance among solutions in the solution space, which induces the neighborhood structure. A guide to the application of VNS to various classic problems can be found in Hansen and Mladenović [39]. The basic principles of VNS have been extended to provide new versions of the algorithm, which have been successfully applied for solving hard optimization problems. One of the most relevant variants is VNDS which explores neighborhoods in a deterministic way [40].

The choice of neighborhood structure is critical to the performance of a local search algorithm. Basically, observation shows that the larger the neighborhood, the better the local optimal solutions. However, the larger the neighborhood, the longer it takes to explore. Thus, efficient search procedures are required to get the most out of exploring large neighborhoods. One useful option for exploring very large-scale neighborhoods is to use network flow techniques, as discussed and applied in the context of the traveling salesman and routing problems by Ahuja et al. [41]. The result of such a combination is a matheuristic algorithm. In this and other similar cases (see, for example, Punnen [42] and Dror and Levy [43]), the so-called related graph or improvement graph is a bipartite graph used to represent assignment and matching problems.

The development of mathematics-based heuristics has focused on studies related to the Vehicle Routing Problem, in general, and to the home health care problem, in particular, in the context of health, where patients are assigned to worker-teams, patient health services are scheduled and routing decisions are made [44–46]. In a related field, Hernández-Leandro et al. [47] developed a matheuristic based on Lagrangian relaxation for the multi-activity shift scheduling problem. Some matheuristic algorithms have been proposed for addressing the nurse rostering problem: A VNS to accelerate a column generation method is developed and also used in [48] and [49], respectively. Santos et al. [50] proposed a Mixed Integer Programming (MIP) model and a heuristic to decomposed the problem to facilitate a local search procedure. The first phase builds a feasible solution by solving the problem considering only the hard constraints, this solution is improved by a VNS with neighborhoods defined by fixing some decision variables of the incumbent solution and optimizing the others by using a MIP solver. This idea has been adapted recently by Wikert et al. [51] to solve a physician scheduling problem for a month time period and three types of shifts.

3. Definition and mathematical modeling of the scheduling problem

The solution to the physician scheduling problem lies in determining which physician will work in each shift of each day throughout the planning horizon. Shifts vary in type: there are day and night shifts, workday and holidays shifts, short and long shifts, etc. Even within these categories, there are differences in terms of the task requirement: from the triage area, to the resuscitation room, to consultation for patients with milder symptoms, etc. There is also variation in the availability and annual working time of the physicians, such that they are not all able to work all types of shifts. Age or work/life balance issues may prevent certain physicians from working night shifts, for example. Physicians can therefore be grouped by availability and annual working time, such that all members of each group are able to work the same number of hours and types of shift.

The objective of the problem is to obtain the fairest feasible schedule. A fair schedule is one that is evenly distributed among physicians, with all members of a group working the same number of hours, public holidays, weekends, nights (unless exempt), and each type of shift, etc. A balance between groups is also required: the ratio of worked to workable shifts for each physician should be kept proportional across the groups. This workload balancing idea is further developed in Section 4.

To offer some idea of the magnitude of this problem, a medium/large size public hospital might have approximately 40 physicians, and approximately 20 different shifts per day. Over a twelve-month planning horizon, this amounts to $365 \times 20 = 7300$ assignments, each with 40 possibilities. The theoretical number of different assignments (40^{7300}) is considerably reduced when different types of constraints are included. However, the number of feasible solutions is still huge.

The general formulation of this scheduling problem considers N physicians groupable into M types with n_r physicians of type G_r, $r = 1, \ldots, M$, and L types of shifts S_j, $j = 1, \ldots, L$, each defined by its duration d_j (in hours), and other characteristics such as night shift, workday shift, the physician's location during the shift, and types of duties required, among others. There are m_j shifts of type S_j in the planning period. Let T be the number of days for the planning horizon.

Each physician type G_r, $r = 1, \ldots, M$ can work a maximum of $h_r = \rho_r H$ hours during the planning horizon (where H is the number of working hours of a full time physician and $\rho_r \leq 1$), in a subset of shifts determined by binary indicators γ_{rj}:

$$\gamma_{rj} = \begin{cases} 1 & \text{if a physician of type } G_r \text{ can work in shift type } S_j \\ 0 & \text{otherwise} \end{cases}$$
$$\forall\, r = 1, \ldots, M; \forall\, S_j; j = 1, \ldots, L \tag{1}$$

Without loss of generality, it is assumed that a subset of shifts $\boldsymbol{S}(t) \subseteq \left\{S_j, j = 1, \ldots, L\right\}$ needs to be assigned each day and that the demand for each type of shift is one. This assumption reflects the high diversity of shifts in the ER, and places the definition of the problem in a worst case scenario, but the algorithm developed in this research can be straightforwardly adapted for a demand level greater than one. This physician scheduling problem can be mathematically modeled as an ILP problem by using the following decision variables X_{ijt}:

$$X_{ijt} = \begin{cases} 1 & \text{if physician P}_i \text{ works S}_j \text{ on day } t \\ 0 & \text{otherwise} \end{cases}$$
$$\forall\, i = 1, \ldots, N; \forall\, S_j \in \boldsymbol{S}(t); \forall\, t = 1, \ldots, T \tag{2}$$

Feasible schedules need to cover all shifts, observe the maximum working hours of each physician, and comply with ergonomic constraints (especially those relating to the length of rest period after some types of shifts). Therefore, constraints are classified by type into (1) coverage, (2) ergonomic, and (3) work balance.

• **Coverage constraints.** The demand rules are the most basic compulsory requirements: each physician can be assigned a maximum of one shift per day, and each shift must be assigned to a single physician.

$$\sum_{S_j \in \boldsymbol{S}(t)} X_{ijt} \leq 1 \;\forall\, i = 1, \ldots, N;\; \forall\, t = 1, \ldots, T \tag{3}$$

$$\sum_{i=1}^{N} X_{ijt} = 1 \;\forall\, S_j \in \boldsymbol{S}(t);\; \forall\, t = 1, \ldots, T \tag{4}$$

• **Ergonomic constraints.** ER Services are available at all hours of the day and night, every day of the year. Having to work long shifts at any part of the day without reasonably-spaced rest periods between shifts turns a poor work schedule into a potential health threat for physicians. To mitigate the effects of a chaotic work shift calendar, further constraints are added (both to meet legal requirements and to accommodate suggestions from physicians) to enable physical and mental recovery as well as a normal social and family life. Specifically, these so-called ergonomic constraints are designed, among other purposes, to avoid consecutive night shifts, to program rest periods after a long or night shift, to plan weekends off, to avoid an excessive number of rest days between working days, to alternate shift lengths, etc.

Ergonomic constraints are classified into three types according to their purpose: to leave a time interval between shifts, to limit the number of shifts within a time window, and to limit the number of consecutive working days. These constraints can be formulated for each shift type, for all shifts in general, or for subsets of D_C- shifts with $C-$ characteristics. For example, D_C ={night shifts worked on public holidays} contains all shifts with $C-$ characteristics={night, public holiday}.

(i) *Minimum days' interval between shifts*. There must be a minimum interval of δ_c days between two shifts belonging to the set D_C.

$$\sum_{t=q-\delta_c}^{q} \sum_{j \in D_c} X_{ijt} \leq 1 \;\;\forall\, q = \delta_c + 1, \ldots, T;\; \forall\, i = 1, \ldots, N;\; \forall\, D_c \tag{5a}$$

For example, in the event of having to spread out night shifts by imposing a two-day interval between two worked night shifts (such that there can be only one night shift in a period of 3 days), $\delta_c = 2$.

This category of constraints includes a compulsory number of days off after certain types of shifts and is formulated as follows when δ_c days' rest are required after a shift S_j in a set D_C.

$$\delta_c X_{ijq} + \sum_{t=q+1}^{q+\delta_c} \sum_j X_{ijt} \le \delta_c \; \forall q = 1, \dots, T - \delta_c; \quad \forall i = 1, \dots, N; \quad \forall S_j \in D_c \tag{5b}$$

(ii) *Maximum number of shifts worked within a time window*. This constraint limits the maximum number of shifts in a set D_c assigned to physicians over a time window of w_{1c} days.

$$\sum_{t=q-w_{1c}+1}^{q} \sum_{j \in D_c} X_{ijt} \le \nu_{1c} \quad \forall \; i = 1, \dots, N; \forall \; q = 1c, \dots, T; \quad \forall D_c \tag{6}$$

This type of constraint is used, say, to limit the number of public holidays worked within a certain period. Suppose that a physician cannot be assigned more than 5 public holiday shifts over a time window of 30 days. Then, $\nu_{1c} = 5$ and $w_{1c} = 30$.

(iii) *Maximum number of consecutive working days*. Physicians cannot work more than w_{2c} consecutive days on any type of shift belonging to a set D_c.

$$\sum_{t=q-w_{2c}}^{q} \sum_{j \in D_c} X_{ijt} \le w_{2c} \quad \forall i = 1, \dots, N; \; \forall q = w_{2c} + 1, \dots, T; \forall D_c \tag{7}$$

Here also, there may be constraints imposing a maximum on the number of days' gap between shifts, a minimum on the number of a certain type of shift that can be assigned within a time window, and a minimum on the number of consecutive days on shifts belonging to a set D_c. The formulation of these constraints is similar to that given in (5a), (6), and (7).

• **Workload balancing constraints.** These constraints are designed to guarantee a fair distribution of the different types of shifts among all physicians.

(i) Fair distribution of working hours on shifts belonging to a set D_c among all physicians.

$$\sum_{t=1,\dots,T} \sum_{j \in D_c} d_j X_{ijt} \le \rho_r H_c^U \quad \forall i = 1, \dots, N \, (P_i \in G_r); \; \forall D_c \tag{8}$$

$$\sum_{t=1,\dots,T} \sum_{j \in D_c} d_j X_{ijt} \ge \rho_r H_c^L \quad \forall i = 1, \dots, N \, (P_i \in G_r); \; \forall D_c \tag{9}$$

H_c^U and H_c^L are variables representing the maximum and minimum number of hours worked on shifts with characteristics in C, respectively. These constraints could also be applied to a single type of shifts S_j or to the entire set of shifts.

(ii) Fair distribution among all physicians of shifts in a set D_c

$$\sum_{t=1,\dots,T} \sum_{j \in D_c} X_{ijt} \le \rho_r J_c^U \quad \forall i = 1, \dots, N \, (P_i \in G_r); \; \forall D_c \tag{10}$$

$$\sum_{t=1,\dots,T} \sum_{j \in D_c} X_{ijt} \ge \rho_r J_c^L \quad \forall i = 1, \dots, N \, (P_i \in G_r); \; \forall D_c \tag{11}$$

These constraints are similar to the previous ones, but are now aimed at balancing the number of shifts rather than the number of working hours. The variables J_c^U and J_c^L, respectively, limit the maximum and minimum number of shifts worked by all physicians.

(iii) Fair distribution of shifts from a set D_c among physicians in the same group. Constraints for balancing the number of shifts can be assigned to particular types of physicians.

$$\sum_{t=1,\dots,T} \sum_{j \in D_c} X_{ijt} \le J_{rc}^U \quad \forall i = 1, \dots, N \, (P_i \in G_r); \quad \forall D_c \tag{12}$$

$$\sum_{t=1,\dots,T} \sum_{j \in D_c} X_{ijt} \ge J_{rc}^L \quad \forall i = 1, \dots, N \, (P_i \in G_r); \quad \forall D_c \tag{13}$$

The variables J_{rc}^U and J_{rc}^L limit the maximum and minimum number of shifts in set D_c worked by physicians P_i in group G_r, $r = 1, \dots, M$, respectively.

The objective function is defined to reach the fairest distribution of the workload among physicians by minimizing the range of the limiting variables H_c^L and H_c^U, J_c^U and J_c^L, J_{rc}^U and J_{rc}^L. Thus, the objective function is the minimization of the sum of all ranges:

$$\min \sum_{i=1}^{\#D} (H_{c_i}^U - H_{c_i}^L) + \sum_{i=1}^{\#D} (J_{c_i}^U - J_{c_i}^L) + \sum_{i=1}^{\#D} \sum_{r=1}^{M} (J_{rc_i}^U - J_{rc_i}^L), \tag{14}$$

where $\#D$ is the number of sets of shifts D_{c_i} involved in the fairness constraints. Different weights may be used in the objective function to reflect the relative importance of the fairness of the shift distribution and working hours among physicians.

Thus, the ILP model for the physician scheduling problem involves the minimization of the objective function (14) subject to a set of constraints (3)–(13), which is fully presented in Appendix C.

4. The hybrid GRASP based algorithm

This section explains the hybrid methodology. Section 4.1 provides a general overview of the algorithm. In Section 4.2 a general covering problem, modeled as an LP problem, is solved to obtain the average number of shifts of each type that should be worked by physicians of each type. These averages are used in Section 4.3 by a greedy random algorithm to construct a full solution. Finally Section 4.4 presents two local search procedures to improve the solution obtained by the greedy algorithm.

4.1. General description of the algorithm

The proposed heuristic algorithm comprises three stages: the first solves a global covering and balancing problem formulated as an LP model; the second is a construction phase, in which a full solution is obtained by applying a greedy randomized algorithm (guided by the solution of the first phase); and the third is an improvement stage, in which the solution provided by the previous stage is used as the input to a cyclic optimization alternating between VNDS and NFO which continues until a feasible solution is obtained; this solution is then improved by means of NFO alone. The first stage is executed only once, while the other two stages are iterated several times to define a multi-start procedure, as illustrated in Fig. 1. This hybrid GRASP-type algorithm will be identified as "Algorithm G+NO".

The proposed methodology starts by determining the number of each type of shift that each physician should work over the entire planning horizon, in order to guarantee coverage of all shifts and a workload balance among physicians, based on a fair distribution of the different types of shifts (nights, weekends, holidays, etc.). This problem is formulated as a continuous LP problem, which, at a very low computational cost, provides the solution to be used in the next phase.

The construction phase is the implementation of a GRASP algorithm to build a solution by assigning shifts to physicians sequentially. The procedure starts with the first day of the planning horizon, assigning all the shifts for that day and progressing day by day until a full assignment is obtained. The list of candidates for each shift assignment is first defined by the feasibility constraints and then by elitism based on a fitness function. This function takes into account the assignments made so far to all physicians and the theoretical number of shifts of each type that each physician should work (obtained as the solution of the LP formulated in the first phase of the algorithm).

The full scheduling obtained in the previous phase is improved by alternating VNDS to repair violations of the constraints (required if the constructive step provides an infeasible solution) with NFO to balance the distribution of shifts and working hours among the physicians. Once a feasible solution is obtained, improvements to the fair distribution of the workload are sought using NFO only.

In the following subsections, a detailed mathematical and algorithmic description is provided for the three components of the heuristic method.

4.2. A linear programming model to solve the general covering problem

The purpose of this optimization step is to obtain the average number Z_{rj} of shifts of type $S_j, j = 1, \ldots, L$, that should be worked by physicians of type G_r, $r = 1, \ldots, M$, in order to cover service demand within the regulatory working hours. Variables Z_{rj} can be positive only if $\gamma_{rj} = 1$, that is, $(1 - \gamma_{rj})Z_{rj} = 0$. In addition, this general planning has to distribute the shifts among physicians as evenly and fairly as possible, for which the decision variables Z_{rj} must fulfill the following constraints:

- Demand Covering constraint

$$\sum_{r=1}^{M} n_r Z_{rj} = m_j \; \forall S_j; j = 1, \ldots, L \tag{15}$$

- Working hours constraint

$$\sum_{j=1}^{L} d_j Z_{rj} \leq h_r \; \forall \; r = 1, \ldots, M \tag{16}$$

- Equitable distribution of shifts

  Some sets of shifts have to be evenly distributed among those physicians who are able to work them. These include holiday shifts ($D_{hol} = \{shifts\ on\ holidays\}$), night shifts ($D_{nig} = \{shifts\ at\ nights\}$), weekend shifts ($D_{wee} = \{shifts\ on\ weekends\}$), etc.

  Let D_c be the set of shifts to be fairly distributed, and let

$$U_c = \frac{\sum_{S_j \in D_c} m_j}{\sum_r \rho_r n_r \left(1 - \prod_{S_j \in D_c} \left(1 - \gamma_{rj}\right)\right)}$$

  be the average number of shifts in D_c per full-time physician able to work such shifts. Some shifts belong to one or more sets D_c while others might belong to none. To impose the equitable distribution of all shifts, two constraints are considered for each set D_c and physician type G_r:

$$\sum_{S_j \in D_c} Z_{rj} - \rho_r U_c \leq F_1 \;\; \forall \; r = 1, \ldots, M; \quad \forall D_c \tag{17}$$

$$\rho_r U_c - \sum_{S_j \in D_c} Z_{rj} \leq F_1 \;\; \forall \; r = 1, \ldots, M; \quad \forall D_c \tag{18}$$

  The deviation variable F_1 bounds the absolute value of the differences between the average number of shifts assigned to each group of physicians and the value of reference $\rho_r U_c$ for all sets of shifts D_c. The deviation variable F_1 is minimized in the objective function of the LP problem.

- Even distribution of each type of shift among all physicians. Let $W_j = \dfrac{m_j}{\sum_r \gamma_{rj} \rho_r n_r}$ be the number of shifts of type S_j that should be worked by each full time physician eligible to do so.

  - Shifts that do not participate in balancing constraints (17) and (18) should also be distributed as fairly as possible. Then,

$$Z_{rj} - \rho_r W_j \leq F_j \rho_r W_j \, \forall \;\; r = 1, \ldots, M; \; \forall \;\; S_j \notin \bigcup_c \{D_c\} \tag{19}$$

$$\rho_r W_j - Z_{rj} \leq F_j \rho_r W_j \, \forall \, r = 1, \ldots, M; \; \forall \;\; S_j \notin \bigcup_c \{D_c\} \tag{20}$$

$$F_j \leq F_2^U \, \forall \, S_j \notin \bigcup_c \{D_c\} \tag{21}$$

$$F_j \geq F_2^L \, \forall \, S_j \notin \bigcup_c \{D_c\} \tag{22}$$

    Each deviation variable F_j bounds the absolute value of the difference between the average number of shifts S_j assigned to each group of physicians and the value of reference $\rho_r U_j$ for all shifts of type S_j that do not belong to any set D_c. These deviation variables are also bounded in the interval (F_2^L, F_2^U). The amplitude and maximum value of this interval are minimized in the objective function of the LP problem.

  - Shifts that do participate in balancing constraints (17) and (18) should be distributed as evenly as possible among all physicians.

$$Z_{rj} - \rho_r W_j \leq F_3 \rho_r W_j \, \forall \, r = 1, \ldots, M; \; \forall \, S_j \in \bigcup_c \{D_c\} \tag{23}$$

$$\rho_r W_j - Z_{rj} \leq F_3 \rho_r W_j \, \forall \, r = 1, \ldots, M; \; \forall \, S_j \in \bigcup_c \{D_c\} \tag{24}$$

    The deviation variable F_3 bounds the absolute value of the differences between the average number of shifts S_j assigned to each group of physicians and the value of reference $\rho_r U_j$ for all shifts of type S_j that are included in any set D_c. The deviation variable F_3 is minimized in the objective function of the LP problem.

    The following objective function (25) minimizes the value of auxiliary variables introduced in constraints (17)–(24), which measure the deviations from both above and below the target that represents the equal distribution of shifts and provide a goal programming approach to this covering problem:

$$\min \beta F_1 + \left(2F_2^U - F_2^L\right) + F_3 \tag{25}$$

    The purpose of the weighting factor β is to give more importance to the first objective than to the others. The first objective balances the distribution of shifts in sets D_c, which are set explicitly by the scheduler, while the other two objectives balance each type of shifts individually, those that belong to a set D_c, and those that do not. The optimization finds the best proportional shift distributions among all those distributions that are optimal in the equitable distribution of shifts in sets D_c. A large enough value for factor β would be the total number of shifts to be assigned.

    The average number of shifts, Z_{rj}, of each type S_j that should be worked by physicians in group G_r is obtained as the solution of the LP problem with objective function (25) and constraints (15)–(24). The full formulation of the LP problem is included in Appendix C.

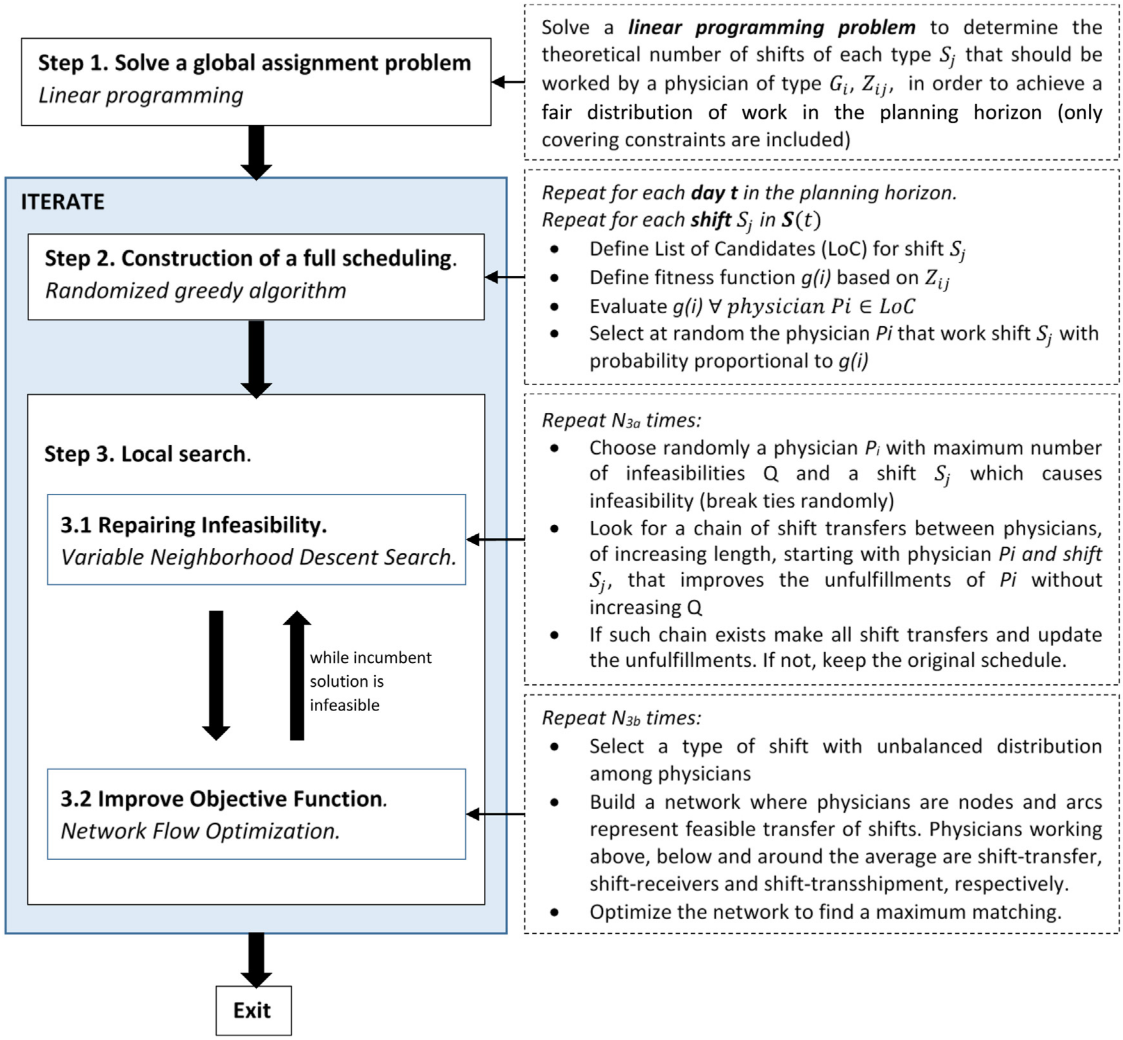

Fig. 1. The three stages of the proposed heuristic algorithm as applied to physician scheduling.

For ease of notation, from this subsection forward, the theoretical average number of each type of shift S_j that should be worked by a physician $P_i \in G_r$ will be denoted by Z_{ij}, which is equal to Z_{rj}.

4.3. Construction of a full scheduling solution by a greedy randomized algorithm

This subsection presents a heuristic to generate solutions by a probabilistic greedy construction method. The heuristic follows the constructive step of the GRASP metaheuristic method [52], which builds a solution one element at a time. In the physician scheduling problem, this is done by successively assigning each of the shifts that must be covered each day, starting with a shift from the first day of the planning horizon and ending with a shift from the last day of the planning horizon. Each day's shifts are assigned in random order.

Let T be the number of days in the planning horizon, *nshifts*(t) the number of shifts for the t-th day, and A_t the set of physicians working on day; then the construction phase proceeds in general as shown in Algorithm 1:

Initialize $X_{ijt} = 0 \;\forall\; i,\, j,\, t$
for $t = 1$ ***to*** T ***do***
  for $j = 1$ ***to*** $nshifts(t)$ ***do***
    Choose at random a shift S_j not yet assigned
    Define the list of candidates LoC;
    Evaluate each physician in LoC by a greedy function $g\colon LoC \to \mathbb{R}$;
    Select a physician $l \in LoC$ by a roulette wheel mechanism
    Add P_l to set A_t
  end
end

Algorithm 1. Initial solution construction algorithm.

The following subsection gives the details for the definition of the List of Candidates, LoC, the definition of the greedy function, and the selection of a physician by a roulette wheel mechanism.

4.3.1. Definition of the list of candidates

A LoC is defined for each shift assignment. A physician is included in the LoC for a shift assignment when all the applicable constraints are fulfilled. If the resulting LoC is empty, then all physicians will be included in the LoC. This process is summarized in Algorithm 2.

$LoC(j,t) = \{P_i | \text{All constraints for shift } j \text{ are fulfilled}\};$
if $LoC(j,t) = \emptyset$ **then**
  $LoC(j,t) = \{P_i, i = 1, \ldots, N | P_i \notin A_t\}$
end

Algorithm 2. List of Candidates for shift j on day t.

4.3.2. Definition of a greedy function $g(i)$

Suppose that a shift of type j has to be assigned on a day t. Let z^*_{ij} be the number of shifts of type j assigned so far to physician P_i and let k be the index of the physician with the maximum value in the following set of ratios:

$$k = \underset{i}{\operatorname{argmax}} \left\{ \frac{z^*_{ij}}{Z_{ij}} \text{ such that } Z_{ij} > 0 \right\} \tag{26}$$

Then, for each physician P_i in the $LoC(j,t)$, the following greedy function $g_j(\mathrm{i})$ is evaluated:

$$g_j(i) = \frac{z^*_{kj}}{Z_{kj}} - \frac{z^*_{ij}}{Z_{ij}} \text{ such that } Z_{kj}, \mathrm{Z_{ij}} > 0 \tag{27}$$

This greedy function measures the difference between the maximum proportion of shifts of type S_j already assigned to a physician (z^*_{kj}/Z_{kj}) and the ratio of shifts assigned to a particular physician. This value is then normalized to the target value for the whole planning horizon, Z_{ij}. Thus, the greater the value of $g_j(i)$ for physician P_i, the greater his/her need to work this shift S_j in order to meet the reference values Z_{ij}. By definition, this greedy function is a non-negative definite function. However, it could occur that $g_j(i) = 0$ for all physicians in the $LoC(j,t)$.

Enhancement of the greedy function. The greedy function was defined based only on already assigned shifts of type S_j. Nevertheless, some shifts are important for the even distribution of other general shift characteristics among physicians. For example, if the shift that is being assigned is a weekend shift and all physicians have to work the same number of weekends within the planning horizon; thus, the greedy function must also take into account the consequences of the assignment for the even distribution of weekend shifts. For this purpose, for each set of shifts D_c that has to be evenly distributed among physicians and $S_j \in D_c$, the following greedy function $g_{D_c}(i)$ is defined:

$$g_{D_c}(i) = \frac{\left(\max_l \left\{ \frac{z^*_{lD_c}}{Z_{lD_c}} \right\} - \frac{z^*_{iD_c}}{Z_{iD_c}}\right)}{\left(\max_l \left\{ \frac{z^*_{lD_c}}{Z_{lD_c}} \right\} - \min_l \left\{ \frac{z^*_{lD_c}}{Z_{lD_c}} \right\}\right)} \tag{28}$$

where,

$$Z_{lD_c} = \sum_{S_j \in D_c} Z_{lj} \text{ and } z^*_{lD_c} = \sum_{S_j \in D_c} z^*_{lj} \tag{29}$$

A normalized greedy function $g_{Nj}(i)$, which ranges in (0,1), is defined as follows:

$$g_{Nj}(i) = \frac{g_j(i)}{\left(\max_l \left\{ \frac{z^*_{lj}}{Z_{lj}} \right\} - \min_k \left\{ \frac{z^*_{lj}}{Z_{lj}} \right\}\right)} \tag{30}$$

The new enhanced greedy function $\mathcal{g}_j(i)$ is defined as:

$$\mathcal{g}_j(i) = g_{Nj}(i) + \sum_c g_{D_c}(i) \tag{31}$$

where the summation is extended to all sets D_c of shifts that need to be balanced and that include the shift S_j.

Then, this greedy function balances the participation of each physician in all shifts and shift characteristics included in the objective function by assigning the shift to the physician who is farthest from meeting all the balancing conditions in which the shift is involved. The balancing assessment takes into account the theoretical values determined by the LP covering problem (Appendix C).

4.3.3. Roulette wheel for the selection of a physician n

In the construction procedure of the basic GRASP, the probability $p(i)$ of selecting a physician $P_i \in LoC(j,t)$ would be chosen at random, with equal probability of being chosen. However, we propose to bias the selection toward the candidates that contribute the most to keep the solution balanced at that moment. The idea of using probability distributions different from the uniform was proposed by Bresina [53]. In particular, we use a power function that extends the polynomial function of order n [54]

$$p(i) = \frac{\left(\mathcal{g}_j(i)\right)^\alpha}{\sum_{P_l \in LoC(j,t)} \left(\mathcal{g}_j(l)\right)^\alpha} \tag{32}$$

Observe that, if $\alpha = 0$, we will have a random construction; if $\alpha = 1$, the probability will be proportional to the greedy value. The greater the value of α is, the more elitist the selection mechanism.

If all physicians in the $LoC(j,t)$ have $\mathcal{g}_j(i) = 0$, then the probability of being chosen is equal among them. We recommend choosing large values for $\alpha\,(\geq 0.9)$ to obtain better initial solutions.

4.4. Improvement of a solution

The feasibility of a solution is improved by decreasing the number of unfulfilled ergonomic constraints by means of a VNDS algorithm, which is followed by a NFO procedure to better fulfill the balancing objectives. These two search mechanisms are applied iteratively (see Fig. 2) until a stop criterion is met (optimization time or iterations with no improvement). The following subsections offer a detailed description of each of these improvement steps.

4.4.1. Variable neighborhood descent search for repairing infeasibility

The construction phase is driven by the solution of the general covering problem and is particularly oriented toward constructing a feasible solution because the LoC in each shift assignment is first defined by physicians who fulfill all constraints. However, in problems with little slack for finding feasible solutions (too small a surplus with respect to the total demand for working hours and very tough ergonomic requirements), the construction phase could provide a solution that fails to meet certain constraints. In this case, the first step of the improvement phase is a repair process, whereby a shift contributing to the infeasibility of one physician's schedule is transferred to another physician. These shift transfers successively involve several physicians and are repeated several times. Fig. 3 represents the logic of these movements: shift S1, which causes the infeasibility of the sequence S1–S2 in physician P_{14}'s schedule (after shift S1, there must be a day off), is transferred to physician P_{23} (causing infeasibility because, two days off are compulsory after shift S7); this requires transferring shift S7 to physician P_9 (again causing an infeasibility), and this, in turn, results in the transfer of shift S3 to physician P_{18}. After these transfers, the initial infeasibility of physician P_{14} is solved without detriment to the total number of non-compliances of the remaining physicians.

The search for sequences of transfers leading to the improvement of the current schedule falls into the category of a VNDS algorithm with rationale as follows.

Let X_i be the set of shifts assigned to physician P_i in the incumbent solution, that is, $X_i = \left\{S_j \in \boldsymbol{S}(t) \,|X_{ijt} = 1\right\}$ and $\rho(X_i, X_i')$ be the distance between solutions for scheduling a physician defined as

$$\rho(X_i, X_i') = |X_i \Delta X_i'| \tag{33}$$

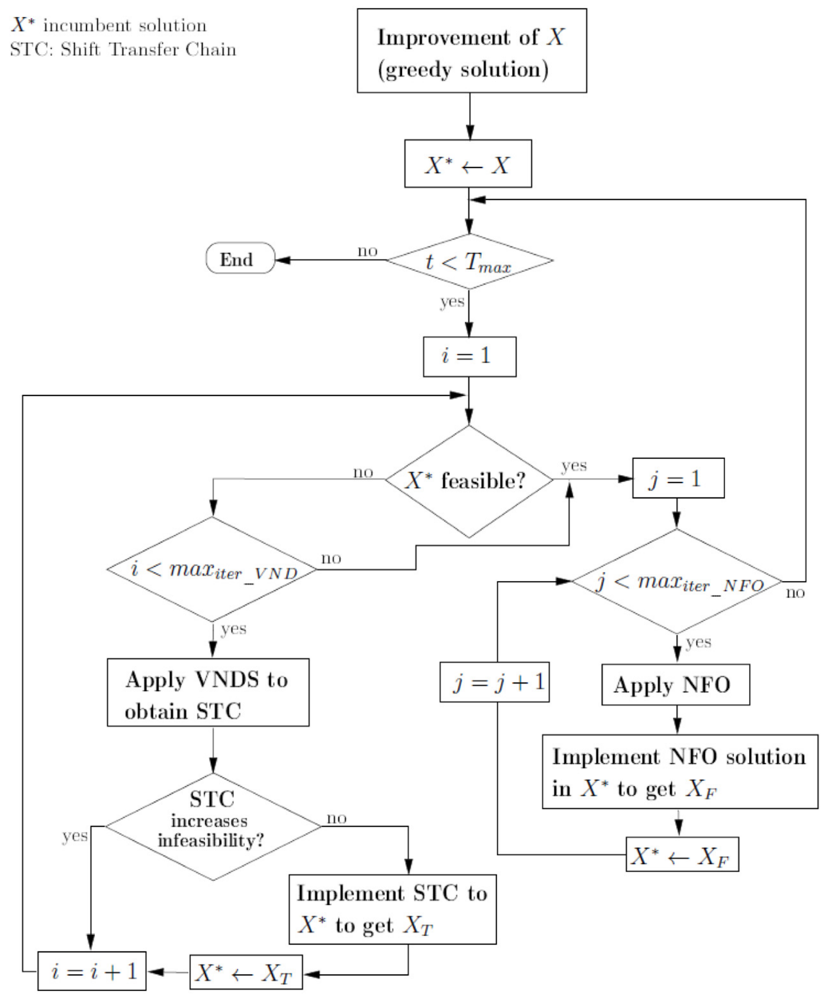

Fig. 2. Flowchart of the solution improvement phase.

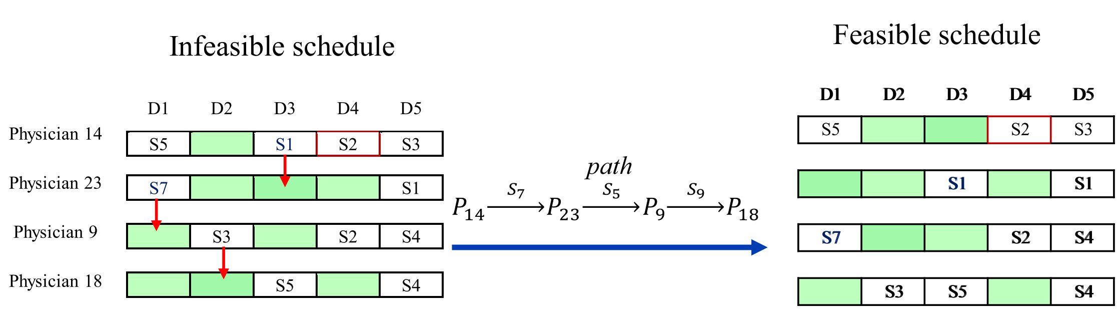

Fig. 3. Example of shift ($S_j : S1 - S7$) transfer among physicians ($P_i : Physician$ 9, 14, 18, 23) on different days ($D_t : D1 - D5$). Ergonomic requirements for the different types of shifts: S7 must be followed by two days off; S1, S5 must be followed by one day off; and S2, S3, S4 do not require the next day to be a rest day.

where $|X_i \Delta X_i'|$ represents the number of shifts that form part of schedule X_i but not of X_i' and those which form part of schedule X_i' but not of X_i. Let us observe that when a physician P_i with schedule X_i transfers a shift to another physician, the resulting schedule for P_i, denoted by X_i', verifies $\rho\left(X_i, X_i'\right) = 1$.

A full schedule X is the aggregation of all the physicians' schedules: $X = (X_1, X_2, \ldots, X_N)$, and then, $\rho\left(\mathrm{X}, \mathrm{X}'\right) = \sum_{i=1}^{N} \rho\left(X_i, X_i'\right)$ represents the distance between two schedules for all physicians. The transfer of a shift from one schedule X to obtain another schedule X' is denoted by $X' = h(X)$. The schedule solution X' resulting from a sequence of k transfers of shifts in which the transferee in one shift transfer becomes the transferor in the next shift transfer is denoted by $X' = h_p^k(X)$. The index p refers to the path p, which determines the transfers of shifts between physicians. For example, in Fig. 3, the path is $P_{14} \xrightarrow{S_1\ of\ D3}$

$P_{23} \xrightarrow{S_7 \text{ of } D1} P_9 \xrightarrow{S_3 \text{ of } D2} P_{18}$. The length of a path is the number of transfers, in the case of Fig. 3 the length is 3.

A neighborhood of depth k is defined as

$$\aleph_k(X) = \{X' | \exists\ p \text{ of length } k \text{ such that } X' = h_p^k(X)\}$$

Let us consider a certain type of constraint that is not fulfilled by a solution X and thus requires repair. Let $Q > 0$ denote the maximum number of unfulfilled constraints among all physicians and P_Q the set of physicians that reach this maximum number of non-fulfillments.

$$P_Q = \{\text{physicians with a number } Q \text{ of non} - \text{fulfillments}\}$$

A recursive function enables fairly easy implementation of this VNDS procedure. In each step, each physician with an infeasible schedule tries to transfer a shift (which is problematic because it causes an infeasibility) to another physician, who is able to accept it, even if this results in an additional infeasibility, and then the infeasibility improvement problem is transferred to another physician, and the process is repeated. The steps of this VNDS algorithm are detailed in Algorithm 3.

4.4.2. A network flow optimization problem for balancing the distribution of shifts and working hours

The goal of this optimization procedure is to transfer shifts of a certain type from physicians with surplus workload, who are working significantly more than average hours and a greater number of that type of shift, to physicians with slack in that type of shift, who are working significantly fewer than average hours. The term "significantly" is used in relation to a zone of indifference surrounding the average number of hours worked, which is defined in order to stabilize the procedure as it progresses. A physician P_i is considered to have an acceptable total of working hours, $H_i(X) = \sum_{t=1}^{T}\sum_{j=1}^{L} d_j X_{ijt}$, in a schedule X when as long as it belongs to this interval of indifference. To formalize this idea, for each iteration l of this optimization procedure ($1 \le l \le \max_{iter\_NFO}$), the lower and upper boundaries of the indifference interval, LH and UH respectively, around the average number of working hours are defined as follows:

$$LH = \rho_r \overline{H}\left(1 - \left(\frac{l}{\max_{iter\_NFO}}\right)\varepsilon\right) \tag{34}$$

$$UH = \rho_r \overline{H}\left(1 + \left(\frac{l}{max_{iter\_NFO}}\right)\varepsilon\right) \tag{35}$$

where ε is the factor defining the final window of indifference. For example, $\varepsilon = 0.0015$ and an average $\overline{H} = 1750$ and a full-time physician ($\rho_r = 1$); the indifference window is $UH - LH \approx$ 5h. The average $\overline{H}$ for a full-time physician can be estimated as $\overline{H} = \frac{\sum_j d_j m_j}{\sum_r \rho_r n_r}$

Given a schedule solution X, these two limits classify the physicians into three groups:

$$P_{TS}(X) = \{P_i | H_i(X) > UH\}$$

$$P_{RS}(X) = \{P_i | H_i(X) < LH\}$$

$$P_{IN}(X) = \{P_i | LH \le H_i(X) \le UH\}$$

The physicians in set $P_{TS}(X)$ can transfer shifts, and those in set $P_{RS}(X)$ can receive shifts. Physicians in the balanced set $P_{IN}(X)$ can play an intermediate role by both receiving and transferring shifts. This condition for transferring a shift is called the *working hours' condition* (*WHC*).

A physician of type G_r can transfer a shift of a certain type S_j when the number of assignments of this type exceeds the theoretical number Z_{ij} determined in the pre-processing optimization phase; and, conversely, a physician can receive a shift of a certain type when the number of assignments of this type is below

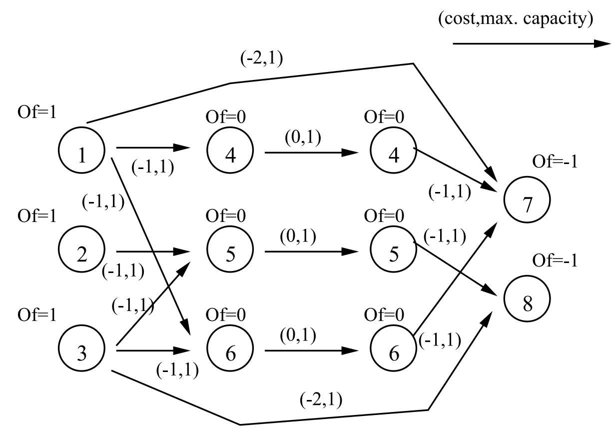

Fig. 4. Example of work-flow network. Physicians 1, 2 and 3 can transfer one shift; physicians 4, 5 and 6 can receive and transfer one shift, and physicians 7 and 8 can receive one shift.

this theoretical figure. In terms of the notation introduced in Section 4.2, a physician P_i is allowed to transfer a shift , S_j, when $z^*_{iD_c} > Z_{iD_c}$, and a physician P_i is allowed to receive a shift S_j when $z^*_{iD_c} < Z_{iD_c}$ for all sets D_c with relevance in the objective function and in which shift S_j participates. This shift transfer condition is named the *balancing shift condition*(*BSC*).

Building the network structure. The nodes represent physicians, and each arc (i, k) represents a possible transfer of a shift S_j from physician P_i to physician P_k. The physician P_i belongs to set $P_{TS}(X)$, and P_k belongs to set $P_{RS}(X) \cup P_{IN}(X)$, or P_i belongs to set $P_{IN}(X)$, and P_k belongs to set $P_{RS}(X)$. To plot an arc on the graph, both physicians, transferor and transferee, must meet the conditions *WHC* and *BSC* defined earlier and the transferee must be feasibly able to work this shift. When there exists more than one arc verifying the conditions between a pair of physicians, one of them is chosen at random (since it is the case that more than one shift could feasibly be transferred from physician P_i to physician P_k). Therefore, the network structure is built randomly and successive iterations of this procedure provide different networks.

Assigning demands, capacities, and costs to the network. Nodes representing a physician in $P_{TS}(X)$ have a demand of -1, nodes representing a physician in $P_{RS}(X)$ have a demand of $+1$, and nodes representing a physician in $P_{IN}(X)$ have a demand of 0 (trans-shipment nodes).

The network is expanded by unfolding each node in the set $P_{IN}(X)$, into two nodes that are connected by an arc.

All arcs in the network have a maximum capacity of 1 and a minimum capacity of 0.

Costs:

- the arcs between a physician in $P_{TS}(X)$ and a physician in $P_{RS}(X)$ have a cost of -2,
- the arcs between a physician in $P_{TS}(X)$ and a physician in $P_{IN}(X)$, or between a physician in $P_{IN}(X)$ and a physician in $P_{RS}(X)$ have a cost of -1,
- the arcs between nodes representing the same physician in $P_{IN}(X)$ have a cost of 0.

Fig. 4 shows a simple example of a flow network with 3 physicians in set $P_{TS}(X)$, 3 physicians in set $P_{IN}(X)$, and 2 physicians in set $P_{RS}(X)$.

Step 0 Initialize $N_{iter} = 0, k = 0$;
Step 1 $N_{iter} = N_{iter} + 1$;
$k = k + 1$;
if $N_{iter} > max_{iter\_VND}$ **then** ***End***;
/* Start new iteration to find a new shift-transfer chain */
Compute set P_Q;
if $P_Q = \emptyset$ **then** Feasible solution, ***End***;
Choose randomly $P_i \in P_Q$ and shift $S_j \in X_i$ causing a constraint infeasibility;
$S_T \leftarrow S_j$;
$k = 0$;
Go to **Step 2**;
Step 2 $k = k + 1$;
if $k \leq max_{depthSearch}$ **then** /* Begin the exploration of the neighborhood of depth k*/
  if $\exists$ *Physician* $P_{i^*} \mid X'_{i^*} = X_{i^*} \cup \{S_T\}$ *does not increase the infeasibilities of* P_{i^*} **then**
    Make definitive all temporal transfers and go to **Step 1**;
  else if $\exists$ *Physician*$P_{i^*} \mid X'_{i^*} = X_{i^*} \cup \{S_T\}$ *does not increase the value of* Q **then**
    Transfer temporarily shift S_T to P_{i^*};
    Select shift $S_j \in X_{i^*}$ $(S_j \neq S_T)$ that causes constraint infeasibility to P_{i^*};
    $i \leftarrow i^*, S_T \leftarrow S_j$; /* Solving the infeasibility problem is transferred from P_i to P_{i^*}*/
    Go to **Step 2**;
  else
    Undo the temporally transfers, keeping the initial schedule and go to **Step 1;**
  end
end

Algorithm 3. VNDS Procedure for repairing solutions. It is based on transferring a shift contributing to the infeasibility of one physician's schedule to another physician. These transfers of shifts successively involve several doctors and are repeated several times.

Solving the network flow problem. The problem is solved by using an algorithm to find the minimum-cost feasible flow. The resulting networks are small in size and can be solved quickly by efficient algorithms such as Network Simplex, Out of Kilter, Cycle Canceling, or Successive Shortest Path (see Thulasiraman et al. [55]). Our implementation uses a successive shortest path algorithm, as described in Ahuja, Magnanti, and Orlin [56]. After network optimization, each physician can transfer and receive, at most, one shift. For this reason, this optimization step is repeated $\max_{iter\_NFO}$ times. In each iteration, the limits that define the partition of physicians into sets $P_{TS}(X)$, $P_{RS}(X)$, and $P_{IN}(X)$ are modified, starting with small values, which are gradually increased. Any fluctuation of the zone of indifference between two values contributes to the variability of the created networks and the stabilization of the shift transfers as the algorithm progresses.

Consecutive iterations of this procedure lead to different networks, which gradually improve the balancing of shifts and working hours. When this NFO phase is iterated with the VNDS algorithm because the solution is still infeasible, the NFO helps the VNDS algorithm by providing new starting solutions from which to search for good shift transfer chains (as in a shaking procedure) and also helping to redress any imbalance in the shift distribution that may be introduced due to the application of VNDS.

5. Computational analysis

This section reports the results of the empirical assessment of the algorithm presented above, which was implemented in Java. Its practical effectiveness is tested in Section 5.1 by solving the problem of scheduling all the ER shifts for the year 2018 among 42 physicians in the Hospital Compound of Navarre (HCN) in Spain. In addition, in Section 5.2, a set of synthetic scheduling problems of varying degrees of difficulty is used to assess the performance of the algorithm under different conditions. The results are compared with those obtained by CPLEX. Finally, Section 5.3 investigates the influence of the different phases of the algorithm on the solutions to the physician scheduling problem as well as the value of its parameters for obtaining good solutions.

5.1. The physician scheduling problem at the Hospital Compound of Navarre (HCN)

The ED of the HCN, which is located in Pamplona (Spain), serves a population of half a million people, and attends to over 140,000 patients per year. This ED is staffed 24 h per day by 42 board-certified emergency physicians. Currently, each year's shift schedule is planned manually by one of the physicians, who dedicates three weeks' work to this task. Although, this person is an experienced physician and has been in charge of schedule planning for many years, the task becomes more complicated every year, because new labor laws create new constraints and new categories of workers with different working conditions. This physician creates the schedule without any technological/computational support, using only large spread sheets, similar to the one shown in Fig. 5, where there is a row for each physician and a column for each day. Starting with simple rotational rules, the scheduler uses his/her own heuristics to consecutively balance holiday shifts, weekend shifts, nights, and, finally, regular shifts, while also trying, to satisfy a large set of constraints (ergonomic, workload, etc. as described in Section 3). The resulting schedule violates many conditions as well as provoking numerous complaints from other physicians, who consider the schedule unbalanced and conditioned by subjective preferences.

Staff characteristics. The staff comprises 42 physicians who can be grouped into two types: (1) a first group G_1 of 3 physicians who are exempt from night shifts (denoted by O, A5, G1, G2 and G3 in

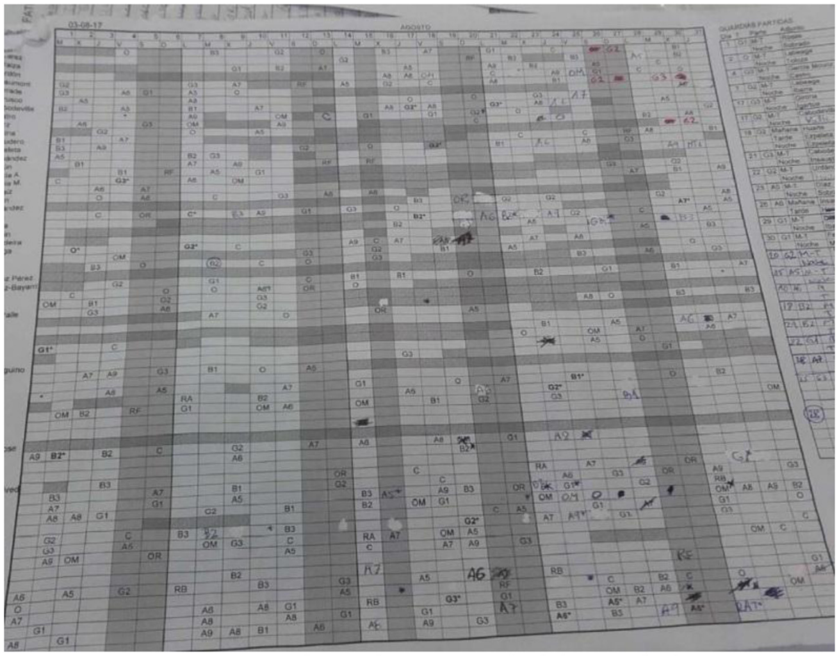

Fig. 5. The hospital's current scheduling method.

Table 1) for reasons of age or various other reasons such as work-family reconciliation and (2) a second group G_2 of 39 physicians who can work any shift.

Shift characteristics. Shifts differ in length and task characteristics. In the ER of the HCN, physicians can be assigned to different areas, such as the resuscitation room, the triage zone, the observation zone, or the severe patient circuit. Each of these locations involves different tasks and responsibilities. In addition, different numbers and types of shifts are scheduled for different types of days. Table 1 includes relevant information about shift length, the type of shifts worked per type of day, and the number of days off after each shift. A balanced distribution of all types of shifts among the physicians must be achieved.

Constraints. There are some compulsory requirements for individual schedules: two days off have to be scheduled after a long shift (19/20 h) and one day off after a 14-hour-shift; schedules must not allow more than two consecutive weekend shifts; or more than 5 holiday shifts in a month (these include Saturdays and Sundays); and must allow a four-day gap between night shifts. In addition, all physicians' schedules must fulfill certain balanced distribution criteria based on the number of shifts of each type worked yearly (13 balance conditions, B1 to B13, defined in Table 2, each one associated to a set of shifts D_c), and all these shifts have to be evenly distributed over the year. There are 5 night shifts (G1, G2, G3, A and O) which should be balanced individually, except G1 and G2, that only require balancing their sum.

Results. The problem was first formulated as a Mixed Integer Linear Programming model (see Appendix C) with over 200,000 variables and 70,000 constraints. CPLEX 12.6.2 solved this problem on an Intel (R) Xeon (R) CPU E5-1630 v4 3.70 GHz and 64.0 GB RAM, and after an entire week of execution time, the best-found integer solution provided an objective function value of 43 (see Table 3), which was obtained after 168 computation hours and remained unchanged for 54 h, until the end of the experiment (see Fig. 6). However, CPLEX was not able to prove optimality of that best-found solution within the computational time limit. In fact, it is not optimal, because the G+NO algorithm obtained a solution with an OFV of 15 within seconds. Fig. 6 shows the best-found solutions obtained by both CPLEX and the G+NO algorithm over time (note that the time axis is expressed in seconds for the G+NO algorithm and in hours for CPLEX).

To apply the G+NO algorithm, the initial LP problem was first formulated in order to obtain the optimum theoretical values of Z_{ij} for each type of shift and physician group $Z_{ij}, i = 1, 2; j = 1, \dots, 19$, which was solved within seconds. Table 3 shows, in row 3, the theoretical optimum value for each shift-balancing goal $B_k, k = 1, \dots, 13$ for the two groups of physicians. These values guided the construction phase and the objective function improvement. The maximum and minimum numbers of shifts worked by a physician in either group according to the solution obtained by CPLEX, in one hour and in one week, are given in rows 4–5, and 6–7, respectively; and in rows 8–9 for a G+NO solution obtained after five minutes' computation time. The column for B13 shows the hours worked annually, and it is here that the G+NO clearly outperforms CPLEX, thus demonstrating the efficacy of the Network improvement phase. The best bound obtained by CPLEX in one week is 4.547. A straightforward analysis of the objective function can provide better bounds; superior to those provided by CPLEX (see Table 3).

The notion underlying this target bound is the following: when the number of shifts participating in a balancing goal is not a multiple of the number of possible shift candidates, it is impossible for them all to be assigned the same number of shifts of this type, and the balanced solution will, therefore, necessarily fall within a range of at least one. However, when the number of shifts is a multiple of the number of candidates then an even distribution among all physicians is possible. This simple analysis provides a minimum bound for the objective function. In the case study, this bound is 11 and G+NO and CPLEX solutions provide a relative gap (36) of 0.27 and 0.74, respectively.

$$GAP = \frac{\{OFV\} - \{Theoretical\ bound\}}{\{OFV\}} \tag{36}$$

Table 1
Shift coverage requirements by type of day. The shift labels (S1–S19) are those used by the ER of HCN (row 2: local description).

Shifts	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19
Local description	G1	G2	G3	A5	O	C	B1	B2	B3	A6	A7	A8	A9	OM	R	OR	RF	RA	RB
Length in hours	19	19	19	19	20	14	14	14	14	14	14	8	8	8	3	14	14	14	14
Workdays	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
Mondays[a]	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X
Holidays	X	X	X	X	X	X				X	X					X	X		
Days off after shift	2	2	2	2	2	1	1	1	1	1	1	0	0	0	0	1	1	1	1

[a]Mondays or any other day following a holiday.

Table 2
The 13 balancing objectives.

Balancing objective name	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
Type of shifts to be balanced	C	B1 + B2 + B3	A6 + A7	A8 + A9	OM	(A6 + A7) −(OR + RF)	O	A5	G3	G1 + G2	D_C = {weekends}	D_C = {holidays}	AWH

Table 3
Case study results: heuristic algorithm and CPLEX results for balancing the different shift sets (B1–B13) included in the objective function OFV. Max. and Min. refer to the maximum and minimum number of balancing goals involving physicians in the respective group. The relative gap (last column) is calculated according to formula (36).

		Obj G1						Obj G2						Obj G1&G2								
Objectives		$B1_1$	$B2_1$	$B3_1$	$B4_1$	$B5_1$	$B6_1$	$B1_2$	$B2_2$	$B3_2$	$B4_2$	$B5_2$	$B6_2$	*B7*	*B8*	*B9*	*B10*	*B11*	*B12*	*B13*	*OFV*	Rel. Gap
Theoretical values		**17.38**	**35**	**32.59**	**23.33**	**11.67**	**0**	**8**	**16.15**	**16.21**	**10.77**	**5.38**	**0**	**9.36**	**9.36**	**9.36**	**18.72**	**25**	**3.57**	**1750.95**		
CPLEX (1 h)	Max.	174	133	41	0	0	41	51	46	44	62	138	19	52	43	56	70	36	8	2515	3451	1
	Min.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78		
CPLEX (1 week)	Max.	25	31	36	23	7	1	8	20	17	12	7	3	10	11	10	21	26	4	1756	43	0.74
	Min.	25	31	36	23	7	0	7	15	14	10	4	0	8	8	8	17	24	3	1745		
G + NO (5 min)	Max.	17	35	33	24	12	0	9	17	17	11	6	1	10	10	10	19	26	4	1752	15	0.27
	Min.	17	35	33	24	12	0	8	16	16	10	5	0	9	9	9	18	24	3	1750		

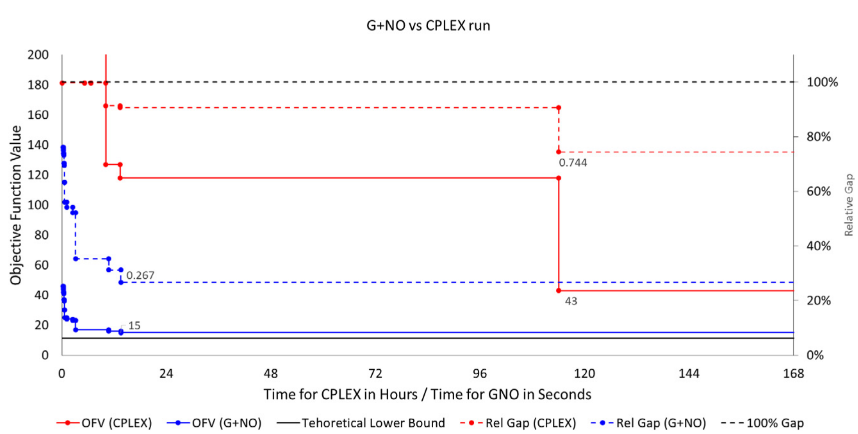

Fig. 6. CPLEX and G+NO algorithm performance: best found solutions obtained by both over time.

The solution obtained with the heuristic obtains the bound for each balancing criterion except for B6, which could theoretically obtain a value of 0 but in fact obtains a range of 1; criterion B11, which could theoretically obtain a value of 0 and actually obtains a range of 2; and criterion B13, which could theoretically obtain a value of 1 and actually obtains a range of 2. These differences increase the global bound of 11 by 4 units to an OFV of 15. In conclusion, the solution may be non-optimal, but, from a practical point of view it is, nevertheless, a very high quality solution compared with those obtained manually by the physician, who accepted solutions within a range of 2 or 3 for goals B1–B12 and a range of 20 for goal B13.

5.2. Additional computational experiments

In this section, the performance and efficacy of the proposed algorithm are evaluated by creating new instances in order to obtain problems of different sizes and degrees of difficulty, while still maintaining the characteristics of a real problem. From the real case detailed in Section 5.1, two more different-sized problems with 20 and 30 physicians, respectively, were created by rescaling all the physician and shift types included in the real

Table 4
Comparison of the solution obtained in five minutes by the heuristic algorithm with the one provided by CPLEX in one hour. "Instances description" includes the number of physicians as well as the ratios of holidays, worked nights, and annual hours worked per physician in each solved instance with respect to the reference problem. The "theoretical bound" refers to the value obtained by the straightforward analysis of the objective function explained in the text. "CPLEX solution" includes the OFV, the relative gap (36), the range of annual hours worked (AWH), and the best bound provided by CPLEX in one hour. "G + NO solution" also includes the OFV, relative gap (36) and the AWH. Column "BEST", "AVERAGE", and "MEDIAN" show the results for the best, average, and median solution after 30 runs of the algorithm G+NO for each instance. The column "Improvement G+NO over CPLEX" is calculated by the difference between the solution provided by G + NO (column BEST) and the solution provided by CPLEX, divided by the latter.

No. of Physicians (Phys)	Instances description (standard = 1)			Theoretical bound	CPLEX solution				G + NO solution (BEST)			G + NO solution (AVERAGE)			G + NO solution (MEDIAN)			Improvement G + NO over CPLEX
	Holidays per Phys ratio	Nights per Phys ratio	AWH ratio		OFV	Rel. Gap	AWH	Best bound	OFV	Rel. Gap	AWH	OFV	Rel. Gap	AWH	OFV	Rel. Gap	AWH	
20	0.80	0.85	0.93	7	20	0.65	9	1.87	8	0.13	2	10.50	0.33	3.4	11	0.36	3	0.60
20	1.00	0.79	0.91	8	16	0.5	7	4.52	9	0.11	2	10.27	0.22	2.13	10	0.2	2	0.44
20	0.80	1.00	0.95	7	15	0.53	7	1.22	8	0.13	1	9.03	0.23	2.1	9	0.22	2	0.47
20	1.00	1.00	0.98	9	23	0.61	6	3.88	10	0.1	1	10.27	0.12	1.07	10	0.1	1	0.57
20	1.00	1.00	1.00	**9**	**24**	**0.63**	**6**	**3.03**	**11**	**0.18**	**2**	**12.20**	**0.26**	**3.4**	**12**	**0.25**	**3**	0.54
20	1.00	1.00	1.07	8	41	0.8	6	2.88	11	0.27	3	14	0.43	5.17	13.5	0.41	5	0.73
20	1.20	1.00	1.05	9	32	0.72	7	1.34	12	0.25	2	14.17	0.36	4.73	14	0.36	5	0.63
20	1.00	1.21	1.09	9	30	0.7	7	1.26	14	0.36	4	16.67	0.46	5.87	16	0.44	5.5	0.53
20	1.20	1.15	1.07	9	23	0.61	14	0.75	11	0.18	2	17.63	0.49	6.2	16.5	0.45	5	0.52
30	0.86	0.91	0.95	9	67	0.87	49	0.52	12	0.25	2	14.47	0.38	3.13	14	0.36	3	0.82
30	1.00	0.87	0.94	12	63	0.81	9	3.18	14	0.14	2	14.97	0.2	3	15	0.2	3	0.78
30	0.86	1.00	0.97	9	22	0.59	11	0.32	12	0.25	1	14.87	0.39	2.23	15	0.4	2	0.45
30	1.00	1.00	0.97	11	26	0.58	13	1.34	14	0.21	2	15.53	0.29	3.23	15.5	0.29	3	0.46
30	1.00	1.00	1.00	**11**	**25**	**0.56**	**12**	**0.63**	**13**	**0.15**	**1**	**14.00**	**0.21**	**1.83**	**14**	**0.21**	**2**	0.48
30	1.00	1.00	1.03	11	3151	1	2402	1.59	14	0.21	1	15.43	0.29	3.03	15	0.27	3	1.00
30	1.14	1.00	1.03	9	2355	1	1932	1.21	13	0.31	1	15.47	0.42	2.2	15.5	0.42	2	0.99
30	1.00	1.13	1.06	11	3077	1	2379	1.28	14	0.21	3	15.30	0.28	3.1	15	0.27	3	1.00
30	1.14	1.09	1.04	10	3038	1	2359	1.06	14	0.29	1	16.13	0.38	1.9	16	0.38	2	1.00
42	0.90	0.93	0.97	12	3387	1	2379	0	14	0.14	1	15.53	0.23	2.83	16	0.25	3	1.00
42	1.00	0.91	0.96	11	3352	1	2427	0	15	0.27	3	16.27	0.32	3.3	16	0.31	3	1.00
42	0.90	1.00	0.98	12	3309	1	2379	0	14	0.14	1	15.67	0.23	1.73	16	0.25	2	1.00
42	1.00	1.00	0.97	11	3129	1	2379	0	14	0.21	3	15.67	0.3	3.23	16	0.31	3	1.00
42	1.00	1.00	1.00	**11**	**3451**	**1**	**2437**	**0**	**15**	**0.27**	**1**	**16.43**	**0.33**	**1.87**	**16**	**0.31**	**2**	1.00
42	1.00	1.00	1.02	11	3516	1	2437	0	14	0.21	2	16.77	0.34	3	17	0.35	3	1.00
42	1.10	1.00	1.01	13	3453	1	2379	0	16	0.19	3	17.47	0.26	3.5	18	0.28	3	1.00
42	1.00	1.09	1.04	10	3292	1	2379	0	14	0.29	2	15.47	0.35	3.53	16	0.38	4	1.00
42	1.10	1.07	1.03	12	3414	1	2413	0	15	0.2	2	16.60	0.28	2.93	17	0.29	3	1.00

case. These three instances (the real case with 42 physicians and the two new rescaled instances with 20 and 30 physicians, respectively) are considered normal-difficulty instances, and highlighted in bold in Table 4.

Eight more instances, all for different sized problems, were designed. Four of them are intended to increase the solving difficulty by increasing the number of shifts to be assigned in total and therefore per physician, thus making the ergonomic constraints more difficult to satisfy. The other four scenarios are designed to facilitate the process by decreasing the number of shifts. Specifically, the new problems are obtained as follows

- The less difficult instances. The number of shifts assigned per day is reduced by one on some days to obtain the four new problems: workday morning shift, holiday morning shift, workday night shift, and holiday night shift, respectively. Thus, the ratio of average annual working hours (AWH) with respect to the initial scenario is less than one.
- The more difficult instances. The number of shifts assigned per day is increased by one on some days to obtain the four new problems: workday morning shift, holiday morning shift, workday night shift, and holiday night shift, respectively. Thus, the ratio of average annual working hours (AWH) with respect to the initial scenario is greater than one.

The increases and reductions in the number of shifts can also change the number of holidays and number of nights worked by a physician, thereby affecting the difficulty of solving the problem. Table 4 compares the results of all instances provided in 5 min by the heuristic algorithm and in one hour by CPLEX on the same computer. The table includes the ratios of physicians, holidays, worked nights, and annual hours worked per physician in each solved instance with respect to the reference problem. The results provide the objective function value (OFV), the range of annual hours worked (explicitly included because of the difficulty involved in balancing it) and the gap with respect to the theoretical bound. The best bound obtained by CPLEX is also included. The heuristic G+NO algorithm is run 30 times for 5 min each. The heuristic algorithm is a multi-start algorithm, set to generate 10 solutions and improve them for a total of 30 s each (easiest problems with *AWH ratio* < 1), or 5 solutions with an improvement time of 1 min (harder problems with *AWH ratio* ≥ 1). The algorithm returns the best of these 5 or 10 solutions. Table 4 presents the results for the best of the 30 runs, the average solution and the median solution. The heuristic algorithm outperforms CPLEX in all instances: the mean and median of the 30 runs of the heuristic algorithm are much lower than the OFV obtained by CPLEX. In all instances, moreover, the 30 runs of the G+NO algorithm provide a better solution than CPLEX.

Observe that, in problems with 20 physicians and fewer/weaker constraints (first four scenarios), the best G+NO solution is only one unit's distance from the theoretical bound, and in all scenarios this distance is less than or equal to 4, except in one where it is 5. As already mentioned, these results are very good from a practical point of view, since they considerably improve the manually designed schedules which were not feasible and had wider-ranging balancing criteria.

The quality of each solution in Table 4 is assessed by the relative gap (36), and the improvement of G+NO over CPLEX is calculated by the difference between the solution provided by G+NO (column BEST) and the solution provided by CPLEX, divided by this latter solution.

5.3. Parameter tuning

In this section we investigate the influence of the different phases of the algorithm and the value of its parameters for obtaining good solutions to the physician scheduling problem. Some parameters are fixed, parameter β in the objective function (25), and parameter α in choice of physicians from the *LoC* (32). The value of parameter β was set to 1000, large enough to guarantee the balanced distribution of shifts at weekends, nights, holidays, and other types of shifts specified as important to balance by the scheduler. We implemented an elitist choice of physicians from the *LoC* by fixing a value $\alpha = 0.95$. In the rest of the Subsection, we deal with the capacity of the algorithm first to achieve feasible solutions and then to improve the value of the objective function.

Fine-tuning of parameters to obtain feasible solutions. The construction phase of the algorithm includes feasibility as the first condition for defining the *LoC* from which a physician will be selected at random to be assigned a shift. Thus, in problems with no heavy constraints, the construction phase is expected to provide a feasible solution. However, this does not occur in problems heavily constrained by strict ergonomic requirements and heavy workloads. To illustrate this, we conducted an experiment using the 27 problems solved in the previous section, obtaining, for each one, 100 different solutions using only the construction phase of the algorithm. Table 5 contains the number of feasible solutions. Clearly, when one extra holiday and night shift are added, and there are fewer physicians to share the extra work, the problem becomes harder to solve. However, when feasibility is not achieved, the number of infeasibilities is low, usually one or two (out of the several tens of thousands of constraints). In the case of the 20-physician problem, with one night shift added on every holiday, none of the 100 solutions provided by the construction phase is feasible. In this worst-case scenario, the number of infeasibilities could reach around 10–15 (Fig. 7 shows the distribution of the number of unfulfilled constraints in the one hundred solutions of the two worst instances: when an extra night shift or an extra day shift is added on holidays for an ED with 20 physicians). In instances with no heavy constraints, the construction phase obtains a feasible solution within 100 runs.

To analyze the performance of the feasibility improvement phase, we use the most difficult problem that of scheduling shifts for 20 physicians, for which no feasible solution was obtained initially. Specifically, we study the influence of two parameters: the number of iterations max_{iter_VND} of the VNDS algorithm; and the number of iterations max_{iter_NFO} of the NFO step. The recursion depth parameter is set as 10, which is large enough to permit a wide search and small enough to avoid excessive memory consumption (higher values can lead to memory allocation problems).

For each combination of the values 1, 5, 10, 20 and 50 for max_{iter_VND} and 1, 5, 25, 50, 100, and 200 for max_{iter_NFO}, 50 solutions are obtained by running the algorithm G+NO for 30 s. Thus, 1500 different solutions are obtained for the same problem. Table 6 shows the percentage of feasible solutions obtained with each combination of parameters. A two-way ANOVA reveals the influence of the value max_{iter_NFO} in the results (p-value<0,001) but not the influence of max_{iter_VND} (p-value = 0,915). The results of a post-hoc analysis of a one-way ANOVA, using only max_{iter_NFO}, and the graph of means (Fig. 8) reveals that results for 1 and 5 are much worse and that significantly better results are obtained for values of 25, 50 and 100 (after which they deteriorate slowly as the number of iterations increases). An explanation for these results is the following: given a schedule, the VNDS tries to sequentially find shift-transfer chains to repair infeasibilities; but, in heavily constrained problems, it is possible that no (or only very few) such chains exist in the current solution. Therefore, it

Table 5
Percentage of feasible solutions reached in the construction phase of the G+NO algorithm.

Instances										
Shifts added Time-slot Type of day		0	+ 1 Day Holiday	+ 1 Night Holiday	+ 1 Day Work day	+ 1 Night Work day	−1 Day Holiday	−1 Night Holiday	−1 Day Work day	−1 Night Work day
Nº . physicians	20	100	28	0	100	84	100	100	100	100
	30	99	89	19	99	81	100	100	100	100
	40	100	99	92	100	99	100	100	100	100

Table 6
% of feasible solutions reached by G+NO algorithm for each configuration.

		max_{iter_VNDS}				
		1	5	10	20	50
max_{iter_NFO}	1	4	6	12	10	4
	5	26	24	28	22	14
	25	78	82	82	90	82
	50	80	78	88	72	78
	100	72	74	74	74	84
	200	76	66	62	78	82

Table 7
Mean, median and minimum values of the 50 iterations of G + NO algorithm.

Mean		max_{iter_VND}				
		1	5	10	20	50
max_{iter_NFO}	25	22.76	21.46	20.93	19.84	20.10
	50	24.12	20.19	25.33	23.82	22.52
	100	20.67	23.82	22.55	20.63	22.28
Median		max_{iter_VND}				
		1	5	10	20	50
max_{iter_NFO}	25	19.0	19.0	18.0	17.5	18.0
	50	22.0	19.0	23.0	22.0	21.0
	100	18.5	21.5	21.0	18.0	19.5
Minimum		max_{iter_VND}				
		1	5	10	20	50
max_{iter_NFO}	25	13	13	13	12	13
	50	13	13	13	12	12
	100	12	13	13	12	13

is necessary to shake the current solution to obtain a new one and then resume the search for the required feasibility-repairing chains. These new schedules are provided by applying the NFO step. The results show that too few iterations max_{iter_NFO} (1–5) do not create significantly different solutions; whereas, above a certain number of iterations, the changes in the solution through the network are minor. These iterations consume computational time and thus influence the total number of global iterations of the algorithm (constructive step, max_{iter_VND} and max_{iter_NFO}). Computational tests, included in Fig. 8, show that above 100 iterations for max_{iter_NFO} the global efficiency decreases. From this analysis, values of 25, 50, and 100 could be considered appropriate.

Fine-tuning to obtain good objective function values. Table 7 shows the average, median, and minimum of the feasible solutions obtained after running the algorithm 50 times for one minute for each combination of max_{iter_NFO} and max_{iter_VND} parameters. We consider the best values (25, 50, and 100) for max_{iter_NFO} and (1, 5, 10, 20, and 50) for max_{iter_VND}. The results show no statistically significant differences. However, in order to fix parameter values, we choose 25 for max_{iter_NFO} and 20 for max_{iter_VND}, because they provide the lowest mean, median, and minimum values.

Execution time. Several experiments were conducted to analyze the computational time required to obtain good solutions. We found that 1 min per solution in the multi-start G+NO algorithm is enough time to achieve the greatest possible improvement of the solution obtained from the construction phase. Fig. 9 shows three 1-minute runs of the real instance, the best solution in each run being obtained in 13.6, 36.5, and 22.58 s.

Fig. 10 shows the G+NO performance for the most difficult problem; that is, scheduling shifts for 20 physicians, as used in the previous analysis. The upper graphs show three 1 min G+NO runs of the instance, which obtains their best values in 18.191, 32.264, and 56.83 s. The second graph is a zoom of the previous graph, showing the points at which feasibility is recovered. The solutions achieve feasibility in 2.359, 1.512, and 3.641 s. The lower graph shows the G+NO run that provided the best solution for that instance in isolation. It reaches feasibility in 2.359 s and its best solution in 18.191 s, which is a value of 12 (the theoretical solution is 9, and the minimum solution provided by CPLEX in an hour is 23).

6. Conclusions

In this paper, we have developed a new hybrid algorithm for solving the physician scheduling problem.

The characteristics of the problems reviewed in the table shown in Appendix A reveal that the problem addressed in our work is one of unique complexity. It has to assign many different types of shifts – of varying lengths –, to accommodate a non-uniform daily shift demand dependent on day type, a one-year planning horizon, a real calendar interspersed with public holidays, and many ergonomic and balancing constraints imposed by mandatory and personnel requirements. Besides, the solution has been implemented in practice the last years, and it will be in the subsequent ones, which is not very common because, as can be seen from the table shown in Appendix A, most papers solve problems based on real data, but only a few report on the practical implementation of the solution. In recent years, the problem has become more complex because there are more different groups of physicians regarding their working hours and the exemption from working certain shifts. This is due to laws on the balance between work and family life and better working conditions for the elderly (for example, exemption of working night shifts). In the words of the scheduler, "currently it is impossible to find, not a good solution, but a solution that could approach the fulfilling of ergonomic constraints, and minimally balanced, to be accepted by the physician staff".

One of the main features of the algorithm is that the number of shifts of each type that must be worked by each physician over the whole planning horizon is used to define the fitness function which determines his/her probability of being selected from the *LoC*. In this way, the construction step creates balanced solutions. In addition, the algorithm also prioritizes the construction of feasible solutions by including in the *LoC* physicians who can feasibly work the shift being assigned in that step. As a result, the construction phase usually obtains good quality solutions, because even infeasible solutions failing in only a few constraints can generally be repaired in the local search step. This step works by combining a shift-transfer process to reduce the number of infeasibilities, with a NFO process, to create new solutions to

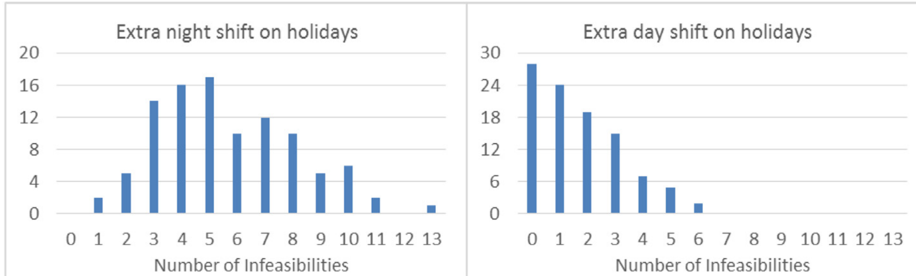

Fig. 7. Distribution of the number of infeasibilities for the two hardest scenarios (they both have 33 379 constraints).

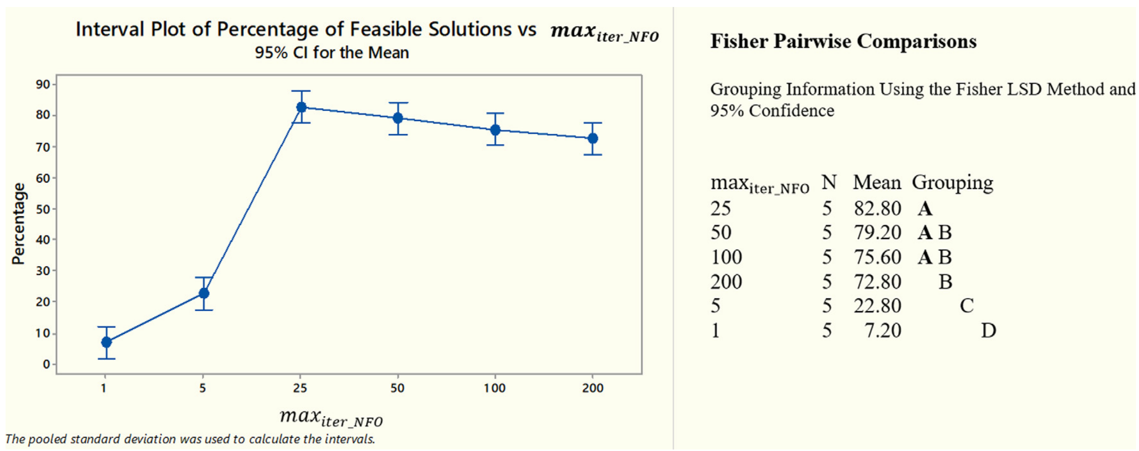

Fig. 8. Graph of the mean % of feasible solutions reached by algorithm for each max_{iter_NFO} parameter value and results of the post-hoc analysis of a one-way ANOVA.

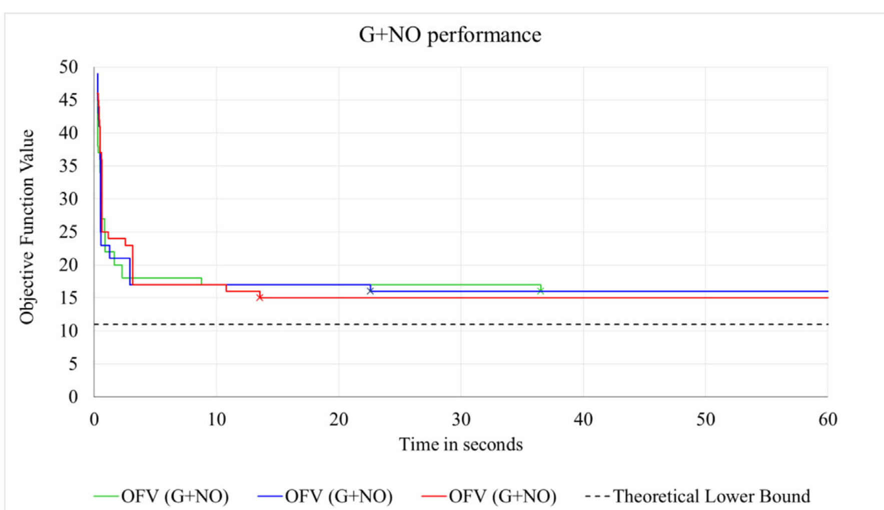

Fig. 9. Three examples of 1 min run of the G+NO algorithm for the real instance.

continue the search for shift-transfer chains. Once feasibility is achieved, the procedure continues with the NFO process alone in order to improve the balance of the solution.

The algorithm is a multi-start algorithm with a greedy constructive phase that looks for good but diversified solutions, that contributes to explore the solution space in the promising areas. This phase takes a very small computational time. For each initial solution, its neighborhood is searched in two ways: the first one, only applied when the incumbent solution is infeasible, it is based on the transfer of single shifts between physicians, so the neighborhood explored is not very large (although the recurrence of the transfer chain can involve several physicians); the second one, it is made by the NFO phase to improve the objective function, which allows for simultaneous transferring of shifts among physicians and explores larger neighborhoods. Computational results show that with the tuned parameters, a few seconds are enough to retrieve a feasible solution and less than one minute to conduct exploitation of the neighborhood of an initial solution (in problems of similar complexity to those more difficult analyzed in this paper).

The results show a clear superiority over ILP for realistically-sized instances; better results being achieved in a few minutes, as opposed to the 168 h (an entire week) taken by CPLEX when real instances are solved. The resolution time, which can be up to

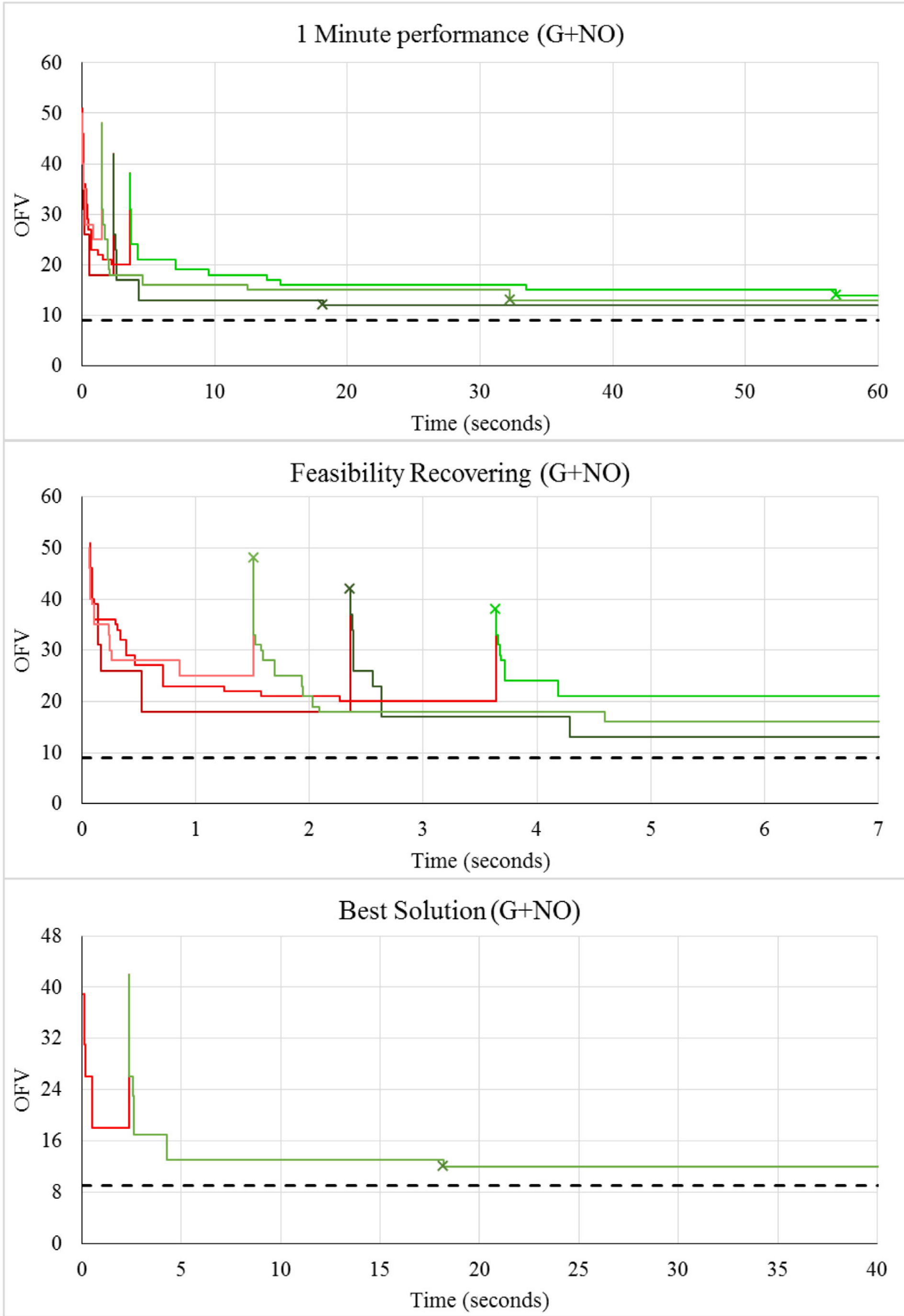

Fig. 10. G+NO performance for the most difficult problem.

several minutes in relatively large, heavily constrained problems, with little slack for physicians' working hours, can be considered satisfactory for use in practice. The algorithm can be applied for solving any scheduling problem that fits the general mathematical model presented in Section 3. It can handle different types of physicians and different types of shifts, with different types of constraints for each pairing (physician type, shift type). Thus, this general framework can fit other contexts, such as the scheduling of physicians in other health departments or police and fire department staff. In fact, the initial motivation of this research was the design of a general physician scheduling algorithm for any hospital department; the ER being the first department for which it was tested.

This study treats ergonomic constraints as hard constraints, although some could also be treated as soft constraints by penalizing any deviation beyond the bounds of the objective function. In this case, weights could be used in the objective function to prioritize the different objectives relative to each other and to other balancing criteria. This extension is quite common and straightforward to apply in the original MIP, but not in the proposed heuristic algorithm. The latter would require adaptation in its three main steps. In the construction step, the *LoC* would include those physicians whose assignment would have the least negative effect on the objective function. In the VNDS step, the

Table A.1
Summary of physician rostering problems literature.

Authors, year	Journal	Problem characteristics								Modeling technique	Solving algorithm	Real data	Implementation
		Planning horizon	Real calendar	Fairness	Employee preferences	N. of different shifts	Variability in shift length	Variability in demand	Working hours balancing				
[22] Beaulieu et al. 2000	HCMS	6 months (SM)	X	X	X	6	X	X	Weekly Monthly	IP	An iterative branch-and-bound (constraints addition)	X	
[10] Bruni and Detti, 2014	ORHC	4–24 Months	X	X	X	2		X		MIP	Branch-and-cut (CPLEX)	X	
[24] Carrasco, 2010	CMPB	12 Months	X	X	X	2 Guard shifts	X	X		MIP	Greedy-random	X	X
[28] Cohn et al. 2009	Interfaces	1 year		X	X	5 On-calls				IP	A multiphase, interactive, iterative branch-and-bound	X	X
[8] Damcı-Kurt et al. 2019	Omega	1 week–12 months	X	X	X	4	Not reported	X		MIP	Enhanced branch-and-cut	X	
[33] Ferrand et al. 2011	Interfaces	8weeks		X	X	3				IP	Branch-and-cut (CPLEX)	X	X (MA)
[57] Gross, Brunner, and Blobner, 2019	HCMS	1 month		X	X	6 Overnights				MILP and MIQP	Branch-and-cut (CPLEX)	X	X
[58] Gunawan and Lau, 2013	JORS	1 week			X	2 in 5 different locations				MIP	- Branch-and-cut (CPLEX): small instances - Greedy + Local Search based algorithm: large-scale problem instances	X	
[59] Hong et al. 2019	INFORMS on AA	1 Month		X	X	7				IP	Branch-and-cut	X	X
[60] Huang, Lee, and Huang, 2016	JIPE	1 Month	X	X	X	3	X	X		IP	Branch-and-bound (LINGO)	X	
[61] Lan et al. 2019	ASOC	1 week		X	X	2				MIP	A hybrid SCA-VNS	X	
[18] Carter and Lapierre, 2001	HCMS	36 weeks		X	X	6	Not reported			IP	Tabu Search	X	X
	HCMS	3 Months	X	X	X	4 + On-call.	Not reported	X		IP	Tabu Search	X	
[23] Puente et al. 2009	C&IE	1 Month	X	X		4	X	X		Not reported	Genetic	X	X

(*continued on next page*)

Table A.1 (*continued*).

Authors, year	Journal	Problem characteristics								Modeling technique	Solving algorithm	Real data	Imple-menta-tion
		Planning horizon	Real calendar	Fairness	Employee preferences	N. of different shifts	Variability in shift length	Variability in demand	Working hours balancing				
[32] Schoenfelder and Pfefferlen, 2018	Service Science	1 Month	X	X	X	19	X	X	Weekly & 26-week	MIP	Branch-and-cut (CPLEX)	X	X (MA)
[20] Tan, Gan, and Ren, 2019	JHE	30 days		X	X	4	X	Not reported		MIP	Branch-and-cut (CPLEX)	X	
[19] Topaloglu, 2006	C&IE	1 Month		X	X	2	X			IP	Branch-and-bound (CPLEX)	X	
[21] Topaloglu, 2009	EJOR	6 Months (SM)	X		X	3	X	X		MIP	Branch-and-bound (CPLEX)	X	X
[62] White and White, 2003	LNCS	28 days	X	X	X	On-calls				Not reported	Constraint logic + Tabu Search	X	X (MA)
[51] Wickert et al. 2020	Annals of OR	4 weeks	X	X	X	36	X		Weekly	LP	Matheuristic: VNS + Small LP	X	
This study	ASOC	1 year	X	X	X	19	X	X	Annually	MIP	A hybrid GRASP based algorithm, G + NO	X	X (Direct)

Mixed-Integer Programming (MIP); Integer Programming (IP); Mixed Integer Linear Program (MILP); Mixed Integer Quadratic Program (MIQP).
SM: Sequence of six 1-month problems.
MA: Manual Adjustment required.

criterion of not transcending the maximum number of infeasibilities among physicians to enable a shift transfer would be replaced by the maximum value of the weighted function of infeasibilities among physicians. In the NFO step, the cost of each arc would be modified to represent the benefit in the objective function of transferring the associated shift from the transferor to the receiver.

The use of NFO models to search large neighborhoods is one of the main features of this methodology. The use of exact methods to solve the network guarantees good, computationally economic, solution improvements, given the small size of the network (there are fewer nodes than physicians). Furthermore, the randomly constructed network favors the repeated use of this improvement step. It is worth mentioning that in the real problem, a narrow range of feasible schedules is obtained for annual hours worked (only two hours in the real case, with a window width of less than 0.05% of the average hours worked, 1751), while the best solutions obtained by the scheduler at the hospital always provide ranges of more than 30 h. Nevertheless, modeling with networks is a rich field that can be exploited to improve the procedure presented here. For example, currently, the costs do not discriminate between arcs, but they could express preferences to balance certain types of shifts or certain types of physicians. The algorithm is designed to build schedules from scratch but to be completely useful in practice; it should also be able to repair solutions. In this case, it would also be used for staff management purposes or for a minimal rearrangement of shifts when a physician is unable to attend work for some reason. However, this is a different problem, which, while requiring its own formulation and solution procedures, can usefully draw on the ideas used to develop the G+NO algorithm. This is a current topic of research.

CRediT authorship contribution statement

M. Cildoz: Methodology, Software, Validation, Writing - original draft. **F. Mallor:** Supervision, Funding acquisition, Project administration, Conceptualization, Methodology, Writing - review & editing. **P.M. Mateo:** Conceptualization, Methodology, Software, Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been supported by grant MTM2016-77015-R (AEI, Spain, FEDER UE). The authors wish to thank all the medical staff of the Emergency Department of the Navarre Hospital Compound, who actively participated in this study.

Appendix A. The literature on physician rostering problems

This appendix includes a table (see Table A.1) containing physician scheduling problems. Their goal is to assign employees to work shifts, considering organizational and regulatory rules, employee skills and preferences, required staffing, and other problem-specific requirements. Column "Publication" cites the paper, and column "Journal" displays the journal in which it is published. The next columns indicate the characteristics of the problem solved in the paper:

- "Planning horizon" is the time period scheduled by the problem (from 1 week to 24 months).
- "Real Calendar": the scheduling problem includes labor days and holidays (Saturday, Sunday and national, regional holidays).
- "Fairness": the workload is balanced (number of different shifts across physicians).
- "Employee preferences": physicians' heterogeneity according to preferences/seniority level/etc. is considered.
- "N. of different shifts": the number of different shifts in the problem solved.
- Variability in shift length: there are shifts with different lengths.
- Variability in demand: the physician-demand is not uniform across different types of days.
- Working hours balancing: the number of hours worked is balanced across physicians.

"Modeling technique" and "solving algorithm" are also included in the table. "Real Data" refers to problems that are based on actual hospital data, and "Implementation" is considered when the paper reports the use of the solution in a real setting. This last column also specifies if the implementation has been direct or a manual adjustment was required.

Appendix B. Notation

Scheduling problem	
Notation	Definition and domain
Parameters	
N	Total number of physicians
P_i	A physician i, $i = 1, \dots, N$,
M	Number of types of physician groups
G_r	Group of physicians of type r, $r = 1, \dots, M$,
n_r	Number of physicians of type r, $r = 1, \dots, M$,
h_r	Workable hours per physician in group G_r over the planning horizon
L	Number of types of shifts
S_j	Group of shifts of type j, $j = 1, \dots, L$
d_j	Length (hours) of shifts of type S_j
m_j	Number of shifts of S_j in the planning period
γ_{rj}	Denotes whether physicians of type r can work a shift S_j (binary)
T	Number of days for the planning horizon. The planning horizon usually spans a year ($T = 365$)
C	Set of shift characteristics
D_c	Set of types of shifts with characteristics in set C
$\#D$	Number of sets of shifts D_c that generate fairness constraints
δ_c	Minimum number of days between shifts that belong to a set D_c
ν_{1c}	Maximum number of shifts in a set D_c assigned to physicians over a time window of w_{1c} days.
w_{1c}	Time window (days) in which there must be no more than a specific number of shifts from set D_c, ν_{1c}
w_{2c}	Time window (consecutive days) that a physician can work a shift belonging to set D_c
U_c	Average number of shifts in D_c per full-time physician able to work such shifts
W_j	Number of shifts of type S_j that should be worked by each full-time physician eligible to do so
β	Weighting factor in the objective function of the general covering problem

Variables of the scheduling problem formulated as Integer Linear Programming (ILP) problem	
X_{ijt}	Binary decision variable which determines whether a physician P_i works the shift Sj on day t
H_c^U	Maximum number of hours worked on shifts with characteristics in C by a physician over the planning horizon
H_c^L	Minimum number of hours worked on shifts with characteristics in C by a physician over the planning horizon
J_c^U	Maximum number of shifts in set D_c worked by a physician over the planning horizon
J_c^L	Minimum number of shifts in set D_c worked by a physician over the planning horizon
$J_{rc}{}^U$	Maximum number of shifts in set D_c worked by a physician P_i, $i = 1, \ldots, N$, of group G_r
$J_{rc}{}^L$	Minimum number of shifts in set D_c worked by a physician P_i, $i = 1, \ldots, N$, of group G_r
Variables of the linear programming model formulated to solve the general covering problem	
Z_{rj}	Decision variables: average number of shifts of type $S_j, j = 1, \ldots, L$, that should be worked by a physician of type G_r, $r = 1, \ldots, M$, in order to cover the demand without exceeding the working hours
F_1	Deviation variable that bounds the absolute value of the differences between the average number of shifts assigned to each group of physicians and the value of reference $\rho_r U_c$ for all sets of shifts D_c.
F_j	Deviation variable that bounds the absolute value of the difference between the average number of shifts S_j assigned to each group of physicians and the value of reference $\rho_r U_j$ for all shifts of type S_j that do not belong to any set D_c
F_2^U, F_2^L	Deviation variables that bound F_j in the interval (F_2^L, F_2^U).
F_3	Deviation variable that bounds the absolute value of the differences between the average number of shifts S_j assigned to each group of physicians and the value of reference $\rho_r U_j$ for all shifts of type S_j that are included in any set D_c.
Greedy random constructive algorithm	
$nshifts\,(t)$	The number of shifts on the tth day
$LoC(j, t)$	List of Candidates who can be assigned shift S_j on day t
z_{ij}^*	Number of shifts of type S_j assigned so far to physician P_i at the moment of assignment, on day t
Z_{iD_c}	Average number of types of shifts S_j in D_c – the set shift type with characteristics in set C - that should be worked by a physician P_i of type G_r in order to cover the demand without exceeding the working hours $Z_{D_c} = \sum_{j \in D_c} Z_{ij}$
$z_{iD_c}^*$	Number of shifts of type S_j in D_c – the set of shift types with characteristics in set C - assigned so far to physician P_i at the moment of assignment, on day t $z_{iD_c}^* = \sum_{j \in D_c} z_{ij}^*$
X_i	Set of shifts assigned to physician P_i in the incumbent solution, that is, $X_i = \{\text{shift } j \text{ of day } t \text{ s.t. } X_{ijt} = 1\}$
$g_j\,(i)$	Greedy function: this is a non-negative definite function. The greater the value of $g_j\,(i)$ for physician P_i, the greater is his/her need to work this shift j in order to meet the reference values Z_{ij}
$g_{Nj}(i)$	Normalized greedy function
$g_{D_c}(i)$	Greedy function for each of the characteristics affected by the assignment of shift j on day t
$g_j\,(i)$	Enhanced greedy function
$p(i)$	The probability of selecting a physician $P_i \in LoC(j, t)$, which depends on his/her value in the greedy function: $p\,(i) = \frac{(g(i))^\alpha}{\sum_{P_i \in LoC(j,t)} (g(i))^\alpha}$
α	Elitism factor of the greedy algorithm construction phase

VNDS: Variable Neighborhood Descent Search	
$\rho(X_i, X_i')$	Distance between schedule solutions for a physician
$\max_{depthSearch}$	Maximum depth in the *VND*
$X' = h_p^k(X)$	Sequence of k shift transfers in which the receiver in one shift transfer is the transferor in the next
$\aleph_k\,(X)$	A neighborhood of depth k
Q	Maximum number of unfulfilled constraints among all physicians
P_Q	The set of physicians that reach this maximum number of non-fulfillments
$\max_{iter\_VND}$	Maximum number of iterations in the *VND*

NFO: Network Flow Optimization	
$H_i\,(X)$	Total working hours
$\overline{H}$	Average working hours
$\max_{iter\_NFO}$	Total iterations of the NFO procedure
LH	Lower limit of the indifference interval
UH	Upper limit of the indifference interval
ε	Factor defining the final window of indifference
$P_{TS}\,(X)$	Group of transferors
$P_{RS}\,(X)$	Group of receivers
$P_{IN}\,(X)$	Group in the indifference interval
WHC	Working hours condition
BSC	Balance shift condition

Appendix C. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2021.107151.

References

[1] P. Knauth, Designing better shift systems, in: Appl. Ergon., Elsevier, 1996, pp. 39–44, http://dx.doi.org/10.1016/0003-6870(95)00044-5.

[2] M. Gendreau, J. Ferland, B. Gendron, N. Hail, B. Jaumard, S. Lapierre, G. Pesant, P. Soriano, Physician scheduling in emergency rooms, in: Pract. Theory Autom. Timetabling VI, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 53–66, http://dx.doi.org/10.1007/978-3-540-77345-0_4.

[3] P. De Bruecker, J. Van Den Bergh, J. Beliën, E. Demeulemeester, Workforce planning incorporating skills: State of the art, European J. Oper. Res. 243 (2015) 1–16, http://dx.doi.org/10.1016/j.ejor.2014.10.038.

[4] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, L. De Boeck, Personnel scheduling: A literature review, European J. Oper. Res. 226 (2013) 367–385, http://dx.doi.org/10.1016/J.EJOR.2012.11.029.

[5] E.K. Burke, P. De Causmaecker, G. Vanden Berghe, H. Van Landeghem, The state of the art of nurse rostering, J. Sched. 7 (2004) 441–449, http://dx.doi.org/10.1023/B:JOSH.0000046076.75950.0b.

[6] B. Cheang, H. Li, A. Lim, B. Rodrigues, Nurse rostering problems—-a bibliographic survey, European J. Oper. Res. 151 (2003) 447–460, http://dx.doi.org/10.1016/S0377-2217(03)00021-3.

[7] M. Erhard, J. Schoenfelder, A. Fügener, J.O. Brunner, State of the art in physician scheduling, European J. Oper. Res. 265 (2018) 1–18, http://dx.doi.org/10.1016/J.EJOR.2017.06.037.

[8] P. Damcı-Kurt, M. Zhang, B. Marentay, N. Govind, Improving physician schedules by leveraging equalization: Cases from hospitals in U.S., Omega 85 (2019) 182–193, http://dx.doi.org/10.1016/j.omega.2018.06.011.

[9] A.J. Ruiz-Torres, J.H. Ablanedo-Rosas, S. Mukhopadhyay, G. Paletta, Scheduling workers: A multi-criteria model considering their satisfaction, Comput. Ind. Eng. 128 (2019) 747–754, http://dx.doi.org/10.1016/j.cie.2018.12.070.

[10] R. Bruni, P. Detti, A flexible discrete optimization approach to the physician scheduling problem, Oper. Res. Health Care 3 (2014) 191–199, http://dx.doi.org/10.1016/j.orhc.2014.08.003.

[11] Ministerio de Empleo Y Seguridad Social, Boletín Oficial Del Estado, BOE-A-2015-11430, 2015, https://www.boe.es/eli/es/rdlg/2015/10/23/2.

[12] R. Stolletz, J.O. Brunner, Fair optimization of fortnightly physician schedules with flexible shifts, European J. Oper. Res. 219 (2012) 622–629, http://dx.doi.org/10.1016/j.ejor.2011.10.038.

[13] A.Y. Zhou, M. Panagioti, A. Esmail, R. Agius, M. Van Tongeren, P. Bower, Factors associated with burnout and stress in trainee physicians, JAMA Netw. Open. 3 (2020) e2013761, http://dx.doi.org/10.1001/jamanetworkopen.2020.13761.

[14] P. Brucker, R. Qu, E. Burke, Personnel scheduling: Models and complexity, European J. Oper. Res. 210 (2011) 467–473, http://dx.doi.org/10.1016/j.ejor.2010.11.017.
[15] M. Gendreau, J.-Y. Potvin, Metaheuristics in combinatorial optimization, Ann. Oper. Res. 140 (2005) 189–213, http://dx.doi.org/10.1007/s10479-005-3971-7.
[16] R.M. Karp, Reducibility among combinatorial problems, in: Complex. Comput. Comput., Springer US, Boston, MA, 1972, pp. 85–103, http://dx.doi.org/10.1007/978-1-4684-2001-2_9.
[17] V. Maniezzo, T. Stützle, S. Voß, Matheuristics: Hybridizing Metaheuristics and Mathematical Programming, 2010, http://dx.doi.org/10.1007/978-1-4419-1306-7.
[18] M.W. Carter, S.D. Lapierre, Scheduling emergency room physicians, Health Care Manag. Sci. 4 (2001) 347–360, http://dx.doi.org/10.1023/A:1011802630656.
[19] S. Topaloglu, A multi-objective programming model for scheduling emergency medicine residents, Comput. Ind. Eng. 51 (2006) 375–388, http://dx.doi.org/10.1016/j.cie.2006.08.003.
[20] M. Tan, J. Gan, Q. Ren, Scheduling emergency physicians based on a multiobjective programming approach: A case study of West China Hospital of Sichuan University, J. Healthc. Eng. 2019 (2019) 1–9, http://dx.doi.org/10.1155/2019/5647078.
[21] S. Topaloglu, A shift scheduling model for employees with different seniority levels and an application in healthcare, European J. Oper. Res. 198 (2009) 943–957, http://dx.doi.org/10.1016/J.EJOR.2008.10.032.
[22] H. Beaulieu, J.A. Ferland, B. Gendron, P. Michelon, A mathematical programming approach for scheduling physicians in the emergency room, Health Care Manag. Sci. 3 (2000) 193–200, http://dx.doi.org/10.1023/A:1019009928005.
[23] J. Puente, A. Gómez, I. Fernández, P. Priore, Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms, Comput. Ind. Eng. 56 (2009) 1232–1242, http://dx.doi.org/10.1016/J.CIE.2008.07.016.
[24] R.C. Carrasco, Long-term staff scheduling with regular temporal distribution, Comput. Methods Programs Biomed. 100 (2010) 191–199, http://dx.doi.org/10.1016/j.cmpb.2010.03.015.
[25] J.O. Brunner, J.F. Bard, R. Kolisch, Flexible shift scheduling of physicians, Health Care Manag. Sci. 12 (2009) 285–305, http://dx.doi.org/10.1007/s10729-008-9095-2.
[26] J.O. Brunner, J.F. Bard, R. Kolisch, Midterm scheduling of physicians with flexible shifts using branch and price, IIE Trans. 43 (2010) 84–109, http://dx.doi.org/10.1080/0740817X.2010.504685.
[27] S. Kraul, Annual scheduling for anesthesiology medicine residents in task-related programs with a focus on continuity of care, Flex. Serv. Manuf. J. (2019) http://dx.doi.org/10.1007/s10696-019-09365-4.
[28] A. Cohn, S. Root, C. Kymissis, J. Esses, N. Westmoreland, Scheduling medical residents at Boston University School of Medicine, Interfaces (Providence) 39 (2009) 186–195, http://dx.doi.org/10.1287/inte.1080.0369.
[29] J.F. Bard, Z. Shu, D.J. Morrice, L.K. Leykum, Constructing block schedules for internal medicine residents, IISE Trans. Healthc. Syst. Eng. 7 (2017) 1–14, http://dx.doi.org/10.1080/19488300.2016.1255284.
[30] J.O. Brunner, G.M. Edenharter, Long term staff scheduling of physicians with different experience levels in hospitals using column generation, Health Care Manag. Sci. 14 (2011) 189–202, http://dx.doi.org/10.1007/s10729-011-9155-x.
[31] L.V. Green, J. Soares, J.F. Giglio, R.A. Green, Using queueing theory to increase the effectiveness of emergency department provider staffing, Acad. Emerg. Med. 13 (2006) 61–68, http://dx.doi.org/10.1197/j.aem.2005.07.034.
[32] J. Schoenfelder, C. Pfefferlen, Decision support for the physician scheduling process at a German Hospital, Serv. Sci. 10 (2018) 215–229, http://dx.doi.org/10.1287/serv.2017.0192.
[33] Y. Ferrand, M. Magazine, U.S. Rao, T.F. Glass, Building cyclic schedules for emergency department physicians, Interfaces (Providence) 41 (2011) 521–533, http://dx.doi.org/10.1287/inte.1110.0563.
[34] E.L. Schreiber, R.E. Korf, L. Angeles, Optimal Multi-Way Number Partitioning, Vol. 65, 2018.
[35] T.A. Feo, M.G. Resende, A probabilistic heuristic for a computationally difficult set covering problem, Oper. Res. Lett. 8 (1989) 67–71, http://dx.doi.org/10.1016/0167-6377(89)90002-3.
[36] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, J. Global Optim. 6 (1995) 109–133, http://dx.doi.org/10.1007/BF01096763.
[37] P. Festa, M.G.C. Resende, Grasp: An Annotated Bibliography, Springer, Boston, MA, 2002, pp. 325–367, http://dx.doi.org/10.1007/978-1-4615-1507-4_15.
[38] N. Mladenović, P. Hansen, Variable neighborhood search, Comput. Oper. Res. 24 (1997) 1097–1100, http://dx.doi.org/10.1016/S0305-0548(97)00031-2.
[39] P. Hansen, N. Mladenović, Variable neighborhood search: Principles and applications, European J. Oper. Res. 130 (2001) 449–467, http://dx.doi.org/10.1016/S0377-2217(00)00100-4.
[40] A. Duarte, J. Sánchez-Oro, N. Mladenović, R. Todosijević, Variable neighborhood descent, in: Handb. Heuristics, Springer International Publishing, Cham, 2018, pp. 341–367, http://dx.doi.org/10.1007/978-3-319-07124-4_9.
[41] R.K. Ahuja, Ö. Ergun, J.B. Orlin, A.P. Punnen, A survey of very large-scale neighborhood search techniques, Discrete Appl. Math. 123 (2002) 75–102, http://dx.doi.org/10.1016/S0166-218X(01)00338-9.
[42] A.P. Punnen, The traveling salesman problem: new polynomial approximation algorithms and domination analysis, J. Inf. Optim. Sci. 22 (2001) 191–206, http://dx.doi.org/10.1080/02522667.2001.10699482.
[43] M. Dror, L. Levy, A vehicle routing improvement algorithm comparison of a greedy and a matching implementation for inventory routing, Comput. Oper. Res. 13 (1986) 33–45, http://dx.doi.org/10.1016/0305-0548(86)90062-6.
[44] F. Grenouilleau, A. Legrain, N. Lahrichi, L.-M. Rousseau, A set partitioning heuristic for the home health care routing and scheduling problem, European J. Oper. Res. 275 (2019) 295–303, http://dx.doi.org/10.1016/j.ejor.2018.11.025.
[45] S.E. Moussavi, M. Mahdjoub, O. Grunder, A matheuristic approach to the integration of worker assignment and vehicle routing problems: Application to home healthcare scheduling, Expert Syst. Appl. 125 (2019) 317–332, http://dx.doi.org/10.1016/j.eswa.2019.02.009.
[46] E. Nikzad, M. Bashiri, B. Abbasi, A matheuristic algorithm for stochastic home health care planning, European J. Oper. Res. 288 (2021) 753–774, http://dx.doi.org/10.1016/j.ejor.2020.06.040.
[47] N.A. Hernández-Leandro, V. Boyer, M.A. Salazar-Aguilar, L.-M. Rousseau, A matheuristic based on Lagrangian relaxation for the multi-activity shift scheduling problem, European J. Oper. Res. 272 (2019) 859–867, http://dx.doi.org/10.1016/j.ejor.2018.07.010.
[48] R.A.M. Gomes, T.A.M. Toffolo, H.G. Santos, Variable neighborhood search accelerated column generation for the nurse rostering problem, Electron. Notes Discrete Math. 58 (2017) 31–38, http://dx.doi.org/10.1016/j.endm.2017.03.005.
[49] A. Legrain, J. Omer, S. Rosat, A rotation-based branch-and-price approach for the nurse scheduling problem, Math. Program. Comput. 12 (2020) 417–450, http://dx.doi.org/10.1007/s12532-019-00172-4.
[50] H.G. Santos, T.A.M. Toffolo, R.A.M. Gomes, S. Ribas, Integer programming techniques for the nurse rostering problem, Ann. Oper. Res. 239 (2016) 225–251, http://dx.doi.org/10.1007/s10479-014-1594-6.
[51] T.I. Wickert, A.F. Kummer Neto, M.M. Boniatti, L.S. Buriol, An integer programming approach for the physician rostering problem, Ann. Oper. Res. (2020) http://dx.doi.org/10.1007/s10479-020-03552-5.
[52] T.A. Feo, M.G.C. Resende, S.H. Smith, A greedy randomized adaptive search procedure for maximum independent set, Oper. Res. 42 (1994) 860–878, http://dx.doi.org/10.1287/opre.42.5.860.
[53] J.L. Bresina, Heuristic-biased stochastic sampling, in: W.J. Clancey, D.S. Weld (Eds.), AAAI/IAAI, Vol. 1, 1996, pp. 271–278.
[54] M.G.C. Resende, C.C. Ribeiro, Greedy randomized adaptive search procedures: Advances and extensions, in: Int. Ser. Oper. Res. Manag. Sci., Springer New York LLC, 2019, pp. 169–220, http://dx.doi.org/10.1007/978-3-319-91086-4_6.
[55] K. "KT" Thulasiraman, S. Arumugam, A. Brandstädt, T. Nishizeki, S. Arumugam, A. Brandstädt, T. Nishizeki, Handbook of Graph Theory, Combinatorial Optimization, and Algorithms, Chapman and Hall/CRC, 2015, http://dx.doi.org/10.1201/b19163.
[56] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows : Theory, Algorithms, and Applications, Prentice Hall, Upper Saddle River, NJ, USA, 1993.
[57] C.N. Gross, J.O. Brunner, M. Blobner, Hospital physicians can't get no long-term satisfaction – an indicator for fairness in preference fulfillment on duty schedules, Health Care Manag. Sci. 22 (2019) 691–708, http://dx.doi.org/10.1007/s10729-018-9452-8.
[58] A. Gunawan, H.C. Lau, Master physician scheduling problem, J. Oper. Res. Soc. 64 (2013) 410–425, http://dx.doi.org/10.1057/jors.2012.48.
[59] Y.-C. Hong, A. Cohn, S. Gorga, E. O'Brien, W. Pozehl, J. Zank, Using optimization techniques and multidisciplinary collaboration to solve a challenging real-world residency scheduling problem, INFORMS J. Appl. Anal. 49 (2019) 201–212, http://dx.doi.org/10.1287/inte.2019.0995.
[60] Y.-C. Huang, P.-T. Lee, T.-L. Huang, A rostering optimization model for physician scheduling in medical department – a case study in district hospital, J. Ind. Prod. Eng. 33 (2016) 533–557, http://dx.doi.org/10.1080/21681015.2016.1180327.
[61] S. Lan, W. Fan, T. Liu, S. Yang, A hybrid SCA–VNS meta-heuristic based on iterated hungarian algorithm for physicians and medical staff scheduling problem in outpatient department of large hospitals with multiple branches, Appl. Soft Comput. J. 85 (2019) 105813, http://dx.doi.org/10.1016/j.asoc.2019.105813.
[62] C.A. White, G.M. White, Scheduling doctors for clinical training unit rounds using tabu optimization, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer Heidelberg, 2003, pp. 120–128, http://dx.doi.org/10.1007/978-3-540-45157-0_8.