Revista de Sistemas de Informação da FSMA
n. 18 (2017) pp. 9-20

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

Fairness in Physician Scheduling Problem in Emergency Rooms

Valdemar Abrão Pedro Anastácio Devesse, *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo*,
Maristela Oliveira dos Santos, *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo*,
Cláudio Fabiano Motta Toledo, *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo*

Fairness in Physician Scheduling Problem in Emergency Rooms

***Abstract*—The aim of fairness in physician scheduling is to build up schedules with high level of compliance of work regulations and individual preferences while violations stemming from requirements are softened and fairly distributed along the workforce. In this paper, we address the fairness in physician scheduling problem in emergency rooms by means of mathematical programming. A minimax MIP and MIP based heuristics relax-and-fix and fix-and-optimize are introduced to create balanced scheduling from the set of physicians. The solutions achieved present high quality in terms of fairness despite the lower bound, which is the main drawback in this problem.**

***Index Terms*—Fairness, Physician Scheduling Problem, MIP, Fix-and-Optimize, Combinatorial Optimization.**

I. Introduction

The aim of fairness in physician scheduling is to build up schedules with high level of compliance of work regulations and individual preferences while violations stemming from requirements are softened and fairly distributed along the workforce. Schedules are usually tailor-made and are designed based on reports of variability and unpredictability of patient demand, what increases the problem's complexity [1].

It is a challenging task to create rosters subject to a huge set of constraints because the constraints are mostly conflicting one to another ([1], [2]). To tackle the complexity, mathematical programming approaches, heuristics and decision making tools have been proposed by operation research and computer science practitioners. However, less attention is specifically devoted to mathematical models considering fairness in physician scheduling problems in emergency rooms, an hospital service that works around the clock and needs prior attention concerning to automation process tools.

In this paper, we address the fairness for physician scheduling problem in emergency rooms by means of mathematical programming. Our approach consists in a mixed-integer-programming (MIP) model and heuristics based on the MIP. First, a minimax model is used to generate feasible schedules, afterwards relax-and-fix and fix-and-optimize heuristics based on the mathematical formulation results are compared with the minimax model. Instances to our experiments are a combination of benchmark instances available at [3], problem characteristics introduced by the present work and data extracted from legislation.

The proposed MIP model handles balance constraints instead of high penalization measure from flexible constraints. The balance constraints generate schedules with evenly distributed deviations among physicians. Furthermore, our research features the ability to generate balanced schedules along the set of physicians and schedules that follow the requirements stated by the problem.

The remainder of our paper is organized as follows. In Section II we present literature review regarding to physician scheduling problem. A problem statement and the relevant mathematical models are presented in Section III. In Section IV we detail the relax-and-fix and fix-and-optimize heuristics applied to tackle our problem. Section V discusses the results achieved and our conclusions follow in Section VI.

II. Literature review

The Physician Scheduling Problem tackled in the literature arise from Personnel Scheduling Problems which deals with the assignment of tasks to be performed by a workforce, while a set of service requirements and contractual agreements is complied. The objective of the assignment is to maximize workers' preferences and/or minimize costs [4]. A schedule can be cyclic or non-cyclic, it is said to be cyclic if physicians are assigned to the same pattern of shifts in a given planning horizon, otherwise the said is said to be non-cyclic.

Likewise, the Physician Scheduling Problem (PSP) can be defined as the allocation of physicians to work shifts and work days considering a range of constraints such as legal regulations, personnel policies, physicians' preferences and other requirements that may be hospital-specific. Physician Scheduling in emergency room is tackled in the literature with certain diligence, in spite of the scarcity. Regarding to PSP as a whole, most of the applied techniques include exact methods, followed by heuristics, according to [5]. In the same paper, authors made the state of the art of the physician scheduling problem.

The first incidence of a mathematical model to obtain feasible schedules considering the case of a major hospital in Montreal region is presented in [6]. The problem was modeled as a multi-objective integer programming model with a set of constraints divided into four categories, namely: compulsory constraints, ergonomic constraints, distribution constraints and goal constraints. In each group of constraints, constraints can be easily modified and new constraints can be included, in case of new scenarios. The approach produced in a shorter time better schedules than those produced by the human expert.

A generic emergency room physician scheduling problem is defined in [7] from characteristics found in six hospitals. The authors proposed a modification on existing scheduling rules that allowed to develop automated solution techniques which produce better schedules and reduced the time spent build up the schedules. The authors in [8] applied an hybridization of Constraint Programming (CP), Local Search and Genetic Algorithm (GA). The model flexibility is handled by CP, this flexibility allows to describe different problems. On the other hand, algorithm flexibility is achieved by the combination of Local Search and GA that provides efficiency for most of the problems with different solution spaces and objective functions.

A general CP is developed by [9] to solve both nurse and physician scheduling problems. The approach is designed to handle a variety of situations, where with global constraints it becomes possible to capture the large number of rules considered in these problems. A Goal Programming model to schedule medicine residents in a emergency room is presented by [10], where monthly schedules are defined within a reduced time and effort. The rules of the problem are divided in two categories, namely, hard and soft constraints, and the objective function minimizes relevant deviations from soft constraints. GA defines schedules for for an emergency room department in [1] . A heuristic-schedule builder produces an initial population of feasible solutions that is evolved by GA by applying crossover and repair operators.

The authors in [11] addressed the problem of building cyclic schedules to emergency department physicians. An integer programming model is proposed to generate such cyclic schedules. To infer the level of satisfaction achieved, interviews with physicians and statistics from the scheduling showed that these schedules provide predictability and well-balanced work patterns.

Mathematical models to deal with master physician schedules are also present in the literature. Three mathematical programming models are formulated by [12] to describe different problem variants. A local search heuristic solves large-scale instances which cannot be dealt with CPLEX solver. The method is tested on real scenarios from a surgery department as well as on randomly generated problem instances.

Besides the physician scheduling, other aspects to take into account for a robust management of the service are planning of limited resources and patients flow. The authors in [13] consider planning procedure for a master planning of elective and emergency patients, while allocating at best the available hospital resources.

The fairness in schedules is studied by [2] applying a methodology for increasing the satisfaction of nurses, regarding to personal schedule. A set of new evaluation models for nurse rostering is proposed from situations found in a hospital from Belgium. Data sets for experiments are generated and cooperative meta-heuristic are applied to generate fair nurse rosters. An agent-based framework for cooperative meta-heuristic search is described to assert fairness. The agent is conceived to execute different meta-heuristic and local search combinations with different parameter settings.

Physician Scheduling Problem is formulated as a MIP model in [4]. The model is flexible since it allows modifications and new additions for representing different constraints. Comparisons are done with literature models in terms of their capability to represent the most practical requirements made. A real case was tested, from which some instances with large planning horizons were solved to optimality applying a Branch & Cut procedure.

Application of MIP based heuristics combined with local search techniques in timetabling as a whole are dealt with in the literature with less emphasis. The authors in [14] introduced a mixed integer programming model and a fix and optimize heuristic combined with variable neighborhood descent to high-school timetabling problem. They applied three types of decomposition, using a fix and optimize heuristic, to solve two sets of benchmark instances.

An integration of Physician and Surgery Scheduling Problem (IPSSP) is introduced by [15]. A model handles both problems considering their main constraints and objectives. The IPSSP is also discussed in [16] with six different decomposition-based heuristic. Solutions are built by the heuristics according rules from surgery and physician scheduling problem.

A stochastic optimization formulation is described in [17] for a shift scheduling problem of emergency department. This problem is solved using a deterministic Sample Average Approximation (SAA) method. Model simplifications are assumed in order to turn it easily tackled by means of linear programming optimization. More recently the case of assigning rooms to physician in emergency departments is studied by [18]. In this work the authors study strategies for room assignment and doctor working routines, where Pareto-efficient combinations are determined with respect to the performance measures.

Comparing with the most of the models in the literature, our model consider dynamic demands and variant number of consecutive days, while most of literature works assume a fixed demand. Regarding to constraints, our problem presents a set of balance constraints to gather the deviations from flexible constraints, while most of the models in the literature consider balance by establishing the minimization of deviations as a measure of fairness.

Tables I and II summarize our literature review.

III. Problem Statement

Each instance of the PSP in emergency room may vary since the related constraints change from one hospital to another. We compiled and extracted general characteristics from models presented by [6] and [10]. The fairness ideas described on this section are based on [2] that studied fairness in nurse scheduling problem.

We also carried out a study at Hospital da Santa Casa de Misericórdia de São Carlos, where other characteristics from the PSP where added to the problem defined on this section. Basically, there is a group of physicians that must be assigned through a time horizon of one month, which can be extended up to a year. The set of physicians can have full-time or part-time contracts. Physicians with full-time contracts work 40 hours a week and they can work extra hours, on the other side physicians with part-time contracts work 24 hours and cannot make extra hours. These physician can be assigned to regular shifts, namely morning, afternoon and night or they can be on duty, day duty or an night duty. For each shift, there is a minimum amount of physician that must be available to cover each shift in a daily basis, which we assume to be the demand. In some hospitals such as Santa Casa, physicians who completed any shifts cannot work in other shifts at the same day.

Every physician has the right to go on holidays or leaves in case of an incident, such as periods of illness, personal issues, and these requirements must be satisfied. Another important rule is that, physicians cannot be assigned to night and in the following day be assigned to an afternoon. These requirements led us to contemplate constraints related to antagonists shifts. There is also a concern with workforce skills level to increase the quality of service. For example, a minimum number of surgeons, specialist in internal medicine and senior physician is mandatory for emergency room where delicate cases can happen. In our problem, we do not label a determined medical specialty for the sake of generalization, but we assume that there is a minimum of senior physicians that must be assigned to regular shift during the week.

Every time a physician is assigned, a maximum number of consecutive days of work is established to allow he/she to rest properly. In order to increase a physician satisfaction level with his/her schedule, offsets are also applied, for example, if a physician was assigned to three night shifts, he/she must rest the three next days. However, this kind of reward is not dealt with in our problem, we only consider the fact that a physician rests at least one day after completing the maximum number of consecutive days. Other relevant welfare/ergonomic constraints are related to intentional leaves on Mondays after being assigned to any shift in the previous Sunday. In our case, if a physician is assigned to any shift in the Sunday, he/she will rest in the following Monday. Moreover, we must consider the fact that physician have a limit of hours on duty, so his/hers circadian cycle may not be compromised.

The mathematical formulation proposed here is divided in three classes of constraints: hard, flexible and balance constraints. Hard constraints are requirements that must be complied:

- Demand coverage must be granted, for example, a minimum of physicians must be assigned to a given shift;
- Physicians must not be assignment to antagonist shifts;
- Workload balance;
- Physicians have the right to vacations, holidays and leaves;
- Physicians have maximum number of consecutive days;
- Whenever a physician is assigned in a Sunday, he must rest in the following Monday;
- Every physician must be assigned but his/her limits of hours on duty must not be exceeded.

Flexible constraints are related to requirements that heads of emergency department wish to accomplish but can be violated. These constraints include weekly hours completeness, desired shift and target duties. In our case, if a physician wishes for a set of days and shifts to work, it is recommended that he/she accomplishes firstly all hard requirements, from which we consider the flexible constraint related to required shift. Thus, we consider that physician's assignment on duties is preferred to reach the target number. If not, a penalty is assumed to each unit of duties that was violated. At last, the target number of hours is an aspect that must be taken into account. In case of violation, a penalty is assumed to each unit of hour. To summarize, our flexible constraints are:

- Complete desired shift of the month;
- Complete the target number of weekly hours;
- Complete the target number of duty of the month.

The balance constraints, in turn, are related to flexible constraints management. These constraints include weekly hours balance, desired shift balance and target duties balance. For example, if there are deviations from the desired amount of weekly hours, the maximum of the deviation is fetched and minimized by the objective function, thus, softening hour assignment. To summarize, our balance constraints are:

- Desired shift of the month balance;
- Target number of weekly hours balance;
- Target number of duty of the month balance.

The minimax approach is suitable for optimal selection of parameters [20] in discrete optimization problems. In our problem, the minimax is modeled based on two main aspects: the maximization of deviations from flexible constraints, followed by their minimization in the objective

TABLE I: Literature review: techniques

Constraint Programming	[8], [9]
Genetic Algorithm	[1],
Goal Programming	[10]
Integer Programming	[6],[4], [7], [16], [19], [15], [12], [11], [2]
Local Search	[8], [14], [16], [2], [7]
Stochastic Optimization	[17]

TABLE II: Literature review: type of problems

Physician Scheduling on Emergency Room	[6], [7], [8], [10],[1] , [11]
Physician Scheduling for General Purpose	[9], [4]
Physicians Scheduling on Surgery Room	[19], [15], [16],
Surgery Room Scheduling	[16], [15],[12]
Others	[2], [14]

function. The proposed mathematical formulation for the PSP is explained next.

In our formulation we assume the first day of the month as a Monday and every month as a set of 28 days.

Sets

I	Set of physicians
J	Set of days
K	Set of shifts
M	Set of months
W	Set of weeks.

Subsets

I^{FT}	Set of physicians with full-time contracts ($I^{FT} \subset I$)
I^{PT}	Set of physicians with part-time contracts ($I^{PT} \subset I$)
JH_i	Set of days of leave required by the i-th physician ($JH_i \subset J$)
$J_{im}^{\mathtt{RM}}$	Set of required days by the i-th physician in the m-th month ($J_{im} \subset J$)
K_i^R	Set of required shifts by the i-th physician ($K_i^R \subset K$)
$K^{\mathtt{M}}$	Morning shifts
$K^{\mathtt{A}}$	Afternoon shifts
$K^{\mathtt{N}}$	Night shifts
$K^{\mathtt{D}}$	Day duty
$K^{\mathtt{E}}$	Night duty

Parameters

c	Maximum of consecutive days.
D_{jk}^-	Minimum demand of k-th shift in the j-th day
D_{jk}^+	Maximum demand of k-th shift in the j-th day
F_{im}	Target number of shift to i-th physician in the m-th month
h_k	Length of k-th shift in hours
H_i	Target hours of i-th physician
S_{im}	Target duties of i-th physician in the m-th month
U_{im}^D	Maximum amount of hours that i-th physician completes in the m-th month in a day duty
U_{im}^E	Maximum amount of hours that i-th physician completes in the m-th month in an evening duty

Penalties

wd_i^+	Positive penalty associated to duty deviations
wd_i^-	Negative penalty associated to duty deviations
wh_i^+	Positive penalty associated to hour deviations
wh_i^-	Negative penalty associated to hour deviations
ws_i^+	Positive penalty associated to required shift deviations.
ws_i^-	Negative penalty associated to required shift deviations.

Decision Variables

x_{ijk}	1, if i-th physician is assigned to k-th shift of j-th day 0, otherwise.

Deviation Variables

dd_{im}^+	Positive deviations for uncompleted duties
dd_{im}^-	Negative deviations for uncompleted duties
hd_{iw}^+	Positive deviations for uncompleted hours
hd_{iw}^-	Negative deviations for uncompleted hours
sd_{il}^+	Positive deviations for uncompleted shifts
sd_{il}^-	Negative deviations for uncompleted shifts

Balance Variables

$maxDD_i^+$	Positive balance for uncompleted duties
$maxDD_i^-$	Negative balance for uncompleted duties
$maxHD_i^+$	Positive balance for uncompleted hours
$maxHD_i^-$	Negative balance for uncompleted hours
$maxSD_i^+$	Positive balance for uncompleted shifts
$maxSD_i^-$	Negative balance for uncompleted shifts

A. Objective Function

$$\begin{aligned} \text{minimize} \sum_{i \in I} \Big(& wd_i^+ maxDD_i^+ + wd_i^- maxDD_i^- \\ & + wh_i^+ maxHD_i^+ + wh_i^- maxHD_i^- \\ & + ws_i^+ maxSD_i^+ + ws_i^- maxSD_i^- \Big) \end{aligned} \tag{1}$$

The objective function is the weighted sum of maximum deviations from the balance constraints (19) - (24).

B. Hard Constraints

H1: Demand coverage

$$\sum_{i \in I} x_{ijk} \geq D^{-}_{jk} \quad j \in J, k \in K \tag{2}$$

$$\sum_{i \in I} x_{ijk} \leq D^{+}_{jk} \quad j \in J, k \in K \tag{3}$$

$$\sum_{i \in I^{\mathtt{S}}} x_{ijk} \geq 1 \quad j \in J, \quad k \in K^{\mathtt{M}} \cup K^{\mathtt{A}} \cup K^{\mathtt{N}} \tag{4}$$

Constraints (2) and (3) are related to demand coverage. The amount of physicians must be between a minimum and maximum values. A minimum of physicians must be usually assigned, but an addition of physician can be needed in some exceptional cases. Constraints (4) are related to the minimum amount of senior physicians in regular shifts.

H2: Workload

$$\sum_{k \in K} x_{ijk} \leq 1 \quad i \in I, j \in J \tag{5}$$

Constraints (5) guarantee that a physician is assigned to only one shift in one day.

H3: Assignment to antagonist shifts

$$\sum_{k \in K^{\mathtt{A}}} x_{i(j-1)k} + \sum_{k \in K^{\mathtt{M}}} x_{ijk} \leq 1 \quad i \in I, j \in J \tag{6}$$

$$\sum_{k \in K^{\mathtt{A}}} x_{i(j-1)k} + \sum_{k \in K^{\mathtt{D}}} x_{ijk} \leq 1 \quad i \in I, j \in J \tag{7}$$

$$\sum_{k \in K^{\mathtt{N}}} x_{i(j-1)k} + \sum_{k \notin K^{\mathtt{N}}} x_{ijk} \leq 1 \quad i \in I, j \in J \tag{8}$$

$$\sum_{k \in K^{\mathtt{E}}} x_{i(j-1)k} + \sum_{k \in K^{\mathtt{M}} \cup K^{\mathtt{A}} \cup K^{\mathtt{D}}} x_{ijk} \leq 1 \quad i \in I, j \in J \tag{9}$$

According to constraints (6) - (9), a physician will not be assigned to antagonists shifts. For example, if a physician is assigned to afternoon shift in a day, he/she cannot be assigned to morning shift or day duty in the following day. These constraints allow physician to have a minimum amount of rest hours.

H4: Holidays and leaves

$$x_{ijk} = 0 \quad \forall i \in I, j \in JH_i, k \in K^R_i \tag{10}$$

Periods of rest and leaves are handled by constraints (10).

H5: Consecutiveness

$$\sum_{f=j-c}^{j} \sum_{k \in K} x_{ifk} \leq c \quad i \in I, j \in J \tag{11}$$

There is a maximum amount of consecutive days of assignment for physician. We handle these requirements by constraints (11).

H6: Hour limits on duty

$$\sum_{j \in J^{\mathtt{M}}} \sum_{k \in K^{\mathtt{D}}} h_k x_{ijk} \leq U^D_{im} \quad i \in I, m \in M \tag{12}$$

$$\sum_{j \in J^{\mathtt{M}}} \sum_{k \in K^{\mathtt{E}}} h_k x_{ijk} \leq U^E_{im} \quad i \in I, m \in M \tag{13}$$

Physician must not exceed the hour limits on duty during a month. This requirement is tackled by constraints (12) and (13).

H7: Mondays after Sundays

$$\sum_{k \in K} (x_{i(j-1)k} + x_{ijk}) \leq 1 \quad i \in I, j \in \{1, 8 \ldots, |J| - 6\} \tag{14}$$

Physicians must not be assigned at Monday if they were assigned in a Sunday.

C. Flexible constraints

F1: Shift completeness during a month

$$\sum_{j \in J^{\mathtt{RM}}_{im}} \sum_{k \in K} x_{ijk} - sd^{+}_{im} + sd^{-}_{im} = F_{im} \quad i \in I^{\mathtt{FT}}, m \in M \tag{15}$$

The first set of flexible constraint is related to shift completeness during a month. It is preferred that a permanent physician completes every shift that he/she has required in a month. From constraints (15), if the target number of shifts is not reached, we add to corresponding surplus variables, sd^{-}_{im}, the number of missing shift, otherwise if the target number of shifts is exceeded, we subtract from the corresponding surplus variable, sd^{+}_{im}.

F2: Hour distribution

$$\sum_{j \in J_w} \sum_{k \in K} h_k x_{ijk} - hd^{+}_{iw} + hd^{-}_{iw} = H_i \quad i \in I^{\mathtt{FT}}, w \in W \tag{16}$$

$$\sum_{j \in J_w} \sum_{k \in K} h_k x_{ijk} + hd^{+}_{iw} = H_i \quad i \in I^{\mathtt{PT}}, w \in W \tag{17}$$

Constraints (16) and (17) are related to hour completeness in a week horizon. Extra hours and missing hours are considered to permanent physician, while part-time physicians consider only missing hours. Variables hd^{+}_{iw} and hd^{-}_{iw} store the values of extra hours and missing hours, respectively, if a physician makes more hours than the targeted or less than expected.

F3: Duty distribution during the month

$$\sum_{j \in J^{\mathtt{M}}} \sum_{k \in K^D \cup K^E} x_{ijk} - dd^{+}_{im} + dd^{-}_{im} = S_{im} \quad i \in I, m \in M \tag{18}$$

According to constraints (18), it is preferable that every physician completes the predefined number of duties. If the target number of duties S_{im} is reached, no deviation exists. Otherwise, surplus variables dd^{-}_{im} stores the amount of missing duties, or the surplus variables dd^{+}_{im} stores the amount of exceeded duties.

B1: Required shifts balance during the month

$$maxSD^{+}_i \geq sd^{+}_{im} \quad \forall i \in I, m \in M \tag{19}$$

$$maxSD^{-}_i \geq sd^{-}_{im} \quad \forall i \in I, m \in M \tag{20}$$

Constraints (19) - (20) check the maximum deviations of shift during a month. This value is then minimized in the objective function.

B2: Weekly hours balance

$$masxHD^{+}_i \geq hd^{+}_{iw} \quad \forall i \in I, w \in W \tag{21}$$

$$maxHD^{-}_i \geq hd^{-}_{iw} \quad \forall i \in I, w \in W \tag{22}$$

The maximum deviations of hour during a week are tackled by constraints (21) - (22). The maximum deviation, stored in variables $masxHD^{+}_i$ and $masxHD^{-}_i$, will be minimized in the objective function in order to balance the amount of deviations.

B3: Duty balance

$$maxDD_i^+ \geq dd_{im}^+ \quad \forall i \in I, m \in M \tag{23}$$

$$maxDD_i^- \geq dd_{im}^- \quad \forall i \in I, m \in M \tag{24}$$

The maximum deviations of duties during a month is controlled by constraints (23) - (24). For each physician i, the variables $maxDD_i+$ and $maxDD_i-$ store the values of the surplus variables that will be minimized in the objective function.

Variables' Domain

$$x_{ijk} \in \{0,1\} \quad \forall i \in I, j \in J, k \in K \tag{25}$$

$$dd_{im}^+, dd_{im}^- \in \mathbb{R}_+ \quad i \in I, m \in M \tag{26}$$

$$sd_{im}^+, sd_{im}^- \in \mathbb{R}_+ \quad i \in I, m \in M \tag{27}$$

$$hd_{iw}^+, hd_{iw}^+ \in \mathbb{R}_+ \quad i \in I, w \in W \tag{28}$$

IV. MIP-Heuristic approaches

In this section we'll describe the two mixed integer programming based heuristics and how we used both to solve our problem.

A. The relax-and-fix heuristic

Relax-and-fix (RF) is a constructive heuristic that was proposed by [21] and operates as follows: from the main problem (Q), the set of integer variables is divided in different subsets R from which we can describe each as Q^r such that $r = 1, \ldots, R$, so $Q = Q^1 \cup \ldots \cup Q^R$ and a sub-problem MIP^r is solved iteratively. Assuming the the first sub-problem MIP^1 from the subset Q^1 the variables are set to integer, the remaining variables from the subsets $(Q^2 \ldots Q^R)$ are linearly relaxed and the sub-problem is solved. After treating MIP^1, the subset of integer variables Q^1 is fixed with the incumbent values. If the solution is feasible, the procedure is repeated for the remaining sets $Q^2 \ldots Q^R$, else, it stops and there is no solution applying the current partitioning strategy. If the partitioning strategy works, the solution of MIP^R solves the original problem. Two types of partitioning are proposed by work: Relax-and-Fix with Week Partitioning (RFWP) and Relax-and-Fix with Shift Partitioning (RFSP).

RFWP is inspired on day decomposition addressed by [14] in a fix-and-optimize heuristic for the high school timetabling problem. We choose a partitioning of 2 weeks without overlapping due to the high number of integer variables to be considered in each iteration. Overlapping consists of a partition where variables are set to be integer but they are not immediately fixed. When using the RFSP, we choose the shifts that are set to integer and fixed after solving the sub-problem.

Algorithm 1 presents the overall Relax-and-Fix heuristic procedure from which the partitioning strategies are applied.

Algorithm 1: *Relax-and-fix* pseudo-code

Set each partition Q^r, $r = 1, \ldots, R$;
Relax all variable from the set Q;
Set the solution as X_{MIP}^r as the solution of MIP^r sub-problem, $r = 1, \ldots, R$;
$r = 0$;
$MIP^r = \emptyset$
while $r < R$ *&* $time \leq Tmax$ **do**
    r=r+1;
    Set to integer the variables from Q^r subset and solve MIP^r subproblem
    Fix the set of variables $Q^1 \cup \ldots \cup Q^{r-1}$ in the value of incumbent solution
    Introduce the incumbent solution as the starting to the new sub-problem
    if MIP^r *is infeasible* **then**
        The MIP problem is infeasible using the selected partitioning strategy;
    end
end

Figures 1 and 2 illustrates the stages of relax and fix heuristic for a planning horizon of 12 periods, partition size of 4 periods, considering every shift of the day, where 3 physicians will be assigned. For each iteration of the method, the number of relaxed variables is reduced according to the size of the partition as we see in Figure 2. The named Free variables will be optimized and they define the sub MILP model to be solved. The other variable are Fixed or Relaxed, where Fixed are those whose binary values were already optimized and Relaxed variable have continuous values within $[0, 1]$. The heuristic ends when there are no more Relaxed variables in the sub-problem.

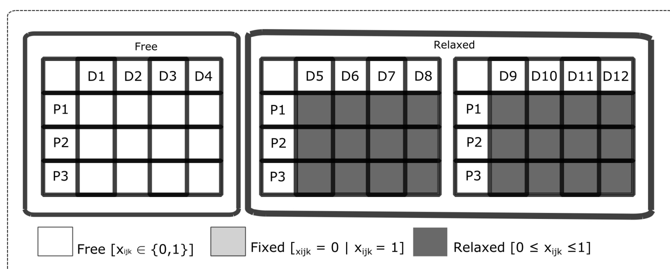

Fig. 1: Relax and fix heuristic - stage 1

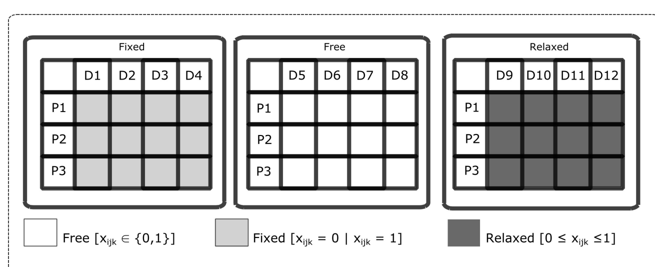

Fig. 2: Relax and fix heuristic - stage 2

B. *Fix and optimize heuristic*

Fix-and-Optimize was first proposed by [22] where, assuming a partitioning strategy of integer variables in the set Q^r such that $r = 1, \ldots, R$, all variables are re-optimized and fixed from an initial solution. The procedure is repeated, using a partitioning strategy, until a stopping criteria is reached.

In the present paper, the partitioning strategies for Relax-and-Fix are the same applied by Relax-and-Fix. Thus, several combinations are evaluated:

- FOWP - Fix-and-Optimize with Week Partitioning: This heuristic uses the week partitioning and initial solution from RFWP.
- FOSP - Fix-and-Optimize with Shift Partitioning: This heuristic uses the shift partitioning and initial solution from RFSP.
- FOSWP - Fix-and-Optimize with Shift Partitioning and Week Partitioning Construction: This heuristic uses the shift partitioning and initial solution from RFWP.
- FOWSP - Fix-and-Optimize with Shift Partitioning: This heuristic uses the week partitioning and initial solution from RFSP.

Algorithm 2 presents the overall Fix-and-Optimize heuristic procedure and Table III describes some variables used by Algorithm 2.

TABLE III: Fix-and-Optimize Variables

P^{fix}_{X,MIP^r}	Set of fixed variables of MIP^r
P^{free}_{X,MIP^r}	Set of free variables of MIP^r
r	Iteration counter
X^{inc}	Incumbent solution
X^{prev}	Former solution
X	Best solution

Algorithm 2: *Fix-and-optimize* pseudo-code

```
r = 0;
Set the solution as X_{MIP_r} as the solution of relax-and-fix
  heuristic;
while r < R & time ≤ Tmax do
    r=r+1;
    X^{prev} ← X^{inc}
    foreach decompositionType do
        foreach MIP_r do
            Compute P^{fix}_{X,MIP_r} and P^{free}_{X,MIP_r}
            Solve MIP_r
            if X^{inc}.value > X^{prev}.value then
                X ← X^{inc}
            end
        end
    end
end
```

Figures 3 and 4 illustrate the two main stages of fix and optimize heuristic. There are no relaxed variables, since this is an improvement heuristic. Thus, variables previously fixed become free aiming to be re-optimized, while the remaining variables stay fixed.

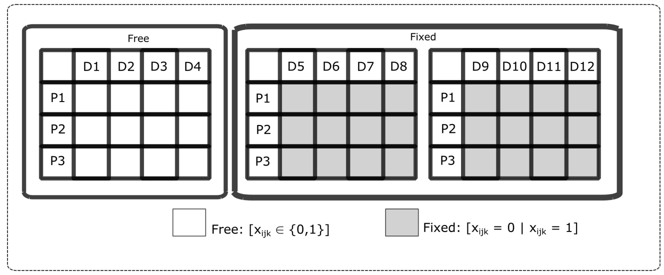

Fig. 3: Fix and optimize heuristic - stage 1

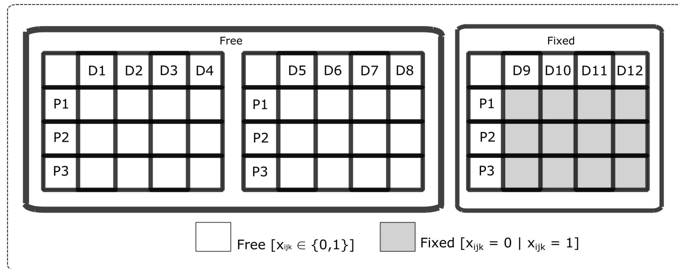

Fig. 4: Fix and optimize heuristic - stage 2

V. Computational Experiments

The experiments were run in a Intel Xeon E5-2680v2 and 2.8 GHz with 128 GB of RAM. The mathematical model was coded using callable libraries from IBM ILOG Cplex 12.6 optimization solver.

A. *Instances Generation*

The instances were generated based on a combination of a real-world schedule and benchmark instances. The real-world schedule was manually generated from a scenario with 10 physicians distributed along 3 shifts and 2 duties during a week. The benchmark instances were taken from [3] and we extracted data related to leaves, holidays and required days of work.

We assumed a demand of 2 physicians in the morning, 1 in the afternoon, 1 in the night, 1 to day duty and 1 to night duty. During the weekend, demand is available only for duties, while demand for shifts are blocked. The schedules are defined for time periods of 1, 2, 3, 6 and 9 months. Data related to maximum number of working hours and other legislation issues were addressed based on [23] and [24].

Details of the instances are presented in the Table IV. Each month is related to a class of instances SET, in which each instance is label $InstanceND\_NW$, where ND is the amount of physicians and NW is the amount of weeks.

The weight setting in the objective function is made upon priority measure of each requirement, inspired in the Analytical Hierarchy Process as proposed by [25]. This process consists in comparing the flexible constraints by establishing the ranking and assignment in order of priority, based on those with greater weights.

The experiments were conducted with the following parameters:

- Execution time limit (TL): 3600 s

TABLE IV: Instances characteristics

Class	Instance	Amount of physicians	Amount of shifts	Amount of Days	Holidays	Desired Days
SET#1	Instance10_4	10	5	28	2	4
	Instance20_4	20	5	28	2	5
	Instance30_4	30	5	28	2	5
	Instance40_4	40	5	28	2	5
SET#2	Instance10_8	10	5	56	-	8
	Instance20_8	20	5	56	-	8
	Instance30_8	30	5	56	-	10
	Instance40_8	40	5	56	2	5
SET#3	Instance10_12	10	5	84	2	2
	Instance20_12	20	5	84	-	14
	Instance30_12	30	5	84	-	9
	Instance40_12	40	5	84	-	14
SET#4	Instance10_24	10	5	168	2	33
	Instance20_24	20	5	168	5	30
	Instance30_24	30	5	168	8	24
	Instance40_24	40	5	168	1	11
SET#5	Instance10_36	10	5	252	2	47
	Instance20_36	20	5	252	8	29
	Instance30_36	30	5	252	8	29
	Instance40_36	40	5	252	4	44

TABLE V: Weights

Constraint	Variable	Priority	Weight
(17)	$maxSD^+_{im}$	3	0,05
(17)	$maxSD^-_{im}$	3	0,05
(18)	$maxHD^-_{iw}$	1	0,5
(18) - (19)	$maxHD^+_{iw}$	2	0,45
(20)	$maxDD^+_{im}$	4	0,015
(20)	$maxDD^-_{im}$	4	0,015

- Memory limit : 128 GB
- Cplex Emphasis: Balanced
- Populate Limit: Default
- Number of Shift Partitions: 5
- Number of Week Partitions: 2
- Heuristics Execution time limit: 1800 s each
- Improvement trials: 3.

B. Experiments

First, we compare the behaviour of Cplex solver when solving our model setting with and without Populate Strategy. This strategy generates multiple solutions to the MIP and fills the pool of solutions. Thus, populate strategy increases the probabilities of the branching strategy, applied by the solver to cover the optimal node. Figures 5 and 6 shows the objective function values (upper bounds) and the lower bounds by, respectively, disabling and enabling populate method from Cplex. The Cplex behaviour is evaluated by scheduling an instance with 10 physicians within 1 month.

There is a quick decrement in the first seconds of execution by disabling the populate method as shown by Figure 5. Also, the objective function values trend to decrease while the lower bounds remains steady.

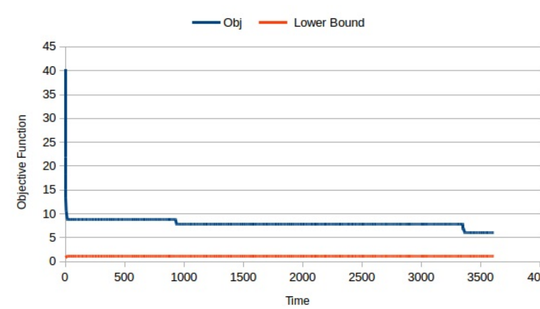

Fig. 5: Minimax behavior without populate Strategy

In the other hand, by enabling the populate method in Figure 6, the objective function value is smaller than the latter one. However the values trend to stay steady after a quick decrement. Thus, when disabling the populate strategy, solutions have a high value of objective function in contrast to enabling the populate strategy.

Figures 7 and 8 illustrates the Cplex performance by enabling the populate method for two other instances. Figure 7 has the results solving an instance with 10 physicians and 1 month, while Figure8 solves an instance with 40 physician and 1 month.

The objective function value decreases quickly during the first 1000s as shown by Figure 7, while there is a steady behavior across the time in Figure8.

Figures 9 and 10 show the same evaluation by solving an instance also with 10 physicians (Figure9) and 40

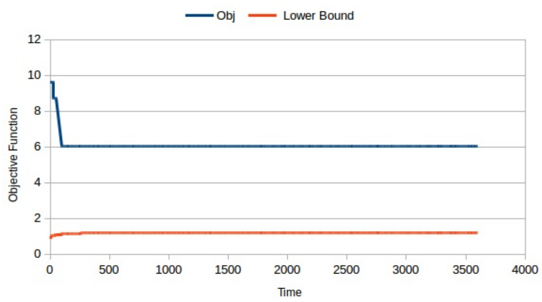

Fig. 6: Minimax behavior with populate Strategy

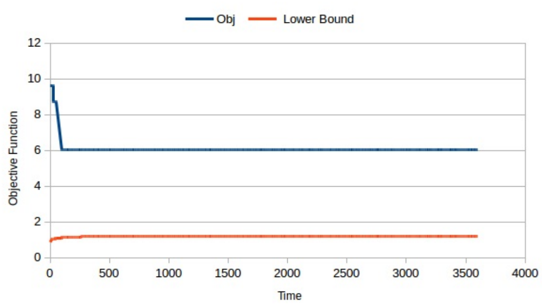

Fig. 7: Cplex with populate strategy for instance with 10 physicians and 1 month.

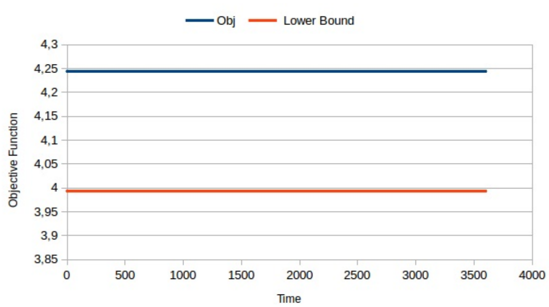

Fig. 8: Cplex with populate strategy for instance with 40 physicians and 1 month.

physicians (Figure10) to be scheduled in 3 months. There is a more regular decrement of the objective function value over time, in spite of ending steady. The lower bound in turn, remains steady in both cases.

The average computational results for all sets of instances are presented in Table VII, while results for each instances are shown by Table VI. From the Table VII we notice that this well constrained problem led the MIP based heuristics to perform poorly on average, as they derive from the linear relaxation of the problem. The partitioning strategies were combined and the difference was not near slight. Regarding to average values of gap, the **FOSP** is more balanced comparing to the rest of the strategies. This strategy performed better in sets 1, 3 and 4, tied with **FOWP** in set 2 and lost in set 5 to the same. The other strategies, **FOSWP** and **FOWSP** had a poor performance on the whole for each set of instances, with

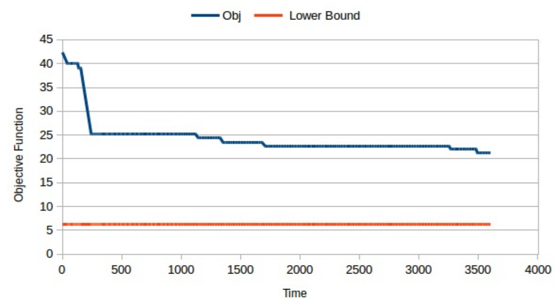

Fig. 9: Minimax on Cplex with populate strategy for instance with 10 physicians and 3 months.

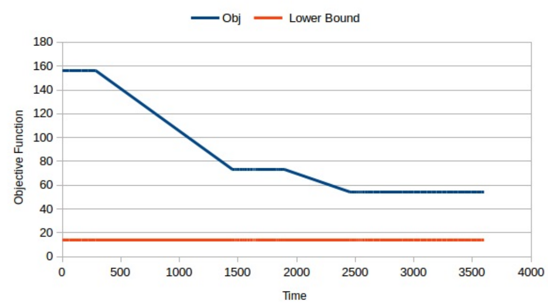

Fig. 10: Minimax on Cplex with populate strategy for instance with 40 physicians and 3 months.

high average values of gap.

Table VI shows that competitive scores comparing to our model were found in some instances applying **FOWP** and **FOWSP** strategies. For example, instances $Instance10\_4$, $Instance30\_4$, $Instance20\_8$, $Instance10\_12$, $Instance10\_24$ returned a smaller value of objective function within a significantly reduced time.

A scheduling in TableVIII, at the end of this paper, illustrates a solution for one week plan. Physicians, labeled as 'PHY#', can be assigned to M - Morning, A - Afternoon, N - Night, D1 - Day Duty or D2 - Night Duty. They can be assigned by displaying a value 'Ass' in the corresponding cell that will be empty, if no assignment is done. Physicians 'PHY[1]' to 'PHY[8]' are senior/full-time physicians and the remaining are part-time. Days 1 to 5 are weekdays starting from Monday. The demand established was fulfilled, but there are deviations. Figure 11 shows that one physician missed 2 hours and 5 physicians completed 1 extra hour each.

Also, from TableVIII and Figure11, there is at least one senior physician (physicians from 1 to 8) for every common shift during the week. This shows that there is no violation of constraints (4). A stretch of 3 consecutive days is not exceeded by any physician, so the constraints of maximum of consecutive day (constraints (11)) are being complied. The assignment of part-time physicians,'PHY[9] and PHY[10]', is made in such a way that they are assigned as less as possible and still completing demanded 24 hours a week. In this example, it is clear that no extra hours were completed by physician 9 and 10. This illustrates the effort to comply most of the constraints, including the flexible ones.

For an accurate appreciation of the impact of balance constraints in our model, Figure 12 shows the difference between the inclusion of balance constraints and the exclusion for all instances. Thus, it is possible to compare the behaviour of the extra extra hours with balance constraints against the deviations of extra hours without balance constraints. From the 1^{st} to the 30^{th} physician the stretch that describes the extra hours is stationary for the balance constraints, in contrast to the stretch without. Thus, if we consider the balance constraints, schedules have an equal or at least very fair distribution of hour deviations. This is the case for 40 physicians and 1 month as shown by Figure 12.

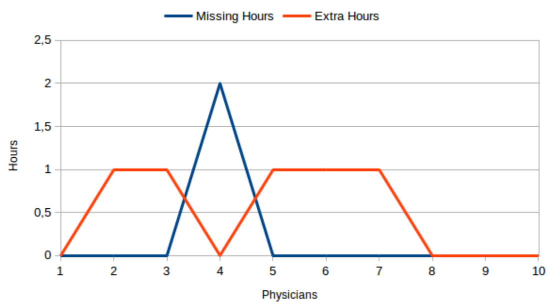

Fig. 11: Hours deviations on the analyzed week for the instance of 10 physicians. In blue the missing hours, in red the extra hours.

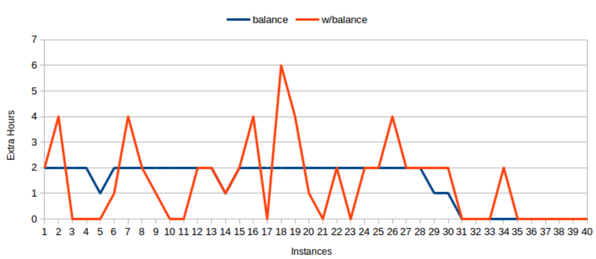

Fig. 12: Difference between the inclusion of balance constraints and the exclusion for all instances.

VI. Conclusions

This paper addressed fairness in physician scheduling problem for emergency rooms. A mixed-integer-programming (MIP) model is introduced to describe such problem by using a minmax approach. Also, relax-and-fix (RF) and a fix-and-optimize (FO) heuristics are combined applying several partitioning strategies.

The experiments are conducted over 20 instances generated extracting features from a real-world schedule and benchmark instances. The proposed MIP model works properly to solve them within 1 hour of execution time using Cplex solver. The model works better enabling the populate strategy of Cplex. When compared against the heuristic methods, the MIP model returned best solutions for the most of the instances.

RF and FO heuristics show a poor performance for the set of instances evaluated. Two main reasons can explain such performance. First, the linear relaxation of our model can lead to poor solutions. This is emphasized on the lower bound values reported on Figures 5-10 as well as gap values on Table VI. Second, heuristics may have a better performance when applied to set of instances larger or more complex than those evaluated so far.

Thus, as future work, we will improve the partitioning strategies aiming a better performance of the RF and FO heuristics. Also, more complex set of instance will be proposed to evaluate the MIP model as well as the proposed heuristics.

References

[1] J. Puente, A. Gómez, I. Fernández, and P. Paolo, “Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms”, *European Journal of Operational Research*, vol. 56, no. 2, pp. 1232–1242, 2009.

[2] D. Ouelhadj, S. Martin, P. Smet, E. Ozcan, and G. Vanden Berghe, “Fairness in nurse rostering”, Feb. 2012.

[3] U. of Nottingham. (2014). Employee shift scheduling benchmark data sets, [Online]. Available: http://www.cs.nott.ac.uk/~psztc/NRP/ (visited on 11/30/2016).

[4] R. Bruni and P. Detti, “A flexible discrete optimization approach to the physician scheduling problem”, *Operations Research for Health Care*, vol. 3, no. 4, pp. 191–199, 2014.

[5] M. Erhard, J. Schoenfelder, A. Fügener, and J. O. Brunner, “State of the art in physician scheduling”, 2016.

[6] H. Beaulieu, J. A. Ferland, B. Gendron, and P. Michelon, “A mathematical programming approach for scheduling physicians in the emergency room”, *Health Care Management Science*, no. 2, pp. 193–200, 2000.

[7] M. W. Carter and S. D. Lapierre, “Scheduling emergency room physicians”, *Health Care Management Science*, vol. 4, no. 4, pp. 347–360, 2001, ISSN: 1572-9389.

[8] L.-M. Rousseau, G. Pesant, and M. Gendreau, “A general approach to the physician rostering problem”, English, *Annals of Operations Research*, vol. 115, no. 1-4, pp. 193–205, 2002, ISSN: 0254-5330.

[9] S. Bourdais, P. Galinier, and G. Pesant, “Hibiscus: A constraint programming application to staff scheduling in health care”, in *Principles and Practice of Constraint Programming – CP 2003: 9th International Conference, CP 2003, Kinsale, Ireland, September 29 – October 3, 2003. Proceedings*, F. Rossi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 153–167.

TABLE VI: Experiments over the sets of instances

Class	Cplex				FOWP			FOSP			FOSWP			FOWSP		
	Obj	Lower Bound	GAP (%)	Execution Time(s)	Obj	GAP (%)	Execution Time(s)	Obj	GAP (%)	Execution Time(s)	Obj	GAP (%)	Execution Time (s)	Obj	GAP (%)	Execution Time(s)
SET#1	8,62	4,75	43,72	3600	8,8	51,06	1928,57	8,8	51,06	2250	8,86	44,55	1928,57	8,8	51,06	2250
SET#2	23,47	9,6	61,15	3600	33,67	63,99	1864,29	33,665	63,99	2025	26,04	65,05	1864,29	30,45	63,41	2025
SET#3	37,16	15,93	56,23	3600	53,02	64,03	1842,86	51,6075	63,63	1950	44,02	59,26	1842,86	51,85	63,88	1950
SET#4	154,73	44,76	63,51	3600	241,1	75,98	1821,43	249,57	76,56	1875	181,77	68,19	1821,43	248,86	76,69	1875
SET#5	296,99	61,57	71,1	3600	517,73	85,43	1814,29	517,7275	85,43	1860	357,2	75,65	1814,29	518,07	85,49	1860

TABLE VII: Experiments for every instances

Class	Instances	Cplex				FOWP			FOSP			FOSWP			FOWSP		
		Obj	Lower Bound	GAP (%)	Execution Time(s)	Obj	GAP (%)	Execution Time(s)	Obj	GAP (%)	Execution Time(s)	Obj	GAP (%)	Execution Time (s)	Obj	GAP (%)	Execution Time(s)
SET#1	Instance10_4	6,05	1,22	79,83	3600	6	79,67	1928,57	6	79,67	2250	6,05	79,83	1928,57	6	79,67	2250
	Instance20_4	7,75	4	48,38	3600	7,75	48,39	1928,57	7,75	48,39	2250	7,75	48,39	1928,57	7,75	48,39	2250
	Instance30_4	16,43	9,77	40,54	3600	14,2	31,2	1928,57	14,2	31,2	2250	17,4	43,85	1928,57	14,2	31,2	2250
	Instance40_4	4,25	3,99	6,12	3600	7,25	44,97	1928,57	7,25	44,97	2250	4,25	6,12	1928,57	7,25	44,97	2250
SET#2	Instance10_8	12,38	2,11	82,96	3600	12,38	82,96	1864,29	12,38	82,96	2025	15,99	86,8	1864,29	12,38	82,96	2025
	Instance20_8	16,63	8,03	51,71	3600	14,13	43,17	1864,29	14,13	43,17	2025	17,43	53,93	1864,29	14,13	43,17	2025
	Instance30_8	30,13	19,89	33,99	3600	33,76	41,08	1864,29	33,76	41,08	2025	34,68	42,65	1864,29	33,76	41,08	2025
	Instance40_8	34,74	8,36	75,94	3600	74,39	88,76	1864,29	74,39	88,76	2025	36,04	76,8	1864,29	61,54	86,42	2025
SET#3	Instance10_12	20,34	6,21	69,47	3600	19,89	68,78	1842,86	19,89	68,78	1950	21,24	70,76	1842,86	19,89	68,78	1950
	Instance20_12	25,36	13,03	48,62	3600	28,96	55,01	1842,86	28,96	55,01	1950	25,86	49,61	1842,86	28,96	55,01	1950
	Instance30_12	43,89	30,57	30,35	3600	55,84	45,25	1842,86	54,84	44,26	1950	45,89	33,38	1842,86	55,84	45,25	1950
	Instance40_12	59,04	13,89	76,47	3600	107,39	87,07	1842,86	102,74	86,48	1950	83,09	83,28	1842,86	102,69	86,47	1950
SET#4	Instance10_24	59,7	14,34	75,98	3600	53,3	73,1	1821,43	52,4	72,63	1875	59,7	75,98	1821,43	54,04	73,46	1875
	Instance20_24	93,42	31,91	65,84	3600	97,02	67,11	1821,43	97,47	67,26	1875	101,67	68,61	1821,43	97,47	67,26	1875
	Instance30_24	119,55	90,25	24,51	3600	327,72	72,46	1821,43	362,07	75,07	1875	146,25	38,29	1821,43	357,57	74,76	1875
	Instance40_24	346,24	42,53	87,72	3600	486,34	91,26	1821,43	486,34	91,26	1875	419,44	89,86	1821,43	486,34	91,26	1875
SET#5	Instance10_36	103,02	29,3	71,56	3600	125,53	76,66	1814,29	125,53	76,66	1860	114,77	74,47	1814,29	126,88	76,91	1860
	Instance20_36	163,99	53	67,68	3600	393,26	86,52	1814,29	393,26	86,52	1860	211,19	74,9	1814,29	393,26	86,52	1860
	Instance30_36	222,77	102,59	53,94	3600	735,41	86,05	1814,29	735,41	86,05	1860	259,67	60,49	1814,29	735,41	86,05	1860
	Instance40_36	698,16	61,37	91,21	3600	816,71	92,49	1814,29	816,71	92,49	1860	843,16	92,72	1814,29	816,71	92,49	1860

TABLE VIII: A Schedule of seven days

	Day 1					Day 2					Day 3					Day 4					Day 5					Day 6					Day 7				
	M	A	N	D1	D2	M	A	N	D1	D2	M	A	N	D1	D2	M	A	N	D1	D2	M	A	N	D1	D2	M	A	N	D1	D2	M	A	N	D1	D2
PHY[1]						Ass										Ass								Ass						Ass					
PHY[2]					Ass			Ass					Ass								Ass														
PHY[3]			Ass											Ass					Ass			Ass													
PHY[4]		Ass										Ass					Ass						Ass											Ass	
PHY[5]	Ass								Ass		Ass														Ass										
PHY[6]	Ass						Ass									Ass					Ass								Ass						
PHY[7]				Ass							Ass									Ass										Ass					
PHY[8]						Ass												Ass											Ass						Ass
PHY[9]										Ass																									Ass
PHY[10]															Ass																			Ass	

[10] S. Topaloglu, "A multi-objective programming model for scheduling emergency medicine residents", *Computers and Industrial Engineering*, vol. 51, no. 3, pp. 375–388, 2006.

[11] Y. Ferrand, M. J. Magazine, U. S. Rao, and T. F. Glass, "Building cyclic schedules for emergency department physicians", *Interfaces*, vol. 41, no. 6, pp. 521–533, 2011.

[12] A. Gunawan and H. C. Lau, "Master physician scheduling problem", *Journal of the Operational Research Society*, vol. 64, no. 3, pp. 410–425, 2013.

[13] I. Adan, J. Bekkers, N. Dellaert, J. Jeunet, and J. Vissers, "Improving operational effectiveness of tactical master plans for emergency and elective patients under stochastic demand and capacitated resources", *European Journal of Operational Research*, vol. 213, no. 1, pp. 290–308, 2011, ISSN: 0377-2217.

[14] Á. P. Dorneles, O. C. de Araújo, and L. S. Buriol, "A fix-and-optimize heuristic for the high school timetabling problem", *Comput. Oper. Res.*, vol. 52, no. PA, pp. 29–38, Dec. 2014, ISSN: 0305-0548.

[15] C. Van Huele and M. Vanhoucke, "Analysis of the integration of the physician rostering problem and the surgery scheduling problem", *Journal of Medical Systems*, vol. 38, no. 6, p. 43, 2014.

[16] ——, "Decomposition-based heuristics for the integrated physician rostering and surgery scheduling problem", *Health Systems*, vol. 4, no. 3, pp. 159–175, 2015.

[17] O. EL-Rifai, T. Garaix, V. Augusto, and X. Xie, "A stochastic optimization model for shift scheduling in emergency departments", *Health Care Management Science*, vol. 18, no. 3, pp. 289–302, 2015, ISSN: 1572-9389.

[18] M. van de Vrugt and R. J. Boucherie, "Assigning treatment rooms at the emergency department", *Operations Research for Health Care*, vol. 8, pp. 62–70, 2016, ISSN: 2211-6923.

[19] C. Van Huele and M. Vanhoucke, "Solving the integrated physician and surgery scheduling problem under stochastic demand", eng, in *Operational Research, 25th European conference, Abstracts*, Vilnius, Lithuania, 2012.

[20] V. F. Dem'yanov and V. N. Malozemov, *Introduction to minimax*. Courier Corporation, 1974.

[21] C. Dillenberger, L. F. Escudero, A. Wollensak, and W. Zhang, "On practical resource allocation for production planning and scheduling with period overlapping setups", *European Journal of Operational Research*, vol. 75, no. 2, pp. 275–286, 1994, ISSN: 0377-2217.

[22] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)*. Se-

caucus, NJ, USA: Springer-Verlag New York, Inc., 2006, ISBN: 0387299599.

[23] P. Planalto, *Medida provisória no 2.164-41, de 24 de agosto de 2001*, "http://www.planalto.gov.br/ccivil_03/MPV/2164-41.htm, 2001.

[24] R. Médica, *Resolução cfm nº 1.763/2005*, "http://www.portalmedico.org.br/resolucoes/cfm/2005/1763_2005.htm, 2005.

[25] S. Topaloglu, "A shift scheduling model for employees with different seniority levels and an application in healthcare", *European Journal of Operational Research*, vol. 198, no. 3, pp. 943–957, 2009, ISSN: 0377-2217.