SAPIENZA
UNIVERSITÀ DI ROMA

Mixed integer linear programming formulations for the Nurse Rostering Problem

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corso di Laurea in Ingegneria Gestionale

Candidate
Fabio Ciccarelli
ID number 1835348

Thesis Advisor
Prof. Ruggiero Seccia

Academic Year 2020/2021

Mixed integer linear programming formulations for the Nurse Rostering Problem
Bachelor's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: ciccarelli.1835348@studenti.uniroma1.it

Contents

Chapter 1

Introduction

When it comes to manage an organization, frequently the planner has to deal with different challenging scenarios. In these cases, the need for a quantitative approach is recently becoming imperative, since it is not possible to rely solely on common sense or on management experience. Firstly, this need arises from the dimension of the problems, in the era of big data, and secondly from the complexity of the constraints that the planner must respect.
The aim of this paper is to provide a basic framework to solve one of the problems mentioned above, which is the personnel scheduling. In fact, optimizing the assignment of shifts to staff can be a large problem (because of the number of employees and the time frame considered), and has very stringent constraints (such as contractual conditions and rest periods). In addition, it is a current issue, since in many organizations situations of staff shortages can occur, and often companies do not have automated and reliable management systems to deal with them (see [3]).
In particular, we focused on the so-called Nurse Rostering Problem, which is a personnel scheduling problem applied to the healthcare context. Problems of this type combine the complexity due to the number of variables (a single hospital can have hundreds of nurses) to the one due to constraints (e.g. rest after night shifts, overtime hours balance or need for extra nurses). Moreover, hospitals and other healthcare facilities are often understaffed, especially when there is a health emergency like the one in progress, due to covid-19. In these cases, it can be helpful for the planner not only to get an efficient shift optimization, but also to know if the department needs extra nurses and, if so, how many of them are needed.

In chapter 2, we describe a standard personnel scheduling mathematical formulation, with emphasis on different objective functions which can be used. We then specialize the standard formulation, applying it to NRP. In this phase, it is crucial to explain why some constraints have been added to the model. We also study the solvability of the problem and the possibility to make some changes in the formulation to have more exhaustive results.
In chapter 3, we introduce an algorithm to find the optimal number of extra-nurses to hire (if they are needed) and we examine how the Key Performance Indicators (KPIs) of the model are influenced by this.
In chapter 4 we address some aspects of software implementation, such as the

libraries used to build the model and to solve it as well, and eventually we analyze the performance of the software with different dimension problems.
Lastly, in chapter 5, we discuss about the effective usefulness of the model and its potential extensions, analyzing pros and cons of applying a similar framework to a real-life scenario.

Chapter 2

Mathematical Formulations

In this chapter, we address a basic mathematical formulation of the personnel scheduling problems mentioned in the previous chapter, and then we deepen the specific one relating to NRP. We underline that this structure is not fixed, since this kind of problems can differ considerably according to the context of application and because the model may be required to evolve over time.
What we are about to introduce belongs to the class of **integer linear programming problems**. This kind of structure is particularly useful both for effectively describing the model and for constructing logical constraints, such as those relating to rest periods.

2.1 General Personnel Scheduling Problems

Before addressing specifically the models for the NRP, let's proceed to define some sets and parameters that are generally present in a model for personnel scheduling. We will then go ahead to introduce the decision variables and some possible forms of objective function. Note that this is an extremely basic model since, as mentioned before, it must be applicable to very different contexts. Therefore, this formulation does not take into account aspects such as overtime hours, the possible presence of night shifts, differences in degree between workers and others.

2.1.1 Sets, Parameters and Variables

$\boldsymbol{I} = \{i_1, ..., i_n\}$ **Set of workers**

$\boldsymbol{S} = \{s_1, ..., s_m\}$ **Set of shifts**

$\boldsymbol{T} = \{t_1, ..., t_p\}$ **Set of days considered**

Note that set T is partitioned into groups, therefore T = $T_1 \cup ... \cup T_q \ \wedge T_h \cap T_k = \emptyset \ , \ h \neq k$, representing the weeks into which the time horizon is divided. In fact, it is crucial to have a time frame divisible into weeks, since working hours are defined on a weekly basis.

$\boldsymbol{h_{min}}$: minimum weekly working hours

$\boldsymbol{h_{max}}$: maximum weekly working hours

$\boldsymbol{a_{s,t}}$: number of workers needed during shift s ∈ S of day t ∈ T

$\boldsymbol{d_s}$: duration of the shift s ∈ S

$$\boldsymbol{x_{i,s,t}} = \begin{cases} 1, & \text{if the i-th workers takes the shifts s} \in \text{S in day t} \in \text{T} \\ 0, & \text{otherwise} \end{cases}$$

2.1.2 Objective Function and Constraints

If we assume that there is no difference between the various work shifts in terms of workload, we can set the goal of balancing the number of shifts performed by each worker as much as possible (a perfect division is not always possible, just think to the need to divide five shifts between two workers).
In order to achieve this result, generally we try to minimize the number of shifts covered by the worker who works more. Therefore, the objective function might look like this:

$$\min \quad \{\max_{i \in I} \sum_{s \in S} \sum_{t \in T} x_{i,s,t}\} \tag{2.1}$$

Another type of balance that we can look for is that relating to working hours. In fact, following a reasoning similar to that illustrated above, we can build the objective function trying to make the amount of working hours homogeneous among employees.

$$\min \quad \{\max_{i \in I} \sum_{s \in S} \sum_{t \in T} d_s \; x_{i,s,t}\} \tag{2.2}$$

We are now ready to define the set of constraints for the first basic formulation:

$$\sum_{s \in S} x_{i,s,t} \leq 1 \qquad i \in I,\ t \in T \tag{2.3}$$

$$\sum_{t \in T_k} \sum_{s \in S} d_s x_{i,s,t} \geq h_{min} \qquad i \in I, \quad k = 1, ..., q \tag{2.4}$$

$$\sum_{t \in T_k} \sum_{s \in S} d_s x_{i,s,t} \leq h_{max} \qquad i \in I, \quad k = 1, ..., q \tag{2.5}$$

$$\sum_{i \in I} x_{i,s,t} \geq a_{s,t} \qquad s \in S,\ t \in T \tag{2.6}$$

Constraint (2.3) requires each worker to perform at most one shift per day, while constraints (2.4) and (2.5) concern the bounds of weekly working hours. Constraint (2.6) instead relates to the need to have at least a certain number of employees covering each shift.

2.1.3 Considerations and Possible Extensions

Although such a formulation is not particularly specific, and may at first appear trivial, it is the basis of each model in the field of personnel scheduling, as complex as it may be. In addition, it proves to be extremely flexible and can be used to solve large problems which a human planner could not solve independently, as long as they do not need additional constraints.
In addition to enriching the model adding night shifts, overtime hours or similar aspects, we can also intervene on the objective function, allowing employees to express totally personalized preferences. For example, let us introduce the parameter $p_{i,s,t} \in [0, 1]$, defined as the satisfaction of worker $i \in I$ for the shift $s \in S$ of day $t \in T$.
We can then proceed to calculate the degree of "dissatisfaction" for each worker, and try to minimize the maximum one (as already done with number of shifts or total hours of work in objective functions (2.1) and (2.2)). We would thus obtain such an objective function:

$$\min \quad \{\max_{i \in I} \quad \sum_{s \in S} \sum_{t \in T} \ (1 - p_{i,s,t}) \ x_{i,s,t}\} \tag{2.7}$$

As is evident, numerous variations can be included in the basic model previously illustrated, that make it more or less suitable for different contexts and management needs.
In addition, we can think to build the model in such a way that it could manage holiday periods, unplanned absences of staff, or identify understaffed departments and the need for new hires.

2.2 NRP formulation

In this phase, we will apply what we saw previously to the specific case of Nurse Rostering Problems. Some of the parameters we are going to introduce are the result of an arbitrary choice: for example, there are no constraints relating to staff vacation periods, unforeseeable absences due to illness or professional hierarchies among nurses. Instead, we will consider aspects such as overtime hours or the possibility to hire new employees.

2.2.1 Sets, Parameters and Variables

$\boldsymbol{I} = \{i_1, ..., i_n\}$ **Nurses Set**

$\boldsymbol{S} = \{morning, afternoon, night\}$ **Shifts set**

$\boldsymbol{T} = \{t_1, ..., t_m\}$ **Days set**

As in the basic case, the set T is partitioned into subsets, (therefore $T = T_1 \cup ... \cup T_q \ \wedge \ T_h \cap T_k = \emptyset \ \ for \ h \neq k$), equivalent to the weeks that make up the time frame.

$\boldsymbol{h_{min}}$: minimum weekly working hours

$\boldsymbol{h_{max}}$: maximum weekly working hours (excluding overtime)

$\boldsymbol{a_s}$: maximum number of patients that a single nurse can take care of during the shift s ∈ S

$\boldsymbol{d_t}$: total number of patients in day t ∈ T

$\boldsymbol{l_s}$: duration in hours of shift s ∈ S

$\boldsymbol{w_{s,t}}$: workload related to shift s of day t, where s ∈ S e t ∈ T

$$\boldsymbol{x_{i,s,t}} = \begin{cases} 1, & \text{if nurse i} \in \text{I covers shift s} \in \text{S during day t} \in \text{T} \\ 0, & \text{otherwise} \end{cases}$$

$\boldsymbol{\beta_{i,k}} \in \mathbb{N}$: overtime hours of nurse i ∈ I during week k = 1, ..., q

2.2.2 Objective Function and Constraints

According to what has been said above, we can identify at least two different objective functions for the model:

$$\min \quad \{\max_{i\in I} \sum_{s\in S}\sum_{t\in T} w_{s,t}\ x_{i,s,t}\} \tag{2.8}$$

$$\min \quad \sum_{i\in I}\sum_{k=1}^{q} \beta_{i,k} \tag{2.9}$$

In (2.8) we follow the same logic as objective functions (2.1) and (2.2), minimizing the maximum workload among nurses. Instead, in (2.9) we try to minimize the amount of overtime hours, assuming that these have a significant cost.
We can combine these two objective functions into one, choosing weights for the addends, in order to give priority to one of the two aspects (reduce workload or overtime). So, we are faced with a multi-objective optimization problem (MOO), and we choose to build the objective function using the weighted sum method (see [7] and [9]).

$$\min \quad \{\max_{i\in I} \sum_{s\in S}\sum_{t\in T} w_{s,t}\ x_{i,s,t}\} + p_1 \sum_{i\in I}\sum_{k=1}^{q} \beta_{i,k} \tag{2.10}$$

We can now introduce the following constraints:

- Each nurse performs at most one shift per day

$$\sum_{s\in S} x_{i,s,t} \leq 1 \quad i \in I,\ t \in T \tag{2.11}$$

- Each nurse must work at least h_{min} hours a week

$$l_s \sum_{s\in S}\sum_{t\in T_k} x_{i,s,t} \geq h_{min} \quad i \in I,\ k = 1, ..., q \tag{2.12}$$

- The number of nurses on duty, multiplied by the number of patients each of them can take care of, must be at least equal to the number of patients in the ward (there must therefore be enough nurses not to leave any patient unattended)

$$a_s \sum_{i\in I} x_{i,s,t}\ \geq d_t \quad s \in S,\ t \in T \tag{2.13}$$

- If a nurse works during the night shift, she must rest at least 24 hours

$$\sum_{s\in S} x_{i,s,t+1} \leq 1 - x_{i,night,t} \quad i \in I,\ t \in T \tag{2.14}$$

- Weekly overtime hours are equal to the number of hours a nurse works during a week minus those required by the contract

$$l_s \sum_{s \in S} \sum_{t \in T_k} x_{i,s,t} - h_{max} \leq \beta_{i,k} \quad i \in I,\ k = 1, ..., q \tag{2.15}$$

According to how we built the objective function, ours is a minimax problem: so let's define a variable z, which represents the maximum workload. By defining z, we get that (2.8) becomes:

$$\min \quad z \tag{2.16}$$

Moreover, the following constraint must be added to the formulation:

$$\sum_{s \in S} \sum_{t \in T} w_{s,t} x_{i,s,t} \ \leq \ z \qquad i \in I \tag{2.17}$$

Which basically means that z is the workload borne by the nurse who bears the maximum workload (that we want to minimize).

The overall formulation will therefore be the following:

$$
\begin{array}{lll}
o.f. & \min \quad z \ + \ p_1 \displaystyle\sum_{i \in I} \sum_{k=1}^{q} \beta_{i,k} & \\
& & \\
s.t. & \displaystyle\sum_{s \in S} x_{i,s,t} \leq 1 & i \in I,\ t \in T \\
& l_s \displaystyle\sum_{s \in S} \sum_{t \in T_k} x_{i,s,t} \geq h_{min} & i \in I,\ k = 1, ..., q \\
& a_s \displaystyle\sum_{i \in I} x_{i,s,t} \ \geq d_t & s \in S,\ t \in T \\
& \displaystyle\sum_{s \in S} x_{i,s,t+1} \leq 1 - x_{i,night,t} & i \in I,\ t \in T \\
& l_s \displaystyle\sum_{s \in S} \sum_{t \in T_k} x_{i,s,t} - h_{max} \leq \beta_{i,k} & i \in I,\ k = 1, ..., q \\
& \displaystyle\sum_{s \in S} \sum_{t \in T} w_{s,t} \ x_{i,s,t} \ \leq \ z & i \in I \\
& & \\
& x_{i,s,t} \in \{0, 1\} & i \in I,\ s \in S,\ t \in T \\
& \beta_{i,k} \in \mathbb{N} & i \in I,\ k = 1, ..., q \\
& z \in \mathbb{R}_+ &
\end{array}
$$

Chapter 3

Extra-Nurses

3.1 Solvability of the basic formulation

It is evident that the formulation defined in the previous chapter does not always admit a solution. In fact, although a single nurse can work for virtually infinite hours (since there is no limit to overtime), constraints (2.11) and (2.14) require rest shifts in any case. Therefore, if there is a high demand for staff due to a high number of patients, these constraints could conflict with constraint (2.13), causing the problem to be inadmissible. In this case, we have to deal with a situation of understaffing, so it might be useful to know if the department needs new staff and, if so, how many new employees should be hired.

The first thing we want to verify is therefore whether the problem admits a solution or not: in order to do this, we introduce a new variable $j_{s,t} \in \mathbb{N}$, which will be inserted in the objective function and in constraint (2.13):

$$\min \qquad z \;+\; p_1 \sum_{i \in I} \sum_{k=1}^{q} \beta_{i,k} \;+\; M \sum_{s \in S} \sum_{t \in T} j_{s,t} \tag{3.1}$$

$$a_s\left(\sum_{i \in I} x_{i,s,t} \;+\; j_{s,t}\right) \geq d_t \qquad s \in S,\; t \in T \tag{3.2}$$

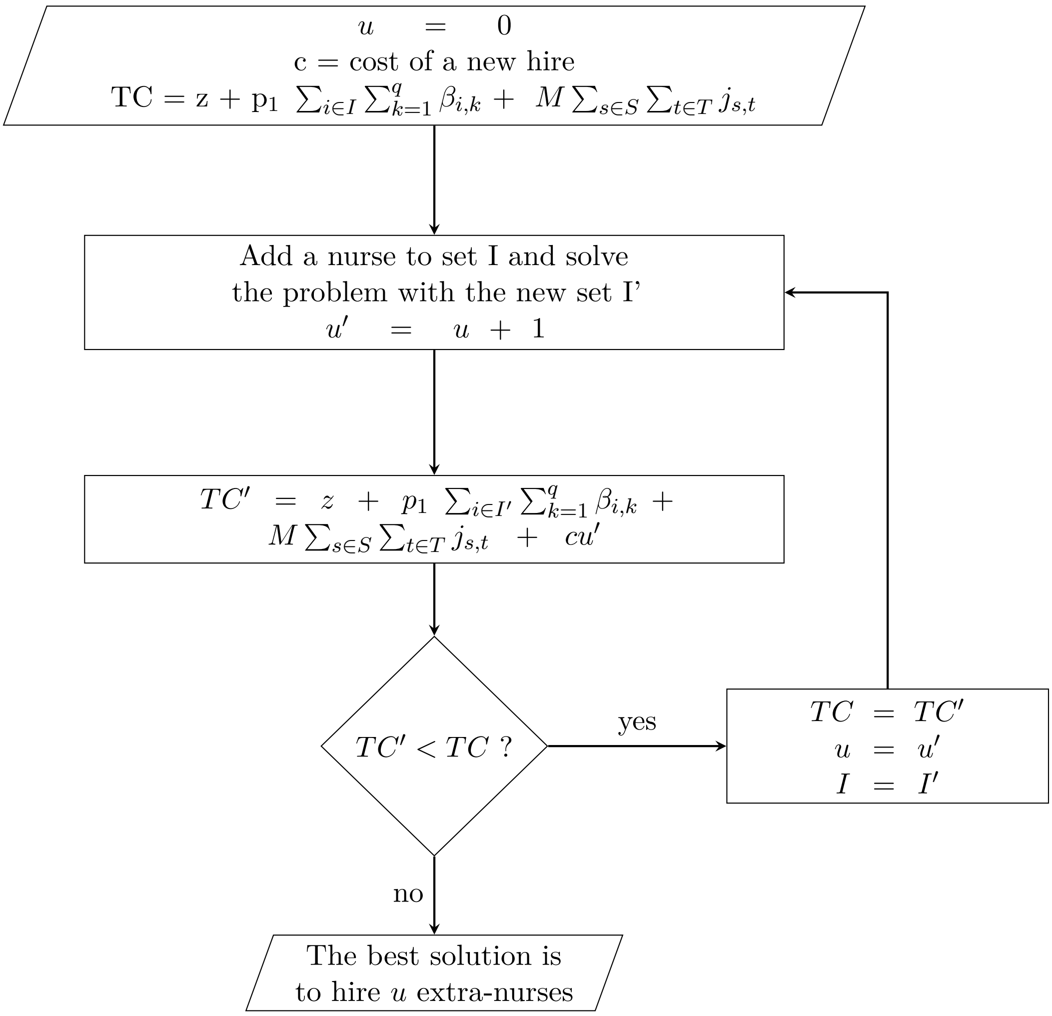
If we consider only the constraint (3.2), we note that this cannot in any way lead to the inadmissibility of the problem. In fact, if there were not enough nurses to assist all patients, the variable $j_{s,t}$ would assume a positive value. We can consider this variable as the number of "dummy" nurses working on shift s of day t. In reality, these nurses do not exist (they do not appear in any of the other constraints), but they allow us to make the problem admissible a priori.
To avoid that the solver chooses to have all the shifts covered by fictitious nurses, we insert the variable $j_{s,t}$ in the objective function (3.1) with a very high weight (the parameter M is usually set at 10^6).

This means that the solver will decide to assign a non-zero value to the variable only if the basic problem does not admit a solution. We therefore use the values assumed by the variable $j_{s,t}$ only to identify situations of understaffing and to communicate them to the decision maker.

3.2 Finding the optimal number of extra-nurses

In case of inadmissibility it becomes necessary to recruit new staff and solve the problem again. It is therefore necessary to find the optimal number of nurses to hire, also taking into account the fact that, once the model is resolved with a new set of nurses, there will be a redistribution of the workload and a decrease in overtime. Consequently, hiring new nurses serves not only to make the problem admissible, but also to find the optimal trade-off between costs incurred for new hires and costs incurred for overtime.

However, since for each nurse added the model must be redefined and solved from scratch (because the set I has undergone a modification), we define the following algorithm.



It is evident that the parameter c strongly influences the progression of the algorithm. In fact, in the event that it has a very low value, or even zero[1], there could be the risk that the set I is updated an infinite number of times (since every time a new nurse is added, this inevitably causes a better distribution of the load of work, a decrease in overtime hours and, therefore, a decrease in the value of the objective function). However, this eventuality cannot occur, due to the presence of restriction (2.12), which imposes a minimum number of hours that each nurse must work (and therefore, indirectly, also a minimum workload). Consequently, we are sure that the previously illustrated algorithm reaches a solution within a finite number of steps.

[1]realistically, hiring a new employee cannot be so cheap as to lead to such a scenario. However, these latter considerations are necessary to discuss the robustness and finiteness of the algorithm.

Chapter 4

Software Implementation

In this chapter we address some aspects of the Python code, implemented with Jupyter notebooks, written with the aim of building a concrete application of the mathematical formulation illustrated in the previous chapters. This code has the objective of finding the optimal solution to the NRP, having received as input some essential data such as the list of employees, the reference time frame and the number of daily patients.

The entire code is available at `https://github.com/FabioCiccarelli/NRP_thesis.git`

Software Tools and Libraries

In order to define the model (to introduce variables, objective function and constraints) we used Pyomo modeling language, an open-source and Pyhton-based software package, and IBM ILOG CPLEX V12.9.0 as a solver to find the optimal solution. We also utilized some other open-source libraries such as Pandas and Numpy.

4.1 Input Data

The input data, necessary for the definition of the model, have been imported via pandas library from specific Excel sheets.

	A	B	C	D	E	F	G	H
1		**ID**	**Age**	**Salary**	**Length of Service**	**Role**	**Ward**	**...**
2	**N1**							
3	**N2**							
4	**N3**							
5	**N4**							
6	**N5**							
7	**N6**							
8	**N7**							
9	**N8**							
10	**N9**							
11	**N10**							

Figure 4.1. Example of data-frame of nurses

It is likely that a company has a similar database, of which, for the purpose of the model presented in this document, only the naming of the rows is of interest (and not the various attributes of the instances, added purely for example).
Another useful data-frame is that concerning the number of patients present in the ward for each day:

	A	B
1		d
2	Monday, 29 March	15
3	Tuesday, 30 March	13
4	Wednesday, 31 March	13
5	Thursday, 1 April	10
6	Friday, 2 April	11
7	Saturday, 3 April	13
8	Sunday, 4 April	10
9	Monday, 5 April	14
10	Tuesday, 6 April	14
11	Wednesday, 7 April	15
12	Thursday, 8 April	11
13	Friday, 9 April	11
14	Saturday, 10 April	15
15	Sunday, 11 April	13
16	Monday, 12 April	10
17	Tuesday, 13 April	10
18	Wednesday, 14 April	10
19	Thursday, 15 April	10
20	Friday, 16 April	11
21	Saturday, 17 April	15
22	Sunday, 18 April	12
23	Monday, 19 April	11
24	Tuesday, 20 April	14
25	Wednesday, 21 April	14
26	Thursday, 22 April	10
27	Friday, 23 April	15
28	Saturday, 24 April	13
29	Sunday, 25 April	15

Figure 4.2. Example of data-frame for daily patients

Also in this case, we can hypothesize other useful attributes for a possible extension of the model, such as the minimum number of nurses required for each day. Note that we chose not to represent the days as a sequence of natural numbers, which would have made it very easy to define some constraints, like (2.14), but as strings. This leads to the need to define the dictionary **day_after** (see the code linked above).

Some parameters, such as the length of shifts, the minimum and maximum number of weekly working hours and the a_s parameter (see 2.2.1) have been chosen arbitrarily, trying to maintain a certain adherence to reality.

4.1.1 Determination of the workload $w_{s,t}$

In subsection 2.2.1 we introduced the $w_{s,t}$ parameter. Not being able to define scientific criteria for determining the "heaviness" of each shift, the weights attributed to the combinations s, t are arbitrary: for example, it was considered that working at night was heavier than working during the day, and that working during the weekend was heavier than working on midweek days. It follows that the shifts having maximum weight will be the nights during the weekends. However, it is possible to hypothesize a customization of the aforementioned weights, which in no way would compromise the functioning of the model. For example, nurses themselves could give weights (perhaps between 1 to 100) to each shift, or they could express their approval rating for shifts, as we saw in (2.7). Another way to proceed could be to customize the workload according to the age of the nurses. For example, the values of $w_{s,t}$ (obtained only from the difference between midweek and weekend days and between day and night shifts) can be updated as follows:

$$w_{i,s,t} = \gamma_i \; w_{s,t} \qquad i \in I, s \in S, t \in T \tag{4.1}$$

$$where \quad \gamma_i = \frac{Age_i}{\frac{1}{n}\sum_{j=1}^{n} Age_j} \tag{4.2}$$

The coefficient γ_i causes an increase in the perceived workload for nurses with an above average age (in fact, in that case, $\gamma_i > 1$), while causing a decrease in the workload for younger nurses ($\gamma_i < 1$).

Those discussed above are different paths that the decision maker can follow to obtain similar results, also based on the context of application of the model. The previous analysis on possible methods for determining the workload parameter has the sole purpose of showing that this is not a trivial aspect at all.

4.2 Results

Our goal, in this section, is to evaluate the performance of the model in terms of readiness in finding the optimal solution and in terms of quality of the solution found. To do this, we solve the problem with different data sets, influencing the number of variables and, consequently, the size of the problem. The following table shows how the time taken by the solver to find a solution (using the branch&cut algorithm) varies according to the size of the problem. Said n the number of nurses and q the number of weeks of the time frame, we obtain the following results:

Table 4.1. Relation between dimension of the problem and time to solve it

n	q	time (s)
10	4	0.32
20	4	0.34
20	6	0.42
40	6	0.47
40	8	0.49

Instead, we evaluate the quality of the solution found through some Key Performance Indicators (KPIs), suitably defined.

4.2.1 Definition of KPIs

Given a feasible solution (x^*, β^*) found by the solver, we define the following KPIs to evaluate it:

- **Minimum workload**:

$$\min_{i \in I} \quad \sum_{s \in S} \sum_{t \in T} w_{s,t} \ x^*_{i,s,t}$$

- **Maximum workload**:

$$\max_{i \in I} \quad \sum_{s \in S} \sum_{t \in T} w_{s,t} \ x^*_{i,s,t}$$

- **Minimum weekly hours**:

$$\min_{i,k} \quad \sum_{s \in S} \sum_{t \in T_k} l_s \ x^*_{i,s,t} \qquad i \in I, \ k = 1, ..., q$$

- **Maximum weekly hours**:

$$\max_{i,k} \quad \sum_{s \in S} \sum_{t \in T_k} l_s \ x^*_{i,s,t} \qquad i \in I, \ k = 1, ..., q$$

- **Total overtime hours**:

$$\sum_{i \in I} \sum_{k=1}^{q} \beta^*_{i,k}$$

By verifying the KPIs of a given solution, our goal is not to draw considerations in an absolute sense. In fact, perfomance indicators prove to be particularly useful more than anything else when different solutions have to be compared.

Case of KPIs' improvement with and without extra-nurses

As previously mentioned, the usefulness of KPIs consists mainly in the possibility of evaluating different solutions, as happens in the case of hiring new nurses. Let us therefore consider the case in which we have the following sets and parameters:

$$I = \{n_1, n_2, ..., n_{10}\} \rightarrow |I| = 10$$
$$T = \{t_1, t_2, ..., t_{28}\} \rightarrow |T| = 28$$
$$q = 4$$
$$h_{min} = 24$$
$$h_{max} = 40$$
$$a = \{Morning\ :\ 5,\ Afternoon\ :\ 5,\ Night\ :\ 10\}$$
$$d_t \in [10, 15] \qquad \forall\ t \in T$$
$$l_s = 8 \qquad \forall\ s \in S$$
$$p_1 = 35$$
$$c = 1000$$

$$w_{s,t} = \begin{cases} 1, & \text{if s day shift and t weekday} \\ 2, & \text{if s night shift and t weekday or s day shift and t weekend day} \\ 4, & \text{if s night shift and t weekend day} \end{cases}$$

In this case[1], the algorithm concludes that the optimal solution is to hire only one extra nurse. Below is a graph of how KPIs evolve following this choice:

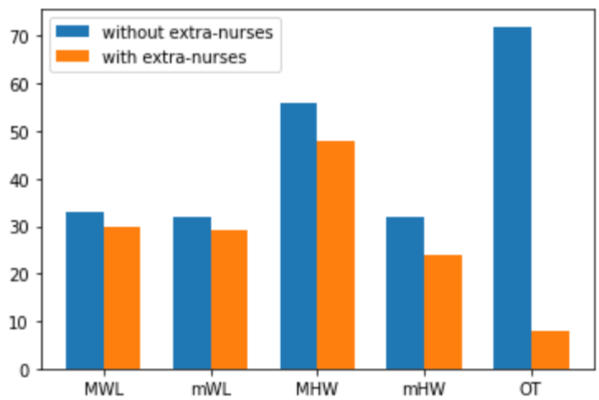


Figure 4.3. Differences in KPIs between the two optimal solutions

The overall improvement in KPIs motivates the need to hire an additional nurse. In particular, note that the decrease in overtime coincides with a greater saving than the cost incurred to hire the new nurse (the value of the objective function also decreases because the variable z takes on a smaller value). The previous one just wants to be, therefore, an example of how the control and evaluation of performance indicators are crucial for a correct understanding of the evolution of the model and in order to assist the planner.

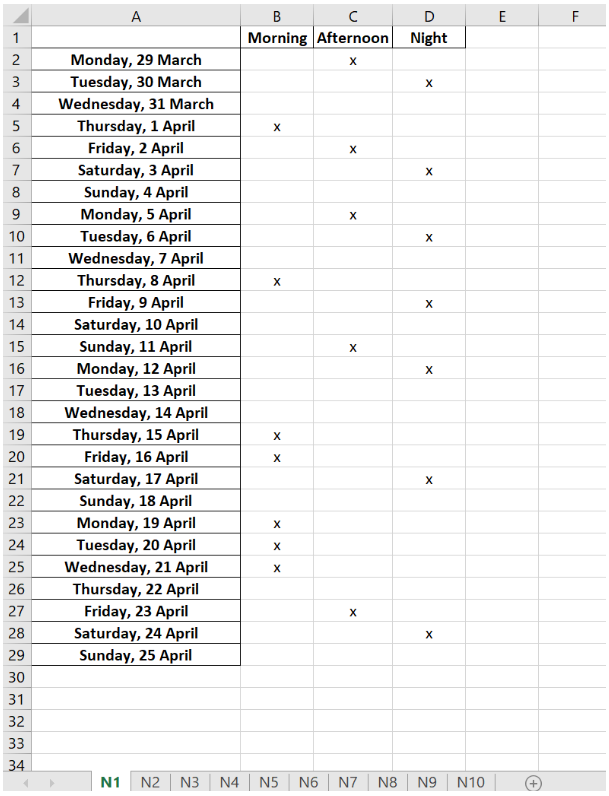
[1]Number of variables: 965

4.2.2 Presentation of results

Once the staff scheduling has been achieved, it is necessary to present the results obtained both to management and to personnel. We have already seen how the general performance indicators can be returned to management, as well as how the DM can be notified of the need to hire new staff. However, it is now necessary to communicate the result obtained in a language different from the mathematical one. Two different ways to do this are shown below.

Shift sheets

The first solution proposed consists in returning as an output of the code, in addition to the values of the KPIs, an Excel workbook with as many sheets as there are nurses. These sheets are then compiled automatically, using the Python Openpyxl library, adding to each sheet the shifts of the i-th nurse over the period of time considered.



	Morning	Afternoon	Night
Monday, 29 March		x	
Tuesday, 30 March			x
Wednesday, 31 March			
Thursday, 1 April	x		
Friday, 2 April		x	
Saturday, 3 April			x
Sunday, 4 April			
Monday, 5 April		x	
Tuesday, 6 April			x
Wednesday, 7 April			
Thursday, 8 April	x		
Friday, 9 April			x
Saturday, 10 April			
Sunday, 11 April		x	
Monday, 12 April			x
Tuesday, 13 April			
Wednesday, 14 April			
Thursday, 15 April	x		
Friday, 16 April	x		
Saturday, 17 April			x
Sunday, 18 April			
Monday, 19 April	x		
Tuesday, 20 April	x		
Wednesday, 21 April	x		
Thursday, 22 April			
Friday, 23 April		x	
Saturday, 24 April			x
Sunday, 25 April			

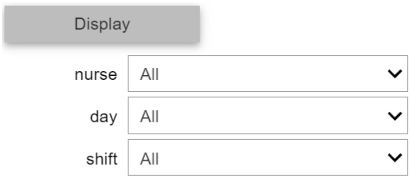
N1 N2 N3 N4 N5 N6 N7 N8 N9 N10

Figure 4.4. Example of shift sheets

Note that this representation of the results allows us to immediately evaluate the correctness of the solution found. In fact, it is possible to visually verify that each nurse performs at most one shift a day, and that rests after night shifts are respected. On the other hand, this kind of solution involves an increase in code execution times, in proportion to the size of the problem. For large problems (several dozen nurses and an extended time period) the Excel document could reach large dimensions and its consultation could become complicated.

Interactive tool

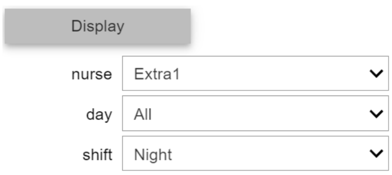
An option that reduces the execution time of the code and obviates the problem of consulting the results is to use the drop-down tool from Ipywidgets library. The user can therefore decide to view the shifts of specific nurses, or which nurses are on duty on a given day.



	Day	Shift	Nurse
1	Monday, 29 March	Morning	N5
2	Monday, 29 March	Morning	N7
3	Monday, 29 March	Morning	N8
4	Monday, 29 March	Afternoon	N1
5	Monday, 29 March	Afternoon	N4
...	...	...	...
199	Sunday, 25 April	Afternoon	N8
200	Sunday, 25 April	Afternoon	N9
201	Sunday, 25 April	Afternoon	N10
202	Sunday, 25 April	Night	N3
203	Sunday, 25 April	Night	N5

Figure 4.5. Drop-down tool to show results

After the execution of the algorithm for determining the optimal number of new hires, the dropdown is updated to take into account the extra nurses.



	Day	Shift	Nurse
1	Monday, 29 March	Night	Extra1
2	Wednesday, 31 March	Night	Extra1
3	Monday, 5 April	Night	Extra1
4	Wednesday, 7 April	Night	Extra1
5	Wednesday, 14 April	Night	Extra1

Figure 4.6. Night shifts performed by Extra1 nurse

4.2.3 Variation of parameters p_1 and c

In section 3.2 we saw that the determination of the optimal number of new nurses to hire is obtained with a trade-off between the cost of new hires and the one associated with overtime hours (as well as the decrease in the objective function due to the redistribution of the workload). It is evident that this balance is strongly influenced by the values assumed by the parameters p_1 and c. It may be of interest to the planner, therefore, to visualize how the solution varies as these parameters vary. To achieve this, we have implemented an additional interactive tool, through which the user can choose the values to be attributed to p_1 and c and view the changes in the solution and in KPIs.

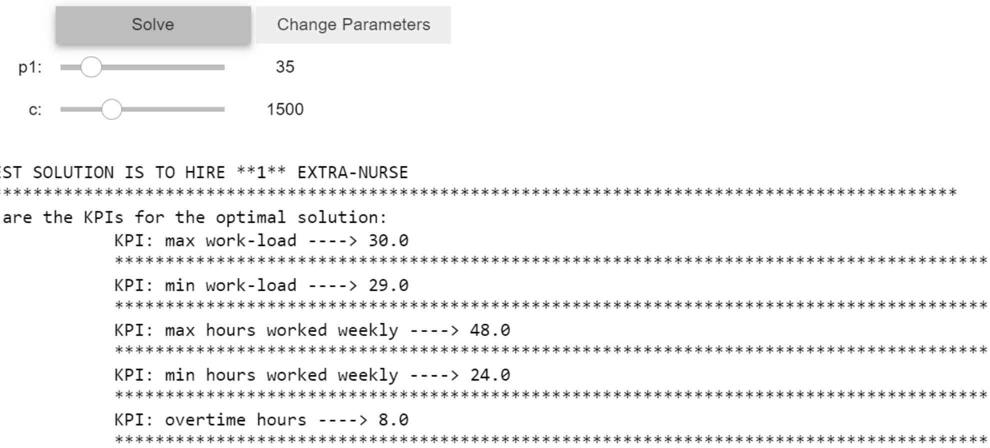


Figure 4.7. Interactive variation of parameters p_1 and c

Chapter 5

Conclusion

What we have studied in depth in this article is the possibility of mathematically modeling a complex situation, such as that relating to NRP, obtaining quick results and helping management to make choices based on concrete data. The formulation we have built, linked to the healthcare context, is actually applicable to different scenarios, with even fewer constraints than those present in our case (there are many contexts in which, for example, there are no night shifts, yet an efficient staff scheduling would be required). The main characteristic that we have tried to attribute to this model is therefore a high flexibility, but without sacrificing rigor and correctness. In fact, tools such as those seen in the previous chapters can be of support in the most varied circumstances (although they require specific technical skills), but become almost necessary when the dimension of the problem is massive and the constraints are stringent: it is therefore not surprising that such tools are in great demand, especially in large companies, where efficient personnel management is a delicate and crucial aspect (workers with stressful shifts are not very motivated and do not work well, while good working conditions can be attractive for the job market).

As we have already said in section 2.1.3, the model can be extended by taking multiple aspects into account. For example, a broader time horizon (six-monthly or even annual) can be considered, with constraints relating to employee vacation periods. Each shift could include the mandatory presence of a foreman or workers with specific skills, and trainees with a lower workload and specific duties could be added to the model. However, we considered it appropriate to investigate only the aspects relating to the possibility of working overtime, the need to hire new nurses in understaffing conditions and the customization of workloads to get a scheduling as fair as possible.

Bibliography

[1] P. Brucker, R. Qu, and E. Burke, *Personnel scheduling: Models and complexity*, European Journal of Operational Research, 210 (2011), pp. 467–473.

[2] E. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Van Landeghem, *The state of the art of nurse rostering*, J. Scheduling, 7 (2004), pp. 441–499.

[3] P. De Causmaecker, P. Demeester, G. Vanden Berghe, E. Berghe, and B. Verbeke, *Analysis of real-world personnel scheduling problems*, (2004).

[4] C. A. Glass and R. A. Knight, *The nurse rostering problem: A critical appraisal of the problem structure*, European Journal of Operational Research, 202 (2010), pp. 379–389.

[5] F. Guerriero and R. Guido, *Modeling a flexible staff scheduling problem in the era of covid-19*, Optimization Letters, (2021).

[6] R. Hoshino, A. Slobodin, and W. Bernoudy, *An automated employee timetabling system for small businesses*, Proceedings of the AAAI Conference on Artificial Intelligence, 32 (2018).

[7] R. T. Marler and J. S. Arora, *The weighted sum method for multi-objective optimization: new insights*, Structural and Multidisciplinary Optimization, 41 (2010), pp. 853–862.

[8] R. Seccia, *The nurse rostering problem in covid-19 emergency scenario*, Technical Report, (2020).

[9] I. Stanimirović, M. L. Zlatanovic, and M. D. Petkovic, *On the linear-weighted summethod for multi-objective optimization* , 2012.