Ph.D. in Electronic and Computer Engineering
Dept. of Electrical and Electronic Engineering
University of Cagliari

Manpower planning optimization in three different real world areas: container terminals, hospitals and retail stores

Simone Zanda

Advisor: Prof. Carla Seatzu

Co-Advisor: Prof. Paola Zuddas

XXXI Cycle
February 2019

This thesis is dedicated to
my family.

Abstract

Problems related to the optimization of human resources in working areas have been extensively studied in the literature with the major goal of guaranteeing the greatest benefits from the efforts of workers, while taking into account their personal skills and requirements. In particular, in this thesis we focus on short-term and long-term manpower planning problems. The main goal consists in appropriately assigning shifts to workers in a given time horizon, taking into account their own requirements, their contractual rules, and the quality and efficiency of the work environment.

In this thesis the manpower planning problem is studied in three different working areas, namely container terminals, hospitals and retail stores. Different solutions are proposed based on mathematical models that allow to describe in linear algebraic terms the set of feasible solutions. An optimal scheduling is then computed using linear integer programming. The proposed policies have been validated on three different real case studies in Cagliari, Italy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Manpower planning and management identify the process of organizing working timetables in order to take into account the workforce demand in all phases of the various activities of a company [15]. Indeed, the efficiency of companies strictly depends on the effectiveness in Manpower Planning (MP), particularly when the labor costs are very high. A low efficiency in MP may result in an expensive workforce surplus/deficit and in a bad quality of the working environment. Frequently, in big companies some human resources are explicitly devoted to organize MPs, producing a timetable after a time consuming effort. Indeed, even if short term scheduling are done, it typically occurs that some unexpected events happen, such as sudden lack of staff, thus requiring the generation of a new timetable which take such events into account. Some research efforts have been devoted to produce efficient tools to support decision-makers [17]. However they are often based on intuitions and skills of the operator devoted to this, and are strictly related to the specific application area. A quite detailed survey on this is reported in the following chapters with special focus on the considered three application areas. As a result, no standard methodologies for MP are available to companies and most of them attempts to adapt similar cases, with big issues deriving from the heterogeneous format of data. Indeed, frequently databases do not exist in a unified format, even among different departments of the same company.

Current commercial MP packages, are tailored to the international market and do not address the unique problem of rostering. Therefore each environment often ends up inventing its own methodology. For example, a consultation with staff of a company in Sardinia indicated that drawing up a week duty roster for a normal sized ward (30 staff members), accounting for all the various requirements, can take up to 3 hours. Rosters are commonly tabulated “by hand” using pencil and customized duty tables.These rosters must then be copied out if different views/parts of the roster are needed. This leads to an endless paper chain of hand-written rosters and accompanying documents, and a lot of repetition of tasks which could be easily automated. Figure 1 shows a real duty roster

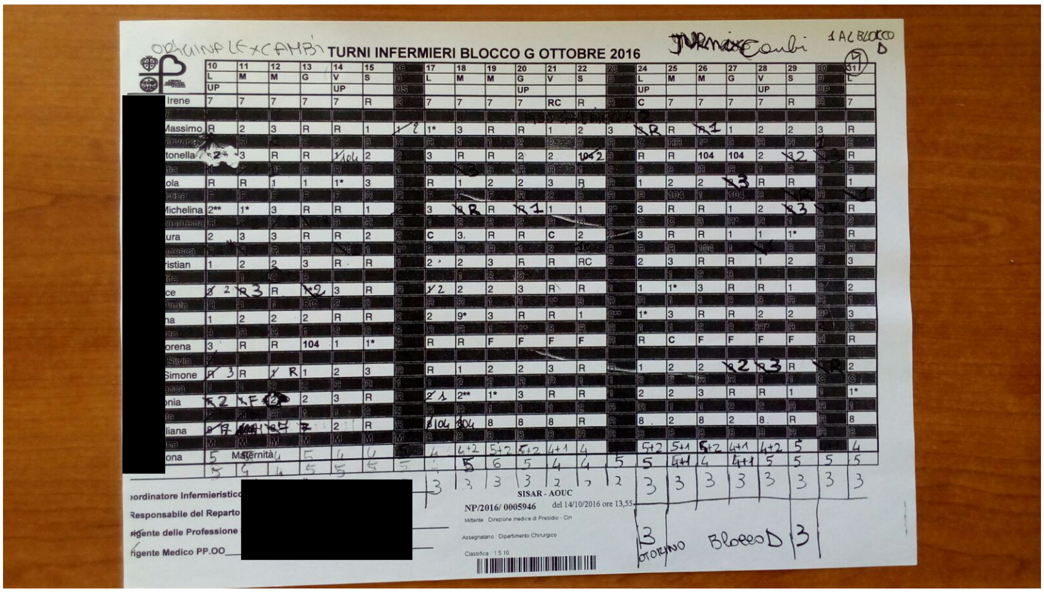


Figure 1.1: Hand written duty rosters

for a ward in the Polytechnic Hospital in Cagliari, one of the cases study discussed in this thesis.

The manual approach creates a number of other problems, apart wasted time. Among them, low quality solutions and unnecessary tensions at work among people due to the inevitable subjectivity of the process. Moreover, the MP is often seen by the operators as a low level secondary job and they have little motivation to dedicate the time required to obtain a good solution.

In this thesis the MP problem is approached as the process of assigning workers to shifts to meet the service demand [21, 27]. To this aim, the time horizon of interest is discretized in elementary sub-periods. During each sub-period, a set of activities have to be performed, each one requiring a certain number of tasks to be completed, and each task requires a specific number of workers. Therefore, the problem can be viewed as the identification of a rostering policy that assigns tasks of the different activities to employees during the different time periods, in order to cover the demand for all the activities. Obviously, this should be done satisfying a series of restrictions of different nature, in particular, personal requirements and contractual rules. Different goals may be considered. In this thesis special attention is devoted to the minimization of the surplus/deficit of workers, and the minimization of some functions that guarantee satisfactory working conditions for employees. A number of different requirements, often unexpected, must be suddenly fulfilled in different sub-periods, such as motivated requests of day-off that must be covered by other staff members. As a consequence the scheduling has to be updated many times and quickly, often forcing inefficient solutions, which are not appreciated by the personnel.

There are two important aspects that heavily contribute to make the MP problem difficult, namely

1. the large number of constraints the working load and scheduling of each roster must satisfy;

2. the different contractual rules and abilities of rosters.

Concerning the first item above, we point out the difference between *hard* constraints and *soft* constraints. Hard constraints correspond to requirements that should be satisfied to make an assignment feasible. For example, having the correct number of employees/nurses with the correct skills on each shift, do not assign shifts the day after a night shift, and so on. On the other side, soft constraints could be violated but a cost is associated to their violation in order to do that only if strictly necessary to find an admissible scheduling. Basically, soft constraints may be seen as strong wishes of the personnel that should be satisfied whenever it is possible. Often soft constraints are contradictory among them. In such a case, solving the MP problem consists in finding the right compromise among them, taking into account the costs deriving from their violation.

In this thesis the MP problem is investigated with reference to three real applications briefly described in the following.

- *Container terminals*. Maritime transport is the backbone of international trade, and containers play an interestingly crucial role in freight transportation [32, 33, 50]. In 2013, the world container port throughput increased by 5.1 percent and reached 651.1 million of twenty-foot equivalent units (TEUs). In maritime transportation networks, shipping liners deploy deep-sea vessels (also called mother vessels) between a limited number of transshipment container terminals (TCTs), whereas smaller vessels (also called feeders) link TCTs to origin and destination ports. The hub-and-spoke topology of maritime networks results in a critical role for TCTs, because of the consolidation of flows along the routes linking TCTs. Moreover, delays at TCTs can negatively impact the reliability of the liner service and generate additional costs for customers. Unlike origin and destination ports, TCTs operate under continuous and heavy competitive pressure, because shipping liners have high bargaining power in redesigning their maritime routes and excluding unsatisfactory TCTs. As a result, TCTs must provide high-performance and cost-effective services and accurately plan the management of their resources to satisfy the demanding requests of shipping liners. Since few ports adopt completely automated systems, human resources are relevant assets for TCTs, particularly in the case of high labour costs, and manpower management is a crucial activity for TCTs.

- *Hospitals*. In this thesis we focus on the long term scheduling problem of the shifts of a team of nurses [64, 75, 58]. We propose a solution based on integer linear programming, which allows to compute a scheduling in a given time horizon, which is optimal with respect to certain criteria, while satisfying a series of constraints imposed by the contractual rules of nurses and that aims to guarantee comfortable working conditions to them. To take into account possible sudden and unpredictable variations in the requirements of the hospital and in the availability of the personnel, we propose a solution based on a Decision Support System (DSS), which splits the scheduling in the long time horizon in several smaller time horizons, and continuously update a series of information relative to the hospital and the team of nurses.
- *Retail stores*. We focus the problem of MP in a big retail store where a short term scheduling (typically one week) should be performed. In this case, each employee has certain skills and is paid according to his/her top skill [23, 45, 48]. Skills uniquelly identify the tasks that can be solved. The objective of the manpower scheduling is that of assigning tasks to employees during the different time periods in order to cover the given workforce demand, which is computed a priori on the basis of hystorical data forcasts techniques. As in the previous cases, contractual rules should be satisfied and the objective of the timetabling problem is to maximize the employee satisfaction, while minimizing the deficit or surplus of employees.

When dealing with the above problems, it is not easy to identify a specific approach that is clearly and always preferable to the other. Indeed, as explained in the following chapters, very different approaches have been proposed in the literature when dealing with MP problems in the above three areas. Furthermore, as in all the cases of very large dimension problems, it may be necessary to look for ad hoc heuristics. Indeed, in general, finding the optimal solution for a large dimension scheduling problem requires analyzing the space of the feasible solutions and selecting the best one, or at least one that is as close as possible to the optimal one (if the given problem is computationally intractable). Some scheduling problems can be efficiently solved by reducing them to combinatorial optimization problems solved in polynomial times, such as linear programming, maximum flow or transport problems. Others can be tackled with standard techniques, such as dynamic programming and branch-and-bound methods. In particular, approaches are usually classified into two macro-classes: exact methods and heuristic methods. Exact methods are unfortunately applicable only to problems with a relatively small number of variables. Therefore, in real problems, they are often unable to provide a solution (even if sub-optimal) in reasonable times.

In this thesis we propose solutions based on linear integer programming and the dimensions of the problems at hand allow the computation of optimal solutions in a reasonable time.

The thesis is organized as follows. In Chapter 2 a brief background on linear integer programming is provided. In Chapter 3 two different polices for the short-term manpower planning problem in terminal containers are described and applied to a real case. In Chapter 4 the MP problem in the department of a hospital is solved in a long-term horizon thanks a Decision Support System. Again, the solution is applied to a real case study. In Chapter 5 a short-term MP problem in retail stores is dealt and implemented using data provided by a real application. Finally, Chapter 6 provides a summary of conclusions and future research perspectives.

Chapter 2

Linear integer programming

Linear programming is concerned with the optimization (minimization or maximization) of a linear function while satisfying a set of linear equality and/or inequality constraints or restrictions. The linear programming problem was first conceived by George B. Dantzig around 1947 [26] while he was working as a mathematical advisor to the United States Air Force Comptroller on developing a mechanized planning tool for a time-staged deployment, training, and logistical supply program. Although the Soviet mathematician and economist L. V. Kantorovich formulated and solved a problem of this type dealing with organization and planning in 1939, his work remained unknown until 1959. Hence, the conception of the general class of linear programming problems is usually credited to Dantzig. Because the Air Force refers to its various plans and schedules to be implemented as "programs," Dantzig's first published paper addressed this problem as "Programming in a Linear Structure." The term "linear programming" was actually coined by the economist and mathematician T. C. Koopmans in the summer of 1948 while he and Dantzig strolled near the Santa Monica beach in California. In 1949 George B. Dantzig published the "simplex method" for solving linear programs. Since that time a number of individuals have contributed to the field of linear programming in many different ways, including theoretical developments, computational aspects, and exploration of new applications of the subject. The simplex method of linear programming enjoys wide acceptance because of (1) its ability to model important and complex management decision problems, and (2) its capability for producing solutions in a reasonable amount of time. In this chapter, we introduce the linear programming problem. The following topics are discussed: basic definitions in linear programming, assumptions leading to linear models, manipulation of the problem, examples of linear problems, and geometric solution in the feasible region space and the requirement space. This chapter is elementary and may be skipped if the reader has previous knowledge of linear programming [9].

2.1 The linear programming problem

We begin our discussion by formulating a particular type of linear programming problem. As will be seen subsequently, any general linear programming problem may be manipulated into this form.

Basic Definitions

Consider the following linear programming problem. Here, $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ is the objective function (or criterion function) to be minimized and will be denoted by z. The coefficients $c_1, c_2, \cdots, c_n$ are the (known) cost coefficients and $x_l, x_2, \cdots, x_n$ are the decision variables (variables, structural variables, or activity levels) to be determined.

$$
\begin{array}{lccccccc}
\text{Minimize} & c_1x_1 & + & c_2x_2 & +\cdots+ & c_nx_n & & \\
\text{subject to} & a_{1,1}x_1 & + & a_{1,2}x_2 & +\cdots+ & a_{1,n}x_n & \geq & \mathsf{b}_1 \\
 & a_{2,1}x_1 & + & a_{2,2}x_2 & +\cdots+ & a_{2,n}x_n & \geq & \mathsf{b}_2 \\
 & \vdots & & \vdots & +\cdots+ & \vdots & & \vdots \\
 & a_{m1,1}x_1 & + & a_{m2,2}x_2 & +\cdots+ & a_{mn,n}x_n & \geq & \mathsf{b}_m \\
 & x_1, & & x_2, & \cdots, & x_n & \geq & 0.
\end{array}
$$

The inequality $\sum_{j=1}^{n} a_{i,j}x_j \geq b_i$ denotes the ith constraint (or restriction or functional, structural, or technological constraint). The coefficients $a_{i,j}$ for $i = 1, \cdots, m, j = 1, \cdots, n$ are called the technological coefficients. These technological coefficients form the constraint matrix A.

$$
\mathsf{A} = \begin{bmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{bmatrix}
$$

The column vector whose ith component is b_i, which is referred to as the right-hand-side vector, represents the minimal requirements to be satisfied. The constraints $x_1, x_2, \cdots, x_n \geq 0$ are the non negativity constraints. A set of values of the variables $x_1, \cdots, x_n$ satisfying all the constraints is called a feasible point or a feasible solution. The set of all such points constitutes the feasible region or the feasible space. Using the foregoing terminology, the linear programming problem can be stated as follows: Among all feasible solutions, find one that minimizes (or maximizes) the objective function.

Example

Consider the following linear problem:

$$
\begin{array}{lcccccc}
\text{Minimize} & 2x_1 & + & 5x_2 & & \\
\text{subject to} & x_1 & + & x_2 & \geq & 6 \\
 & -x_1 & + & -2x_2 & \geq & \text{-18} \\
 & x_1, & & x_2, & \geq & 0.
\end{array}
$$

In this case, we have two decision variables x_1 and x_2. The objective function to be minimized is $2x_1 + 5x_2$. The constraints and the feasible region are illustrated in Figure 2.1. The optimization problem is thus to find a point in the feasible region having the smallest possible objective value.

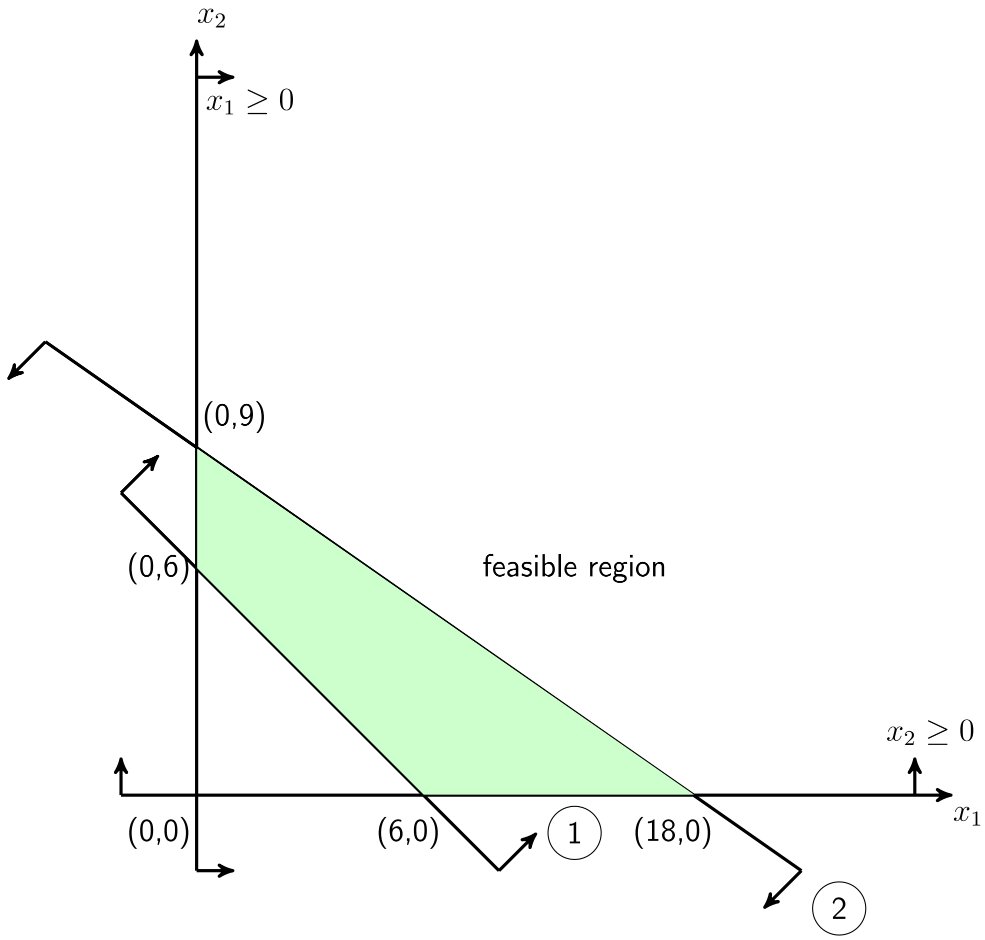


Figure 2.1: Illustration of the feasible region

Assumptions of Linear Programming

To represent an optimization problem as a linear program, several assumptions that are implicit in the linear programming formulation discussed previously are needed. A brief discussion of these assumptions is given next.

1. Proportionality. Given a variable x_j, its contribution to cost is $c_j x_j$ and its contribution to the ith constraint is $a_{i,j} x_j$. This means that if x_j is doubled, say, so is its contribution to cost and to each of the constraints. To illustrate, suppose that x_j is the amount of activity j used. For instance, if $x_j = 10$, then the cost of this activity is $10c_j$. If $x_j = 20$, then the cost is $20c_j$, and so on. This means that no savings (or extra costs) are realized by using more of activity j; that is, there are no economies or returns to scale or discounts. Also, no setup cost for starting the activity is realized.

2. Additivity. This assumption guarantees that the total cost is the sum of the individual costs, and that the total contribution to the ith restriction is the sum of the individual contributions of the individual activities. In other words, there are no substitution or interaction effects among the activities.

3. Divisibility. This assumption ensures that the decision variables can be divided into any fractional levels so that non-integral values for the decision variables are permitted.

4. Deterministic. The coefficients c_j, $a_{i,j}$, and b_i are all known deterministically. Any probabilistic or stochastic elements inherent in demands, costs, prices, resource availabilities, usages, and so on are all assumed to be approximated by these coefficients through some deterministic equivalent.

It is important to recognize that if a linear programming problem is being used to model a given situation, then the aforementioned assumptions are implied to hold, at least over some anticipated operating range for the activities. When Dantzig first presented his linear programming model to a meeting of the Econometric Society in Wisconsin, the famous economist H. Hotelling critically remarked that in reality, the world is indeed nonlinear. As Dantzig recounts, the well-known mathematician John von Neumann came to his rescue by countering that the talk was about "Linear" Programming and was based on a set of postulated axioms. Quite simply, a user may apply this technique if and only if the application fits the stated axioms. Despite the seemingly restrictive assumptions, linear programs are among the most widely used models today. They represent several systems quite satisfactorily, and they are capable of providing a large amount of information besides simply a solution, as we shall see later, particularly in Chapter 6. Moreover, they are also often used to solve certain types of nonlinear optimization problems via (successive) linear approximations and constitute an important tool in solution methods for linear discrete optimization problems having integer-restricted variables.

Problem Manipulation

Recall that a linear program is a problem of minimizing or maximizing a linear function in the presence of linear inequality and/or equality constraints. By simple manipulations the problem can be transformed from one form to another equivalent form. These manipulations are most useful in linear programming, as will be seen throughout the text.

INEQUALITIES AND EQUATIONS

An inequality can be easily transformed into an equation. To illustrate, consider the constraint given by $\sum_{j=1}^{n} a_{i,j}x_j \geq b_i$. This constraint can be put in an equation form by subtracting the nonnegative surplus or slack variable x_{n+i} (sometimes denoted by S_i) leading to $\sum_{j=1}^{n} a_{i,j}x_j - x_{n+i} = b_i$ and $x_{n+i} \geq 0$. Similarly, the constraint $\sum_{j=1}^{n} a_{i,j}x_j \leq b_i$ is equivalent to$\sum_{j=1}^{n} a_{i,j}x_j - x_{n+i} = b_i$ and $x_{n+i} \geq 0$. Also, an equation of the form $\sum_{j=1}^{n} a_{i,j}x_j = b_i$ can be transformed into the two inequalities $\sum_{j=1}^{n} a_{i,j}x_j \leq b_i$ and $\sum_{j=1}^{n} a_{i,j}x_j \geq b_i$,although this is not the practice.

NONNEGATIVITY OF THE VARIABLES

For most practical problems the variables represent physical quantities, and hence must be nonnegative. The simplex method is designed to solve linear programs where the variables are nonnegative. If a variable x_j is unrestricted in sign, then it can be replaced by $x_j^{'} - x_J^{''}$ where $x_j^{'} \geq 0$ and $x_j^{''} \geq 0$. If $x_1, \cdots, x_k$ are some K variables that are all unrestricted in sign, then only one additional variable $x^{''}$ is needed in the equivalent transformation: $x_j = x_j^{'} - x^{''}$ for $j = 1, \cdots, k$, where $x_j^{'} \geq 0$ for $j = 1, \cdots, k$, and $x'' \geq 0$. (Here, $-x$" plays the role of representing the most negative variable, while all the other variables x_j are x_j' above this value.) Alternatively, one could solve for each unrestricted variable in terms of the other variables using any equation in which it appears, eliminate this variable from the problem by substitution using this equation, and then discard this equation from the problem. However, this strategy is seldom used from a data management and numerical implementation viewpoint. Continuing, if $x_j \geq l_j$, then the new variable $x_j' = x_j - l_j$ is automatically nonnegative. Also, if a variable x_j is restricted such that $x_j \leq u_j$, where we might possibly have $u_j \leq 0$, then the substitution $x_j' = u_j - x_j$ produces a nonnegative variable x_j'.

MINIMIZATION AND MAXIMIZATION PROBLEMS

Another problem manipulation is to convert a maximization problem into a minimization problem and conversely. Note that over any region,

$$\text{maximum} \sum_{j=1}^{n} c_j x_j = -\text{minimum} \sum_{j=1}^{n} -c_j x_j$$

Hence, a maximization (minimization) problem can be converted into a minimization (maximization) problem by multiplying the coefficients of the objective function by -1 . After the optimization of the new problem is completed, the objective value of the old problem is -1 times the optimal objective value of the new problem.

Standard and Canonical Formats

From the foregoing discussion, we have seen that any given linear program can be put in different equivalent forms by suitable manipulations. In particular, two forms will be useful. These are the standard and the canonical forms [4]. A linear program is said to be in standard format if all restrictions are equalities and all variables are nonnegative. The simplex method is designed to be applied only after the problem is put in standard form. The canonical form is also useful, especially in exploiting duality relationships. A minimization problem is in canonical form if all variables are nonnegative and all the constraints are of the $\geq$ type. A maximization problem is in canonical form if all the variables are nonnegative and all the constraints are of the $\leq$ type. The standard and canonical forms are summarized in Table 2.1

Linear Programming in Matrix Notation

A linear programming problem can be stated in a more convenient form using matrix notation. To illustrate, consider the following problem:

	MINIMIZATION PROBLEM	MAXIMIZATION PROBLEM
STANDARD FORM	Minimize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{i,j} x_j = b_i, \quad i = 1, \cdots, m$ $x_j \geq 0, \quad j = 1, \cdots, n$	Maximize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{i,j} x_j = b_i, \quad i = 1, \cdots, m$ $x_j \geq 0, \quad j = 1, \cdots, n$
CANONICAL FORM	Minimize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{i,j} x_j \geq b_i, \quad i = 1, \cdots, m$ $x_j \geq 0, \quad j = 1, \cdots, n$	Maximize $\sum_{j=1}^{n} c_j x_j$ subject to $\sum_{j=1}^{n} a_{i,j} x_j \leq b_i, \quad i = 1, \cdots, m$ $x_j \geq 0, \quad j = 1, \cdots, n$

Table 2.1: Standard and Canonical Forms

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

$$\begin{array}{llcll} \text{Minimize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{subject to} & \sum_{j=1}^{n} a_{i,j} x_j & = & b_i, & i = 1, \cdots, m \\ & x_j & \geq & 0, & j = 1, \cdots, n \end{array}$$

Denote the row vector $(c_1, c_2, ..., c_n)$ by **c**, and consider the following column vectors **x** and **b**, and the mx n matrix **A**.

Then the problem can be written as follows:

$$\begin{array}{lccc} \text{Minimize} & \mathbf{cx} & & \\ \text{subject to} & \mathbf{Ax} & = & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

The problem can also be conveniently represented via the columns of **A**. Denoting A by $[a_1, a_2, \cdots, a_n]$ where a_j, is the jth column of **A**, the problem can be formulated as follows:

$$\begin{array}{llcll} \text{Minimize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{subject to} & \sum_{j=1}^{n} \mathbf{a}_j x_j & = & \mathbf{b} & \\ & x_j & \geq & 0, & j = 1, \cdots, n \end{array}$$

2.2 Linear programming modeling

The modeling and analysis of an operations research problem in general, and a linear programming problem in particular, evolves through several stages. The problem formulation phase involves a detailed study of the system, data collection, and the identification of the specific problem that needs to be analyzed (often the encapsulated problem may only be part of an overall system problem), along with the system constraints, restrictions, or limitations, and the objective function(s). Note that in real-world contexts, there frequently already exists an operating solution and it is usually advisable to preserve a degree of persistency with respect to this solution, i.e., to limit changes from it (e.g., to limit the number of price changes, or decision option modifications, or changes in percentage resource consumptions, or to limit changing some entity contingent on changing another related entity). Such issues, aside from technological or structural aspects of the problem, should also be modeled into the problem constraints. The next stage involves the construction of an abstraction or an idealization of the problem through a mathematical model. Care must be taken to ensure that the model satisfactorily represents the system being analyzed, while keeping the model mathematically tractable. This compromise must be made judiciously, and the underlying assumptions inherent in the model must be properly considered. It must be borne in mind that from this point onward, the solutions obtained will be solutions to the model and not necessarily solutions to the actual system unless the model adequately represents the true situation. The third step is to derive a solution. A proper technique that exploits any special structures (if present) must be chosen or designed. One or more optimal solutions may be sought, or only a heuristic or an approximate solution may be determined along with some assessment of its quality. In the case of multiple objective functions, one may seek efficient or Pareto-optimal solutions, that is, solutions that are such that a further improvement in any objective function value is necessarily accompanied by a detriment in some other objective function value. The fourth stage is model testing, analysis, and (possibly) restructuring. One examines the model solution and its sensitivity to relevant system parameters, and studies its predictions to various what-if types of scenarios. This analysis provides insights into the system. One can also use this analysis to ascertain the reliability of the model by comparing the predicted outcomes with the expected outcomes, using either past experience or conducting this test retroactively using historical data. At this stage, one may wish to enrich the model further by incorporating other important features of the system that have not been modeled as yet, or, on the other hand, one may choose to simplify the model. The final stage is implementation. The primary purpose of a model is to interactively aid in the decision-making process. The model should never replace the decision maker. Often a "frank-factor" based on judgment and experience needs to be applied to the model solution before making policy decisions. Also, a model should be treated as a "living" entity that needs to be nurtured over time, i.e., model parameters, assumptions, and restrictions should be periodically revisited in order to keep the model current, relevant, and valid. We describe several problems that can be formulated as linear programs. The purpose is to exhibit the varieties of problems that can be recognized and expressed in precise mathematical terms as linear programs.

2.3 Geometric solution

We describe here a geometrie procedure for solving a linear programming problem. Even though this method is only suitable for very small problems, it provides a great deal of insight into the linear programming problem. To be more specific, consider the following problem:

$$\begin{array}{rrcl} \text{Minimize} & \mathbf{cx} & & \\ \text{subject to} & \mathbf{Ax} & = & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

Note that the feasible region consists of all vectors **x** satisfying **Ax** $=$ **b** and **x** $\geq$ **0**. Among all such points, we wish to find a point having a minimal value of **cx**. Note that points having the same objective value z satisfy the equation **cx**$=z$, that is, $\sum_{j=1}^{n} c_j x_j = z$. Since z is to be minimized, then the plane (line in a two-dimensional space) $\sum_{j=1}^{n} c_j x_j = z$ must be moved parallel to itself in the direction that minimizes the objective the most. This direction is -**c**, and hence the plane is moved in the direction of -**c** as much as possible, while maintaining contact with the feasible region. This process is illustrated in Figure 1.3. Note that as the optimal point **x*** is reached, the line $c_1x_1 + c_2x_2 = z^*$, where $z^* = c_1x_1^* + c_2x_2^*$ cannot be moved farther in the direction -**c** $= (-c_1, -c_2)$, because this will only lead to points outside the feasible region. In other words, one cannot move from **x*** in a direction that makes an acute angle with -c , i.e., a direction that reduces the objective function value, while remaining feasible. We therefore conclude that **x*** is indeed an optimal solution. Needless to say, for a maximization problem, the plane **cx** $=$ z must be moved as much as possible in the direction **c**, while maintaining contact with the feasible region. The foregoing process is convenient for problems having two variables and is obviously impractical for problems with more than three variables. It is worth noting that the optimal point **x*** in Figure 2.2 is one of the five corner points that are called extreme points.

2.4 Modeling with integer variables

Consider the manufacture of television sets. A linear programming model might give a production plan of 205.7 sets per week. In such a model, most people would have no trouble stating that production should be 205 sets per week (or even roughly 200 sets per week). On the other hand, suppose we were buying warehouses to store finished goods, where a warehouse comes in a set size. Then a model that suggests we purchase 0.7

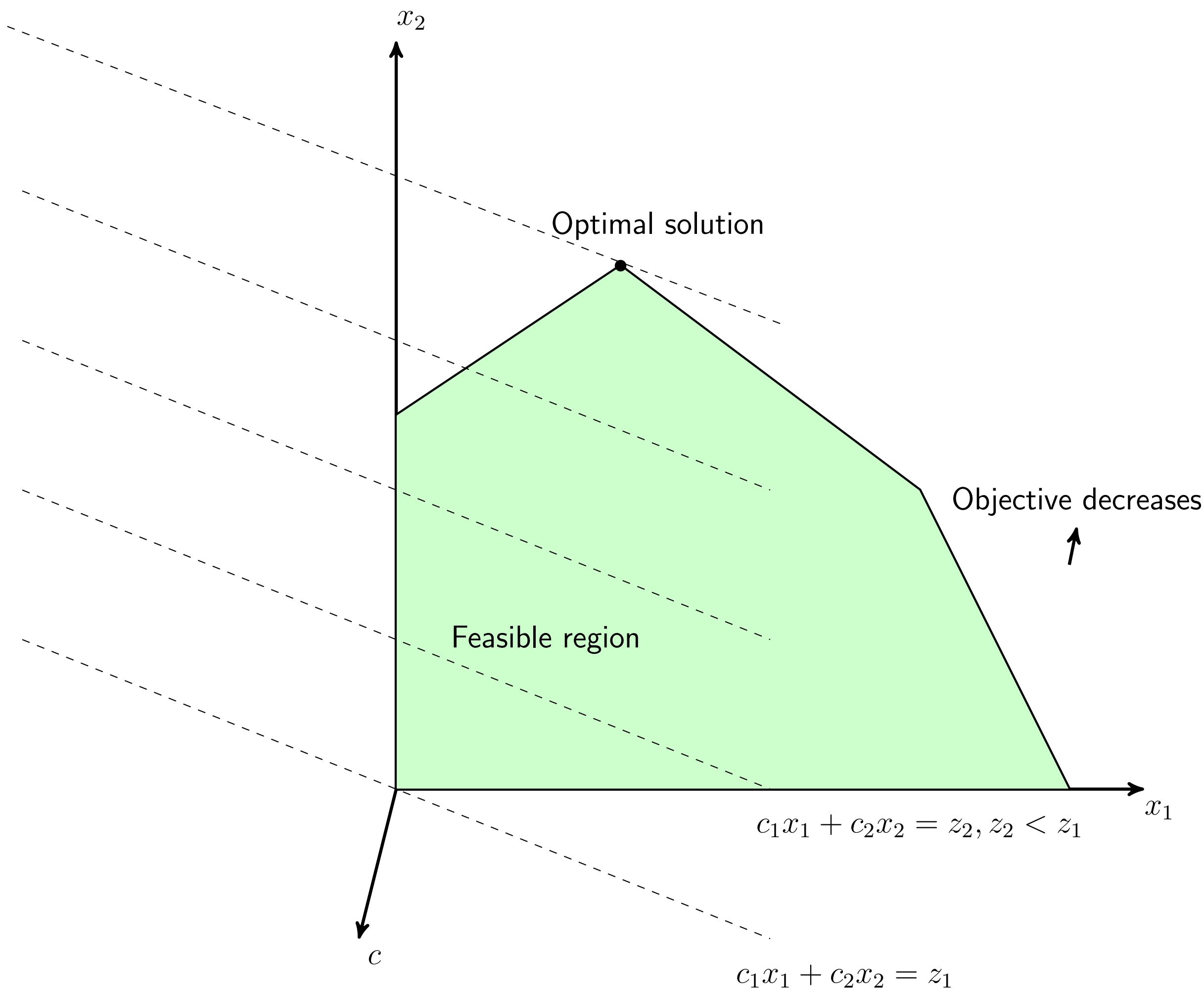


Figure 2.2: Geometric solution

warehouse at some location and 0.6 somewhere else would be of little value. Warehouses come in integer quantities, and we would like our model to reflect that fact. This integrality restriction may seem rather innocuous, but in reality it has far reaching effects. On one hand, modeling with integer variables has turned out to be useful far beyond restrictions to integral production quantities. With integer variables, one can model logical requirements, fixed costs, sequencing and scheduling requirements, and many other problem aspects. In AMPL, one can easily change a linear programming problem into an integer program. The downside of all this power, however, is that problems with as few as 40 variables can be beyond the abilities of even the most sophisticated computers. While these small problems are somewhat artificial, most real problems with more than 100 or so variables are not possible to solve unless they show specific exploitable structure. Despite the possibility (or even likelihood) of enormous computing times, there are methods that can be applied to solving integer programs. The CPLEX solver in AMPL is built on a combination of methods, but based on a method called branch and bound. The purpose of this chapter is to show some interesting integer programming applications and to describe some of these solution techniques as well as possible pitfalls. First we introduce some terminology. An integer programming problem in which all variables are required to be integer is called a pure integer programming problem. If some variables are restricted to be integer and some are not then the problem is a mixed integer programming problem. The case where the integer variables are restricted to be 0 or 1 comes up surprising often. Such problems are called pure (mixed) 0-1 programming problems or pure (mixed) binary integer programming problems. The use of integer variables in production when only integral quantities can be produced is the most obvious use of integer programs.

2.5 Branch and bound

We will illustrate branch and bound by using an example . In that problem, the model is

$$\begin{array}{lrcl} \text{Minimize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 & & \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 & \leq & 14 \\ & x_j \in \{0,1\} j = 1, \cdots, 4. & & \end{array}$$

The linear relaxation solution is $x_1 = 1, x_2 = 1, x_3 = 0.5, x_4 = 0$ with a value of 22. We know that no integer solution will have value more than 22. Unfortunately, since x_3 is not integer, we do not have an integer solution yet. We want to force $x3$ to be integer. To do so, we branch on $x3$, creating two new problems. In one, we will add the constraint

$x3 = 0$. In the other, we add the constraint $x3 = 1$. Note that any optimal solution to the overall problem must be feasible to one of the subproblems. If we solve the linear relaxations of the subproblems, we get the following solutions:

- $x_3 = 0$: objective $21.65, x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0.667$;
- $x_3 = 1$:objective $21.85, x_1 = 1, x_2 = 0.714, x_3 = 1, x_4 = 0$

At this point we know that the optimal integer solution is no more than 21.85 (we actually know it is less than or equal to 21 (Why?)), but we still do not have any feasible integer solution. So, we will take a subproblem and branch on one of its variables. In general, we will choose the subproblem as follows:

- We will choose an active subproblem, which so far only means one we have not chosen before, and
- We will choose the subproblem with the highest solution value (for maximization) (lowest for minimization).

In this case, we will choose the subproblem with $x_3 = 1$, and branch on x_2. After solving the resulting subproblems, we have the branch and bound tree. The solutions are:

- $x_3 = 1, x_2 = 0$: objective $18, x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$;
- $x_3 = 1, x_2 = 1$: objective $21.8, x_1 = 0.6, x_2 = 1, x_3 = 1, x_4 = 0$

We now have a feasible integer solution with value 18. Furthermore, since the $x_3 = 1, x_2 = 0$ problem gave an integer solution, no further branching on that problem is necessary. It is not active due to integrality of solution. There are still active subproblems that might give values more than 18. Using our rules, we will branch on problem $x_3 = 1, x_2 = 1$ by branching on x_1 to get Figure 4. The solutions are:

- $x_3 = 1, x_2 = 1, x_1 = 0$:objective $21, x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$;
- $x_3 = 1, x_2 = 1, x_1 = 1$: infeasible.

Our best integer solution now has value 21. The subproblem that generates that is not active due to integrality of solution. The other subproblem generated is not active due to infeasibility. There is still a subproblem that is active. It is the subproblem with solution value 21.65. By our rounddown result, there is no better solution for this subproblem than 21. But we already have a solution with value 21. It is not useful to search for another such solution. We can fathom this subproblem based on the above bounding argument and mark it not active. There are no longer any active subproblems, so the optimal solution value is 21. We have seen all parts of the branch and bound algorithm. The essence of the algorithm is as follows:

1. Solve the linear relaxation of the problem. If the solution is integer, then we are done. Otherwise create two new subproblems by branching on a fractional variable.

2. A subproblem is not active when any of the following occurs:

   (a) You used the subproblem to branch on,

   (b) All variables in the solution are integer,

   (c) The subproblem is infeasible,

   (d) You can fathom the subproblem by a bounding argument.

3. Choose an active subproblem and branch on a fractional variable. Repeat until there are no active subproblems.

Thats all there is to branch and bound! Depending on the type of problem, the branching rule may change somewhat. For instance, if x is restricted to be integer (but not necessarily 0 or 1), then if $x = 4.27$ your would branch with the constraints $x \leq 4$ and $x \geq 5$ (not on $x = 4$ and $x = 5$). In the worst case, the number of subproblems can get huge. For many problems in practice, however, the number of subproblems is quite reasonable. For an example of a huge number of subproblems, try the following in AMPL:
var x0 binary;
var x{1..17} binary;

maximize z: -x0 + sum{j in 1..17} 2 * x[j];
subject to c: x0 + sum{j in 1..17} 2 * x[j] ¡= 17;

Note that this problem has only 18 variables and only a single constraint. CPLEX looks at 48,619 subproblems, taking about 90 seconds on a Sun Sparc 10 workstation, before deciding the optimal objective is 16. LINGO (another math programming package) on a 16MHz 386 PC (with math coprocessor) looks at 48,000+ subproblems and takes about five hours. The 100-variable version of this problem would take about 1029 subproblems or about 31018 years (at 1000 subproblems per second). Luckily, most problems take far less time.

2.6 Cutting plane techniques

There is an alternative to branch and bound called cutting planes which can also be used to solve integer programs. The fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. Of course, we have to be careful which constraints we add: we would not want to change the problem by adding the constraints. We will add a special type of constraint called a cut. A cut relative to a current fractional solution satisfies the following criteria:

1. every feasible integer solution is feasible for the cut, and
2. the current fractional solution is not feasible for the cut.

This is illustrated in Figure 2.3.

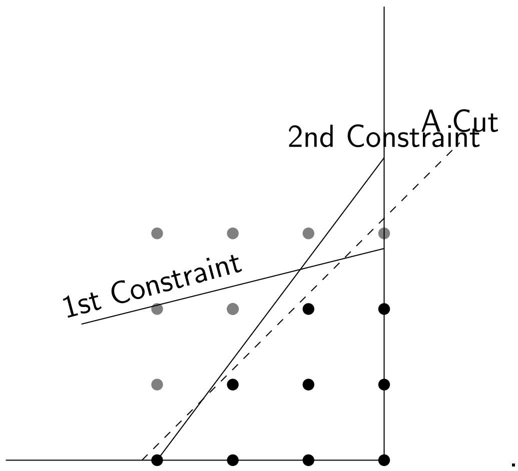


Figure 2.3: A cut

There are two ways to generate cuts. The first, called Gomory cuts, generates cuts from any linear programming tableau. This has the advantage of solving any problem but has the disadvantage that the method can be very slow. The second approach is to use the structure of the problem to generate very good cuts. The approach needs a problem-by-problem analysis, but can provide very efficient solution techniques.

Chapter 3

Short-term manpower planning in transhipment container terminals

Nowadays, maritime transport is the backbone of international trade, and containers play an interestingly crucial role in freight transportation. In 2013, the world container port throughput increased by 5.1 percent and reached 651.1 million TEUs [[76]]. In maritime transportation networks, shipping liners deploy deep-sea vessels (also called mother vessels) between a limited number of transsshipment container terminals (TCTs), whereas smaller vessels (also called feeders) link TCTs to origin and destination ports. The hub-and-spoke topology of maritime networks results in a critical role for TCTs, because of the consolidation of flows along the routes linking TCTs. Moreover, delays at TCTs can negatively impact the reliability of the liner service and generate additional costs for customers [[60], [80]]. Unlike origin and destination ports, TCTs operate under continuous and heavy competitive pressure, because shipping liners have high bargaining power in redesigning their maritime routes and excluding unsatisfactory TCTs. As a result, TCTs must provide high-performance and cost-effective services and accurately plan the management of their resources to satisfy the demanding requests of shipping liners. Since few ports adopt completely automated systems, human resources are relevant assets for TCTs, particularly in the case of high labour costs, and manpower management is a crucial activity for TCTs.

The workload of TCTs is typically organized in 24/7 shifts. Due to union and work rules, shifts must be planned several months before their implementation. However, when shifts are built, there is little or no knowledge about the arrival times of vessels and the final workload of TCTs, because maritime logistics is affected by both uncertainty and vulnerability, which may also result in port disruptions due to natural causes (wind, etc.), equipment failures, labour disputes, and geopolitical factors. As a result, the relevant uncertainty in maritime logistics causes frequent priority changes, and shipping liners often request service variations to TCTs, which must adapt internal processes to these external changes in order to remain competitive. In the case of the manpower problem, this situation can be addressed by its separation into two planning stages:

Long-term plan. This plan consists of a sequence of working and free days, which spans several months. In this plan, a TCT's worker is denoted as *fixed* in a shift of a day if he must be on-duty during that shift, whereas he is classified as *flexible* in a day if he must be on-duty during that day, but his shift in that day will be determined in the next planning stage, when there will be more precise information on the actual workload. The missing information on the real workload prevents deciding in the long-term plan what each worker is required to do and results in the risk of personnel overmanning and undermanning.

Short-term plan. This plan is typically performed 24 hours before the day in question, when the workload is almost certain. It is required to inherit the separation between fixed and flexible operators from the long-term plan, to determine shifts for flexible workers and to decide what each worker must do in the next workday. Moreover, in the short-term plan, TCTs must decide how many external workers should be hired and compute personnel undermanning and overmanning.

This thesis aims to model the short-term manpower planning problem and evaluate its effectiveness as it determines the final manpower costs for TCTs. Although there is much literature on workforce management in several fields [[27], [59]], little attention has been devoted thus far to the specific context of container terminals [[72], [25]].
Some studies addressed the scheduling of preselected operators to handling activities, whereas our problem setting is different, as we aim to select which workers will be employed by a TCT. [54] studied an allocation problem in which preselected servicemen are dispatched to locations in the yard, while minimizing the number of servicemen, distances, travel and waiting times. [47] investigated the operator-scheduling problem, in which each preselected operator is assigned to time slots of handling equipment. [41] proposed a model for scheduling and assigning a set of jobs on reefer containers to preselected operators.
A more similar study was done by [53], who focused on both the short-term and long-term manpower planning problems, but they neglected personnel overmanning and undermanning. [30] considered personnel undermanning in the short-term manpower planning problem, but ignored overmanning. The optimization model by [30] was also used in [69] to investigate the introduction of different levels of manpower flexibility.

In our opinion, the limited amount of studies on the manpower planning problem has resulted in few tools to support decision-making processes regarding this problem, whereas TCTs keep on facing it by ad hoc policies, which are based on hunch and experience. In addition, no studies shed light on these policies and evaluate how good they are. In this thesis, the manpower policy adopted in a real case study is illustrated and compared to the optimal solutions of an optimization model, which is proposed to minimize the assignment costs of workers to shifts, tasks and vessels, while avoiding both overmanning and undermanning.

In this chapter we present the results of two papers [32],[33] dealing with the problem of manpower planning in TCT. Two different manpower policies are proposed and validated in a real case study in Cagliari.

3.1 The manpower planning problem in transhipment container terminals

In this section, we describe the main components and decisions of the manpower planning problem. Decision-making processes in this problem should take into account a large array of information on the TCT workers from the long-term plan. An explicit example of the long-term plan of a worker is indicated in Figure 3.1. Each line represents a month, and each column is a day of that month. Six types of entries are used:

- I means that the worker is fixed in the first shift;
- II means that the worker is fixed in the second shift;
- III means that the worker is fixed in the third shift;
- FLX means that the worker is flexible in that day;
- RIC and FER denote days off. The first string represents days off already set since the first release of the long-term plan; the second one indicates additional days off added upon a worker's request.

Although external workers can also also employed when the inner manpower is insufficient, the number of inner and external operators is limited; thus workforce undermanning may occur. It must be avoided because TCTs cannot afford to pay penalties for delays produced on vessels. Personnel overmanning may also occur because the long-term plan is built when the workload is not yet known. It must be avoided because TCTs cannot pay

Mese	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	III	RIC	RIC	I	II	FLX*	III	RIC	RIC	I	FLX	II	III*	RIC	RIC	I	II	FLX	III	RIC*	RIC	I	FLX	II	III	RIC	RIC*	I	RIC	FLX	III
2	RIC	RIC	I*	FLX	II	III	RIC	RIC	I	II*	FLX	III	RIC	RIC	I	FLX	II*	III	RIC	RIC	I	RIC	FLX	III*	RIC	RIC	I	FLX			
3	II	III	RIC*	RIC	I	II	FLX	III	RIC	RIC*	I	FLX	II	III	RIC	RIC	RIC*	II	FLX	III	RIC	RIC	I	FLX*	II	III	RIC	RIC	I	II	FLX*
4	III	RIC	RIC	I	FLX	II	III*	RIC	RIC	I	II	FLX	III	RIC*	RIC	I	FLX	II	III	RIC	RIC*	I	II	FLX	III	RIC	RIC	I*	FLX	II	
5	III	RIC	RIC	I	RIC*	FLX	III	RIC	RIC	I	FLX	RIC*	III	RIC	RIC	I	II	FLX	III*	RIC	RIC	I	FLX	II	III	RIC*	RIC	I	II	FLX	III
6	RIC	RIC*	I	FLX	II	III	RIC	RIC	I*	II	FLX	III	RIC	RIC	I	FLX*	II	III	RIC	RIC	I	II	FLX*	III	RIC	RIC	I	FLX	II	III*	
7	RIC	RIC	I	II	FLX	III	RIC*	RIC	I	FLX	II	III	RIC	RIC*	FER	FER	FER	FER	RIC	RIC	FER*	FLX	II	III	RIC	RIC	I	II*	FLX	III	RIC
8	RIC	I	FLX	II*	III	RIC	RIC	I	II	FLX	RIC*	RIC	RIC	I	FLX	II	III	RIC*	RIC	FER	FER	FER	FER	RIC	RIC*	I	FLX	II	III	RIC	RIC
9	I*	II	FLX	III	RIC	RIC	I	FLX*	RIC	III	RIC	RIC	I	II	FLX*	III	RIC	RIC	I	FLX	II	III*	RIC	RIC	I	II	FLX	III	RIC*	RIC	
10	I	FLX	II	III	RIC	RIC*	I	RIC	FLX	III	RIC	RIC	I*	FER	FER	FER	RIC	RIC	FER	FER*	FLX	III	RIC	RIC	I	FLX	II*	III	RIC	RIC	I
11	II	FLX	RIC*	RIC	RIC	I	FLX	II	III	RIC*	RIC	I	II	FLX	III	RIC	RIC*	I	FLX	II	III	RIC	RIC	RIC*	II	FLX	III	RIC	RIC	I	
12	FLX*	II	III	RIC	RIC	I	II	FLX*	III	RIC	RIC	I	FLX	II	III*	RIC	RIC	FER	FER	FER	FER	RIC*	RIC	I	FLX	II	III	RIC	RIC*	I	II

Figure 3.1: An example of a real long-term plan

workers for doing nothing.

Overmanning and undermanning are the main issues of the short-term manpower problem. They are faced with swapping a day off with a workday in the long-term plan:

- In the case of undermanning, a workday is added to some workers on a day off according to the long-term plan, provided that there is sufficient rest time before and after their new shift. At the same time, a future workday is changed into a day off in the long-term plan.
- In the case of overmanning, a day off is added to some workers who were on duty according to the long-term plan. At the same time, a future day off is changed into a workday in the long-term plan.

Changing a close workday in a day off and vice versa is a viable modus operandi; in fact, it is already adopted by some TCTs. However, workers must be timely informed on these possible changes in the long-term plan in order to make them smoothly implementable and correct undermanning and overmanning promptly.

Two main types of activities are performed in maritime TCTs: vessel activities, i.e. loading/discharging operations, and housekeeping activities. A vessel activity is a sequence of handling operations in which containers are discharged from and loaded onto vessels. A housekeeping activity is a sequence of container transfers along the yard, which are performed when the area of the incoming and outgoing vessel activities differ [[51]]. Some activities have high priorities, whereas other activities are less important and can be postponed. Generally speaking, vessel activities have higher priority because shipping liners request them, whereas housekeeping must be unavoidably performed to be in the position of properly performing future vessel operations. Mother vessels have the highest priority and must be served on time, due to relevant penalties negotiated between shipping liners and TCTs on possible operation delays.

Each activity is carried out by a team (or gang) of workers, each of which is in charge of one task. Generally speaking, tasks have a tight hierarchy, and inner operators are paid according the top task they can do. They can be employed in lower-level tasks, but they cannot be assigned to upper lever tasks.

In the short-term plan, TCTs are required to inherit from the long-term plan the separation between fixed and flexible operators. While fixed operators in a specific day are already assigned to shifts in that day, the so-called flexible operators in a day will work in that day, and their shift must be determined in the short-term plan. According to work rules, flexible operators must be informed about their shifts at least 24 hours before their beginning. Once an inner worker is employed as a fixed or a flexible worker, he must be off duty for a minimum number of rest shifts. Although the long-term plan already guarantees sufficient rest times between consecutive workdays in fixed shifts, this is not always true for workers in flexible shifts, because they often cannot be assigned to any shift due to insufficient rest times. Therefore, the short-term plan must enforce the rest times before and after any flexible duty.

The workload in TCTs depends on the number of containers to be handled. However, since TCTs know how many containers can be handled in average by a gang, the workload demand can be described as number of workers required to perform a task for an activity in a shift. The objective of the short-term plan is to assign internal and external workers to shifts, tasks and activities at the lowest operating cost, as well as to minimize personnel undermanning and overmanning.

Unlike inner operators, the costs of external workers do not have individual attributes, because TCTs do not know a priori which external person will be employed. They can perform only a limited number of tasks and must be paid according to shifts, tasks and activities.

3.2 First manpower policy

In this section we present the short term manpower planning policy proposed in [32]. Different requirements are take into account to properly assign workers to shifts, tasks and operations, while accounting for some decisions already made in the long-term plan. An optimization model is formulated to discuss the decision-making ability of the real terminal in dealing with this problem. The experimentation shows that its current policy in the short-term plan is effective, but some improvements can be obtained if unnecessary restrictions were removed for the long term plan and some changes are accepted in the current modus opearndi of the terminal.

3.2.1 Mathematical model

In this paragraph, we present an optimization model for the short-term planning of the TCTs' manpower problem. Let I be the set of internal operators. They can be employed on a set J of shifts. Let d be the number of shifts in a day and t the index of the T days in the planning horizon, whose values range from 0 to $T-1$. It is worth noting that, in this thesis, the planning horizon of the short-term plan is longer than one day, provided that the workload is almost certain in this time interval. This novelty is motivated by the advantage of rapidly alerting TCTs on the possible occurrence of undermanning and overmanning in future days.
Let Z be the set of operations to be performed, which are divided into vessel and housekeeping operations. Since several tasks are requested to perform any operation $z \in Z$, let K^z be the set of tasks required for operation $z \in Z$ and $K_o^z \subseteq K^z$ the set of tasks that can be outsourced for activity $z \in Z$. Moreover, let $I_k \subseteq I$ be the set of operators able to perform task k, $I_j \subseteq I$ the set of fixed workers in shift $j \in J$, $I_t \subseteq I$ the set of flexible workers in day t.
The following data are defined. Let n_{jkz} be the number of operators required to perform task $k \in K^z$ for activity $z \in Z$ at shift $j \in J$. This work demand can be met by internal workers who can be deployed at shift $j \in J$ and are able to perform task k. They can be employed as fixed operators at shift $j \in J$ or flexible operators in the day including shift $j \in J$. Moreover, the work demand n_{jkz} can be met by external operators if $k \in K_o^z$. According to work rules, a rest of $r(j)$ shifts must be guaranteed to workers after a duty on shift $j \in J$.
Five types of variables are defined:

x_{ijkz} Operator selection variable, which takes value 1 if worker $i \in I$ is employed as a flexible worker in shift $j \in J$ to perform task $k \in K^z$ for operation $z \in Z$, 0 otherwise. Let $c_i \geq 0$ be the related unitary cost.

y_{ijkz} Operator selection variable, which takes value 1 if worker $i \in I$ is employed as a fixed worker in shift $j \in J$ to perform task $k \in K^z$ for operation $z \in Z$, 0 otherwise. Let $c_i \geq 0$ be the related unitary cost.

v_{jkz} Number of external workers deployed in shift $j \in J$ to perform task $k \in K_o^z$ for activity $z \in Z$. Let also d_{jkz} be the related unitary cost and w_j the maximum number of external workers who can be employed in shift $j \in J$.

u^+_{jkz} Number of workers in surplus in shift $j \in J$, task $k \in K^z$ for operation $z \in Z$. Let f^+_{jkz} be the related unitary cost.

u^-_{jkz} Number of workers in shortage in shift $j \in J$, task $k \in K^z$ for operation $z \in Z$. Let f^-_{jkz} be the related unitary cost.

The fulfillment of workload demand can be enforced as follows:

$$\sum_{i \in I_k} x_{ijkz} + \sum_{i \in I_k} y_{ijkz} + v_{jkz} - u^+_{jkz} + u^-_{jkz} = n_{jkz} \forall j \in J, \forall z \in Z, \forall k \in K_o^z \tag{3.1}$$

When tasks cannot be outsourced, constraint 3.1 takes the following form:

$$\sum_{i \in I_k} x_{ijkz} + \sum_{i \in I_k} y_{ijkz} - u^+_{jkz} + u^-_{jkz} = n_{jkz} \qquad \forall j \in J, \forall z \in Z, \forall k \in K^z - K_o^z \tag{3.2}$$

Constraint 3.3 enforces the employment of flexible operators, who can perform only one task and one activity in each day:

$$\sum_{j=td+1}^{(t+1)d} \sum_{z \in Z} \sum_{k \in K^z} x_{ijkz} = 1 \qquad \forall i \in I_t, t = 0, ..., |T| - 1 \tag{3.3}$$

Constraint 3.4 enforces the employment of fixed operators in the shift established in the long-term plan. They also perform only one task and one activity:

$$\sum_{z \in Z} \sum_{k \in K^z} y_{ijkz} = 1 \qquad \forall i \in I_j, j \in J \tag{3.4}$$

The assignment of workers to shifts should guarantee sufficient rest shifts between consecutive workdays for each operator. Each operator $i \in I$ must be off duty for $r(j)$ consecutive shifts after a duty as a flexible worker in shift $j \in J$:

$$\sum_{z \in Z} \sum_{k \in K^z} x_{ijkz} + \sum_{\rho=1}^{r(j)} (x_{i(j+\rho)kz} + y_{i(j+\rho)kz}) \leq 1$$

$$\forall i \in I, j = 1, ..., |J| - r(j) \tag{3.5}$$

Each fixed operator $i \in I$ must be off duty for $r(j)$ consecutive shifts after a duty in shift $j \in J$:

$$\sum_{z \in Z} \sum_{k \in K^z} y_{ijkz} + \sum_{\rho=1}^{r(j)} x_{i(j+\rho)kz} \leq 1$$

$$\forall i \in I, j = 1, ..., |J| - r(j) \tag{3.6}$$

It is worth noting that the constraints on the rest shifts can be ignored between two consecutive fixed shifts, because the long term-plan already guarantees sufficient rest times in this case.

The number of external sub-contracted workers in each shift $j \in J$ is limited:

$$\sum_{z \in Z} \sum_{k \in K^z_o} v_{jkz} \leq w_j \qquad \forall j \in J \tag{3.7}$$

The objective is to minimize the assignment costs of internal and external operators, as well as personnel workforce undermanning and overmanning:

$$\begin{aligned} min \sum_{i \in I} \sum_{j \in J} \sum_{z \in Z} \sum_{k \in K^z} c_i (x_{ijkz} + y_{ijkz}) + \sum_{j \in J} \sum_{z \in Z} \sum_{k \in K^z_o} d_{jkz} v_{jkz} + \\ \sum_{j \in J} \sum_{z \in Z} \sum_{k \in K^z} f^+_{jkz} u^+_{jkz} + \sum_{j \in J} \sum_{z \in Z} \sum_{k \in K^z} f^-_{jkz} u^-_{jkz} \end{aligned} \tag{3.8}$$

The model is used in a rolling horizon fashion, i.e., decisions are taken for all days of the planning horizon, but only the decisions on the first day of the planning horizon are implemented. This model is tested with a planning horizon of two days in the real case study described in Section 3.2.2.

3.2.2 The case study

In this section we illustrate and analyse the policy of a real container terminal for addressing the short-term manpower planning problem. In order to explain its policy, some details are provided on its organization.

Although many tasks are performed in the terminal, only three of them are taken into account in this analysis, because they are supposed to be the most important ones for this terminal: the quay cranes (QC) driver, who moves containers between vessels and terminal berths, the rubber-typed gantry crane (RTG) driver, who stores containers in the yard, and the driver of internal transfer vehicles (ITV), which are low-cost tractors providing horizontal transport. The QC is the top-level task; in fact, QC operators can also perform any other task, whereas RTG operators can also be employed as ITV ones, and ITV operators can perform this task only.

The long-term plan is taken for granted from the terminal, which divided each workday into 3 shifts of 8 hours each. The main task of each inner operator is known, and external workers can carry out the ITV task only in the case of workforce undermanning. The typical housekeeping gang is made up of 5 operators: 2 of them are deployed in the RTG task and 3 in the ITV one. The standard vessel gang operating on a quay crane is made up of 6 operators: 1 deployed in the QC task, 2 in the RTG one and 3 in the ITV one. However, 2 operators in each vessel gang must be able to perform the QC task: due to the physiological impossibility of keeping high handling rates for 8 hours in this crucial task, QC operators are deployed for 4 hours each in this task, which is swapped after half of the shift. Therefore, in the model of Section 3.2.1, the work demand is derived from vessel gangs with 2 QCs, 2 RTGs and 2 ITVs.

Although the precise costs cannot be disclosed for confidentiality purposes, we are allowed to provide information on the ratios between these costs and the cheapest one (i.e., the cost of internal workers in the ITV task): 1.25 for internal operators in the QC task, 1.0625 for internal operators in the RTG task, 1.625 for external operators (in the ITV task only), 12.5 for undermanning and overmanning for all tasks and activities.

3.2.2.1 The manpower policy

The terminal does not adopt any tool for addressing the manpower problem, and its policy on this problem is organized as follows.

First, the terminal assigns all operators in fixed shifts to their main tasks and to the most important activities, which are ranked by priority. This assignment starts from the QC task, continues with the RTG one and, next, with the ITV one. Therefore, the terminal

first assigns QC operators in fixed shifts to the most important activities: if the workforce demand in this task is perfectly met, the terminal switches to the analysis of the RTG task; in the case of overmanning, QC operators in surplus could be later selected for employment in the RTG task and/or the ITV one; in the case of undermanning, shortages will be addressed by the following assignment of QC operators in flexible shifts.
Next, the terminal checks the RTG task: if the workforce demand is perfectly met by adding RTG operators in fixed shifts, the terminal switches to the ITV task; in the case of overmanning, RTG operators in surplus could be later selected for employment in the ITV task; in the case of undermanning, shortages will be addressed by possible QC operators in surplus and, if this is not sufficient and/or possible, by the following assignment of flexible operators. Next, the terminal checks the ITV task: if the workforce demand is perfectly met by adding ITV operators in fixed shifts, the terminal switches to the analysis of operators in flexible shifts; in the case of overmanning, ITV operators are kept in surplus; in the case of undermanning, shortages will be addressed by the following assignment of operators in flexible shifts.
Next, the focus switches to operators in flexible shifts. They are assigned to their main tasks and activities according to the priority list, as long as some shortages occur at this stage after the previous assignment of fixed operators. First, the terminal checks the QC task and works as follows: if the workforce demand is perfectly met by adding QC operators in flexible shifts, the terminal switches to RTG operators; in the case of overmanning, it checks if QC operators in surplus can be assigned to the RTG task and/or the ITV one, and, if this is not sufficient or possible, all QC operators in surplus are put in a day off and a future rest day in the long-term plan is changed into a workday; in the case of undermanning, the terminal deploys QC operators having a day off, provided that they have sufficient rest time before and after the new shift. The procedure is then repeated for RTG operators and ITV ones. In the last case, shortages are addressed first by external operators and, if they are not sufficient, by operators who were supposed to have a day off.

3.2.2.2 Analysis of the manpower policy

The terminal decisions and the optimization model in Section 3.2.1 are compared in this section. The model is implemented by the PuLP library, an open source package that allows mathematical programs to be described in the Python computer programming language. In this thesis, PuLP is set to call the freeware solver GLPK 4.46 running on a PC with a 2.3 GHz processor and 8 GB of memory. We consider a set of 18 real instances denoted in Table 3.1 from P_1 to P_{18}, each of which represents a typical workday. All

instances are optimally solved in less than two minutes.
In each instance, the daily workforce inherited from the long-term plan is made up with:

- 8 operators, 4 RTG operators and 1 ITV operator in the first shift;
- 8 QC operators, 4 RTG operators and 1 ITV operator in the second shift;
- 8 QC operators, 4 RTG operators and 1 ITV operator in the third shift;
- 11 QC operators, 4 RTG operators and 5 ITV operators in flexible shifts.

All instances are divided into three shifts denoted by j=1, j=2 and j=3, each of which is associated with a manpower demand expressed in terms of number of gangs for vessels and housekeeping activities. For example, in the third shift of instance P_5, the workforce demand is 1 vessel gang and 3 housekeeping ones.
In Table 3.1, terminal decisions and model solutions are denoted by T and $\mathtt{M}$ respectively, whereas **CF** concerns the "clustered flexibility" described in Section 3.2.3. The columns of Table 3.1 have the following meanings:

- QC is the number of operators available for the QC task;
- RTG is the number of operators available for the RTG task;
- ITV is the number of inner operators available for the ITV task;
- EXT is the number of external operators deployed in the ITV task;
- QC^- is the number of QC operators in shortage;
- RTG^- is the number of RTG operators in shortage;
- ITV^- is the number of inner ITV operators in shortage;
- QC^+ is the number of QC operators in surplus;
- RTG^+ is the number of RTG operators in surplus;
- ITV^+ is the number of inner ITV operators in surplus.

Therefore, the total number of operators required in the QC task is $QC+QC^-$ in the case of undermanning and $QC - QC^+$ in the case of overmanning. The same computation can also be repeated for RTG and ITV tasks. For example, in the second shift of problem P_1, the terminal and the model recommend employing 5 internal operators in the QC task, 11 in the RTG one and 7 in the ITV one. Interestingly, although 35 QC operators are available, in the solution of P_1 only $6 + (5 - 1) + 6 = 16$ are deployed in this task, whereas the remaining $35 - 16 = 19$ are considered for deployment in lower-level tasks.
The comparison in Table 3.1 shows that the terminal's decisions and the model solutions are identical. Hence, one may infer that the manpower policy seems to be very effective; in fact, no improvement is found by the optimization model, which was exactly solved. From an algorithmic viewpoint, this policy is a greedy heuristic, which solves the problem because of:

- The clear priority rank among tasks (the topmost task is the QC, next the RTG and, finally, the ITV);
- The evaluation of all operators in their main tasks and in all possible lower-level tasks, to minimize undermanning and overmanning;
- The employment of external operators, only if necessary, to avoid undermanning.

Therefore, if one aims to pursue some improvements, they cannot be found in the short-term plan. This thesis shows that they can be observed if some changes are made in the long-term plan, whose construction presents two drawbacks:

- the frequent impossibility to deploy in any duty some workers in a flexible shift;
- the deployment of 2 QCs in each vessel gang, as this is the most expensive task;

These drawbacks are discussed and analyzed in Section 3.2.3 and 3.2.4.

3.2.3 The clustered flexibility

The terminal schedules flexible shifts in the long-term plan by the replication of predefined sequences of shifts, but some of them prevent assigning operators to any shift in a day. For example:

- 1 2 F 3, which means that an operator is fixed in the first shift of a day, fixed in the second shift of the next day, flexible in the following day and fixed in the third shift of the last day. Since the rest time after a duty is two shifts, in this case the operator can be deployed only in the second and the third shift of the third day.

		QC		RTG		ITV		EXT		QC^-		RTG^-		ITV^-		QC^+		RTG^+		ITV^+	
		T,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**	*T*,M	**CF**
P_1	j=1: 3, 0	6,6	**6**	7,7	**6**	10,10	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**0**	4,4	**0**
	j=2: 2, 0	5,5	**8**	11,11	**10**	7,7	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**4**	7,7	**6**	3,3	**1**
	j=3: 3, 0	6,6	**6**	6,6	**6**	1,1	**6**	5,5	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_2	j=1: 2, 0	5,5	**4**	8,8	**4**	10,10	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**0**	4,4	**0**	6,6	**1**
	j=2: 2, 0	8,8	**5**	5,5	**12**	10,10	**12**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	4,4	**1**	1,1	**8**	6,6	**8**
	j=3: 1, 0	3,3	**3**	5,5	**7**	5,5	**7**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**1**	3,3	**5**	3,3	**5**
P_3	j=1: 7, 0	7,7	**3**	12,12	**9**	6,6	**1**	8,8	**13**	7,7	**11**	2,2	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 5, 0	10,10	**10**	10,10	**10**	1,1	**4**	9,9	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 6, 0	4,4	**7**	8,8	**12**	1,1	**3**	11,11	**9**	8,8	**5**	4,4	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_4	j=1: 7, 0	5,5	**4**	10,10	**11**	4,4	**1**	10,10	**13**	9,9	**10**	4,4	**3**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 6, 0	12,12	**12**	12,12	**12**	3,3	**3**	9,9	**9**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 3, 0	6,6	**6**	6,6	**6**	1,1	**4**	5,5	**2**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_5	j=1: 7, 0	13,13	**14**	14,14	**13**	5,5	**1**	9,9	**13**	1,1	**0**	0,0	**1**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 2, 2	4,4	**4**	8,8	**8**	2,2	**5**	6,6	**3**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 1, 3	2,2	**2**	8,8	**8**	3,3	**4**	5,5	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_6	j=1: 2, 0	5,5	**4**	13,13	**5**	15,15	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**0**	9,9	**1**	11,11	**0**
	j=2: 1, 0	2,2	**2**	6,6	**4**	5,5	**7**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	4,4	**2**	3,3	**5**
	j=3: 6, 0	4,4	**12**	8,8	**12**	1,1	**9**	11,11	**3**	8,8	**0**	4,4	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_7	j=1: 1, 0	5,5	**4**	8,8	**3**	19,19	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	3,3	**2**	6,6	**1**	17,17	**4**
	j=2: 1, 0	3,3	**8**	2,2	**7**	9,9	**7**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**6**	0,0	**5**	7,7	**5**
	j=3: 1, 0	3,3	**3**	4,4	**6**	6,6	**15**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**1**	2,2	**4**	4,4	**13**
P_8	j=1: 5, 0	10,10	**10**	10,10	**10**	8,8	**10**	2,2	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 3, 0	6,6	**6**	6,6	**6**	6,6	**4**	0,0	**2**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 1, 0	2,2	**2**	6,6	**4**	5,5	**7**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	4,4	**2**	3,3	**5**
P_9	j=1: 2, 0	5,5	**5**	4,4	**4**	5,5	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**1**	0,0	**0**	1,1	**0**
	j=2: 8, 0	11,11	**13**	15,15	**14**	6,6	**4**	10,10	**12**	5,5	**3**	1,1	**2**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 3, 0	6,6	**6**	6,6	**6**	1,1	**3**	5,5	**3**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{10}	j=1: 9, 0	13,13	**11**	14,14	**12**	6,6	**2**	12,12	**16**	5,5	**7**	4,4	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 1, 0	5,5	**2**	4,4	**7**	4,4	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	3,3	**0**	2,2	**5**	2,2	**2**
	j=3: 4, 0	5,5	**8**	7,7	**8**	1,1	**5**	7,7	**3**	3,3	**0**	1,1	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{11}	j=1: 2, 0	5,5	**4**	9,9	**4**	11,11	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**0**	5,5	**0**	7,7	**1**
	j=2: 3, 0	6,6	**9**	9,9	**10**	6,6	**9**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**3**	3,3	**4**	0,0	**3**
	j=3: 3, 0	6,6	**6**	6,6	**6**	1,1	**6**	5,5	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{12}	j=1: 5, 0	9,9	**6**	10,10	**6**	6,6	**1**	4,4	**9**	1,1	**4**	0,0	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 5, 0	10,10	**10**	10,10	**10**	1,1	**4**	9,9	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 5, 0	5,5	**9**	7,7	**10**	1,1	**3**	9,9	**7**	5,5	**1**	3,3	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{13}	j=1: 2, 0	6,6	**5**	14,14	**4**	13,13	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	2,2	**1**	10,10	**0**	9,9	**0**
	j=2: 2, 0	4,4	**4**	4,4	**4**	5,5	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**1**
	j=3: 9, 0	5,5	**15**	7,7	**12**	1,1	**6**	17,17	**12**	13,13	**3**	11,11	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{14}	j=1: 2, 0	4,4	**4**	5,5	**4**	5,5	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**0**	1,1	**1**
	j=2: 6, 0	12,12	**12**	12,12	**12**	8,8	**7**	4,4	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 3, 0	6,6	**6**	6,6	**6**	1,1	**3**	5,5	**3**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{15}	j=1: 4, 0	8,8	**7**	11,11	**8**	13,13	**2**	0,0	**6**	0,0	**1**	0,0	**0**	0,0	**0**	0,0	**0**	3,3	**0**	5,5	**0**
	j=2: 1, 0	3,3	**3**	6,6	**5**	5,5	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	1,1	**1**	4,4	**3**	3,3	**3**
	j=3: 6, 0	5,5	**12**	7,7	**12**	1,1	**5**	11,11	**7**	7,7	**0**	5,5	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{16}	j=1: 1, 2	2,2	**2**	6,6	**6**	8,8	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	2,2	**0**
	j=2: 5, 0	10,10	**10**	10,10	**10**	10,10	**10**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 1, 2	2,2	**2**	6,6	**6**	5,5	**7**	1,1	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**1**
P_{17}	j=1: 3, 1	6,6	**6**	8,8	**8**	8,8	**2**	0,0	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 1, 3	4,4	**2**	10,10	**8**	10,10	**8**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	2,2	**0**	2,2	**0**	2,2	**0**
	j=3: 5, 0	4,4	**10**	8,8	**10**	1,1	**5**	9,9	**5**	6,6	**0**	2,2	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
P_{18}	j=1: 2, 2	4,4	**4**	8,8	**8**	5,5	**2**	3,3	**6**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=2: 6, 0	12,12	**12**	12,12	**12**	5,5	**8**	7,7	**4**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
	j=3: 1, 3	2,2	**2**	8,8	**8**	3,3	**3**	5,5	**5**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**	0,0	**0**
Total		*321*	**352**	*442*	**431**	*299*	**279**	*213*	**192**	*78*	**45**	*41*	**27**	*0*	**0**	*23*	**21**	*71*	**46**	*100*	**59**

Table 3.1: Terminal (T) vs Model (M) vs Clustered Flexibility (**CF**)

- 1 F 2, which means that an operator is fixed in the first shift of a day, flexible in the following day and fixed in the second shift of the last day. In this case, the operator can be deployed only in the first and second shift of the second day.

Generally speaking, flexibility opportunities are not completely exploited. This shortcoming is corrected by the proposed variant, which is called *clustered flexibility*. It is based on clustering flexible shifts in the long-term plan, as already done in the Gioia Tauro Terminal [[53]], while enforcing that 25% of shifts of each operator are flexible in a month, as imposed by work rules. In this study, we meet this requirement; in fact, each week 25% of operators are set to be on-duty in flexible shifts for 5 consecutive days, which precede and follow two off-duty days.
The clustered flexibility is denoted by **CF** in Table 3.1. The last row of this table shows that the clustered flexibility changes the deployment of inner operators: in the QC task, they are on duty for 352 times instead of 321, in the RTG task 431 times instead of 442, in the ITV task 279 times instead of 299. Interestingly, the number of inner operators deployed by the terminal and the clustered flexibility policy is identical (1062), but the clustered flexibility employs QC operators more frequently, and they can also be used in RTG and ITV tasks. Unsurprisingly, the recourse to external workforce for the ITV task dropped from 213 to 192 operators. Personnel undermanning decreases for QC operators from 78 to 45 and for RTG operators from 41 to 27. Finally, personnel overmanning decreases for QC operators from 23 to 21, for RTG operators from 71 to 46 and for ITV operators from 100 to 59.

3.2.4 QCs shift splitting

QC operators perform the most important task in TCTs because their productivity affects the overall terminal performance. However, it is not physiologically possible for them to keep high handling rates in shifts of 8 hours. The terminal policy accounts for these fatigue issues by vessel gangs with two operators able to perform the QC task. More precisely, each QC operator drives a quay crane for 4 hours a day and carries out a simpler task in the remaining 4 hours of his shift, even if he is paid as a QC for 8 hours a day. Therefore, 6 workers operate a quay crane for 4 hours each on a daily basis.
However, the QC is the most expensive task for TCTs, and it is beneficial to deploy the minimum number of QC operators. In order to face this drawback, we investigate a strategy called *QCs Shift Splitting*. It is based on the splitting of shifts of QC operators into two intervals of 4 hours, during which they only perform this task. The two intervals are separated by a break of 8 hours for workers in fixed duties and 4 or 8 hours for those

in flexible duties. According to this strategy, 3 workers operate a quay crane daily and, thus, the number of QC operators is halved.
The *QCs Shift Splitting* is investigated by a new optimization model, whose periods represent intervals of 4 hours. Therefore, consider the same notation in Section 3.2.1 and let J' be the set of periods of 4 hours, which is used instead of set J. As a result, the work demand is expressed with respect to index $j' \in J'$ and is denoted by $n_{j'kz}$, which represents the number of operators requested to perform task $k \in K^z$ for activity $z \in Z$ at period $j' \in J'$. Moreover, the number of periods in a day becomes $2d$ instead of d, which still denotes the number of shifts in a day, and the QC task is denoted by 1.
The new discretization of the planning horizon can be exploited if the long-term plan is changed accordingly. QC operators with fixed duty in the first shift have a fixed duty in the first and the fourth period of the modified long-term plan. If they are fixed in the second shift, the modified long-term plan reports a fixed duty in the second and fifth period; if they are fixed in the third shift, the modified plan reports a fixed duty in the third and sixth period. RTG and ITV operators with a fixed duty in the first shift have fixed duties in the first and second periods; if they are fixed in the second shift, they become fixed in the third and fourth periods; if they are fixed in the third shift, they become fixed in the fifth and sixth periods. More formally, let $I_{j'}$ the set of operators fixed in period $j' \in J'$. QC operators in flexible shifts are required to be on duty twice a day, but the work periods cannot be consecutive. RTG and ITV operators in flexible shifts are also required to be on duty twice a day, but the work periods must be consecutive.
The new model is formulated as follows:

$$min \sum_{i \in I} \sum_{j' \in J'} \sum_{z \in Z} \sum_{k \in K^z} c_i (x_{ij'kz} + y_{ij'kz}) + \sum_{j' \in J'} \sum_{z \in Z} \sum_{k \in K^z_o} d_{j'kz} v_{j'kz} + \\ + \sum_{j' \in J'} \sum_{z \in Z} \sum_{k \in K^z} f^+_{j'kz} u^+_{j'kz} + \sum_{j' \in J'} \sum_{z \in Z} \sum_{k \in K^z} f^-_{j'kz} u^-_{j'kz} \tag{3.9}$$

$$\sum_{i \in I_k} x_{ij'kz} + \sum_{i \in I_k} y_{ij'kz} + v_{j'kz} - u^+_{j'kz} + u^-_{j'kz} = n_{j'kz} \qquad \forall j' \in J', \forall z \in Z, \forall k \in K^z_o \tag{3.10}$$

$$\sum_{i \in I_k} x_{ij'kz} + \sum_{i \in I_k} y_{ij'kz} - u^+_{j'kz} + u^-_{j'kz} = n_{j'kz} \qquad \forall j' \in J', \forall z \in Z, \forall k \in K^z - K^z_o \tag{3.11}$$

$$\sum_{j'=2td+1}^{(t+1)2d} \sum_{z \in Z} \sum_{k \in K^z} x_{ij'kz} = 2 \qquad \forall i \in I_t, t = 0, ..., |T|-1 \tag{3.12}$$

$$x_{ij'1z} + x_{i(j'+1)1z} \leq 1 \qquad \forall i \in I_t, t = 0, ..., |T|-1, j = 2td+1...(t+1)2d, \forall z \in Z \tag{3.13}$$

$$x_{ij'kz} = x_{i(j'+1)kz} \qquad \forall i \in I_t, t = 0, ..., |T|-1, \forall j' mod 2 = 1, \forall z \in Z, \forall k \in K^z - 1 \tag{3.14}$$

$$\sum_{z \in Z} \sum_{k \in K^z} y_{ij'kz} = 2 \qquad \forall i \in I'_j, j' \in J' \tag{3.15}$$

$$\sum_{z \in Z} \sum_{k \in K^z} (x_{ij'kz} + y_{ij'kz} + x_{il'kz} + y_{il'kz}) \leq 1 \qquad \forall i \in I, j' = (t+1)d, ..., (t+1)2d,$$
$$l' = (t+1)2d+1, ..., (t+1)2d+d, t = 0, ..., |T|-1 \tag{3.16}$$

The objective function (3.9) minimizes the assignment costs of internal and external operators, as well as manpower undermanning and overmanning. The fulfillment of the workload demand is enforced by constraint (3.10) for tasks that can be outsourced and (3.11) in the opposite case. According to (3.12), each operator in flexible shifts must be deployed in two periods of 4 hours each. These periods cannot be consecutive for the QC task (constraint (3.13)), whereas they must be consecutive for RTG and ITV tasks (constraint (3.14)). According to (3.15), each operator in fixed shifts must be deployed in two periods of 4 hours each. Constraint (3.16) guarantees a rest time of 12 hours between two duties in consecutive workdays.

The scenario of QCs' shift splitting is compared to the current terminal policy, which was described in Section 3.2.2. Results are shown in Table 3.2, in which the terminal policy is denoted by *T* and the QCs' shift splitting by **SS**. This table presents the same notation of Table 3.1 on workforce availability (QC, RTG and ITV), outsourcing (EXT), undermanning (QC^-, RTG^- and ITV^-) and overmanning (QC^+, RTG^+ and ITV^+). Note that the figures on the terminal policy in Table 3.2 are exactly twice that of Table 3.1, due to the different discretization of the planning horizon.

According to the experimentation, the lessening in the shortage of operators is $(156 - 14)/156 = 91.03\%$ in the QC task and $(82-5)/82 = 93.90\%$ in the RTG one. No shortage is yet observed on ITV operators. These figures are certainly better than those in Table 3.1 in the case of clustered flexibility; in fact, the lessening in shortages was $(78-45)/78 = 42.31\%$ in the QC task and $(41-27)/41 = 34.15\%$ in the RTG one. Moreover, the lessening in the surplus of operators is $(46-22)/46 = 52.17\%$ in the QC task, $(142-89)/142 = 37.32\%$ in the RTG one and $(200-177)/200 = 11.50\%$ in the ITV one. In the case of clustered flexibility (Table 3.1), the lessening in shortages was $(23-21)/23 = 8.69\%$ in the QC task, $(71-46)/71 = 35.21\%$ in the RTG one, whereas the decrease was larger in the case of the ITV task ($(100-59)/100 = 41.00\%$).

	QC		RTG		ITV		EXT		QC^-		RTG^-		ITV^-		QC^+		RTG^+		ITV^+	
	T	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**	*T*	**SS**
Total	*642*	**384**	*884*	**908**	*598*	**832**	*426*	**581**	*156*	**14**	*82*	**5**	*0*	**0**	*46*	**22**	*142*	**89**	*200*	**177**

Table 3.2: The effect of QCs' shift splitting on the overall set of 18 instances

The QC shift splitting strategy is actually investigated by the terminal, as it follows the most important rules already adopted in this case study, such as informing workers on their shifts and providing sufficient rest times. Although the terminal is requested to plan daily duties of 8 hours, nothing is said about their possible division.
From a practical viewpoint, the QC shift splitting strategy results in the need to change the organization of the long-term plan for QC workers, but this is an easy modification in the system data. The decision-making process of the terminal is also recommended to change, as it can be effectively supported by optimization-based solutions instead of specific policies. The final acceptance of the QC splitting strategy depends on the final approval of QC workers: although some workers may not be willing to accept the new strategy, it produces some savings, which can be used as incentives for QC workers.

3.3 Second manpower policy

In this section we present the short term manpower planning policy proposed in [33]. It consists in determining shifts, tasks and activities of the manpower working in these terminals in order to serve vessels in time intervals, which typically do not overlap with personnel shifts. This complex problem is modelled by an integer linear programming formulation. The optimal solutions of the model are compared with the decisions made in accordance with the manpower policy adopted by a real transhipment container terminal. The experimentation sheds light on when its policy is effective or when there is room for optimisation. The computational tests indicate that the model can be optimally solved even in the case of huge transhipment container terminals.

3.3.1 Mathematical model

In this section, we propose a general mathematical programming model for the short-term manpower planning problem, which can be used for any configuration of personnel shifts, vessel activities and their possible overlapping. Let I be the set of the internal workers. They can be employed on a set J of periods, which span over a planning horizon of several days. Each period $j \in J$ represents a time interval, which is equal to the greatest common divisor between the duration of the shifts of the internal workers and the shortest vessel activity. In the example of Table 3.3.2, each period j is a time interval of 2 hours, which is the greatest common divisor between 8 and 6.

Let d be the number of periods in a day, s the number of periods in each shift of the internal workers, the number of shifts in a day and τ be the number of days in the planning horizon. Moreover, let e be the number of periods in a work time of the external workers. In the example of Table 3.3.2 , the number d of periods in a day is 12, the number s of periods in the shift of the internal workers is 4, the number ρ of daily shifts of the internal workers is 3 and the number τ of days in the planning horizon is 2. Since external workers must be on-duty for 6 hours, thus e is equal to 3 in this case study. According to this notation, in the first day of the planning horizon, the shifts of the internal workers start in periods $1 + (p - 1)s$, where p is the index of the daily shift and ranges from 1 to ρ. More generally, in day t, the shifts of the internal workers start in periods $[(t-1)d+1]+(p-1)s$, where t is the day index ranging from 1 to τ. Let Z_j be the set of activities to be performed in period $j \in J$. Since several tasks are requested to perform any activity $z \in Z_j$ in period $j \in J$, let K_z be the set of tasks requested for activity $z \in Z_j$ in period $j \in J$ and $\bar{K}_z \subseteq K_z$, the subset of tasks that can be outsourced for activity $z \in Z_j$ in period $j \in J$. For example, an element of K_z is the QC task, which is denoted by qc. Moreover, let $I_k \subseteq I$ be the set of the internal workers able to perform task $k \in K_z$ for activity $z \in Z_j$ in period $j \in J, I_j \subseteq I$ be the set of fixed workers in period $j \in J$ and $I_t \subseteq I$ be the set of flexible workers in day t. Let $\tilde{J}$ be the last period in which a worker can be deployed in the QC task (for example, if each period $j \in J$ is a time interval of 2 hours, Table 3.3.2 shows that QC workers must interrupt their activity at the end of period $j = 2$, thus $j = 2 \in \tilde{J}$). Let $\ddot{J}$ be the periods at the end of which QC workers must keep performing the current activity (for example, if each period $j \in J$ is a time interval of 2 hours, Table 3.3.2 shows that QC workers must perform the same activity in periods $j = 1$ and $j = 2$, thus $j = 1 \in \ddot{J}$). In addition, let $n_{k,z,j}$ be the number of internal workers requested to perform task $K \in K_z$ for activity $z \in Z_j$ in period $j \in J$. The following five types of variables are defined:

- $x_{i,k,z,j}$ is the internal worker selection variable, which takes value 1 if worker $i \in F \cup I_k$ is employed as a flexible worker to perform task $k \in K_z$ on activity $z \in Z_j$ in period $j \in J$, 0 otherwise. Let $c_{i,k,z,j} \geq 0$ be the related unitary cost.

- $y_{i,k,z,j}$ is the internal worker selection variable, which takes value 1 if worker $i \in I_j \cup I_k$ is employed as a fixed worker to perform task $k \in K_z$ on activity $z \in Z_j$ in period $j \in J$, 0 otherwise. Let $c_{i,k,z,j} \geq 0$ be the related unitary cost.

- $v_{k,z,j}$ is an integer non-negative variable representing the number of external workers hired to perform task $k \in \bar{K}_z$ on activity $z \in Z_j$ in period $j \in J$. Let w_j be the maximum number of external workers who can be hired in period $j \in J$ and $d_{k,z,j}$ be the associated unitary cost.

- $u^+_{k,z,j}$ is an integer non-negative variable representing number of workers in surplus in task $k \in K_z$ for activity $z \in Z_j$ in period $j \in J$. Let $f^+_{k,z,j}$ be the associated unitary cost.

- $u^-_{k,z,j}$ is an integer non-negative variable representing number of workers in shortage in task $k \in K_z$ for activity $z \in Z_j$ in period $j \in J$. Let $f^-_{k,z,j}$ be the associated unitary cost.

1. The workforce demand $n_{k,z,j}$ can be met by internal workers in flexible shifts, internal workers in fixed shifts and external workers, if tasks can be outsourced. As the available manpower is typically different from the workload demand, the surplus of workers or the deficiency of workers must be computed in each task $k \in K_z$, each activity $z \in Z_j$ and each period $j \in J$, in order to obtain exact demand satisfaction. This constraint can be enforced as follows:

   1.a

$$\sum_{i \in I_k \cup I_t} x_{i,k,z,j} + \sum_{i \in I_k \cup I_j} y_{i,k,z,j} + v_{k,z,j} - u^+_{k,z,j} + u^-_{k,z,j} = n_{k,z,j} \quad j = 1 + (t-1)d \cdots dt,$$
$$t = 1 \cdots \tau, \forall z \in Z_j, \forall k \in \bar{K}_z$$

   1.b If tasks cannot be outsourced, constraint (1.a) is formulated as follows:

$$\sum_{i \in I_k \cup I_t)} x_{i,k,z,j} + \sum_{i \in I_k \cup I_j} y_{i,k,z,j} - u^+_{k,z,j} + u^-_{k,z,j} = n_{k,z,j}$$
$$j = 1 + (t-1)d \cdots dt, t = 1 \cdots \tau, \forall z \in Z_j, \forall k \in K_z \setminus \bar{K}_z$$

2. Each internal worker flexible in day t must start his duty in the first period of a shift of that day:

$$\sum_{k\in K_z}\sum_{z\in Z_j}\sum_{j=[(t-1)d+1]+(p-1)s \quad p=1,\cdots,\rho} x_{i,k,z,j} = 1 \forall i \in I_t$$

3. Each internal worker flexible in day t must be on duty for s consecutive periods in that day:

$$\sum_{k\in K_z}\sum_{z\in Z_j} x_{i,k,z,j} = \sum_{k\in K_z}\sum_{z\in Z_j} x_{i,k,z,(j+1)} \forall j \bmod s \neq 0, (t-1)d+1 \leq j \leq td, t=1,\cdots\tau, \forall i \in I_t$$

4. Each fixed internal worker starts his duty in the first period of a shift decided in the long-term plan:

$$\sum_{k\in K_z}\sum_{z\in Z_j} y_{i,k,z,j} = 1 \quad j=[(t-1)d+1]+(p-1)s, p=1,\cdots,\rho, t=1,\cdots\tau, \forall i \in I_j$$

5. Each fixed internal worker must be on-duty in his shift for s consecutive periods a day:

$$\sum_{k\in K_z}\sum_{z\in Z_j} y_{i,k,z,j} = \sum_{k\in K_z}\sum_{z\in Z_j} y_{i,k,z,(j+1)} \quad \forall i \in I_j, \forall j \bmod s \neq 0$$

6. Each worker $i \in I$ must have a rest time of $r(j)$ periods after a duty finishing in period $j \in J$:

$$\sum_{k\in K}\sum_{z\in Z}\Big[x_{i,k,z,j}+y_{i,k,z,j}+\sum_{=1}^{r(j)}(x_{i,k,z,(j+)}+y_{i,k,z,(j+)})\Big] \leq 1 \quad j \bmod s = 0, \forall z \in Z_j, k \in K_z, \forall i \in I$$

7. The number of external workers hired in each period is limited:

$$\sum_{k\in \bar{K}_z}\sum_{z\in Z_j} v_{k,z,j} \leq w_j \quad \forall j \in J$$

8. Whenever external workers are hired, they must complete their shift, thus the number of external workers must not change during their shift. As they must work for e consecutive periods, the following constraint is introduced:

$$\sum_{k\in\bar{K}_z}\sum_{z\in Z_j} v_{k,z,j} = \sum_{k\in\bar{K}_z}\sum_{z\in Z_j} v_{k,z,(j+1)} \quad \forall j \bmod e \neq 0$$

9. The rotation in the QC task must be enforced in the middle of shifts for workers in flexible duties:

$$\sum_{z\in Z_j}(x_{i,k,z,j} + x_{i,k,z,(j+1)} \leq 1 \quad j \in \tilde{J}, \forall z \in Z_j, k = qc \in K_z, \forall i \in I_t \cup I_k$$

10. In the example of Table 3.3.2, if each period $j \in J$ is a time interval of 2 hours, the periods in $\ddot{J}$ take values $(t-1)d + b, b = 2, 6, 10$, where the day index t takes values 1 and 2. When this rotation must not be performed, the flexible workers in the QC task are enforced to keep performing the current activity:

$$x_{i,k,z,j} = x_{i,k,z,(j+1)} \quad j \in \ddot{J}, \forall z \in Z_j, k = qc \in K_z, \forall i \in I_t \cup I_k$$

11. In the example of Table 3.3.2, if each period $j \in J$ is a time interval of 2 hours, the periods in $\ddot{J}$ take values $(t-1)d+b, b = 1, 5, 7, 11$, where the day index t takes values 1 and 2. The same restrictions must be enforced for QC workers in fixed shifts:

$$\sum_{z\in Z_j}(y_{i,k,z,j} + y_{i,k,z,(j+1)} \leq 1 \quad j \in \bar{J}, \forall z \in Z_j, k = qc \in K_z, \forall i \in I_t \cup I_k$$

12. y$_{i,k,z,j} = y_{i,k,z,(j+1)} j \in \ddot{J}, \forall z \in Z_j, k = qc \in K_z, \forall i \in I_t \cup I_k$

13. The objective is to minimise the assignment costs of internal as well as external workers to tasks, activities and periods, as well as the costs of personnel under-manning and over-manning:

min$\sum_{i\in I_k}\sum_{k\in K_z}\sum_{z\in Z_j}\sum_{j\in J} c_{i,k,z,j}(x_{i,k,z,j} + y_{i,k,z,j}) + \sum_{k\in\bar{K}_z}\sum_{z\in Z_j}\sum_{j\in J} d_{k,z,j}v_{k,z,j} + \sum_{k\in K_z}\sum_{z\in Z_j}\sum_{j\in J} f^{+}_{k,z,j}u^{+}_{k,z,j} + + \sum_{k\in K_z}\sum_{z\in Z_j}\sum_{j\in J} f^{-}_{k,z,j}u^{-}_{k,z,j}$

3.3.2 Case study

This section presents a case study on the short-term manpower problem in a real TCT, where each workday of internal workers is divided into three consecutive shifts of 8 hours each. Moreover, vessel services are typically provided to shipping liners in time intervals multiple of 6 hours. External workers could also be hired to overcome under-manning in vessel services and they must be on-duty for 6 hours a day whenever they are hired. If all vessel services are 6 hours long, all time intervals can be represented as shown in Table 3.3.2 in the case of available information regarding vessel arrival and departure times for the next two days. This example indicates that several criticalities take place because of the different durations of internal workers shifts and vessel services. For example, some vessels may berth at the end of the 6th hour of the planning horizon and internal workers must be assigned to these new activities during their shift. The same problem occurs at the end of the 12th, 18th, 30th, 36th and 42th hour of the planning horizon. In addition, vessel services must be interrupted at the end of the 8th, 16th, 32th and 40th hour of the planning horizon owing to shift change. Although many tasks are performed in this case study, only the most important are investigated: the Quay Crane (QC) driver, who picks up and drops of containers from vessels, the Rubber Typed Gantry crane (RTG) driver, who stores containers in the yard and the driver of Internal Transfer Vehicles (ITV), which are low-cost tractors providing horizontal transport. The QC is the top-level task, in fact QC workers can also perform any other task; RTG workers can also be employed in the ITV task and ITV workers can perform this task only. External workers are allowed to carry out the ITV task only. In this case study, housekeeping gangs are made up of 5 workers: 2 of them are deployed in the RTG task and the remaining 3 in the ITV task. The standard vessel gang is made up of 6 workers: 1 deployed in the QC task, 2 in the RTG task and 3 in the ITV task. For example, consider this sequence of vessel activities in the first day of the planning horizon: 9 activities in the first service interval of 6 hours, 3 in the second, 7 in the third and 5 in the fourth interval. Consequently, the manpower demand becomes:

- 9 QCs, 18 RTGs and 27 ITVs in the first service interval;
- 3 QCs, 6 RTGs and 9 ITVs in the second service interval;
- 7 QCs, 14 RTGs and 21 ITVs in the third service interval;
- 5 QCs, 10 RTGs and 15 ITVs in the fourth service interval.

DAY 1			DAY 2		
THE FIRST SHIFT OF INTERNAL WORKERS	THE SECOND SHIFT OF INTERNAL WORKERS	THE THIRD SHIFT OF INTERNAL WORKERS	THE FOURTH SHIFT OF INTERNAL WORKERS	THE FIFTH SHIFT OF INTERNAL WORKERS	THE SIXTH SHIFT OF INTERNAL WORKERS
8 h	8 h	8 h	8 h	8 h	8 h

THE FIRST	THE SECOND	THE THIRD	THE FOURTH	THE FIFTH	THE SIXTH	THE SEVENTH	THE EIGHTH
WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS	WORK PERIOD OF EXTERAL WORKERS
6 h	6 h	6 h	6 h	6 h	6 h	6 h	6 h
THE FIRST	THE SECOND	THE THIRD	THE FOURTH	THE FIFTH	THE SIXTH	THE SEVENTH	THE EIGHTH
VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL	VESSEL SERVICE INTERVAL
6 h	6 h	6 h	6 h	6 h	6 h	6 h	6 h

Figure 3.2: Discretisation of the planning horizon

Moreover, in this case study, two workers in each vessel gang must be able to perform the QC task, owing to the physiological impossibility of keeping high handling rates for 8 hours in this crucial task. Consequently, two workers are deployed in the QC task for 4 consecutive hours in their shift. Therefore, in this example,18 workers must be able to perform the QC task in the first service interval, 6 in the second, 14 in the third and 10 in the fourth interval. The example in Table 3.3.2 also provides some guidelines to minimise the time lost by QC workers during the change of vessel activities: if all requested vessel services are 6 hours long, then the workers in the QC task are recommended to perform the same vessel activity in the first 4 hours of the first vessel service, in the last four hours of the second vessel service, in the first 4 hours of the third vessel service, in the last four hours of the fourth vessel service and so on. In this case study, the daily manpower demand can be typically met by 60 internal workers: 52 workers able to perform the QC task, as well as the RTG and the ITV task, 7 workers for the RTG and the ITV task, 1 for the ITV task. In addition, 13 workers are fixed in the first shift, 13 workers in the second shift, 13 workers in the third shift and 21 workers are employed on a flexible duty.

This numerical example also points out the need of removing the overlapping between internal workers shifts and service intervals, as this overlapping involves the quantification of the manpower demands by time intervals of 8 hours. In order to minimise the risk of personnel under-manning, the TCT of this case study usually considers the maximum number of activities between the consecutive intervals of 6 hours: since max ((9, 3)=9, max (3,7)=7 and max (7,5)=7, in the case of overlapping, the sequence of vessel activities would become 9 activities in the first service interval of 8 hours, 7 in the second and 7 in the third. As a result, this modus operandi overestimates the number of vessel activities in the last two hours of the first shift, in the first four hours of the second shift and in the last six hours of the third shift. Generally speaking, removing the overlapping is beneficial in the short-term manpower planning problem, because the workforce can be assigned to vessels services of any duration requested by shipping liners.

3.3.3 Testing

The policy of this TCT regarding the short-term manpower problem can be described by a simple greedy heuristic algorithm, which works as follows. At first, the TCT assigns all workers in fixed shifts to tasks and activities, which are ranked and served by priority. This assignment starts from the workers capable of performing the QC task, goes on with those capable of performing the RTG task and continues to those capable of performing the ITV task. In case of over-manning on a specific task, the surplus of workers may be corrected by the employment in lower-level tasks. If this is not possible, they are put in a day-off and a future rest day in the long-term plan is changed into a workday. In the case of under-manning on a specific task, deficiencies will be later addressed by the assignment of workers in flexible shifts. Subsequently, the TCT focuses on workers with flexible shifts and determines when they must be on-duty in order to have the same workforce in all shifts. This choice is motivated by its simplicity and the current lack of decision tools, which are able to set and evaluate different manpower configurations. However, the uniform manpower supply may disclose its ineffectiveness whenever the manpower demand is not uniform in the daily planning horizon. After that, the terminal policy determines the tasks and activities of workers in flexible shifts. These decisions start from the QC task, go on with the RTG task and end with the ITV task. In the last case, the possible under-manning is addressed by external workers and if they are still insufficient, the under-manning is addressed by the overtime of workers in a day off, provided they have sufficient rest time before and after the new shift. In the case of over-manning, workers are put in a day off and a future rest day in the long-term plan is changed into a workday. The TCT policy and the solutions of the optimisation model are compared in a set of problem instances taken from the case study in Section 2.2. The model is implemented using the PuLP library, which is an open-source package that allows mathematical programmes to be described in the Python computer programming language. In this thesis, PuLP is set to call the solver Cplex 12.6.2 running on a PC with 1,3 GHz Intel Core I5, 4GB of RAM and 4 cores. Thirty real instances are taken from 30 consecutive days in the long-term plan. Since the model is implemented with a planning horizon of two days ($|J| = 24$ with periods of 2 hours) and the TCT policy is implemented for a single day, the comparison is performed only on the first day of each instance. To clarify, the model is used in a rolling horizon fashion: decisions are taken for all days in the planning horizon, but only the decisions made in the first day are implemented on the three investigated tasks ($|K| = 3$). The instances are reported in the lines of Tables 2, 3 and 4. They are introduced by $|I| = 60, w_j = 40 \forall j$, as this case study has 60 internal workers in each day and 40 external workers at most in each period j. The columns denoted by Z1 report the sequence of vessel and housekeeping activities performed by the terminal in the first day of the planning horizon. For example, the line (5,0), (8,0), (6,0), (1,0) means 5 vessel activities and 0 housekeeping activities in the first service interval, 8 vessel activities and 0 housekeeping activities in the second service interval, 6 vessel activities and 0 housekeeping activities in the third service interval, 1 vessel activity and 0 housekeeping activities in the fourth service interval. Since the planning horizon spans two days, the activities in the second day are reported in the following line. For example, in the first instance the activities in the first day are (1,0), (1,0), (1,0), (1,0) and the activities in the second day are (5,0), (5,0), (5,0), (5,0). All instances are optimally solved in a few seconds, as detailed in section 4.2. As the terminal policy provides a uniform manpower supply, it is worth discussing its effectiveness with respect to different values of variance in the manpower demand. Instances in Table 3.3.3 present a variance lower than 1, those in Table 3.3.3 present a variance ranging between 1 and 5, whereas those in Table 3.3.3 present a variance larger than 5. The decisions derived from the terminal policy and those obtained by the optimal model solutions are denoted by $t$and M respectively. Moreover, the following notations are adopted:

- QC is the number of workers used in the QC task in all periods;
- RTG is the number of workers used in the RTG task (even if their top task may be QC) in all periods;
- ITV is the number of internal workers used in the ITV task (even if their top task may be QC or RTG) in all periods;
- EXT is the number of external workers (hired for the ITV task only) in all periods;
- U$+$ number of internal workers in surplus in all periods;
- QC- is the number of QC workers deficient in all periods;
- RTG- is the number of RTG workers deficient in all periods;
- ITV- is the number of internal ITV workers deficient in all periods.

$\lvert I\rvert=60, w_j = 40\ \forall j$	QC		RTG		ITVS		EXT		U+		QC-		RTG-		ITVS-	
Z	T	M	T	M	T	M	T	M	T	M	T	M	T	M	T	M
(1,0),(1,0),(1,0),(1,0)	12	12	24	24	36	36	0	0	120	120	0	0	0	0	0	0
(5,0),(5,0),(5,0),(5,0)	60	60	120	120	0	0	180	180	0	0	0	0	0	0	0	0
(2,0),(3,0),(3,0),(3,0)	33	33	66	66	63	63	36	36	6	6	0	0	0	0	0	0
(2,0),(2,0),(1,0),(1,0)	18	18	36	36	54	54	0	0	72	72	0	0	0	0	0	0
(3,0),(2,0),(3,0),(2,0)	30	30	60	60	72	78	18	12	18	12	0	0	0	0	0	0
(4,0),(4,0),(5,0),(5,0)	54	54	102	104	12	9	150	153	0	1	0	0	6	4	0	0
(7,0),(5,0),(6,0),(5,0)	69	69	111	111	0	0	207	207	0	0	0	0	27	27	0	0
(3,0),(5,0),(3,0),(3,0)	42	42	84	84	54	54	78	78	0	0	0	0	0	0	0	0
(1,4),(3,2),(5,0),(5,0)	42	42	120	120	78	78	102	102	0	0	0	0	0	0	0	0
(2,2),(3,1),(3,2),(4,1)	36	36	108	108	96	96	66	66	0	0	0	0	0	0	0	0
(5,2),(3,2),(4,2),(4,1)	48	48	138	138	54	54	153	153	0	0	0	0	0	0	0	0
Total	444	444	969	971	519	522	990	987	216	211	0	0	33	31	0	0

Figure 3.3: Instances with low variance in the manpower demand

$\lvert I\rvert=60, w_j = 40\ \forall j$	QC		RTG		ITVS		EXT		U+		QC-		RTG-		ITVS-	
Z	T	M	T	M	T	M	T	M	T	M	T	M	T	M	T	M
(4,0),(6,0),(4,0),(6,0)	60	60	96	102	12	0	168	180	0	6	0	0	24	18	0	0
(7,0),(4,0),(4,0),(4,0)	57	57	93	105	18	0	153	171	0	6	0	0	21	9	0	0
(3,0),(6,0),(3,0),(6,0)	54	54	84	87	30	21	132	141	0	6	0	0	24	21	0	0
(5,0),(1,0),(1,0),(5,0)	36	36	66	72	18	21	90	87	48	39	0	0	6	0	0	0
(4,0),(1,0),(6,0),(1,0)	36	36	63	69	27	18	78	90	54	57	0	0	6	3	0	0
(1,0),(1,0),(1,0),(6,0)	27	27	45	54	27	51	54	30	81	48	0	0	9	0	0	0
(2,2),(3,3),(3,1),(6,0)	42	42	120	120	78	78	102	102	0	0	0	0	0	0	0	0
(2,3),(1,0),(1,0),(5,0)	27	27	72	72	57	141	51	30	84	63	0	0	0	0	0	0
(3,1),(1,0)(4,2),(0,1)	27	27	72	72	57	137	51	12	84	41	0	0	0	0	0	0
Total	366	366	711	753	324	467	879	843	351	266	0	0	90	51	0	0

Figure 3.4: Instances with medium variance in the manpower demand

The last line of each table reports the sum of the results over all problem instances. Table 3.3.3 shows, in the case of low variance, the terminal policy returns similar results with respect to the optimisation model. The model slightly decreases under-manning in the RTG task, over-manning and the use of external workers.

Table 3.3.3 shows that in case of medium manpower variance, the model is much more able to reduce deficiencies in the RTG task. No deficiency is observed on QCs and ITV tasks owing to the high manpower supply in the QC task as well as external workers hired in the ITV task respectively. However, the model makes use of larger number of external workers to perform the ITV task, whereas the terminal policy deploys larger number of internal workers in this task.

Table 3.3.3 shows that in the case of high variance, the model is much more efficient in reducing under-manning and over-manning while hiring similar number of external workers. The higher quality of model solutions depends on the possibility of employing flexible workers in peak periods of the manpower demand, whereas this option is not

|I|=60,w_j = 40 ∀j	QC		RTG		ITVS		EXT		U+		QC-		RTG-		ITVS-	
Z	*T*	**M**	*T*	**M**	*T*	**M**	*T*	**M**	*T*	**M**	*T*	**M**	*T*	**M**	*T*	**M**
(5,0),(8,0),(6,0),(1,0)	*60*	**60**	*84*	**93**	*9*	**9**	*171*	**171**	*27*	**18**	*0*	**0**	*36*	**18**	*0*	**0**
(6,0),(1,0),(7,0),(1,0)	*45*	**45**	*63*	**74**	*18*	**18**	*117*	**117**	*54*	**43**	*0*	**0**	*27*	**16**	*0*	**0**
(7,0),(1,0),(1,0),(7,0)	*48*	**48**	*66*	**81**	*18*	**18**	*126*	**126**	*60*	**45**	*0*	**0**	*30*	**15**	*0*	**0**
(9,0),(1,0),(4,0),(1,0)	*45*	**45**	*54*	**81**	*27*	**18**	*108*	**117**	*54*	**36**	*0*	**0**	*36*	**9**	*0*	**0**
(1,0),(1,0),(8,0),(7,0)	*51*	**51**	*51*	**73**	*18*	**18**	*135*	**135**	*48*	**26**	*0*	**0**	*57*	**26**	*0*	**0**
(1,0),(10,0),(1,0),(2,0)	*42*	**42**	*39*	**54**	*36*	**27**	*99*	**99**	*63*	**57**	*0*	**0**	*45*	**30**	*0*	**0**
(1,0),(1,0),(1,0),(10,0)	*39*	**39**	*33*	**72**	*27*	**27**	*90*	**90**	*81*	**42**	*0*	**0**	*45*	**6**	*0*	**0**
(4,2),(0,1),(6,1),(1,0)	*33*	**33**	*90*	**90**	*30*	**117**	*105*	**105**	*87*	**87**	*0*	**0**	*0*	**0**	*0*	**0**
(6,3),(1,0),(3,1),(0,1)	*30*	**30**	*78*	**90**	*45*	**120**	*90*	**81**	*87*	**66**	*0*	**0**	*12*	**0**	*0*	**0**
(1,0),(1,0),(0,1),(7,3)	*27*	**27**	*57*	**78**	*27*	**135**	*90*	**69**	*129*	**87**	*0*	**0**	*21*	**0**	*0*	**0**
Total	*420*	**420**	*615*	**786**	*255*	**507**	*1131*	**1110**	*690*	**507**	*0*	**0**	*309*	**120**	*0*	**0**

Figure 3.5: Instances with high variance in the manpower demand

possible according to the terminal policy.

In this section, the proposed optimisation model is also tested on larger TCTs in order to evaluate its reproducibility. As the worlds largest transhipment terminal has 40 quay cranes [https://www.singaporepsa.com/our-business/terminals] and the experimentation in section 4.1 was done on a TCT equipped with 10 quay cranes, we have also investigated new problem instances, which are two, three and four times larger than those reported in section 4.1. More precisely, the cardinality of set I of internal workers, the maximum number of external workers w_j in each period $j \in J$ and the manpower demand n_{kzj} are supposed to be two, three and four times larger than the values adopted in section 4.1. All problem instances are optimally solved by Cplex 12.6.2 running on a PC with 1,3 GHz Intel Core I5, 4GB of RAM and 4 cores. The results are reported in Table 3.3.3, in which each problem instance is described by the quantification of the number of workers and activities, whereas the number of periods $|J|$ is 24 and the number of tasks $|K|$ is 3 for the overall experimentation. The columns of Table 3.3.3 are divided into four groups, each representing different terminal class size:

- The first group of columns concerns the real instances described in section 4.1 on a terminal with 10 quay cranes and is denoted by $|I| = 60, w_j = 40 \forall j \in J$, as it has 60 internal workers and 40 external workers at most in each period j. Columns report the sequence of vessel and housekeeping activities performed by the TCT in the first day of the planning horizon (denoted by Z1), the solution time (denoted by T[s]) and the gap between the optimum and the LP relaxation (denoted by LG[%]). For example, line (5,0), (8,0), (6,0), (1,0) in Z1 means 5 vessel activities and 0 housekeeping activities in the first service interval, 8 vessel activities and 0 housekeeping activities in the second service interval, 6 vessel activities and 0 house-

$\lvert I\rvert$=60,w_j = 40 ∀j			$2\lvert I\rvert$, $2w_j$, $2Z_1$		$3\lvert I\rvert$, $3w_j$, $3Z_1$		$4\lvert I\rvert$, $4w_j$, $4Z_1$	
Z_1	T[s]	LG[%]	T[s]	LG[%]	T[s]	LG[%]	T[s]	LG[%]
(1,0),(1,0),(1,0),(1,0)	0.67	0.00%	6.15	1.40%	11.11	0.00%	28.27	0.00%
(5,0),(5,0),(5,0),(5,0)	0.99	0.00%	5.55	0.00%	13.68	0.00%	41.67	0.00%
(2,0),(3,0),(3,0),(3,0)	0.14	0.00%	1.67	0.00%	3.71	0.00%	7.99	0.00%
(2,0),(2,0),(1,0),(1,0)	0.56	1.51%	1.22	0.00%	7.34	0.49%	9.89	0.37%
(3,0),(2,0),(3,0),(2,0)	0.66	0.99%	4.30	0.00%	24.30	0.32%	49.40	0.24%
(4,0),(4,0),(5,0),(5,0)	5.20	7.57%	38.09	0.11%	166.54	0.05%	154.53	0.00%
(7,0),(5,0),(6,0),(5,0)	7.30	8.32%	33.77	0.03%	140.42	0.10%	493.85	0.06%
(3,0),(5,0),(3,0),(3,0)	0.19	0.00%	2.20	0.00%	4.07	0.00%	9.38	0.00%
(1,4),(3,2),(5,0),(5,0)	0.28	0.00%	2.88	0.00%	8.59	0.00%	17.00	0.00%
(2,2),(3,1),(3,2),(4,1)	1.04	0.00%	5.28	0.00%	48.87	0.09%	31.38	0.00%
(5,2),(3,2),(4,2),(4,1)	0.80	0.00%	5.61	0.00%	10.58	0.00%	79.56	0.21%
(4,0),(6,0),(4,0),(6,0)	1.04	0.00%	9.03	0.00%	29.70	0.00%	109.61	19.13%
(7,0),(4,0),(4,0),(4,0)	1.00	0.00%	7.33	0.00%	20.06	0.00%	56.22	0.00%
(3,0),(6,0),(3,0),(6,0)	0.66	0.00%	3.96	0.00%	11.37	0.00%	27.53	0.00%
(5,0),(1,0),(1,0),(5,0)	0.42	0.00%	2.72	0.00%	6.49	0.00%	16.07	0.00%
(4,0),(1,0),(6,0),(1,0)	0.22	0.00%	1.99	0.00%	4.41	0.00%	9.25	0.00%
(1,0),(1,0),(1,0),(6,0)	0.71	0.00%	5.11	0.00%	13.80	0.00%	40.69	0.00%
(2,2),(3,3),(3,1),(6,0)	2.40	1.46%	5.81	0.00%	16.12	0.00%	38.87	0.00%
(2,3),(1,0),(1,0),(5,0)	0.41	0.00%	1.91	0.00%	11.42	0.00%	11.06	0.00%
(3,1),(1,0)(4,2),(0,1)	0.27	0.00%	1.02	0.00%	2.86	0.00%	9.15	0.00%
(5,0),(8,0),(6,0),(1,0)	0.54	0.00%	5.85	0.00%	15.01	0.00%	41.71	0.00%
(6,0),(1,0),(7,0),(1,0)	0.97	13.11%	28.52	0.27%	96.38	0.09%	423.81	0.17%
(7,0),(1,0),(1,0),(7,0)	1.37	3.23%	21.10	0.22%	75.55	0.03%	309.71	0.19%
(9,0),(1,0),(4,0),(1,0)	0.28	0.00%	3.58	0.00%	9.88	0.00%	28.28	0.00%
(1,0),(1,0),(8,0),(7,0)	0.49	0.00%	3.05	0.00%	9.20	0.00%	22.66	0.00%
(1,0),(10,0),(1,0),(2,0)	0.24	0.00%	1.37	0.00%	4.80	0.00%	18.14	0.00%
(1,0),(1,0),(1,0),(10,0)	0.76	17.22%	6.15	0.00%	11.41	0.00%	43.58	0.00%
(4,2),(0,1),(6,1),(1,0)	0.37	0.00%	2.95	0.00%	8.71	0.00%	20.28	0.00%
(6,3),(1,0),(3,1),(0,1)	0.56	0.00%	4.55	0.00%	9.33	0.00%	23.15	0.00%
(1,0),(1,0),(0,1),(7,3)	0.63	0.00%	3.76	0.00%	19.64	0.00%	34.79	0.00%

Figure 3.6: A set of instances on larger TCTs.

keeping activities in the third service interval, 1 vessel activity and 0 housekeeping activities in the fourth service interval. This instance is solved in 0.54 seconds and LP relaxation is integer.

- The second group of columns concerns a TCT equipped with 20 quay cranes and is denoted by $2*\lvert I\rvert$, $2w_j$, $2Z1$, as it has $2*60 = 120$ internal workers, $2*40 = 80$ external workers at most in each period and the number of activities is twice larger than the values reported in the first group of columns. For example, line (5,0), (8,0), (6,0), (1,0) in Z1 means, in this case, 10 vessel activities and 0 housekeeping activities in the first service interval, 16 vessel activities and 0 housekeeping activities in the second service interval, 12 vessel activities and 0 housekeeping activities in the third service interval, 2 vessel activities and 0 housekeeping activities in the fourth service interval. This instance is solved in 5.85 seconds and LP relaxation is integer.

- The third group of columns concerns a TCT equipped with 30 quay cranes and is denoted by $3|I|, 3w_j, 3Z1$, as it has $3 * 60 = 180$ internal workers, $3 * 40 = 120$ external workers at most in each period and the number of activities is three times larger than the values reported in the first group of columns. For example, line (5,0), (8,0), (6,0), (1,0) in Z1 means, in this case, 15 vessel activities and 0 housekeeping activities in the first service interval, 24 vessel activities and 0 housekeeping activities in the second service interval, 18 vessel activities and 0 housekeeping activities in the third service interval, 3 vessel activities and 0 housekeeping activities in the fourth service interval. This instance is solved in 15.01 seconds and LP relaxation is integer.

- The fourth group of columns concerns a TCT equipped with 40 quay cranes and is denoted by $4|I|, 4w_j, 4Z1$, as it has $4 * 60 = 240$ internal workers, $4 * 40 = 160$ external workers at most in each period and the number of activities is four times larger than the values reported in the first group of columns. For example, line (5,0), (8,0), (6,0), (1,0) in Z1 means, in this case, 20 vessel activities and 0 housekeeping activities in the first service interval, 32 vessel activities and 0 housekeeping activities in the second service interval, 24 vessel activities and 0 housekeeping activities in the third service interval, 4 vessel activities and 0 housekeeping activities in the fourth service interval.

Table 3.3.3 shows that all instances are optimally solved within an acceptable time interval for TCTs. In fact, the LP relaxation is often integer and if this is not the case, the insertion of cuts at the root node rapidly leads to the optimum. Hence, the size of the branch-and-bound tree is always zero.

Chapter 4

Long-term nurse scheduling

In this chapter we present the results proposed in [62] where, motivated by a real case study in Cagliari (Italy), we focus on the nurse scheduling problem in a department of a hospital. We deal with the long term scheduling problem of the shifts of a team of nurses. We propose a solution based on integer linear programming, which allows to compute a scheduling in a given time horizon, which is optimal with respect to certain criteria, while satisfying a series of constraints imposed by the contractual rules of nurses and that aims to guarantee comfortable working conditions to them.

To reduce the computational complexity of the approach, and to take into account possible sudden and unpredictable variations in the requirements of the hospital and in the availability of the personnel, we propose a solution based on a Decision Support System (DSS), which splits the scheduling in the long time horizon in several smaller time horizons, and continuously update a series of information relative to the hospital and the team of nurses. For sake of simplicity and inspired by the considered case study, we assume that the long term horizon corresponds to one year, while the sub-intervals are equal to one month.

Three main features of the proposed approach have to be highlighted since, even if fundamental in real situations, are often neglected in the literature (see e.g., [12, 58, 68, 70, 75, 79] mentioned in the following subsection).

- First, we assume that external personnel may be taken from other departments or other structures if the current team of nurses does not allow to satisfy certain hard constraints. However, this leads to high costs for the department and typically to a lower quality of service. Our approach allows to keep this explicitly into account and look for a solution that minimizes such an occurrence.

- Second, our approach allows to keep into account the work load at the end of a given period and automatically send such an information as an input to the scheduling in the following period. This problem is also addressed in [70] but from a different perspective as mentioned in the following subsection.

- Third, in several approaches in the literature differences among nurses are not taken into account. On the contrary, our model allows to distinguish among nurses in maternity, in part time, or nurses that benefit of special reductions of the work for several reasons (e.g., because they have a close relative who suffers a serious illness).

We finally remark that the proposed tool could also be successfully used as a support to find out redundancies or weakness in the staff.

A real case study is considered, namely the surgery department at the University of Cagliari (Italy). A series of real data provided by the person who currently manually assigns shifts, enables us to confirm its effectiveness. Not only the desired goals are more satisfactorily reached, but all the constraints are met, despite of the solutions computed manually.

The testing phase also shows the scalability of the method. In particular, it shows that scheduling (optimal in the short time horizon) may be obtained in reasonable time even in the case of hypothetical problems of dimensions much larger than the case study at hand.

The following subsection provides a survey of relevant contributions strictly related to the topic in this thesis. As it clearly appears, a huge amount of problem formulations, goals, and perspectives have been considered, so it is really hard to make a fair comparison among the different contributions. Typically, the most appropriate one highly depends on the specific requirements and operating conditions.

4.1 Literature review

An excellent survey of the approaches proposed to solve the problem of nurse rostering, updated at 2004, has been proposed by Burke *et al.* in [17]. Here methods that span from operations research techniques to artificial intelligence approaches are recalled and critically evaluated. Furthermore, a huge amount of problem formulations are mentioned that span from *staffing* (i.e., the computation of the number of personnel of the required skills in order to meet predicted requirements) to specific nurse rostering problems based on peculiar administrative modes of operation (centralized scheduling, unit scheduling, self-scheduling, etc.). Typically such formulations are inspired by real case studies with their own features/rules/constraints. In the following, without the pretension of being exhaustive, we mainly provide details on a series of contributions that are based on integer or linear/mixed integer programming, being them the mostly related to our contribution, at least in terms of modeling approach.

Several interesting contributions based on integer or linear/mixed integer programming are mentioned by Burke *et. al* in [17]. See e.g., [43, 57]. However, as Burke *et al.* pointed out, almost all of them focus on short-term scheduling or cyclic re-scheduling and do not keep into account exhaustively the constraints that currently influence a feasible solution in a real hospital.

Integer programming is also used in [64] to solve the cyclic preference scheduling problem for hourly workers. In particular, a branch-and-price algorithm is developed that makes use of several branching rules and an extremely effective rounding heuristic.

Trilling *et al.* [75] focus on the anaestehesiology nurse scheduling problem of a French public hostital. As well known, and as pointed out in the paper, anaestehesiology nurses constitute one of the most shared resourses. Therefore the goal in [75] is that of establishing how to assign nurses to the different departments.

Moz and Pato [58] solve a problem that is essentially a rescheduling problem. In more detail, they assume that shifts in a given period (a week) are already assigned to nurses and one of the nurses could no more respect his/her assignment, thus the whole scheduling should be recomputed. The approach they propose allows to minimize differences with respect to the original scheduling while guaranteeing the satisfaction of a series of constraints.

Berrada *et al.* [12] solve the problem of assigning shifts to nurses under the assumptions that each nurse always works during the same shift (i.e. there is no rotation). Hence, each shift corresponds to a separate problem.

Satheesh *et al.* [68] deal with a problem that is similar to ours but with much less details, which are however fundamental when applying the approach in real case applications. Furthermore, authors in [68] apply their method to a numerical example that is not taken from a real case study. On the contrary in our thesis, the validation is performed via a real case study, comparing the results obtained with the proposed approach with those actually used in the hospital.

Smet *et al.* [70] investigate the effects of the scheduling on short time horizons over long time horizons. In particular, the concepts of local and global consistency in constraint evaluation processes are introduced and a general methodology to address these challenges in integer programming approaches is proposed. As mentioned above, our approach also deals with this problem, even if in different terms and without resorting to constraints classifications, but guarantees the satisfaction of all the constraints also on the long time horizon.

Valouxis *et al.* [79] provide a way to address the computational complexity of the problem in large scale systems, proposing a two stage strategy to compute shifts to be assigned to nurses: the first phase decides the workload for each nurse and for each day, while the second phase assigns the specific daily shifts.

Special attention deserves the contribution by Maenhout and Vanhoucke [55]. Here the authors propose an integrated methodology for allocating a given workforce over *multiple* departments based on the hospitals nurse staffing policies, each wards shift scheduling policies and the nurses characteristics. The model decides at the staff planning level on the workforce size and on which nurses are assigned to each ward. This staffing plan is developed based on an initial baseline roster that is created at the shift scheduling level for a heterogeneous set of nurses and indicates, for each ward, which nurses will be working each shift. As a result, the baseline roster consists of a configuration of individual nurse schedules that is generated by incorporating multiple objectives, such as cost, schedule desirability and quality nursing care.

Furthermore, Carrasco [65], inspired by a real case study in Spain, focused on the problem of assigning guard shifts to the physicians in a department starting from the assumption that employees prefer that their guard duties are regularly distributed in time. The proposed solution efficiently combines random and greedy strategies with heuristics, allowing to keep into account a series of specific constraints.

We finally mention the recent contribution by Bagheri *et al.* [5] and some references therein. In particular, Bagheri *et al.* propose a stochastic optimization model which accounts for uncertainties in the demand and stay period of patients over time. Sample Average Approximation method is used to obtain an optimal schedule for minimizing the regular and overtime assignment costs. Results have been validated via numerical experiments on a real case study. In [5] Bagheri *et al.* also propose an excellent, up to date survey of the literature on the nurse scheduling problem. Here a series of models are classified based on the considered objective, the main properties, and the proposed approach. Here we limit to briefly recall the most recent contributions, and address to [5] for a more exhaustive discussion. Tapaloglu and Selim [74] use fuzzy mathematical programming and fuzzy goal programming to minimize deviations of nurse preferences and hospital regulations. Ohki *et al.* [61] propose a cooperative genetic algorithm to minimze a penality function for evaluating shift schedules. Zhang *et al.* [82] aim at maximizing the quality of objectives with respect to the importance of constraints using genetic algorithms and a variable neighborhood search. Finally, Fan *et al.* [38] maximize the nurse satisfactions and hospital regulations thanks to an approach based on binary integer linear programming.

4.2 Problem description

A Database contains three kinds of information: the specific requirements/rules of the department, the information on the available team of nurses, and the information on the current month. In more detail, the main information on the specific requirements/rules of the department are: the definition of the shifts (starting and end time of each shift), the maximum number of call days per month, the maximum number of night shifts per month, and the maximum (minimum) number of consecutive working days to be assigned to nurses, where all such numbers depend on the working position of nurses. The main information on the team of nurses are: the number of regular nurses, the number of maternity nurses, the scheduled free days (e.g. vacations or sick days) of each nurse, and the expected (average) monthly number of working hours. Finally, the most significant information on the current month are: the number of days in the current month, the weekday or public holidays, and so on.

All such information are used to formulate a linear integer programming model that characterizes the set of admissible solutions in the current month, illustrated in detail in the next section, which keeps into account all the constraints mentioned above. This clearly requires the formulation of a series of constraints that account a series of information coming from the scheduling of the previous month. The scheduling in each month is thus computed solving an optimization problem whose objective function takes into account the three main requirements previously mentioned. As shown in Fig. 4.1, the scheduling resulting from the optimization in one month, is not only used to define the scheduling in the long term horizon, but is also used to update a series of information in the database.

4.3 Mathematical model

In this section we propose a mathematical model to solve the problem of scheduling working hours of nurses in a given time horizon. Without loss of generality, we assume that the time horizon is equal to one month.

We first set input data, including some input data from the working plan of the previous month, then we define decision variables and introduce the objective function that we want to minimize. Finally, we define the constraints in order to fulfill the restrictions

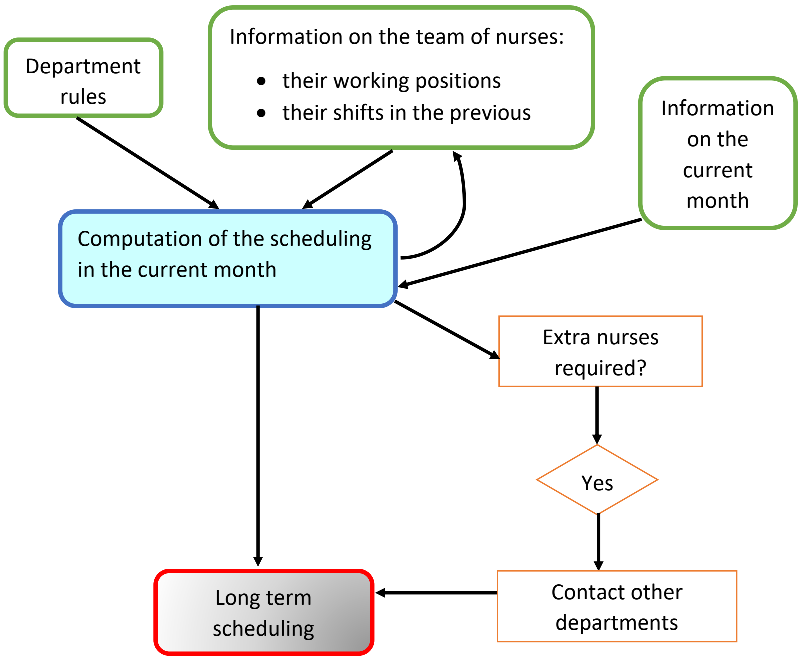


Figure 4.1: The proposed Decision Support System

required by the hospital manager.

Input data

- n is the number of days in the current month.
- $\mathcal{D} = \{1, \cdots, n\}$ is the set of days in the current month.
- $\mathcal{I}$ is the set of indices associated with nurses.
- $\mathcal{I}_{mat} \subset \mathcal{I}$ is the set of indices associated with maternity nurses.
- $A_{i,d} \in \{0, 1\}$ is equal to 1 if d is a scheduled free day for nurse i in the current month; 0 otherwise.
- d-off$_i$ is the number of free days of nurse $i \in \mathcal{I}$ during the current month.
- $\overline{hm}_i$ is the expected (average) monthly number of working hours of nurse $i \in \mathcal{I}$.
- $\mathcal{J} = \{1, 2, 3\}$ is the set of shifts.
- p_j is the amount of hours of the shift $j \in J$.

- R_{max} is the maximum number of call days per month assigned to each nurse.

- $Night_{max}$ is the maximum number of night shifts per month assigned to each nurse.

- D_{max} (D_{min}) is the maximum (minimum) number of consecutive working days assigned to each nurse.

- $N_{d,j}$ is the number of nurses required at shift $j \in \mathcal{J}$ on day $d \in \mathcal{D}$.

- $Z_i \in \{0,1, \cdots, D_{max} - 1\}$ is equal to the maximum number of admissible consecutive working days of nurse $i \in I$ at the beginning of the current month.

  (This parameter allows us to impose that the constraint on the maximum number of consecutive working days is satisfied also at the beginning of the month. Therefore, if a nurse i worked D_{max} consecutive days at the end of the previous month, Z_i should be equal to 0 because the nurse could not work more than D_{max} consecutive days. If the last but one day of the previous month was the last free day for nurse i at that month, then $Z_i = D_{max} - 1$.)

- $LJ_i \in \{1, 2, 3\}$. In particular, $LJ_i = 2$ (3) if nurse $i \in \mathcal{I}$ worked at the second (third) shift during the last day of the previous month; it is equal to 1 if the last day of the previous month was a call day for nurse $i \in \mathcal{I}$.

- $LBOJ_i \in \{0, 1\}$. is equal to 1 if nurse $i \in \mathcal{I}$ worked at the third shift during the last but one day of the previous month; 0 otherwise.

Decision variables and related costs

- $X_{i,d,j} \in \{0, 1\}$ is equal to 1 if nurse $i \in \mathcal{I}$ has to work at shift $j \in \mathcal{J}$ of day $d \in \mathcal{D}$, and the day d is not a call day for nurse i; 0 otherwise.

- $R_{i,d} \in \{0, 1\}$ is equal to 1 if $d \in \mathcal{D}$ is a call day for nurse $i \in \mathcal{I} \setminus \mathcal{I}_{mat}$. In the considered case study, a nurse in a call day should be available during the third shift of the day. Furthermore, a day could be a call day for a nurse i only if he/she was working during the first shift of that day; 0 otherwise. No call day could be assigned to maternity nurses.

- $Y_{i,d} \in \{0, 1\}$ is equal to 1 if $d \in \mathcal{D}$ is a working day for nurse $i \in \mathcal{I}$; 0 otherwise.

- $Uplus_{d,j} \in \mathbb{N}$ is a positive integer variable that counts the number of nurses in surplus during the shift $j \in \mathcal{J}$ of day $d \in \mathcal{D}$. In the considered case study no surplus may be used during shift $j = 3$. Therefore, $Uplus_{d,j}$ is only defined for $j \in \{1, 2\}$.

  We denote as $f^+_{d,j} > 0$ the cost associated with a nurse in surplus at shift $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

- $Uminus_{d,j} \in \mathbb{N}$ is a positive integer variable that counts the number of nurses in deficit during the shift $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

  We denote as $f^-_{d,j} > 0$ the cost associated with a nurse in deficit at shift $j \in \mathcal{J}$ of day $d \in \mathcal{D}$. This corresponds to the cost deriving from the recruitment of a nurse from other departments or structures.

- $W^+_{i,d} \in \{-1, 0, 1\}$ is equal to -1 if nurse $i \in \mathcal{I}$ works at day $d \in \mathcal{D}$ but does not work at day $d - 1 \in \mathcal{D}$; is equal to 0 if days $d, d - 1 \in \mathcal{D}$ are either working days or rest days for nurse $i \in \mathcal{I}$; it is equal to 1 if nurse $i \in \mathcal{I}$ works at day $d - 1 \in \mathcal{D}$ but does not work at day $d \in \mathcal{D}$.

- $HM^+_i \in \mathbb{N}$ is a positive integer variable counting the number of working hours of nurse $i \in \mathcal{I}$ during the current month, exceeding $\overline{hm_i}$.

  We denote as $fH^+_i > 0$ the cost associated with a working hour of nurse i exceeding $\overline{hm_i}$.

- $HM^-_i \in \mathbb{N}$ is a positive integer variable counting the number of working hours of nurse $i \in \mathcal{I}$ during the current month, in deficit with respect to $\overline{hm_i}$.

  We denote as $fH^-_i > 0$ the cost associated with a working hour of nurse i in deficit with respect to $\overline{hm_i}$.

- $S_{i,d} = Y_{i,d} + Y_{i,d+1}$ is a dummy variable that will be used to give priority to a schedule where free days assigned to nurse $i \in \mathcal{I}$ are not isolated, i.e., if $d \in \mathbb{D} \setminus \{n\}$ is a free day (and $d - 1$ was not a free day), $d + 1$ is desirable to be a free day.

  We denote as $fS^+_{i,d} > 0$ the cost associated with the case where d is an isolated free day for nurse $i \in \mathcal{I}$.

Objective Function

We have three main requirements.

1. Minimize the number of nurses in deficit.
2. Minimize the difference between the number of hours that each nurse currently works during a month and the number of hours he/she is expected to work.
3. Give priority to a schedule where free days are assigned consecutively (and not isolated).

This corresponds to minimize the following objective function:

$$\begin{aligned} f \quad &= \textstyle\sum_{d\in\mathcal{D}} \sum_{j\in\mathcal{J}} f^{-}_{d,j} \cdot Uminus_{d,j} + \\ &+ \textstyle\sum_{i\in\mathcal{I}} HM^{+}_{i} \cdot fH^{+}_{i} + \sum_{i\in\mathcal{I}} HM^{-}_{i} \cdot fH^{-}_{i} + \sum_{i\in\mathcal{I}} S_{i,d} \cdot fS^{+}_{i,d} \end{aligned}$$

Constraints

1. During shift $j \in \mathcal{J}$ of day $d \in \mathcal{D}$ the number of working nurses should be equal to $N_{d,j}$.

   1.1 Shift $j = 1$:

   $$\sum_{i\in\mathcal{I}} X_{i,d,1} + \sum_{i\in\mathcal{I}\setminus\mathcal{I}_{mat}} R_{i,d} - Uplus_{d,1} + Uminus_{d,1} = N_{d,1} \quad \forall d \in \mathcal{D}$$

   1.2 Shift $j = 2$:

   $$\sum_{i\in\mathcal{I}} X_{i,d,2} - Uplus_{d,2} + Uminus_{d,2} = N_{d,2} \quad \forall d \in \mathcal{D}$$

   1.3 Shift $j = 3$:

   $$\sum_{i\in\mathcal{I}} X_{i,d,3} + Uminus_{d,3} = N_{d,3} \quad \forall d \in \mathcal{D}$$

2. No more than one shift a day should be assigned to nurses. We need to distinguish non maternity (regular) and maternity nurses. Indeed, the former ones may work during any shift, while the latter ones may only work during shifts $j = 1$ or $j = 2$.

2.1 Regular nurses:

$$R_{i,d} + \sum_{j \in \mathcal{J}} X_{i,d,j} \leq 1 - A_{i,d} \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D}$$

2.2 Maternity nurses:

$$\sum_{j=1}^{2} X_{i,d,j} \leq 1 - A_{i,d} \quad \forall i \in \mathcal{I}_{mat} \quad \forall d \in \mathcal{D}$$

By definition of $A_{i,d}$, the above constraints guarantee that no shift is assigned to a nurse who is in vacation, or in rest, or is sick at that day.

3. The following constraints enable us to count the number of hours each nurse works during the current month, in addition, or in deficit, with respect to the expected monthly amount of hours.

3.1 Regular nurses:

$$\sum_{d \in \mathcal{D}} \sum_{j \in \mathcal{J}} p_j \cdot X_{i,d,j} + \sum_{d \in \mathcal{D}} p_1 \cdot R_{i,d} = \overline{hm}_i + HM_i^+ - HM_i^- \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat}$$

3.2 Maternity nurses:

$$\sum_{d=1}^{\mathcal{D}} \sum_{j=1}^{2} p_j \cdot X_{i,d,j} = \overline{hm}_i + HM_i^+ - HM_i^- \quad \forall i \in \mathcal{I}_{mat}$$

The correctness of the above constraints follows from the fact that the considered objective function aims at minimzing HM_i^+ and HM_i^+ for all $i \in \mathcal{I}$.

4. When a nurse works in a given shift, he/she cannot work in the subsequent two shifts. Also in this case, we have to distinguish between regular and maternity nurses.

4.1 Regular nurses:

4.1.a

$$X_{i,d,2} + X_{i,d+1,1} + R_{i,d+1} \leq 1 \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D} \setminus \{n\}$$

4.1.b

$$R_{i,d} + X_{i,d+1,1} + R_{i,d+1} \leq 1 \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D} \setminus \{n\}$$

4.1.c

$$R_{i,d} + X_{i,d+1,2} \leq 1 \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D} \setminus \{n\}$$

The first set of inequalities requires that if nurse i works at shift $j = 2$ in a given day, he/she cannot work at shift $j = 1$ the day after. We do not need to impose that he/she doesn't work at shift $j = 3$ of the same day because this is already guaranteed by constraint (2.1). Note that constraint (2.1) guarantees that a nurse i benefits of two following rest shifts if he/she works at shift $j = 1$.

Constraints (4.1.a) and (4.1.b) impose that if nurse i is in a call day in d (so he/she may potentially work during the night of day d), then he/she cannot work at shifts $j = 1$ the day after.

Similarly, constraint (4.1.c) imposes that if a nurse i is in a call day in d, then he/she cannot work at shift $j = 2$ the day after.

Note that we are not imposing that a nurse who works during the night in a given day cannot work during the first and the second shift of the day after, because this would be a redundant constraint. Indeed (see constraints (9)), a nurse who works during the night must be at rest the two days after.

4.2 Maternity nurses.

$$X_{i,d,2} + X_{i,d+1,1} \leq 1 \quad \forall i \in \mathcal{I}_{mat} \quad \forall d \in \mathcal{D} \setminus \{n\}$$

In this case we only have the above set of constraints because night shifts cannot be assigned to maternity nurses.

5. Each day, exactly one nurse should be in a call day. Maternity nurses cannot be in a call day.

$$\sum_{i \in \mathcal{I}} R_{i,d} = 1 \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D}$$

6. At most R_{max} call days per month can be assigned to regular nurses.

$$\sum_{d \in \mathcal{D}} R_{i,d} \leq R_{max} \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat}$$

7. At most $Night_{max}$ night shifts per month can be assigned to regular nurses.

$$\sum_{d \in \mathcal{D}} X_{i,d,3} \leq Night_{max} \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat}$$

8. The following equality constraints introduce $|\mathcal{I}| \times |\mathcal{D}|$ binary variables. The generic variable $Y_{i,d}$, associated with nurse i and day d, is equal to 1 when nurse i is working at day d; 0 otherwise. Obviously, such constraints need not be currently implemented when solving the optimization problem but are introduced here to more clearly explain the remaining constraints.

   8.1 Regular nurses:

$$Y_{i,d} = \sum_{j \in \mathcal{J}} X_{i,d,j} + R_{i,d} \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D}$$

   8.2 Maternity nurses:

$$Y_{i,d} = \sum_{j=1}^{2} X_{i,d,j} \quad \forall i \in \mathcal{I}_{mat} \quad \forall d \in \mathcal{D}$$

9. If a regular nurse i works in a night shift in a given day, then he/she must be at rest during the subsequent two days.

   9.1

$$2\ X_{i,d,3} + Y_{i,d+1} + Y_{i,d+2} \leq 2 \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad \forall d \in \mathcal{D} \setminus \{n-1, n\}$$

   9.2

$$X_{i,d,3} + Y_{i,(d+1)} \leq 1 \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad d = n-1$$

10. Each nurse could not work more than D_{max} consecutive days.

$$Y_{i,d} + \sum_{t=1}^{D_{max}} Y_{i,d+t} \leq \ D_{max} \quad \forall i \in \mathcal{I} \quad \forall d \in \{1, \cdots, n - D_{max} + 1\}$$

11. Each nurse could not work less than D_{min} consecutive days.

   11.1

$$W^{+}_{i,d} = Y_{i,d} - Y_{i,d-1} \quad \forall i \in \mathcal{I} \quad d \in \{2, \cdots, n\}$$

   11.2 When a nurse moves from a rest day to a working day, the following $D_{min} - 1$ days should be working days as well. If the remaining time horizon is smaller than $D_{min}-1$ days, then the constraint applies to the whole remaining horizon.

11.2.a

$$\sum_{t=2}^{D_{min}} Y_{i,d+t} \geq (D_{min}-1) \cdot W^{+}_{i,d} \quad \forall i \in \mathcal{I} \quad d \in \{1, \cdots, n-D_{min}+1\}$$

11.2.b

$$\sum_{t=d+1}^{n} Y_{i,d+t} \geq (n-d) \cdot W^{+}_{i,d} \quad \forall i \in \mathcal{I} \quad d \in \{n-D_{min}+2, \cdots, n-1\}$$

12. An additional constraint is necessary because constraint (9) imposes to a nurse i to have two rest days, after the night shift, therefore it is clear that to respect the minimum consecutive D_{min} days, a nurse can work the night shift if and only if he/she worked the D_{min}-1 previous days.

$$\sum_{t=1}^{D_{min}-1} Y_{i,d-t} \geq (D_{min}-1) \cdot X_{i,d,3} \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad d \in \{n-D_{min}+2, \cdots, n\}$$

13. We need to add the following constraint because constraints (4.1) imposes that a nurse i in a call day d either works in the night shift of d or is at rest in $d+1$. Therefore it is clear that to respect the minimum consecutive D_{min} days, a nurse can work in a call day if and only if he/she worked the D_{min}-2 previous days.

$$D_{min} \cdot R_{i,d} \leq \sum_{t=1}^{D_{min}-2} Y_{i,d-t} \quad \forall i \in \mathcal{I} \setminus \mathcal{I}_{mat} \quad d \in \{n-D_{min}+2, \cdots, n\}$$

Border constraints

We finally introduce a series of constraints, (called *"border constraints"*) that take into account the programming in the previous time horizon. They allow to automatically run the optimization problem *"month after month"*.

14. If the last day of the previous month was a call day for a regular nurse i, then the first and the second shift of the first day of the current month could not be assigned to him/her.

$$X_{i,1,1} + X_{i,1,2} = 0 \quad \forall i \in \mathcal{I} \ : \ LJ_i = 1$$

15. If a regular nurse worked during the second shift of the last day of the previous month, in the first day of the current month he/she should not work during the first shift and should not be in call day.

$$X_{i,1,1} + R_{i,1} = 0 \quad \forall i \in \mathcal{I} \ : \ LJ_i = 2$$

16. If a regular nurse worked during the third shift of the last day of the previous month, in the first and in the second day of the current month he/she should be at rest.

$$Y_{i,1} + Y_{i,2} = 0 \quad \forall i \in \mathcal{I} \ : \ LJ_i = 3$$

17. The above two constraints reduce to the following one in the case of a maternity nurse since he/she cannot be in a call day and cannot work during the third shift.

$$X_{i,1,1} = 0 \quad \forall i \in \mathcal{I}_{mat} \ : \ LJ_i = 2$$

18. If a regular nurse worked in the third shift of the last but one day of the previous month, then the first day of the current month he/she must be at rest.

$$Y_{i,1} = 0 \quad \forall i \in \mathcal{I} \ : \ NLJ_i = 0$$

19. Each nurse should work at most D_{max} consecutive days. To ensure that this constraint is also guaranteed when switching from one month to the next one, variable Z_i has been introduced. It is an input variable to the scheduling of the current month and keeps into account the number of consecutive days that the generic nurse i could work at the beginning of the current month. The following inequality imposes the satisfaction of such a constraint:

$$\sum_{d=1}^{Z_i+1} Y_{i,d} \le Z_i \quad \forall i \in \mathcal{I}$$

20. Working days of nurses could not be isolated. In particular, D_{min} is the minimum number of consecutive days that he/she should work. The following constraint guarantees that this holds also when switching from one month to the next one.

$$\sum_{d=1}^{[D_{max}-(D_{min}-1)]-D_{min}} Y_{i,d} = [D_{max} - (D_{min} - 1)] - D_{min}$$
$$\forall i \in \mathcal{I} \ : \ Z_i \ge D_{max} - (D_{min} - 1)$$

Concluding, the number of decision variables is $10 \cdot |\mathcal{I}| \cdot |\mathcal{D}|$, while the number of constraints is $9 \cdot |\mathcal{I}| \cdot |\mathcal{D}| + 3 \cdot |\mathcal{D}| + |\mathcal{I} \setminus \mathcal{I}_{mat}| \cdot |\mathcal{D}|$, where $| \cdot |$ denotes cardinality. Therefore, both numbers are $O(|\mathcal{I}| \cdot |\mathcal{D}|)$.

4.4 The case study

The proposed approach has been tested on a real working environment, specifically, the surgery department of the University Hospital in Cagliari (Italy).

Working days are partitioned in three shifts: 7:00 am - 2:00 pm ($j = 1$), 2:00 pm - 10:00 pm ($j = 2$), 10:00 pm - 7:00 am ($j = 3$). Therefore the amount of hours of shifts $j = 1, 2, 3$, is equal to $p_1 = 7$, $p_2 = 8$, $p_3 = 9$, respectively.

Furthermore, a regular nurse ($i \in \mathcal{I} \setminus \mathcal{I}_{mat}$) is expected to work 36 hours per week, while a maternity nurse ($i \in \mathcal{I}_{mat}$) is expected to work 30 hours per week. The length of the planning horizon is one month, therefore it is $\overline{hm}_i = (n - \text{d-off}_i) \cdot 36/7$ for $i \in \mathcal{I} \setminus \mathcal{I}_{mat}$, and $\overline{hm}_i = (n - \text{d-off}_i) \cdot 30/7$ for $i \in \mathcal{I}_{mat}$. Moreover, regular nurses may work during any shift, while maternity nurses may only work during shifts 1 and 2.

For each regular nurse, the maximum number of call days per month is 6, as well as the maximum number of night shifts per month. Furthermore, the maximum number of consecutive working days is 6, while the minimum number of consecutive working days is 3.

Finally, shifts should be assigned to nurses in order to guarantee the following requirements:

- at least 4 nurses should be simultaneously present during shifts $j = 1$ and $j = 2$;
- exactly 3 nurses should be simultaneously present during shift $j = 3$;
- if a day is an urgent day, at least 5 nurses should be simultaneously present during shift $j = 1$.

4.5 Testing

The testing phase has been performed with the aim of comparing the solutions obtained using the proposed optimization approach with those currently used in the considered department, which have been computed manually by a person responsible to do that. Due to the difficulty of fulfilling all the requirements and constraints when operating manually, the person devoted to this first takes into account "hard" constraints, then she/he tries to rearrange the scheduling in order to also meet "soft" constraints (if possible). Referring to the enumeration used in Section 3, Constraints (6), (11), and (12) may be considered as soft constraints when computing the solution manually, while all the others should be always considered as hard constraints. We also note that the manual approach could be very time consuming, usually many hours.

Results are reported in Tables 4.1 to 4.4. Each table contains information relative to two months. Let us explain the notation used in these tables by looking at Table 4.1 and focusing on February. Analogous considerations repeat for March in Table 4.1, and for the other months in the remaining tables.

The first column identifies the 21 nurses working in the considered department.

Column 2 summarizes the values of HM_i^+ resulting from the scheduling currently used in the hospital (H). Analogously, column 4 summarizes the values of HM_i^+ resulting from the scheduling obtained using the proposed model (M). HM_i^+ is equal to the number of working hours of nurse i during the current month, exceeding the average monthly number of working hours, which is equal to 36 (for regular nurses) or 30 (for maternity nurses) in the considered case study.

Column 3 summarizes the values of HM_i^- resulting from the scheduling currently used in the hospital (H). Analogously, column 5 summarizes the values of HM_i^- resulting from the scheduling obtained using the proposed model (M). HM_i^- is equal to the number of working hours of nurse i during the current month, in deficit with respect to the average monthly number of working hours.

Columns 6, 7 and 8 contain information on the soft constraints not satisfied by the scheduling used in the hospital (H): A_i (B_i, C_i, respectively), is equal to the number of times constraint (11) ((12), (6), respectively) is not met during the current month by nurse i. Zeros are not reported in these columns to make tables more readable. On the contrary, all the constraints are met when using the proposed approach. Therefore, in such a case for all i and all months, it is $A_i = B_i = C_i = 0$.

Looking at Tables 1 to 4, we observe that (see last row), during almost all months $\sum_{i=1}^{21} HM_i^+$ and $\sum_{i=1}^{21} HM_i^-$ are smaller when computed using the proposed approach. The only exception is given by July. However, as summarized in Table 4.3, this is obtained with the violation of a huge number of constraints when operated manually as it actually occurs in the hospital (H).

A clear comparison between the scheduling resulting from the proposed approach (M) and the scheduling used in the hospital (H) is also reported in Fig. 4.2 and Fig. 4.3. These figures show in the two cases (M and H) the values of function f_i, for $i = 1, \ldots, 21$, defined as follows:

Table 4.1: Comparison of the scheduling currently used in the hospital (H) and the scheduling obtained using the proposed model (M): February and March

Nurse$_i$	FEBRUARY							MARCH						
	H		M		constraints not satisfied by hospital			H		M		constraints not satisfied by hospital		
	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i
1	0	13	0	0				6	0	0	0		1	
2	0	1	0	0				25	0	0	0			1
3	9	0	0	0			1	10	0	0	0			
4	0	3	0	0				10	0	0	0		1	
5	0	5	0	0				16	0	0	1			
6	0	3	0	0	3			13	0	1	0			
7	0	2	0	0				8	0	0	0			
8	6	0	0	0				8	0	0	0			
9	7	0	0	0				15	0	0	5			
10	1	0	0	0				14	0	0	0			
11	1	0	0	0				13	0	0	0	2		
12	0	1	0	0				9	0	0	0	2		1
13	0	6	0	0	1	1	1	5	0	0	0	1		
14	0	8	0	0				0	0	0	0		1	1
15	0	3	0	0		1		12	0	0	0	2		
16	1	0	0	0	1	1		9	0	0	2			
17	0	4	0	0				0	0	0	0		1	
18	0	7	0	0				9	0	0	14	4		
19	0	8	0	0				14	0	0	0			
20	0	4	0	0				28	0	0	0	2		
21	0	5	0	0				9	0	0	0			
	25	73	0	0	6	3	2	233	0	1	22	13	4	3
	98		0		11			233		23		20		

Table 4.2: Comparison of the scheduling currently used in the hospital (H) and the scheduling obtained using the proposed model (M): April and May

Nurse_i	APRIL							MAY						
	H		M		constraints not satisfied by hospital			H		M		constraints not satisfied by hospital		
	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i
1	15	0	3	0				0	15	1	0		2	
2	16	0	4	0				0	10	0	0		2	
3	17	0	1	0		1	1	15	0	0	0	2		
4	0	10	0	0	1	2		2	0	0	0		1	
5	17	0	2	0				0	4	0	0		2	
6	19	0	3	0	2			19	0	0	0	2		
7	0	1	3	0	2	1		0	2	0	0		1	
8	16	0	0	0				0	15	0	0		2	
9	14	0	0	0	1			0	13	0	0	1	1	1
10	13	0	4	0				0	0	0	0			
11	14	0	3	0	2			8	0	8	0	1		
12	23	0	0	0	2			0	5	1	0	1		
13	15	0	0	0				3	0	0	0			
14	1	0	0	0		2		2	0	1	0			
15	19	0	3	0				1	0	0	0	1	1	
16	12	0	11	0			1	2	0	0	0	1	1	
17	0	0	0	0	2			6	0	0	0		1	
18	18	0	0	0				4	0	0	2			
19	9	0	0	0	1			2	0	0	0			
20	22	0	0	0				9	0	0	0	2		
21	8	0	2	0				23	0	1	0			
	267	11	39	0	13	6	2	97	64	12	2	11	14	1
	278		39		21			161		14		26		

Table 4.3: Comparison of the scheduling currently used in the hospital (H) and the scheduling obtained using the proposed model (M): June and July

Nurse_i	JUNE							JULY						
	H		M		constraints not satisfied by hospital			H		M		constraints not satisfied by hospital		
	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i
1	18	0	15	0		1		2	0	3	0		2	
2	7	0	0	5	2		1	5	0	19	0		1	
3	23	0	19	0	1		1	16	0	16	0		2	
4	0	0	0	6		3		16	0	7	0		1	2
5	0	2	0	1	2	1		13	0	20	0			
6	9	0	6	0	3			23	0	20	0	2		
7	19	0	0	1		1		5	0	20	0	1	1	1
8	0	2	0	1		1		3	0	16	0	1		
9	0	1	3	0	2		2	11	0	18	0	1		2
10	0	0	0	0				14	0	20	0	1	1	1
11	1	0	0	2		1		0	0	10	0		1	1
12	11	0	15	0	1			9	0	19	0			
13	11	0	0	0		1		38	0	16	0		1	
14	12	0	0	4	1		2	13	0	20	0	4	1	1
15	12	0	17	0	2		1	6	0	20	0	2	1	
16	0	0	0	0				8	0	14	0			
17	19	0	0	0				0	0	6	0		1	
18	0	0	0	0				17	0	0	0			
19	5	0	20	0	1			4	0	20	0	2		
20	9	0	20	0	1		1	0	4	20	0	1		
21	0	11	20	0	1			10	0	20	0			
	153	16	135	20	17	9	8	211	4	324	0	15	13	8
	169		155		34			215		324		36		

Table 4.4: Comparison of the scheduling currently used in the hospital (H) and the scheduling obtained using the proposed model (M): August and September

Nurse$_i$	AUGUST							SEPTEMBER						
	H		M		constraints not satisfied by hospital			H		M		constraints not satisfied by hospital		
	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i	HM_i^+	HM_i^-	HM_i^+	HM_i^-	A_i	B_i	C_i
1	20	0	0	12				9	0	1	0	3	1	
2	22	0	0	1			1	9	0	3	0			
3	22	0	0	6			1	12	0	0	1			1
4	23	0	0	5			1	0	3	0	0	1		1
5	22	0	3	0				13	0	1	0	1		
6	5	0	1	0	2		2	28	0	0	0			
7	11	0	0	2		1		0	4	0	0	1	1	
8	19	0	0	2				0	1	0	0	1	1	1
9	28	0	11	0				13	0	2	0	1		1
10	30	0	2	0				11	0	1	0	1		
11	7	0	0	12			1	6	0	0	0		1	
12	22	0	0	6			1	5	0	0	0		1	1
13	6	0	0	5	2			5	0	0	0			
14	13	0	2	0			1	3	0	0	1			
15	11	0	0	3	1	1		18	0	1	0			1
16	24	0	0	6				6	0	1	0			
17	22	0	0	2				7	0	0	0			1
18	6	0	0	0		1		8	0	0	0	1		
19	18	0	17	0	2			18	0	0	0			
20	7	0	3	0	1			24	0	13	0			
21	15	0	5	0				0	6	0	0			1
	348	0	44	62	8	3	8	194	14	23	2	10	5	8
	348		106		19			208		25		23		

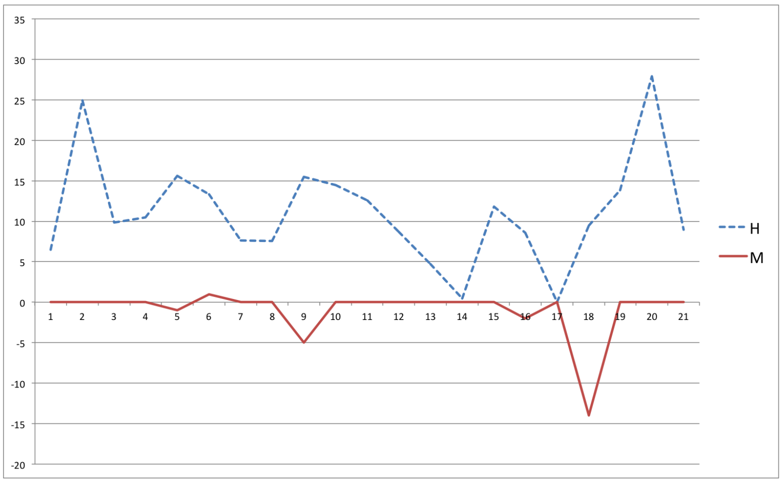


Figure 4.2: Summary of f_i defined as in eq. (4.1) for $i = 1, \cdots, 21$ in March, using the scheduling provided by the hospital (H) and the scheduling obtained using the proposed model (M).

$$f_i = \begin{cases} HM_i^+ & \text{if} \quad HM_i^+ > 0 \\ -HM_i^- & \text{if} \quad HM_i^- > 0 \end{cases} \tag{4.1}$$

In more detail, Fig. 4.2 refers to March, while Fig. 4.3 refers to July. In simple words, based on the above definition, if f_i is positive, it means that nurse i worked more than expected in the considered month. Alternatively, if f_i is negative, it means that he/she worked less that expected, and the absolute value is equal to the deficit of hours with respect to the expected value. Obviously the target value is 0 for all i.

Finally, to validate the applicability of the proposed approach in larger dimension problems, different scenarious have been considered, and the computational times (in seconds) as well as the percentage distance (GAP $\%$) with respect to the optimal solution, are computed solving instances by Cplex 12.6.2 on a mac air with 1.3 GHz Intel Core I5, 4 GB of RAM and 4 cores. Results are summarized in Table 4.5. The first series of simulations $(\times 1)$ refer to the current situation where the number of nurses is equal to 21. As it can be seen, all instances are solved in less than 1.25 hours.

The other series of simulations $(\times 2, \times 3, \times 4)$ refer to hypothetical situations where the current number of nurses is multiplied by 2, 3, and 4, respectively. For sake of brevity, computational times and gaps in these cases are reported for only two months, namely March (3) and July (7). The most critical instance is relative to 168 nurses and month 3, and is solved in slightly more than 2 hours.

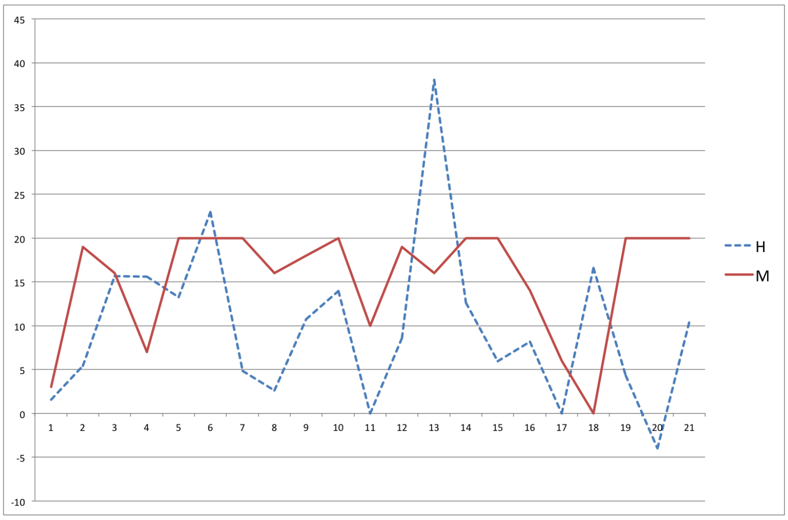


Figure 4.3: Summary of f_i defined as in eq. (4.1) for $i = 1, \cdots, 21$ in July, using the scheduling provided by the hospital (H) and the scheduling obtained using the proposed model (M).

	Number of Nurses	Month	Time (s)	GAP %
X1	21	2	770	0,00
	21	3	369	0,01
	21	4	345	0,00
	21	5	420	0,01
	21	6	9	0,00
	21	7	13	0,00
	21	8	64	0,01
	21	9	4394	0,04
X2	42	3	540	0,00
	42	7	170	0,20
X4	84	3	4225	0,00
	84	7	735	0,23
X8	168	3	7321	0,00
	168	7	3287	0,00

Table 4.5: Summary of the results of a series of experiments carried out to test the proposed approach in larger dimension problems.

Chapter 5

Short-term manpower scheduling in retailer stores

This chapter focuses on the problem of short-term manpower scheduling in retailer stores. A team of empolyees, each one with its on skills, should be assigned to different shifts taking into account the tasks that should be solved, the skills of the employees, and their benefit in terms of working conditions.

A real case study in Cagliari is used to test the approach, still based on linear integer programming. A web application has also been developed, with a user friendly interface, which makes the tool ready to be used by the considered retailer store.

As already discussed in the previous chapters, a huge literature exists devoted to the assignment of shifts to working staff. However, while the literature focused on certain specific working areas, in particular hospitals and health care in general, is extremely rich, despite of its significance, schedule optimization for retail stores has received relatively weak attention. Furthermore, most of the contributions are focused on the problem of profit maximization. We address to Chapados *et al.* [23] for a good, detailed, and quite recent survey of the literature on this topic.

An important remark that still holds in this area, as in the areas investigated in the previous chapters, is that it is not easy to provide a well structured literature review where contributions are compared in terms of generality and effectiveness, e.g., in terms of optimality of the solution or computational complexity. Indeed, very different problem statements are considered having different goals and perspectives, based on ad hoc assumptions coming from specific applications and selled products, considering different time horizons (long term and short term scheduling), and using completely different approaches for the computation of an optimal or suboptimal solution.

This chapter considers a problem formulation that is quite different from the rest of the literarure. Indeed, it assumes that the team of employees is given and each employee has his/her own skills and personal requirements. He/she should be asked to work as much as possible to solve the tasks that are more appropriate to his/her skills, and in any case he/she should never be devoted to tasks requiring a skill that is more payed that that considered when computing his/her salary. Furthermore, a series of requirements in terms of breaks, number of working Sunday per month, and so on, should be satisfied in order to guarantee a good quality of working conditions, which obviously has an impact on his/her effectiveness at work and in the quality of the service to costumers [81].

The rest of the chapter is structured as follows. The problem statement and the proposed solution are described in detail in Section 5.1. The integer linear programming model is illustrated in detail in Section 5.2. Finally, Section 5.3 presents the considered case study, the web application that has been developed, and the results of some scheduling.

5.1 Problem Statement and proposed solution

In this chapter we deal with the problem of automatically scheduling the shifts of employees in a big store, taking into account a series of activities that should be performed, and the individual skills of employees.

Each activity consists of a certain number of tasks, which could be performed by employees having certain skills. Skills are ordered according to a top/down list that is used to compute salaries. Each employee may be devoted to a certain number of tasks depending on his own skills. However, if his salary has been computed based on one skill, he/she cannot be devoted to tasks requiring skills in a higher position of the above mentioned list. On the contrary, he can be devoted to tasks requiring skills in a lower position in the list, but this produces a lost for the company. Indeed, the company is paying a work more than necessary, as if it requires a skill in a higher position in the list.

We focus on short-term scheduling and, as it typically occurs in this framework, the time horizon is taken equal to one week. However, the proposed solution is general and can be used even if the time horizon is multiple of one week.

We assume that a series of information on the current week are taken into account, e.g., the number of working days and the time period of the year, which clearly affects the requirements in terms of workload. In more detail, we assume that the *workforce demand* is given. This consists in a table that specifies the workforce required during each day of the week, as a function of the hour of the day, devoted to each activity. Such a table is computed taking into account historical data on analogous weeks in previous years, and eventually using ad hoc forecasts approaches. Several methods can be used in this respect. For an extensive discussion on this we address the reader to [45, 48] and the references therein.

We also assume that a series of fundamental information on the team of employees are available, in particular, we know:

- which employees are in a working week or in vacation,
- which employees requested some day-off during the week or are in a sick day,
- which are the skills of each employee in the team,
- which is the last day in the previous week each employee has worked,
- the number of Sunday in which each employee has worked in the previous weeks, and so on.

As typically occurs in big stores, each employee has his/her own skills (e.g., cashier, warehouse worker, accountant, shop assistant, and so on) and can solve a certain number of activities depending on them. However, his/her salary is established based on the most expensive and qualified skill. So, whenever, a scheduling is computed, which assigns an employee to an activity that requires a skill less qualified than his/her top skill, the store is not taking the maximum benefit from its own human resources. To avoid this, a penalty is associated with such an occurrence and this is taken into account in the performance index to be optimized.

Furthermore, each employee has his/her own contract that could be a full time or a part time.

Moreover, based on the specific store a series of hard constraints should be imposed, namely constraints that identify feasible solutions since they cannot be violated. In particular:

- the opening and the closing time of the store are given;
- shifts could either be totally free or they should be multiple of one hour or multiple of half an hour;
- there could be an upper (lower) bound on the maximum (minimum) number of consecutive and total hours in a working day;
- there could be an upper bound on the maximum number of consecutive working days;

- no more that two Sunday per month could be working days;
- on Sunday and other holidays the over-manning (namely, a number of employees larger than necessary) is not allowed because of economical reasons.

Each store also has some *soft* business constraints that could be taken into account in the performance index to be minimized:

1. avoid breaks between shifts assigned to the same employee in the same working day,
2. avoid under-manning and over-manning. The number of employees should be as close as possible to a given recommended value defined by the workforce demand. Such a value varies with the day, the time, and the activity;
3. the maximum number of working hours per day should not exceed a certain value, unless strictly necessary to find out a feasible solution;
4. minimize the number of employees that are assigned to activities not requiring their top skill.

The main contribution of this thesis consists in the formulation a of mathematical model that allows to compute an optimal solution using integer linear programming. Details on the constraints and on the objective function are provided in the following section.

Fig. 5.1 presents via a flow chart, how the proposed mathematical model interacts with a database which contains information on the current week, on the team of employees, on the store rules, and on the workload demand.

Note that, to make the approach user friendly, an interface is fundamental, as shown in Fig. 5.1.

5.2 The integer linear programming model

In this section we illustrate the integer linear programming model that allows us to solve the scheduling problem introduced in the previous section.

We first introduce input data, including a series of data from the working plan of the previous weeks; we define all the decision variables and some cost functions; then, we introduce the objective function that we want to minimize; finally, we define all the constraints that allow us to take into account the restrictions required by contractual rules.

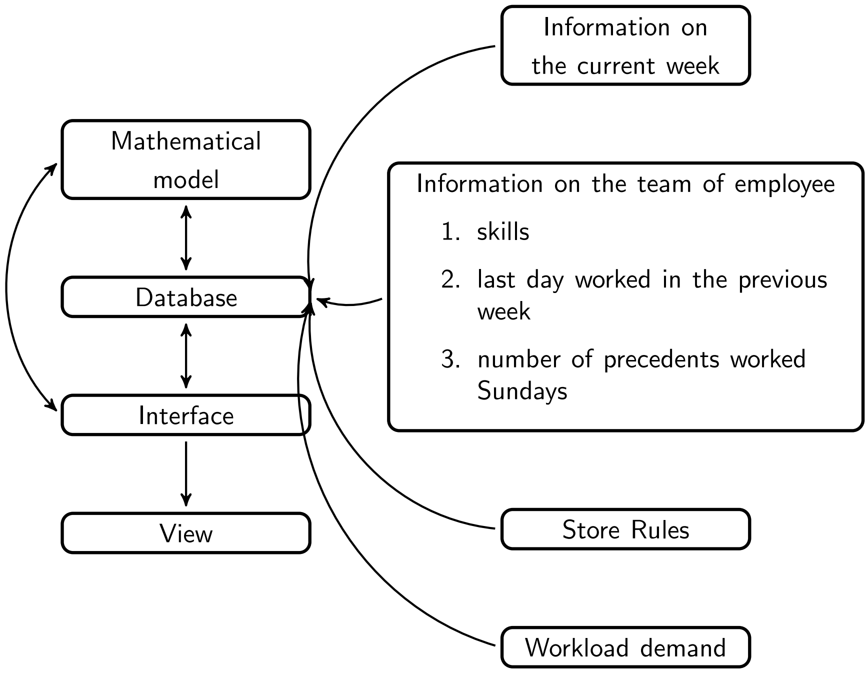


Figure 5.1: Flow chart of the proposed approach

Input data

- $\mathcal{I}$: set of indices associated with employees.
- s: number of weeks to be scheduled.
- $\mathcal{S} = \{0, 1, \cdots, s-1\}$: indices associated with the weeks in the considered time horizon.
- $ns \in \{4, 5\}$: number of weeks in a month.
- $d = s * 7$: number of days in the considered time horizon.
- $\mathcal{D} = \{1, \cdots, d\}$: indices associated with the days in the considered time horizon.
- HS_i: number of weekly hours worked by the ith employee.
- HN: number of working hours in each day (eventually some reduction could occur in some days, e.g., Sunday, but for simplicity they are taken into account via appropriate constraints).
- $HDMAX$: maximum number of working hours per day assigned to each employee (differences among employees may be easily accounted adding appropriate constraints).

- $DCmax$ ($DCmin$): number of maximum (minimum) consecutive working hours per day.

- $DAYCmax$: number of maximum consecutive working days.

- $HBREAKmin$: minimum number of consecutive hours of break between two shifts in the same day.

- $\mathcal{J} \in \{1, \dots, HN\}$: set of indices associated with the time periods in each day.

- $\mathcal{K}$: set of tasks to be executed;

- $\mathcal{Z}$: set of activities, where each activity is defined as a set of tasks.

- $\mathcal{Z}_j \subseteq \mathcal{Z}$: set of activities to be performed in period $j \in \mathcal{J}$.

- $\mathcal{K}_z$: set of tasks to be executed to perform activity $z \in \mathcal{Z}$.

- $N_{d,j,k,z}$: number of employees required to execute task $k \in \mathcal{K}_z$ inside activity $z \in \mathcal{Z}$ in period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

- $C_{i,d,j,k,z}$: cost of employee $i \in \mathcal{I}$ assigned to task $k \in \mathcal{K}_z$ inside activity $z \in \mathcal{Z}$ in period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

- $Hday_{i,d} \in \{0, 1\}$: 1 if employee $i \in \mathcal{I}$ is in vacation on day $d \in \mathcal{D}$ (or he/she got some special permission that allows him/her not to work, e.g., he/she is sick); 0 otherwise.

Decision variables and related costs

- $X_{i,d,j,k,z} \in \{0, 1\}$: 1 if employee $i \in \mathcal{I}$ has to work in period $j \in \mathcal{J}$ at activity $z \in \mathcal{Z}_j$ to perform task $k \in \mathcal{K}_z$ on day $d \in \mathcal{D}$; 0 otherwise.

- $LX_{i,d,j} \in \{0, 1\}$: 1 if employee $i \in \mathcal{I}$ has to work in period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$; 0 otherwise.

- $Uplus_{d,j,k,z} \in \mathbb{N}$: counts the number of employees in surplus (with respect to $N_{d,j,k,z}$) devoted at activity $z \in \mathcal{Z}_j$ to perform task $k \in \mathcal{K}_z$ in period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

  We denote as $f^+_{d,j} > 0$ the cost associated with the surplus of one employee in the period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

- $Uminus_{d,j,k,z} \in \mathbb{N}$: counts the number of employees in deficit (with respect to $N_{d,j,k,z}$) devoted at activity $z \in \mathcal{Z}_j$ to perform task $k \in \mathcal{K}_z$ in period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

  We denote as $f^-_{d,j} > 0$ the cost associated with the deficit of one employee in the period $j \in \mathcal{J}$ of day $d \in \mathcal{D}$.

  Note that, since typically stores manager want to penalize deficit more than surplus, it is $f^-_{d,j} > f^+_{d,j}$.

- $Y_{i,d} \in \{0, 1\}$: 1 if employee $i \in \mathcal{I}$ works in day $d \in \mathcal{D}$; 0 otherwise.

- $W^+_{i,d,j} \in \{0, 1\}$: 1 if employee $i \in \mathcal{I}$ works in day $d \in \mathcal{D}$ and starts in period $j \in \mathcal{J}$; 0 otherwise.

- $W^-_{i,d,j} \in \{-1, 0\}$: -1 if employee $i \in \mathcal{I}$ works in day $d \in \mathcal{D}$ and finishes in period $j \in \mathcal{J}$; 0 otherwise.

  The following Fig. 5.2 explains the meaning of the above variables. Assume that the working day starts at time 8.30 and ends at 20.00. Furthermore, assume that shifts should be multiple of half an hour. Finally, let the first and the second shift of a certain employee i in day d be as shown in Fig. 5.2. Then, it is $W^+_{i,d,4} = W^+_{i,d,14} = 1$ and $W^-_{i,d,9} = W^-_{i,d,18} = -1$.

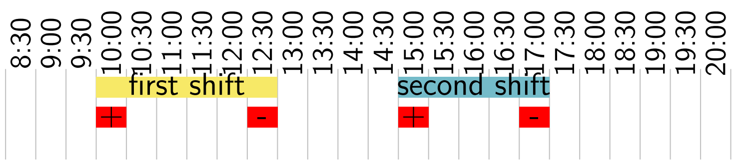


Figure 5.2: $W^+_{i,d,j}$ ($W^-_{i,d,j}$) is equal to 1 (-1) for the value of period j corresponding to the beginning (end) of the shift.

Objective Function

As explained in Section 4 we have four soft constraints, 1 to 4, which can be easily imposed via the minimization of the following performance index consisting in the summation of four terms, each one associated with a different soft constraint:

$$
\begin{aligned}
f \quad &= \textstyle\sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}} W^+_{i,d,j} \\
&+ \textstyle\sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}}\sum_{z\in\mathcal{Z}_j}\sum_{k\in\mathcal{K}_z} Uminus_{d,j,k,z}\cdot f^-_{d,j,k,z} + Uplus_{d,j,k,z}\cdot f^+_{d,j,k,z} \\
&+ \textstyle\sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}} Hplus_{i,d}\cdot fH^+_{i,d} \\
&+ \textstyle\sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}}\sum_{z\in\mathcal{Z}_j}\sum_{k\in\mathcal{K}_z} X_{i,d,j,k,z} * C_{i,d,j,k,z}
\end{aligned}
$$

Constraints

In the following all the hard constraints mentioned in Section 4 are formalized in terms of linear equalities or inequalities.

1. During activity $z\in\mathcal{Z}_j$ in period $j\in\mathcal{J}$ of day $d\in\mathcal{D}$, the number of working employees required to perform task $k\in\mathcal{K}_z$ should be possibly equal to $N_{d,j,k,z}$. If such is not feasible, under-manning or over-manning occurs:

$$
\begin{gathered}
\textstyle\sum_{i\in\mathcal{I}} X_{i,d,j,k,z} - Uplus_{d,j,k,z} + Uminus_{d,j,k,z} = N_{d,j,k,z}, \\
\forall d\in\mathcal{D},\ \forall j\in\mathcal{J},\ \forall z\in\mathcal{Z}_j,\ \forall k\in\mathcal{K}_z.
\end{gathered}
$$

2. In period $j\in\mathcal{J}$ of day $d\in\mathcal{D}$, employee $i\in\mathcal{I}$ could be devoted to only one task $k\in\mathcal{K}_z$ in one activity $z\in\mathcal{Z}_j$:

$$
\sum_{k\in\mathcal{K}_z}\sum_{z\in\mathcal{Z}_j} X_{i,d,j,k,z} \le LX_{i,d,j}, \quad \forall i\in\mathcal{I},\ \forall d\in\mathcal{D},\ \forall j\in\mathcal{J}.
$$

3. In day $d\in\mathcal{D}$ employee $i\in\mathcal{I}$ should not exceed a $HDMAX$ value, unless strictly necessary to find out a feasible solution:

$$
\sum_{j\in\mathcal{J}} LX_{i,d,j} \le HDMAX + Hplus_{i,d}, \quad \forall i\in\mathcal{I},\ \forall d\in\mathcal{D}.
$$

4. In day $d\in\mathcal{D}$ employee $i\in\mathcal{I}$ could not work less than $DCmax$ hours:

$$
LX_{i,d,j} + \sum_{t}^{DCmax} LX_{i,d,(j+t)} \le DCmax, \quad \forall i\in\mathcal{I},\ \forall d\in\mathcal{D},\ \forall j\in\mathcal{J}-DCmax.
$$

5. Variables $Y_{i,d}$ and $\sum_{j\in\mathcal{J}} LX_{i,d,j}$ should be related. In particular, it should be:

$$Y_{i,d} \leq \sum_{j\in\mathcal{J}} LX_{i,d,j} \leq M * Y_{i,d}, \quad \forall i \in \mathcal{I},\ \forall d \in \mathcal{D},$$

where M is a positive integer greater than or equal to the number of periods in day d.

6. Employee $i \in \mathcal{I}$ should work at least $DCmin$ consecutive hours in day $d \in \mathcal{D}$. This can be imposed by the following four constraints:

   (a) First, we should impose that $W^+_{i,d,j} = 1$ if $LX_{i,d,j-1} = 0$ and $LX_{i,d,j} = 1$; $W^-_{i,d,j} = -1$ if $LX_{i,d,j-1} = 1$ and $LX_{i,d,j} = 0$ (see Figure 5.2):

   $$LX_{i,d,j-1} - LX_{i,d,(j)} = W^+_{i,d,(j)} + W^-_{i,d,(j)}, \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \setminus \{1\}.$$

   (b) If variable $W^+_{i,d,j} = 1$, then the consecutive $DCmin$ periods of time must be equal to 1:

   $$\sum_{t=1}^{DCmin} LX_{i,d,(j+t)} \geq (DCmin) * W^+_{i,d,j},$$
   $$\forall i \in \mathcal{I},\ \forall d \in \mathcal{D},\ \forall j \in \mathcal{J} \setminus \{|\mathcal{J}| - DCmin, |\mathcal{J}| - DCmin + 1, \cdots, |\mathcal{J}|\}$$

   (c) In a given day $d \in \mathcal{D}$, if employee works in the first period, then the consecutive $DCmin - 1$ periods of time must be equal to 1:

   $$\sum_{t=1}^{DCmin-1} LX_{i,d,(j+t)} \geq (DCmin - 1) * LX_{i,d,0} \quad \forall i \in \mathcal{I} \quad \forall d \in \mathcal{D}$$

   (d) In a given day $d \in \mathcal{D}$, if employee works in the last period l, then the consecutive previous $DCmin - 1$ periods of time must be equal to 1:

   $$\sum_{t=1}^{DCmin-1} LX_{i,d,(j-t)} \geq (DCmin - 1) * LX_{i,d,l} \quad \forall i \in \mathcal{I} \quad \forall d \in \mathcal{D}$$

7. At least $HBREAKmin$ periods of break between two shifts in the same day $d \in \mathcal{D}$ should be assigned to employee $i \in \mathcal{I}$:

$$W^-_{i,d,j} + \sum_{t=1}^{HBREAKmin} LX_{i,d,(j+1)} \leq 0,$$
$$\forall i \in \mathcal{I},\ \forall d \in \mathcal{D},\ \forall j \in\ \{DCmin, DCmin + 1, \cdots, |\mathcal{J}| - DCmin]\}.$$

8. Each employee $i \in \mathcal{I}$ could not work more than $DAYCmax$ consecutive days. To enforce this constraint we need to take into account the shifts assigned to employee i during the previous week:

(a) For the first day, b_i measures the difference between the number of days in a week (7) and the number of days that employee i worked after the last day of rest in the previous week.

$$\sum_{t=1}^{d+7-b_i} Y_{i,(d+t)} = DAYCmax - b_i, \quad \forall i \in \mathcal{I},\ d = 1.$$

(b) For the remaining days:

$$\textstyle\sum_{t=0}^{DAYCmax} Y_{i,(d+t)} \leq DAYCmax,$$
$$\forall i \in \mathcal{I},\ \forall d \in \mathcal{D} \setminus \{|\mathcal{D}| - DAYCmax, |\mathcal{D}| - DAYCmax + 1, \cdots, |\mathcal{D}|\}.$$

9. Employee $i \in \mathcal{I}$ must work exactly $HSMAX_i$ hours per week.

$$\sum_{d=(s*7)+1}^{7*(s+1)} \sum_{j \in \mathcal{J}} LX_{i,d,j} = HSMAX_i \quad \forall i \in \mathcal{I} \quad \forall s \in \mathcal{S}$$

10. Each employee $i \in \mathcal{I}$ should work at most two Sunday per month.

$$\sum_{d=1}^{s} Y_{i,7*d} \leq 2 - \sum_{d=0}^{ns-s-1} Y_{i,7*d} \quad \forall i \in \mathcal{I}$$

11. If an employee $i \in \mathcal{I}$ is in vacation on day d, or he/she got some special permission that allows him/her not to work, then $Y_{i,d}$ should be equal to 0; otherwise, it could either be 0 or 1.

$$Y_{i,d} \leq 1 - Hday_{i,d} \quad \forall i \in \mathcal{I} \quad \forall d \in \mathcal{D}$$

5.3 Case study

The above mathematical model has been used to solve a scheduling problem on a real case study in Cagliari (Italy). The considered big store has 11 retail stores in Sardinia (Olbia, Carbonia, Sestu, Oristano, etc.), including the one in Cagliari. It sells a huge variety of products for the house and the garden, e.g., furnishings, carpentry, bricolage, paintings, lighting systems, animal foods and equipments, gardening, and so [22]. Each store has its own rules. Our model is suitable for all of them even if numerical results are presented in the following only for the largest and oldest store in Cagliari.

In the considered big store the team of employees consists of $|\mathcal{I}| = 20$ persons: 18 working full time ($HS_i = 40$ hours) and 2 working in part time ($HS_i = 20$ hours). Each employee (full time and part time) could not work more than $HDMAX = 9$ hours per day. The minimum number of consecutive hours of break between two shifts in the same day is equal to $HBREAK_{min} = 1.5$ hours.

For each employee, the maximum number of consecutive days per week is $DAYCmax = 6$. Furthermore, the maximum (minimum) number of consecutive working hours per day is $DCmax = 6.5$ hours ($DCmin = 3$ hours).

The considered store opens at 8:30 am and closes at 8:30 pm. We assume that shifts are multiple of half an hour. As a consequence, the interval of time in which the store is opened in partitioned into 24 subintervals, i.e., $\mathcal{J} = \{1, \cdots, 24\}$.

The scheduling is performed over a time horizon of one week, namely it is $s = 1$. Moreover, we assume that the current month has 4 weeks, which implies $ns = 4$.

We deal with the $|\mathcal{K}| = 9$ different tasks summarized in Table 5.1, each one requiring a specific skill.

K	Task
1	store manager
2	cashier 1
3	cashier 2
4	technical sale agent 1
5	technical sale agent 2
6	technical sale agent 3
7	furniture sale agent 1
8	furniture sale agent 2
9	warehouse worker

Table 5.1: Tasks definition

Table 5.2 summarizes the tasks that each employee may execute based on his/her own skills. Tasks are divided into "main task" and "other tasks". The "main task" corresponds to the main skill of the employee and it is the task for which the employee has been hired. In other words, it is the task that has been considered when fixing his/her salary. The "other tasks" are tasks that typically require a less qualified skill, so that are usually less paid. This is taken into account in the problem formulation assigning a smaller value to the cost $C_{i,d,j,k,z}$ if k is a "main task" rather than an "other task" for employee i (for a given triple of values d, j and z).

Employees working part-time are identified by $i = 18, 19$.

i	main task	other tasks
1	1	2-3
2,3	2	3
4	1	2-3-4-5-6-7-8-9
5	4	5-6
6,11,12,13,20	1	2-3-4-5-6-7-8-9
7,8,9,10	4	5-6
14,15	1	4-5-6
16	2	3
17	4	5
18	6	7-8-9
19	6	7-8

Table 5.2: Tasks that can be assigned to employees

We consider $|\mathcal{Z}| = 4$ different activities defined as in Table 5.3.

z	set of tasks
1	cashier 1, technical sale agent 1,furniture sale agent 1 warehouse worker, store manager
2	cashier 1, cashier 2, technical sale agent 1 technical sale agent 2, furniture sale agent 1, warehouse worker, store manager
2	cashier 1, cashier 2, technical sale agent 1 technical sale agent 2, furniture sale agent 1, warehouse worker, store manager
4	cashier 1, cashier 2, technical sale agent 1 technical sale agent 2, technical sale agent 3, furniture sale agent 1 furniture sale agent 2, warehouse worker, store manager

Table 5.3: Activities definition

The Figure 5.3 is an example of workforce demand $N_{d,j,k,z}$ where the day d is fixed. Periods of time j are reported in the abscissa while in the ordinate we have the number of employees required for the different tasks k.

For the sake of brevity we do not report the costs $C_{d,j,k,z}$'s for all d, j, k, and z. We only provide in Figure 5.4, the cost $C_{d,j,k,z}$ for employee $i = 5$ in a certain day d for the different tasks he/she can be execute.

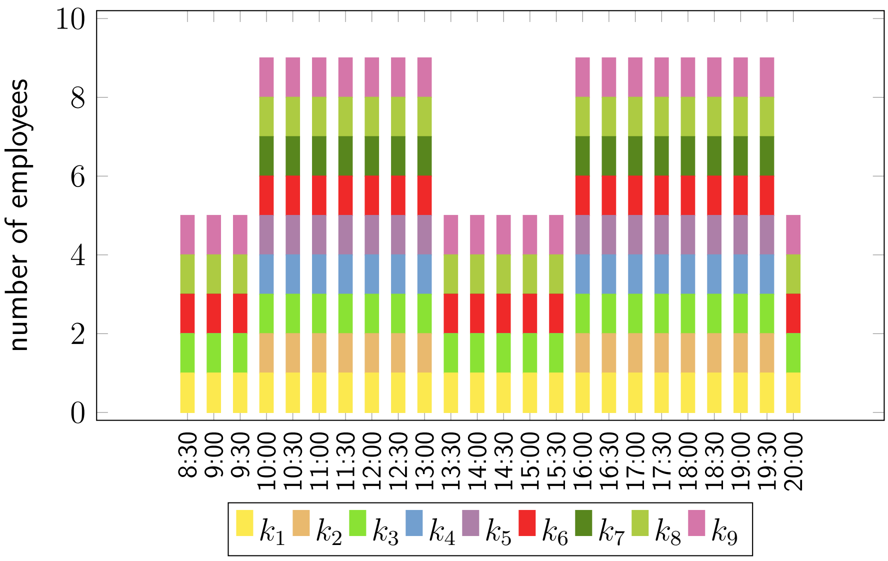


Figure 5.3: An example of worksforce demand for a given day

5.3.1 Web application and numerical simulations

In this subsection we present a web application that has been developed to implement the proposed approach in a user friendly manner. For a better understanding its description is done with reference to the considered case study. Therefore, we simultaneously illustrate the web application and the numerical simulations.

The web application architecture is sketched in Figure 5.5. Python is the main language. Furthermore, the micro-service Flask is used to keep the core simple but extensible. The database mySql is used to store data that are easily input via two interfaces built using bootsrap and html5.

The mathematical model is developed using the library PULP. Different optimization software packages can be selected, including Cplex, Gurobi, Xpress, GLPK.

Two screenshots of the interface that we use to enter input data are given in the following. Figure 5.6 shows how some preliminary settings relative to the big store are entered. Figure 5.7 shows how data relative to employees are entered, in particular: their names, their maximum number of working hours per week, their skills.

Two screenshots of the interface that provide output data, namely the results of the scheduling, are given in the following.

Figure 5.8 shows the screenshot of a summary of the scheduling during a day. Each column refers to a different employee: here we can read in which time intervals of the day he/she should work and, inside the green rectangles, the activity (first entry in the rectangle) and the task (second entry in the rectangle) to whom he/she is devoted. A

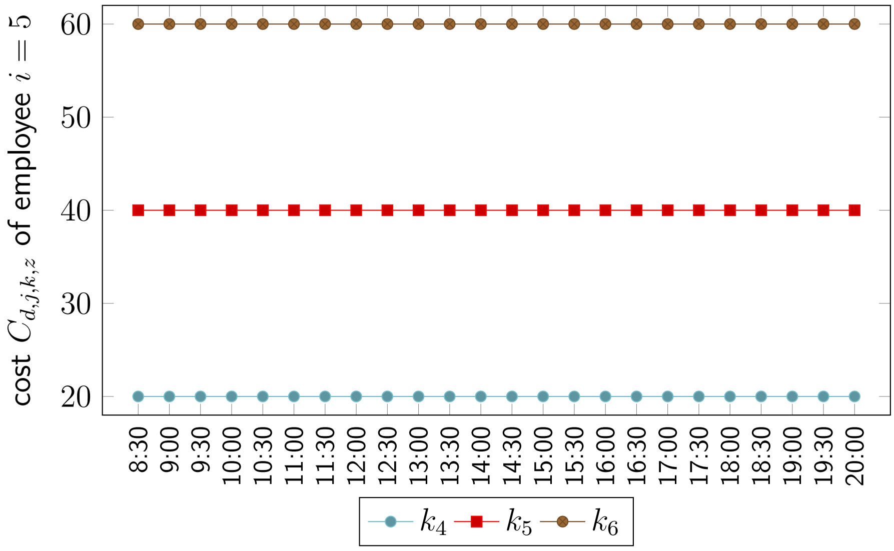


Figure 5.4: An example of the cost $C_{d,j,k,z}$ for employee i_5 in a day d where in the abscissa there are the periods of time j in the ordinate we have the costs for different skills that is able to do.

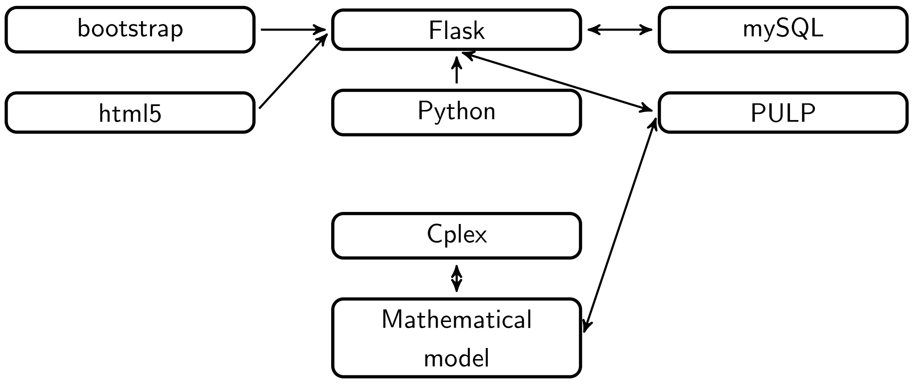


Figure 5.5: Architecture of the web application

slightly different notation is used here for activities: letters A, B, C, D refer to activities 1, 2, 3, 4, respectively. Figure 5.9 provides a zoom of a part of Figure 5.8. The two last rows provide a summary of the number of working hours in the day and in the week, respectively. Again, Figure 5.10 provides a zoom of a part of Figure 5.8.

Finally, the last two columns show how many employees are in surplus (blue rectangles) and how many employees are in deficit (red rectangles) in each period of time.

Other views, which are not reported here for sake of brevity, can be obtained to get a clear summary of several issues. Finally, given the current structure of the web application, it is very easy to further enrich the set of output views, depending on the requirements of the manager of the retail store.

To show the effectiveness of the proposed approach when applied to the considered

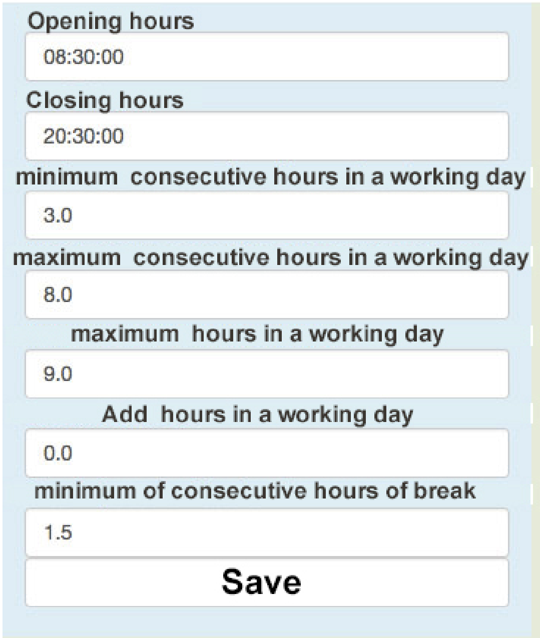
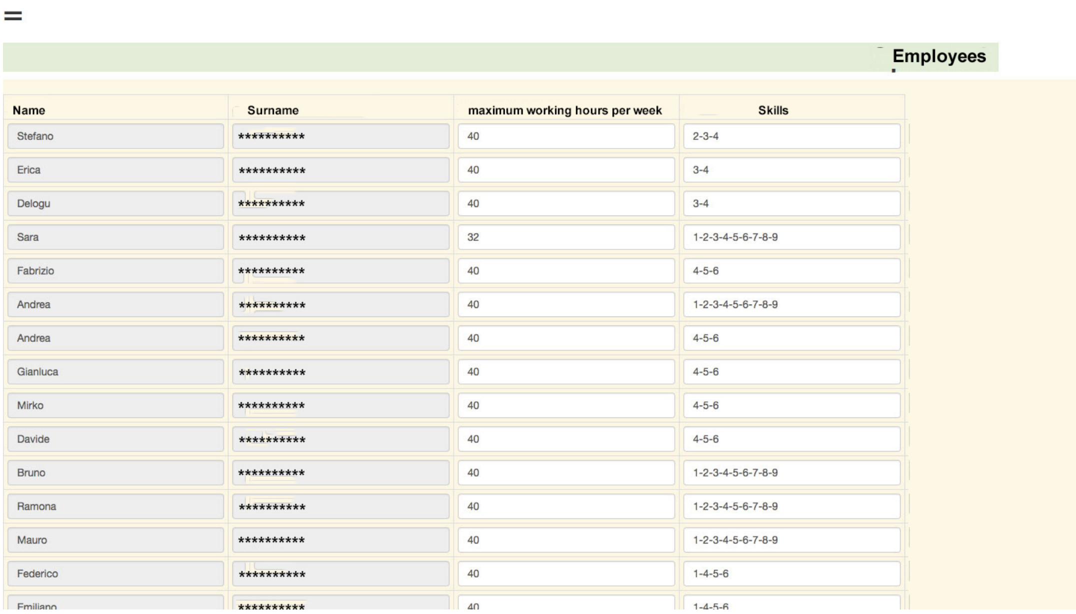
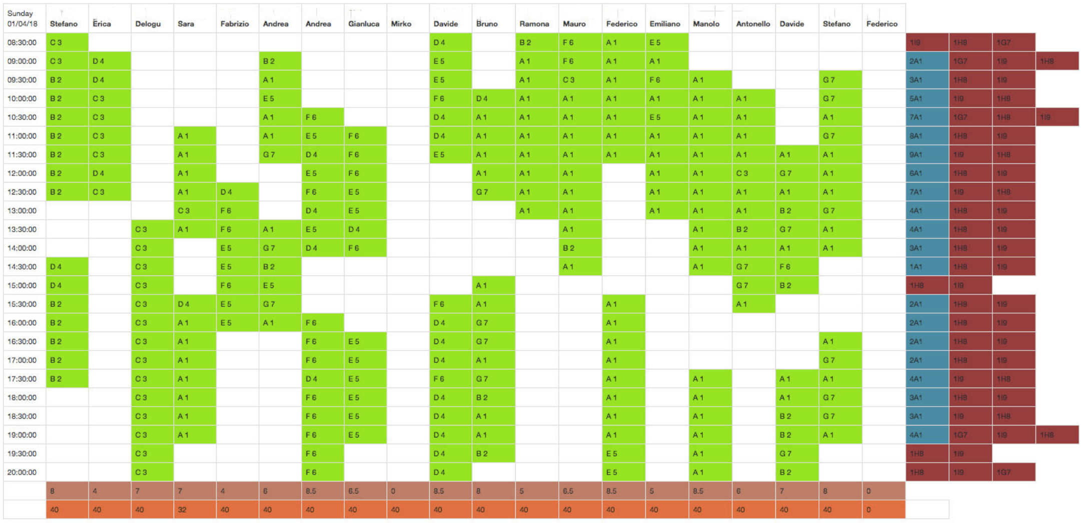


Figure 5.6: Screenshot of the interface used to enter preliminary settings on the big store



Name	Surname	maximum working hours per week	Skills
Stefano	**********	40	2-3-4
Erica	**********	40	3-4
Delogu	**********	40	3-4
Sara	**********	32	1-2-3-4-5-6-7-8-9
Fabrizio	**********	40	4-5-6
Andrea	**********	40	1-2-3-4-5-6-7-8-9
Andrea	**********	40	4-5-6
Gianluca	**********	40	4-5-6
Mirko	**********	40	4-5-6
Davide	**********	40	4-5-6
Bruno	**********	40	1-2-3-4-5-6-7-8-9
Ramona	**********	40	1-2-3-4-5-6-7-8-9
Mauro	**********	40	1-2-3-4-5-6-7-8-9
Federico	**********	40	1-4-5-6
Emiliano	**********	40	1-4-5-6

Figure 5.7: Screenshot of the interface used to enter data relative to employees

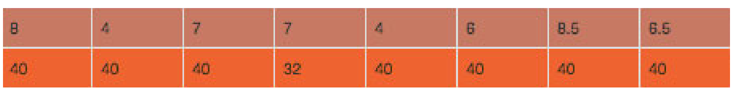


Sunday 01/04/18	Stefano	Erica	Delogu	Sara	Fabrizio	Andrea	Andrea	Gianluca	Mirko	Davide	Bruno	Ramona	Mauro	Federico	Emiliano	Manolo	Antonello	Davide	Stefano	Federico				
08:30:00	C 3									D 4		B 2	F 6	A 1	E 5						1I9	1H8	1G7	
09:00:00	C 3	D 4				B 2				E 5		A 1	F 6	A 1	A 1						2A1	1G7	1I9	1H8
09:30:00	B 2	D 4				A 1				E 5		A 1	C 3	A 1	F 6	A 1			G 7		3A1	1H8	1I9	
10:00:00	B 2	C 3				E 5				F 6	D 4	A 1	A 1	A 1	A 1	A 1	A 1		G 7		5A1	1I9	1H8	
10:30:00	B 2	C 3				A 1	F 6			D 4	A 1	A 1	A 1	A 1	E 5	A 1	A 1		A 1		7A1	1G7	1H8	1I9
11:00:00	B 2	C 3		A 1		A 1	E 5	F 6		D 4	A 1	A 1	A 1	A 1	A 1	A 1	A 1		G 7		8A1	1H8	1I9	
11:30:00	B 2	C 3		A 1		G 7	D 4	F 6		E 5	A 1	A 1	A 1	A 1	A 1	A 1	A 1	A 1	A 1		9A1	1I9	1H8	
12:00:00	B 2	D 4		A 1			E 5	F 6			A 1	A 1	A 1		A 1	A 1	C 3	G 7	A 1		6A1	1H8	1I9	
12:30:00	B 2	C 3		A 1	D 4		F 6	E 5			G 7	A 1	A 1		A 1	A 1	A 1	A 1	A 1		7A1	1I9	1H8	
13:00:00				C 3	F 6		D 4	E 5				A 1	A 1		A 1	A 1	A 1	B 2	G 7		4A1	1H8	1I9	
13:30:00			C 3	A 1	F 6	A 1	E 5	D 4					A 1			A 1	B 2	G 7	A 1		4A1	1H8	1I9	
14:00:00			C 3		E 5	G 7	D 4	F 6					B 2			A 1	A 1	A 1	A 1		3A1	1H8	1I9	
14:30:00	D 4		C 3		E 5	B 2							A 1			A 1	G 7	F 6			1A1	1H8	1I9	
15:00:00	D 4		C 3		F 6	E 5					A 1						G 7	B 2			1H8	1I9		
15:30:00	B 2		C 3	D 4	E 5	G 7				F 6	A 1			A 1			A 1				2A1	1H8	1I9	
16:00:00	B 2		C 3	A 1	E 5	A 1	F 6			D 4	G 7			A 1							2A1	1H8	1I9	
16:30:00	B 2		C 3	A 1			F 6	E 5		D 4	G 7			A 1					A 1		2A1	1H8	1I9	
17:00:00	B 2		C 3	A 1			F 6	E 5		D 4	A 1			A 1					G 7		2A1	1H8	1I9	
17:30:00	B 2		C 3	A 1			D 4	E 5		F 6	G 7			A 1		A 1		A 1	A 1		4A1	1I9	1H8	
18:00:00			C 3	A 1			F 6	E 5		D 4	B 2			A 1		A 1		A 1	G 7		3A1	1H8	1I9	
18:30:00			C 3	A 1			F 6	E 5		D 4	A 1			A 1		A 1		B 2	G 7		3A1	1I9	1H8	
19:00:00			C 3	A 1			F 6	E 5		D 4	A 1			A 1		A 1		B 2	A 1		4A1	1G7	1I9	1H8
19:30:00			C 3				F 6			D 4	B 2			E 5		A 1		G 7			1H8	1I9		
20:00:00			C 3				F 6			D 4				E 5		A 1		B 2			1H8	1I9	1G7	
	8	4	7	7	4	6	8.5	6.5	0	8.5	8	5	6.5	8.5	5	8.5	6	7	8	0				
	40	40	40	32	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	0				

Figure 5.8: The resulting scheduling in a certain day

Sunday 01/04/18	Stefano	Erica	Delogu	Sara
08:30:00	C 3			
09:00:00	C 3	D 4		
09:30:00	B 2	D 4		
10:00:00	B 2	C 3		
10:30:00	B 2	C 3		
11:00:00	B 2	C 3		A 1
11:30:00	B 2	C 3		A 1
12:00:00	B 2	D 4		A 1
12:30:00	B 2	C 3		A 1
13:00:00				C 3
13:30:00			C 3	A 1
14:00:00			C 3	

Figure 5.9: A zoom of the first columns and rows of Figure 5.8



8	4	7	7	4	6	8.5	6.5
40	40	40	32	40	40	40	40

Figure 5.10: A zoom of the last two rows of Figure 5.8

numb. of employees	time horizon [weeks]	numb. of tasks	Gap	Time [seconds]
20	1	1512	0.0	91.32
20	2	3024	0.0	1026,97

Table 5.4: Average computational times to compute the scheduling

case study, both in terms of computational times and distance from the optimality, Table 5.4 has been reported. A team of $N = 20$ employees has been considered (first column), and two different time horizons, one and two weeks (second column) with a number of tasks equal to 1512 and 3024, respectively (third column). In both cases (1 week and 2 weeks), a set of real data from four different time horizons have been considered to implement the approach. The average computational times are reported in the last column. Obviously, it is not linear with the length of the time horizon. Finally, in all cases, the percentage distance (Gap, fourth column) between the best possible objective and the best found objective, is null. In all cases, optimization has been performed using Cplex 12.6.2 on a mac air with 1.3 GHz Intel Core I5, 4 GB of RAM and 4 cores.

We conclude this section pointing out that a numerical comparison with real data is not provided, since the company currently computes the shifts assignment by hand, so it is not able to implement all the policies described in Sections 5.2 and 5.3.

Chapter 6

Conclusions and future work

Manpower planning or scheduling or rostering is the process of constructing work timetables for its staff so that an organization can satisfy the demand for its goods or services. It is extremely difficult to find good solutions to these highly constrained and complex problems and even more difficult to determine optimal solutions that minimize costs, meet employee preferences, distribute shifts equitably among employees and satisfy all the workplace constraints. In general, the unique characteristics of different industries and organizations mean that specific mathematical models and algorithms must be developed for personnel scheduling solutions in different areas of application.

The application areas in this thesis are: container terminals, hospitals, retail shops.

A short summary of the main contributions is provided in the following items.

- The short-term manpower planning problem plays a crucial role for TCTs, but it was rarely investigated in the literature. It consists of determining shifts, tasks and activities of workers, while avoiding both personnel under-manning and over-manning. The case study of a real TCT has revealed that personnel shifts and vessel services typically do not overlap, but this problem setting was not investigated. In order to cover this gap, the problem has been modelled by an integer linear programming formulation, which can be used for any configuration of personnel shifts, vessel activities and their possible overlapping.

  The optimal solutions of the model have been compared to the decision policy of this TCT, which ranks activities in a priority list and assigns workers to activities starting from the topmost task. A key critically in this policy is the unexploited option of flexible workers, who are assigned to shifts in order to provide a uniform manpower supply in each day. The motivation of this choice is just the simplicity of implementation for the TCT, which has no planning tools to evaluate other manpower configurations rapidly.

- A Decision Support System to solve the nurse scheduling problem in a long time horizon is proposed. It is based on the idea of splitting the problem in the whole time interval (typically one year) in several sub-problems (each one relative to one month) and update a series of information that could vary in the long time horizon. A linear integer programming problem has been formulated to optimally solve the problem in the short time horizon. All short time decisions take into account decisions adopted in the previous period while inheriting information to be implemented in the current period. This allows to minimize deficit and surplus in assigning shifts to nurses in a typical department, while taking into account the required constraints. This enables to automatically compute a solution that is optimal in the short time period according to certain criteria, e.g., (1) minimize the number of nurses in deficit, (2) minimize the difference between the number of hours that each nurse currently works during a month and the number of hours he/she is expected to work, (3) give priority to a schedule where free days are assigned consecutively.

  Results are validated using real data provided by the surgery department of the University Hospital in Cagliari, Italy, where shifts are currently assigned manually by a person devoted to this. This kind of approach could be time consuming (usually many hours) and, even more, could lead to inefficient schedules, namely schedules where some constraints are not satisfied. On the contrary, using the proposed approach, it is possible to reach in few minutes a solution that fulfills all the constraints.

  Thanks to its nice computational performance the proposed model could be usefully adopted in many occurrences, such as:

  – to highlight, face and forecast critical events in a department by simulating emergency situations;

  – to adopt, almost in real time, decisions different from those planned, if necessary, quickly re-optimizing with respect to new unexpected requirements;

  – to verify different policies of human resources assignment;

  – to avoid unbalances in shifts assignments.

- The problem of automatically scheduling the shifts of employees in a big store is proposed, taking into account a series of activities that should be performed, and the individual skills of employees. We focus on short-term scheduling and, as it typically occurs in this framework, the time horizon is taken equal to one week. However, the proposed solution is general and can be used even if the time horizon is multiple of one week. We assume that a series of information on the current week are taken into account. The main contribution is a mathematical model that allows to compute an optimal solution using integer linear programming. Note that, to make the approach user friendly, an interface is built.

As a future work we plan to extend the above results in the directions summarized in the following items.

- Research in the field is in progress to model activities, which may not necessarily overlap with personnel shifts. These activities may span several days and may interplay with other activities, which may gain priority in the uncertain environment of the shipping industry. In a future work, we aim to investigate the short-term manpower planning problem with stochastic workforce demand and planning horizons longer than that adopted in this thesis.

- The experimentation in [32] has shown that the proposed TCT policy is very effective in case of low variance in the daily manpower demand, as it returns the similar solutions to those obtained by the optimization model. However, the uniform manpower supply may be ineffective whenever the manpower demand is not uniform in the daily planning horizon. In fact, in case of medium and high variance, the model outperforms the terminal policy significantly, because it is able to increase the manpower in peak demand periods. The experimentation has revealed that this model can also be used in the case of huge TCTs; in fact all instances are optimally solved within an acceptable time interval. Research in the field is in progress to model account for possible sources of randomness, such as vessel delays or service disruptions. Finally, the model generality may be of interest for other application areas (e.g. the healthcare).

- As a future work related to nurse rostering, we first plan to consider more general scenarios, simultaneously dealing with sets of departments who share resources. Second, we plan to extend the proposed approach in order to be useful for the computation of summer and winter holidays, keeping into account the data from the previous years and the specific/personal requirements of the staff. Finally, we would like to define a user friendly interface that makes the resulting tool appealing for hospital operators.

- As a future work related to retailer stores, we plan to consider sets of shops sharing common resources. Then we plan to improve the way the workforce demand is computed since it has a strong impact on the effectiveness of the solution. Finally, we plan to improve our web application making it more flexible and useful in quite different environments.

Bibliography

[1] Alfares HK "Survey, categorization, and comparison of recent tour scheduling literature." *Annual Opereration Research*, Vol. 127, N. 1-4, pp. 145–175, 2004.

[2] P. Shahnazari-Shahrezaeia, R. Tavakkoli-Moghaddamb, H. Kazemipoor, "Solving a multi-objective multi-skilled manpower scheduling model by a fuzzy goal programming approach," *Applied Mathematical Modelling*, Vol. 37(1), pp. 5424-5443, 2013.

[3] J.P. Arabeyre, J. Fearnley, F.C. Steiger andW. "Teather, The airline crew scheduling problem: A survey",*Transportation Science 3*, pp. 140163, 1969.

[4] Ahuja-Magnanti-Orlin, "Network Flows", Prentice Hall (1993).

[5] M. Bagheri, A. Gholinejad Devin, A. Izanloo, "An application of stochastic programming method for nurse scheduling problem in real word hospital," *Computers & Industrial Engineering* , Vol. 96, pp. 192-200, 2016.

[6] Bard JF, Purnomo HW. " Cyclic preference scheduling of nurses using a Lagrangian-based heuristic ", *J Sched Archive*, Vol. 10, pp. 523, 2007.

[7] Bard JF, Binici C, desilva AH. "Staff scheduling at the United States postal service." Comput. Oper. Res., Vol. 30, pp. 745–771, 2003.

[8] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh and P.H. Vance, "Branch-and-price: column generation for solving huge integer programs", *Operations Research*, Vol. 46, pp. 316–329, 1998.

[9] Bazaraa-Jarvis-Sherali, "Linear Programming and Network Flows", Wiley(1994)

[10] J. Van den Bergh, J. Belin, P. De Bruecker, E. Demeulemeester, and L. De Boeck." Personnel scheduling: A literature review." *European Journal of Operational Research*, Vol. 226, N. 3, pp. 367–385, 2013.

[11] Bergh JVd, Belien J, Bruecker PD, Demeulemeester E, Boeck LD, "Personnel scheduling: a literature review," *Eur. J. Oper. Res.*, Vol. 226, N. 3, pp. 367–385, 2013.

[12] I. Berrada, J.A. Ferland, P. Michelon, "A multi-objective approach to nurse scheduling with both hard and soft constraints," *Socio-Economic Planning Sciences*, Vol. 30, N. 3, pp. 183-193, 1996.

[13] L. Bianco, M. Bielli, A. Mingozzi, S. Ricciardelli and M. Spadoni, "A heuristic procedure for the crew rostering problem",*European Journal of Operational Research* Vol. 58, pp. 272–283, 1992.

[14] I. Blöchliger, "Modeling staff scheduling problems. A tutorial", *European Journal of Operational Research*, Vol. 158, pp. 533-542, 2004.

[15] P. Bruckera, R. Qub, E. Burkeb, "Personnel scheduling: models and complexity," *European Journal of Operational Research*, Vol. 210, N. 3, pp. 467–473, 2011.

[16] E.K. Burke, P. De Causmaecker, G.Vanden Berghe, H. Van Landeghem, "The state of the art of nurse rostering," *Journal of Scheduling*, Vol. 7, N. 6, pp. 441–499, 2004.

[17] E K Burke, P De Causmaecker, and G Vanden Berghe." A hybrid tabu search algorithm for the nurse rostering problem," *In Selected Papers from the 2nd Asia Pacific Conference on Simulated Evolution and Learning* , volume 1585 of LNAI, pp. 187–194. Springer Verlag, 1998.

[18] E K Burke, P I Cowling, P De Causmaecker, and G Vanden Berghe. "A memetic approach to the nurse rostering problem," *Applied Intelligence*, pp. 199–214, 2001.

[19] A. Caprara, M. Fischetti, P.L. Guida, P. Toth and D. Vigo, "Solution of large-scale railway crew planning problems: The Italian experience," *Computer-Aided Transit Scheduling, ed. N.H.M. Wilson, Lecture Notes in Economics (Springer, Berlin, 1999)* pp. 118.

[20] Carlo, H.J., Vis, I.F.A. Roodbergen, K.J., "Literature overview, trends, research directions and classification scheme," *European Journal of Operational Research*, Vol. 236, N. 1, pp.1-13.

[21] P. De Causmaecker, P. Demeester, G. Vanden Berghe, B. Verbeke, "Analysis of real-world personnel scheduling problems," *5th Int. Conf. on the Practice and Theory of Automated Timetabling*, Pittsburgh, PA, USA, 2004.

[22] https://www.cfadda.com

[23] N. Chapados, M. Joliveau, P. L'Ecuyer, L.-M. Rosseau. "Retail store scheduling for profit," *European Journal of Operation Research*, Vol. 239, pp. 609-624, 2014.

[24] B. Cheang, H. Li, A. Lim, B. Rodrigues, "Nurse rostering problems a bibliographic survey," *European Journal of Operational*, Vol. 151, N. 3, pp. 447–460, 2003.

[25] Carlo, H.J., Vis, I.F.A. and Roodbergen, K.J. "Transport operations in container terminals: Literature overview, trends, research directions and classification scheme", *European Journal of Operational Research*, Vol. 236, No. 1, pp.1–13, 2014.

[26] Dantzig, G.B. (1987b). Simplex Method for Solving Linear Programs, in J. Eatwell, M. Milgate, and P. Newman (eds.), The New Palgrave: A Dictionary of Economic Theory and Doctrine, Vol. 4, The Macmillian Press, Ltd., London, 337340.

[27] De Bruecker, P., Van den Bergh, J, Beliën, J. and Demeulemeester, E. "Workforce planning incorporating skills: State of the art", *European Journal of Operational Research*, Vol. 243, N. 1, pp.1–16, 2015.

[28] De Bruecker, P., Van den Bergh, J, Belin, J. Demeulemeester, E., "Workforce planning incorporating skills: State of the art," *European Journal of Operational Research*, Vol. 243, N. 1, pp.1-16.

[29] Demassey, S., Pesant, G., Rousseau, L. M.: "Constraint programming based column generation for employee timetabling," *7th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* (CPAIOR05, Prague, 2005)

[30] Di Francesco, M., Fancello, G., Serra, P. and Zuddas P., "Optimal management of human resources in transhipment container ports", *Maritime Policy & Management*, Vol. 42, No. 2, pp.127-144, 2015.

[31] Di Francesco, M., Fancello, G., Serra, P. Zuddas, "Optimal Management of Human Resources in Transhipment Container Ports.," *Maritime policy management*, Vol. 42, N. 2, pp.127-144.

[32] M. Di Francesco, N. Diaz-Maroto Llorente, S. Zanda, P. Zuddas, "Planning and optimising manpower management in a transshipment container terminal," *European J. of Industrial Engineering (EJIE)*, Vol. 10, No. 6, 2016

[33] M. Di Francesco, N. Diaz-Maroto Llorente,S. Zanda , P. Zuddas , "An optimization model for the short-term manpower planning problem in transhipment container terminals," *Computers & Industrial Engineering*, Vol. 97, pp. 183–190, 2016.

[34] K A Dowsland and JM Thompson. "Solving a nurse scheduling problem with knapsacks, networks and tabu search.," *Journal of the Operational Research Society*, Vol. 51, pp. 825–833, 2000.

[35] Kathryn Dowsland. "Nurse scheduling with tabu search and strategic oscillation.," *European Journal of Operational Research*, Vol. 106, pp. 393–407, 1998.

[36] Ernst, A., Jiang, H., Krishnamoorthy, M., Sier, D.: "Staff scheduling and rostering: A review of applications, methods and models.," *European Journal of Operational Research*, Vol. 153, pp. 3–27, 2014.

[37] A.T. Ernst, H. Jiang, M. Krishnamoorthy, D. Sier, "Staff scheduling and rostering: A review of applications, methods and models,", *European Journal of Operational Research*, Vol. 153, pp. 327, 2004.

[38] N. Fan, S. Mujahid, J. Zhang, P. Georgiev, P. Papajorgji, P. Pardlos, "Nurse scheduling problem: An integer programming model with a practical application", *Systems analysis tools for better health care delivery*, Vol. 74, pp 65-98, New York: Springer, 2013.

[39] M. Gamache and F. Soumis, "A method for optimally solving the rostering problem," *in: OR in Airline Industry,* (Kluwer Academic, Boston, 1998) pp. 124–157.

[40] Glover F., McMillan C.,"The general employee scheduling problem: an integration of management science and artificial intelligence." *Comput Oper Res*, Vol. 13, pp. 563–593, 1986.

[41] Hartmann, S. "A general framework for scheduling equipment and manpower at container terminals", *OR Spectrum*, Vol. 26, No. 1, pp.51–74, 2004.

[42] Hartmann, S., "A general framework for scheduling equipment and manpower at container terminals.Container Terminals and Automated Transport Systems.," *Springer Berlin Heidelberg*, pp.207-230

[43] M. Isken, "An implicit tour scheduling model with applications in healthcare," *Annals of Operations Research*, Vol. 128, pp. 91–109, 2004.

[44] Isken M.W. "An implicit tour scheduling model with applications in healthcare." *Ann Oper Res* Vol. 128, pp. 91–109, 2004.

[45] zgr Kabak, Fsun lengin, Emel Akta, ule nsel ,Y. Ilker Topcu. "Efficient shift scheduling in the retail sector through two-stage optimization." *European Journal of Operational Research*, Vol. 184, pp. 76–90, 2008.

[46] Kim, K.H., Kim, K.W., Hwang, H. Ko, C.S., "Operator-scheduling using a constraint satisfaction technique in port container terminals.," *Computers Industrial Engineering*, Vol. 46, N. 2, pp.373–381.

[47] Kim, K.H., Kim, K.W., Hwang, H. and Ko, C.S. "Operator-scheduling using a constraint satisfaction technique in port container terminals", *Computers & Industrial Engineering*, Vol. 46, No. 2, pp.373–381, 2004.

[48] Shunyin Lam, Mark Vandenbosch, Michael Pearce. "Retail sales force scheduling based on store traffic forecasting." *European Journal of Operational Research*, Vol. 74, pp. 61–88, 1998.

[49] Lauer J, Jacobs LW, Brusco MJ, Bechtold S.E. "An interactive, optimization-based decision support system for scheduling part-time, computer lab attendants." *Omega*, Vol. 22, pp. 613–626, 1994.

[50] Legato, P. Monaco, M.F., "Human resources management at a marine container terminal.," *European Journal of Operational Research*, Vol. 156, N. 3, pp.769-781.

[51] Li, M-K. "A method for effective yard template design in container terminals", *European Journal of Industrial Engineering*, Vol. 8, No. 1, pp.1–21, 2014.

[52] Lim, A., Rodrigues, B., Song, L., "Manpower allocation with time windows," *Journal of the Operational Research Society*, Vol. 55, N. 11, pp.1178–1186.

[53] Legato, P. and Monaco, M.F. "Human resources management at a marine container terminal', *European Journal of Operational Research*, Vol. 156, No. 3, pp.769–781, 2004.

[54] Lim, A., Rodrigues, B., and Song, L. "Manpower allocation with time windows', *Journal of the Operational Research Society*, Vol. 55, No. 11, pp.1178–1186, 2004.

[55] B. Maenhout, M. Vanhoucke, "An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems," *Omega*, Vol. 41, n. 2, pp. 485–499, 2013.

[56] H.H. Millar, M. Kiragu, "Cyclic and non-cylcic scheduling of 12h shift nurses by network programming," *European Journal of Operational Research*, Vol. 104, n. 3, pp. 582–592, 1998.

[57] H.H. Millar, M. Kiragu, "Cyclic and non-cylcic scheduling of 12h shift nurses by network programming," *European Journal of Operational Research*, Vol. 104, N. 3, pp. 582–592, 1998.

[58] M. Moz, M. Pato, "Solving the problem of rerostering Nurse schedules with hard constraints: New multicommodity flow models," *Annals of Operations Research*, Vol. 128, pp. 179–197, 2004.

[59] Nearchou, A.C. and Lagodimos, A.G. "Heuristic solutions for the economic manpower shift planning problem", *European Journal of Industrial Engineering*, Vol. 7, No. 6, pp.657–686, 2013.

[60] Notteboom, T.E. "The Time Factor in Liner Shipping Services", *Maritime Economics & Logistics*, Vol. 8, No. 1, pp.19–39, 2006.

[61] M. Ohki, S.-Y. Uneme, H. Kawano, "Effective mutation operator and parallel processing for nurse scheduling", In V. Sgurev, M. Hadjiski, J. Kacprzyk (Eds.) *Intelligent systems: From theory to practice*, Vol. 299, pp 229–242, Berlin, Heidelberg: Springer, 2010.

[62] S. Zanda, C. Seatzu, P. Zuddas, "Long term nurse scheduling via a decision support system based on linear integer programming: A case study at the University Hospital in Cagliari," *Computers & Industrial Engineering*, 2018.

[63] H.W. Purnomo, J.F. Bard, "Cyclic preference scheduling for nurses using branch and price," *Naval Research Logistics*, Vol. 54, N. 2, pp. 200–220, 2007.

[64] Purnomo HW, Bard JF "Cyclic preference scheduling for nurses using branch and price." *Nav Res Logist* Vol. 54, pp. 200–220, 2007.

[65] R.C. Carrasco, "Long-term staff

[66] Roja Z, Kalkis V, Vain A, Kalkis H, Eglite M "Assessment of skeletal muscle fatigue of road maintenance workers based on heart rate monitoring and myotonometry." *J Occup Med Toxicol*, pp. 1–20, 2007.

[67] M. Sanja Petrovic, “You have to get wet to learn how to swim applied to bridging the gap between research into personnel scheduling and its implementation in practice,” PATAT 2016.

[68] K. Satheesh Kumar, G. Nagalakshmi, S. Kumaraguru, “A shift sequence for nurse scheduling using linear programming problem,” *Journal of Nursing and Health Science*, Vol. 3, N. 6, pp. 24–28, 2014.

[69] Serra, P., Fadda, P. and Fancello, F. “Evaluation of alternative scenarios of labour flexibility for dockworkers in maritime container terminals’, *Maritime Policy & Management*, DOI: 10.1080/03088839.2015.1043752, 2015.

[70] P. Smet, F. Salassa, G. Vanden Berghe, “Local and global constraint consistency in personnel rostering,” *Int. Trans. in Operation Research*, pp. 1–19, 2016.

[71] Stahlbock, R. Voss, S., Operations research at container terminals: a literature update,” *OR Spectrum*, Vol. 30, N. 1, pp.1–52. Stahlbock, R. Voss, S. (2008). Operations research at container terminals: a literature update. OR Spectrum, Vol. 30, N. 1, pp.1–52.

[72] Stahlbock, R. and Voss, S. “Operations research at container terminals: a literature update’, *OR Spectrum*, Vol. 30, No. 1, pp.1–52, 2008.

[73] Thompson GM, Goodale JC (2006) Variable employee productivity in workforce scheduling. Eur J Oper Res, vOL. 170, PP. 376–390, 2006.

[74] S. Topaloglu, H. Selim, “Nurse scheduling using fuzzy modeling approach,” *Fuzzy Sets and Systems*, Vol. 161, n. 11, pp. 1543–1563, 2010.

[75] L. Trilling, A. Guinet, D. Le Magny, “Nurse scheduling using integer linear programming and constraint programming,” 12th IFAC Symposium on Information Control Problems in Manufacturing, Saint-Etienne, France, 2006.

[76] UNCTAD “Review of Maritime Transport”, *Technical report, United Nations*, New York and Geneva, 2014.

[77] Vakharia AJ, Selim HS, Husted RR, Efficient scheduling of part-time employees. Omega Vol. 20, pp. 201–213, 1992.

[78] Christos Valouxis, Christos Gogos, George Goulas, Panayiotis Alefragis, Efthymios Housos, “A systematic two phase approach for the nurse rostering problem,” *European Journal of Operational Research*, Vol. 219, pp. 425–433, 2012.

[79] Christos Valouxis, Christos Gogos, George Goulas, Panayiotis Alefragis, Efthymios Housos, “A systematic two phase approach for the nurse rostering problem,” *European Journal of Operational Research*, Vol. 219, pp. 425–433, 2012.

[80] Vernimmen, B., Dullaert, W. and Engelen S. ‘Schedule Unreliability in Liner Shipping: Origins and Consequences for the Hinterland Supply Chain’, *Maritime Economics & Logistics*, Vol. 9, No. 3, pp.193–213, 2007.

[81] S. Zanda, P. Zuddas, C. Seatzu, "Short term manpower scheduling in retailer stores", *computers and industrial engineering* (in preparation)

[82] Z. Zhang, Z. Hao, H. Huang, “Hybrid swarm-based optimization algorithm of GA & VNS for nurse scheduling problem”, In B. Liu, C. Chai (Eds.) *Information computing and applications*, Vol. 7030, pp 375–382, Berlin, Heidelberg: Springer, 2011.