Mixed Integer Linear Programming Models for Scheduling Elective Surgical Procedures

Hanna Pamplona Hortencio(✉) and Débora Pretti Ronconi

Department of Production Engineering, EP-USP, University of São Paulo,
Av. Prof. Almeida Prado, 128, Cidade Universitária, São Paulo 05508-900, Brazil
hannapamplona@gmail.com

Abstract. The problem of scheduling surgeries consists of allocating patients and resources to each surgical stage, considering the patient's needs, as well as sequencing and timing constraints. This problem is classified as NP-hard and has been widely discussed in the literature for the past 60 years. Nevertheless, many authors do not take into account the multiple stages and resources required to address the complex aspects of operating room management. The general goal of this paper is to propose a mathematical model to represent and solve this problem. Computational tests were also performed to compare the proposed model with a similar model from the literature, with a 64% average reduction in computational time.

Keywords: Scheduling · Surgery · Mixed integer linear programming · Multiple resources · Multiple stages

1 Introduction

The aging population, a higher incidence of chronic diseases and reduced health care budgets are some of the reasons why hospitals need to improve their productivity [1,2]. In a hospital, the surgery department accounts for the highest revenue (around 70%), although it also represents the highest costs (approximately 40%) [3]. A survey conducted by the Brazilian Federal Council of Medicine and Data Folha Institute [4] on the Brazilian public health care system (SUS) revealed that 45% of patients wait longer than six months to book a surgery. Thus, it is extremely important that public and private hospitals have an efficient management system to reduce patients' waiting time and earn good profits.

Higher productivity in surgery departments depends on some factors, like efficient scheduling of activities, fast setups, and punctuality [5]. Operation research tools used in other service sectors like hotels or restaurants can be used in hospitals to find good results [6]. In this paper, we propose a mixed integer linear

This work has been partially supported by the Brazilian funding agencies FAPESP (grants 2013/07375-0 and 2016/01860-1), CNPq (grants 306083/2016-7), and CAPES.

O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12251, pp. 632–647, 2020.
https://doi.org/10.1007/978-3-030-58808-3_46

programming (MILP) model to solve the Problem of Scheduling Multiple Surgery Resources.

1.1 Problem Definition

The surgery scheduling problem consists of allocating resources to meet patients' needs, when and where these resources are required [7]. This work focuses on the offline operational hierarchical level, because its goal is to define the scheduling of known patients, considering all the necessary resources and the sequence of activities.

Surgical processes addressed in this paper include three stages: preoperative, perioperative, and postoperative. Many resources may be required to perform each stage and a resource can only be used by one patient at a time. The resource is released when the patient moves to the next stage; this constraint is called a blocking constraint [9]. Figure 1 shows a Gantt chart with an example of scheduling involving three patients. The first column shows patient stages, while the second presents the resources required for each stage. After determining which resources will be used by each patient at each stage, the start and end times of each stage are established, taking into account time constraints.

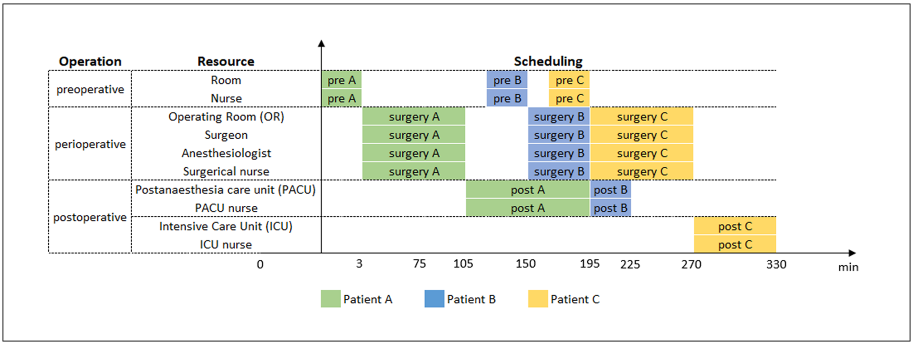

Fig. 1. Gantt chart: example with three patients.

This paper encompasses four sections. Section 1.1 brings a literature review on surgery scheduling at the offline hierarchical level. Section 2 presents two models - one from the literature and the proposed model. Section 3 reports computational experiments to validate and evaluate our model. Final remarks are given in Sect. 4.

1.2 Literature Review

This section consists of a brief literature review of the procedures used to schedule elective surgeries at the offline operational level. These papers have various degrees of complexity, ranging from one single operation and resource to multiple operations and resources. Different solution strategies were found, such as mathematical modeling, exact methods, heuristic approaches, and simulation. Furthermore, distinct objective functions can be defined for the surgery scheduling problem. A literature review shows that 78% of the papers published between 2000 and 2017 use objective functions to efficiently manage resources [8]. Weinbroum, Ekstein and Ezri [9] investigated the reasons that lead to low efficiency in operating rooms and those authors concluded that the unavailability of operating rooms and staff accounts for approximately 66% of delays, while other causes are related to activities that take place in the operating room before or after the surgery, such as the cleaning activities and preoperative and postoperative stages.

Conforti, Guerriero and Guido [10] presented a multi objective model that determines how medical specialties are distributed in surgical rooms and schedules surgeries for elective patients; therefore, it simultaneously addresses problems at the tactical and offline operational levels. Surgery scheduling is based on patients' clinical priorities and considers one stage and one resource. In order to minimize total costs, Fei *et al.* [11] presented a branch-and-price approach to solve a surgery scheduling problem with just one stage and one resource. In a subsequent publication [12], those authors developed a heuristic to define the tactical plan for the surgery department and an exact algorithm to define the patients' schedule. Two resources and just one stage were considered in [12].

Testi and Tafani [13] developed an MILP model that not only schedules surgeries at the operational level but also determines the tactical distribution of specialties. The model works with two stages - perioperative and postoperative -, and a single resource is used in each stage. The objective function seeks to maximize patients' well-being, based on waiting time indicators. Cardoen, Demeulemeester and Beliën [14] developed a model to solve a multiple-objective surgery sequencing problem, where two stages (perioperative and postoperative) are considered with one resource each. The same authors [15] proposed a branch-and-price approach model to solve a day-care scheduling problem. In addition to the perioperative and postoperative stages, the cleaning procedure was also considered. Both studies used data from a Belgian hospital. Perdomo, Augusto and Xie [16] formulated a surgery scheduling problem model that uses two stages (perioperative and postoperative) and two resources (operating room and PACU (Post-Anesthesia Care Unit)). In a subsequent work [17], those authors consider the same stages and resources, but allow a fraction of the postoperative stage to be performed in an operating room, if necessary.

To maximize the marginal financial contribution of each patient, Gartner and Kolisch [18] developed a model to solve the surgery scheduling problem. They described the problem in a Flexible Job Shop machine environment and took into account several stages. Riise, Mannino and Burke [19] also proposed a multi-resource, multi-stage model. The multi-objective function has four terms that are normalized and weighted according to the hospital's needs, therefore being a priority multi-objective optimization method. As pointed out by T'Kindt and

Billaut [20], this method is sensitive and can lead to different results depending on the weighting values used. Pham and Klinkert [21] introduced an MILP model which, in addition to preoperative, perioperative and postoperative stages, also considers cleaning and setup stages, thus addressing around 88% of the causes of delay (see [9]). The model also allows the inclusion of several resources in each stage to better represent an actual hospital setting.

It can be observed from the literature review presented previously in this paper that several surgery scheduling research studies do not consider multiple stages and multiple resources. This paper analyzes the model of Pham and Klinkert [21] and presents suggestions for improvements. This paper also proposes an alternative MILP model for the problem.

2 Mixed Integer Linear Programming Models

This section presents and discusses two MILP models. The first one was developed by Pham and Klinkert [21] and uses the concept of resource modules to simplify resource allocation. The second model was developed by the authors of this paper and aims to improve and expand Pham and Klinkert's model.

2.1 Model Proposed by Pham and Klinkert (2008)

The model presented in [21] for the elective surgery scheduling problem is called Multi-Mode Blocking Job Shop (MMBJS). The authors described the problem as a Flexible Job Shop machine environment, where multiple resources are required to process each task (job). The Job Shop is an environment with m machines and n tasks, where each task has its own route (sequence of operations) to follow. When an operation can be performed by more than one machine, this environment is called the Flexible Job Shop (FJS).

For the problem at hand, patients are equivalent to tasks in the FJS problem and operations correspond to stages in a surgical process. For the execution of each stage, a set of different resources is needed and they are grouped into modules, similarly to machines in the FJS environment. Once a stage is started, all the resources that make up the module cannot be allocated to another patient until the stage is completed and the patient is transferred to the next stage or leaves the system. This restriction is called blocking constraint and aims to eliminate patient waiting time between two stages of the surgical process without being allocated to any module. The MMBJS model and the notations used (Table 1) will be presented next.

Set $\Im$ contains all patients. For each patient $J \in \Im$ there is a set of processing stages (operations); the present work defines three stages for each patient: preoperative, perioperative, postoperative. The set of operations for all patients is called I. The set of two consecutive operations for the same patient J is defined as O_J. The patient to which operation i belongs is called J_i. Set M represents all the resources available in the system. Each operation requires a different set of resources for its execution. A module is a set of resources that meet the requirements for an operation and are available in the same period of time. Modules

Table 1. Notation of the MMBJS.

Sets	
$\Im$	Patients
I	Operations
O_j	Consecutive pairs of operations of patient $J \in \Im$
M	Resources
M_i^r	Resources contained in module r that serves operation $i \in I$, $r \in R_i$, $M_i^r \subseteq M$
R	Module indexes, $R = \cup_{i \in I} R_i$
R_i	Possible module indexes for operation $i \in I$
Parameters	
p_i^r	Processing time of operation i in module r, $i \in I, r \in R_i$
p_i^{su}	Setup time before operation $i \in I$
p_i^{cl}	Cleanup time after operation $i \in I$
b_i	Maximum waiting time allowed for operation $i \in I$ before the patient is moved to the next stage
e^r	Start time of the availability range of module $r \in R$
f^r	End time of the availability range of module $r \in R$
H	Big number: $H = \sum_{i \in I} \max_{r \in R_i}\{p_i^r\} + \max_{r \in R_i, i \in I}\{p_i^r\} + \max_{i \in I}\{p^s u_i\} + \max_{i \in I}\{p^c l_i\}$
α	Small weight factor
J^i	Patient which operation $i \in I$ belongs to
Decision variables	
z_i^r	1, if module r is assigned to operation $i \in I$; 0, otherwise
y_{ij}	1, if operation j is processed after operation i; 0, otherwise $i, j \in I$, $i < j$
x_i	Start time of operation $i \in I$
l_i	End time of operation $i \in I$
τ	Makespan

that share the same resource cannot be used by two patients simultaneously. The set of resources that can process operation i is defined as $M_i^r \in M$, where r belongs to the set of indexes of the modules that can process operation i. A fictional module is available at the end of the time horizon. Unscheduled patients are assigned to an artificial module that does not consume any system resources, which is called fictitious module. The set of all indexes of the modules is defined by R. Each operation $i \in I$ has the following parameters: processing time (p_i^r), which is dependent on the module to be used, setup time (p_i^{su}), cleaning (p_i^{cl}), and maximum waiting time (b_i). Modules are available in an interval indicated by $[e^r, f^r]$, in which $0 \leq e^r \leq f^r$. Parameters H and $alpha$ are a large number and a small number, respectively.

Decision binary variables z_i^r and y_{ij} indicate whether module r will execute operation i and determine the precedence between two operations, respectively. Integer variables x_i and l_i determine the beginning and end of each operation. The makespan is represented by τ and indicates the moment when the last patient leaves the system.

MMBJS model:

$$\min \qquad \tau + \alpha \sum_{i \in I} x_i \tag{1}$$

subject to:

$$\sum_{r \in R_i} z_i^r = 1 \quad i \in I \tag{2}$$

$$l_i - x_i - p_i^r z_i^r \geq 0 \quad i \in I, r \in R_i \tag{3}$$

$$l_i - x_i - \sum_{r \in R_i} p_i^r z_i^r \leq b_i \quad i \in I \tag{4}$$

$$l_i - x_j = 0 \quad (i, j) \in O_j, J \in \Im \tag{5}$$

$$x_j - l_i + H(2 - z_i^r - z_j^s) + H(1 - y_{ij}) \geq p_i^{cl} + p_j^{su} \quad \begin{aligned} &i, j \in I, i < j, J^i \neq J^j, \\ &r \in R_i, s \in R_j, M_i^r \cap M_j^s \neq 0 \end{aligned} \tag{6}$$

$$x_i - l_j + H(2 - z_i^r - z_j^s) + H y_{ij} \geq p_j^{cl} + p_i^{su} \qquad \begin{aligned} &i, j \in I, i < j, J^i \neq J^j, \\ &r \in R_i, s \in R_j, M_i^r \cap M_j^s \neq 0 \end{aligned} \tag{7}$$

$$x_i - e^r z_i^r \geq p_i^{su} \quad i \in I, r \in R_i \tag{8}$$

$$\sum_{r \in R_i} f^r z_i^r - l_i \geq p_i^{cl} \quad i \in I \tag{9}$$

$$\tau - l_i \geq p_i^{cl} \quad i \in I \tag{10}$$

$$x_i, l_i, \tau \geq 0 \quad i \in I \tag{11}$$

$$y_{ij} \in \{0, 1\} \quad i, j \in I, i < j \tag{12}$$

$$z_i^r \in \{0, 1\} \quad i \in I, r \in R_i \tag{13}$$

The objective function (1) minimizes the makespan, and then the sum of all start times, weighted by factor *alpha*. Constraints (2) assign a single module to each operation. Restrictions (3) and (4) determine the time of entry and exit of an operation. Restrictions (5) determine that there is no time interval between two subsequent operations. Restrictions (6) and (7) prevent two modules that share the same resource from being scheduled simultaneously. Restrictions (8) and (9) guarantee that setup and cleaning times will be carried out within the module's availability interval. Constraints (10) define the makespan value. Restrictions (11), (12) and (13) indicate the domain of decision variables.

Objective Function Analysis. The makespan is defined as $max(C_1, C_2, ..., C_k, ..., C_n)$, where C_k indicates the end time of operation k, that is, the makespan is when the last operation leaves the system. In the model analyzed in this paper, the makespan is calculated based on the set of restrictions (10) and (11) and on the minimization of the objective function. Restrictions (10) indicate that all operations are considered for the evaluation of the makespan, including those that are in the fictional module which does not represent real resources.

Therefore, patients allocated to this module should not be considered in the calculation of the makespan. Moreover, as the fictitious module is at the end of the time horizon, whenever an operation is rejected, the makespan is associated with an unscheduled patient. The following set of constraints can calculate the actual makespan value:

$$\tau \geq l_i + p_i^{cl} - G \sum_{r \in R_i^r} z_i^r \quad i \in I \tag{14}$$

where R_i^r is the set of all non-fictitious modules and G is a sufficiently large number.

The sum of start times of operations aims to reduce the number of patients allocated in the fictitious module. The definition of α is a fundamental task, since it is intended to weigh the values of the two terms of the objective function and guarantee an optimal schedule. By analyzing the objective function, it can be noted that, when the value of α is large, the generated solution can anticipate the beginning of operations without worrying about finishing them as early as possible, worsening the makespan. On the other hand, considering that the makespan calculation is performed in the operations allocated in the real modules, when the α factor is very small, many patients may not be scheduled, i.e. they are allocated in the fictitious module.

Constraint Analysis. The set of constraints (3) and (4) determines the start and end times of operations. Although both constraints have similar functions, they have different structures. Note that the device used in the second expression reduces the number of restrictions. The same occurs for the set of restrictions (8) and (9) that defines that the activities must be performed within the time window of the module. Thus, restrictions (3) and (8) can be replaced with restrictions (15) and (16), respectively.

$$l_i - x_i - \sum_{r \in R_i} p_i^r z_i^r \geq 0 \qquad i \in I \tag{15}$$

$$x_i - \sum_{r \in R_i} e^r z_i^r \geq p_i^{su} \qquad i \in I \tag{16}$$

These alternative constraints may lead to a gain in computational time while solving this problem.

Additional Remarks. By analyzing the objective function (1), it can be seen that there is no prioritization rule for patients. This fact can benefit patient scheduling with shorter processing times. In a hospital environment, the lack of a patient prioritization rule may be undesirable, since patients who require more complicated and therefore longer procedures may not be scheduled. Pham and Klinkert [21] presented a solution to this problem by establishing patient priorities i by fixing variable $y_{ij} = 1$ for all j. By predetermining these variables, it is possible to schedule priority patients when they compete with other patients for the same resource, ensuring that they are scheduled. However, previously determining the sequence of patients reduces the search space and can exclude solutions that include priority patients and have better makespan.

2.2 Proposed Model

The proposed model considers all the analyses presented in the previous section to improve the MMBJS model. First, in order to guarantee the correct makespan evaluation, the dummy module was excluded from the model and an alternative strategy to select the patients was implemented. Furthermore, considering that the definition of the parameter α is not a trivial task and has a great influence on the quality of the solution, a new objective function is proposed. This function also minimizes the number of unscheduled patients but does not use parameter α. A weighting factor associated with each patient was also included in the objective function to consider priority patients.

The set of constraints (3) and (8) can be reduced if replaced with restrictions (15) and (16), respectively. As observed in preliminary tests, fewer restrictions reduced the model's execution time to find an optimal solution. Therefore, this replacement was carried out in the proposed model.

Table 1 and Table 2 show all the notations used in the proposed model. Table 2 presents notations that are exclusive to the proposed model. The makespan value is indicated by mkp to differentiate it from the makespan value of the MMBJS model.

Proposed model:

$$min \quad mkp - G \sum_{i \in I} \sum_{r \in R_i} w_i z_i^r \tag{17}$$

Table 2. Notation of the proposed model.

Parameters	
G	Big number used in the objective function $G = \max_{r \in R}\{f^r\}$
w_i	Weighted priority factor for operation $i \in I$
Decision variables	
mkp	Makespan

subject to:

$$\sum_{r \in R_i} z_i^r \leq 1 \quad i \in I \tag{18}$$

$$l_i - x_i - \sum_{r \in R_i} p_i^r z_i^r \geq 0 \quad i \in I \tag{19}$$

$$l_i - x_i - \sum_{r \in R_i} p_i^r z_i^r \leq b_i \quad i \in I \tag{20}$$

$$l_i - x_j = 0 \quad (i,j) \in O_j, J \in \Im \tag{21}$$

$$x_j - l_i + H(2 - z_i^r - z_j^s) + H(1 - y_{ij}) \geq p_i^{cl} + p_j^{su} \quad i,j \in I, i < j, J^i \neq J^j, \tag{22}$$

$$r \in R_i, s \in R_j, M_i^r \cap M_j^s \neq 0$$

$$x_i - l_j + H(2 - z_i^r - z_j^s) + H y_{ij} \geq p_j^{cl} + p_i^{su} \quad i,j \in I, i < j, J^i \neq J^j, \tag{23}$$

$$r \in R_i, s \in R_j, M_i^r \cap M_j^s \neq 0$$

$$x_i - \sum_{r \in R_i} e^r z_i^r \geq p_i^{su} \sum_{r \in R_i} z_i^r \quad i \in I \tag{24}$$

$$\sum_{r \in R_i} f^r z_i^r - l_i \geq p_i^{cl} \sum_{r \in R_i} z_i^r \quad i \in I \tag{25}$$

$$mkp - l_i \geq p_i^{cl} \quad i \in I \tag{26}$$

$$x_i, l_i, mkp \geq 0 \quad i \in I \tag{27}$$

$$y_{ij} \in \{0,1\} \quad i,j \in I, i < j \tag{28}$$

$$z_i^r \in \{0,1\} \quad i \in I, r \in R_i \tag{29}$$

In the objective function (17), the number of scheduled patient activities, weighted by the priority factor, is calculated by the term $\sum_{i \in I} \sum_{r \in R_i} w_i z_i^r$. This part is multiplied by constant G to ensure that this term will be the problem at the top level, while makespan remains at the bottom level. Note that, in the proposed model, patients' priority is established through factor (w_i). Thus, it is possible to schedule a higher priority patient in the time horizon without necessarily getting this patient to be the first to be scheduled.

Observe that the set of constraints (18) was converted into inequalities when compared with an analogous set of constraints of the MMBJS model. Given the absence of the fictitious module, these restrictions guarantee that each operation should only be scheduled by no more than one module. Constraints (19) and (20) determine the time at which an operation starts and ends, respecting the maximum allowed waiting time. As previously discussed, the set of restrictions (19) was reduced in relation to the MMBJS model. The set of constraints (21) guarantees that there will be no waiting time for the patient between two consecutive operations. Constraints (22) and (23) ensure that modules that share the same resource are not scheduled simultaneously. Restriction sets (24) and (25) ensure that activities will be allocated to modules within their availability range. Two changes were made to these sets with respect to the MMBJS model. The set of restrictions (24) was reduced and the term $\sum_{r \in R_i} z_i^r$ added to the end of the restrictions, thus eliminating the fictitious module of the model. Constraints (26) determine that the makespan is the longest time to end all system activities. These restrictions do not need require any changes in relation to the MMBJS model since there is no longer a fictitious module at the end of the time horizon. Finally, restrictions (27), (28) and (29) determine the domain of decision variables.

Objective Function Analysis. The objective function of the proposed model maximizes the number of scheduled patients and, among those solutions with the largest number of patients, it chooses one that minimizes makespan. It is worth noticing that the modeled problem is not equivalent to the problem of finding a single schedule that maximizes the number of scheduled patients while, in a second stage and preserving the same sequence, minimizes the makespan.

Here is an example to illustrate the reason to use this function. Two situations are analyzed: (*i*) using the proposed model to maximize only the number of scheduled patients and, subsequently, making an adjustment to reduce the makespan and (*ii*) analyzing the problem in two levels using the objective function (17). In the example, there are five patients available at the beginning of the horizon and five modules: 1 preoperative, 2 perioperative, and 2 postoperative modules. Figure 2(a) shows the Gantt chart that represents the solution obtained by solving the proposed model to maximize only the number of scheduled patients. In this solution, four patients are scheduled and the makespan value is 1050. Figure 2(b) presents the same sequence of patients' activities adjusted to reduce idle times and, consequently, the makespan. The value of the makespan of this schedule is 980 units of time. Figure 2(c) shows the solution obtained through the resolution of the complete proposed model. The number of scheduled patients remains four, as expected, and the makespan value is 940 units of time. The makespan value is reduced because this solution was found using the proposed MILP model which, among all schedules for 4 patients, seeks one that minimizes makespan.

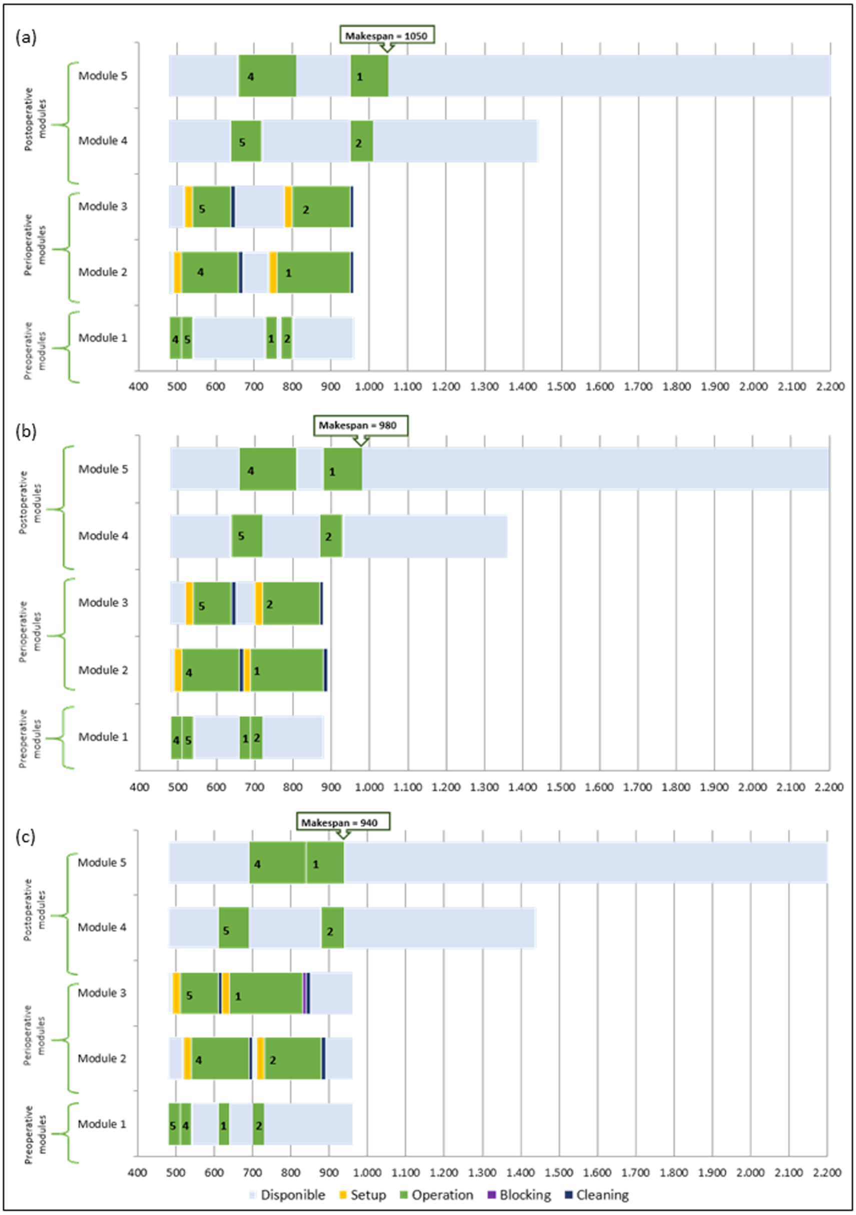

Fig. 2. Gantt charts: (a) optimal solution that only minimizes the number of rejected patients, (b) solution with the same sequence presented in a Gantt chart of (a) without idle times, and (c) optimal solution found by solving the proposed MILP model.

3 Numerical Experiments

The MILP models were implemented in CPLEX ILOG version 12.6. The default configuration parameters have not been changed, except for a CPU time limit of two hours per instance and a suboptimality tolerance level of 0.005%. The experiments were carried out with an Intel (R) Xeon (R) processor, 6 cores, 4.20 20 GHz, 512 GB of RAM, and on a Linux operating system.

Since the problem of scheduling elective surgeries with multiple resources and multiple stages has not yet been widely studied in the literature, new instance sets were generated for this problem based on instances from the literature. Ozkarahan [22] proposed instances with surgery processing time divided by specialty. Pham and Klinkert [21] completed the data from Ozkarahan [22], adding the processing times of preoperative and postoperative stages, maximum waiting time, and setup and cleaning times. Since full instances have not been made available, the present work also includes similar instances that were randomly generated. The proposed set of instances considers three medical specialties in which patients are distributed. The random number generator and the seeds proposed by Taillard [23] were used to generate processing, setup, cleaning, and maximum waiting times, as well as a weighted priority factor for patients using a discrete uniform distribution. The resources required for each stage were determined as suggested by Pham and Klinkert [21] and the time horizon considered was five days.

In the first part of the numerical experiments, a small set of instances was generated to conduct a comparative assessment of the models. Next, a sensitivity analysis was carried out to investigate the performance of the proposed model in different scenarios. With this purpose, 245 instances were generated and the obtained results will be briefly discussed in Sect. 3.2.

3.1 Comparative Evaluation

In this section, numerical experiments were carried out to compare the proposed model and the MMBJS model. Since the MMBJS model cannot properly calculate the makespan value if a patient is rejected, instances with enough resources were generated to avoid unscheduled patients. Moreover, in the experiments with this model, each instance was tested several times in order to find the appropriate value of α.

Table 3 shows the results obtained using the MILP models presented in the previous section. In the instances considered, there are 10 patients, 16 resources, and 47 modules distributed in 5 days. The MMBJS* model corresponds to the MMBJS model with the replacement of constraints (3) and (8) with constraints (15) and (16), respectively. The first column identifies the instance. The following columns show the running time required to solve each instance and the column named GAP (%) indicates the percentage difference between the incumbent solution and the lower bound determined by CPLEX.

Figures in the table show that the solver execution time using the MMBJS model is greater than the corresponding time of the proposed model for all instances. The use of the proposed model led to a 64% reduction in the average computational time. Furthermore, the proposed model was able to find the optimal solution for all instances, while with the MMBJS model, three optimal solutions could not be found within the limit time, with an average gap of 22.34%. By comparing models MMBJS* and MMBJS, it can be seen that, in most instances, the computational time of CPLEX is shorter using the MMBJS* model. The average reduction in computational time was 54%, indicating that a reduction in the restriction set size was beneficial. On the other hand, when comparing this model with the proposed model, it can be noted that, in general, the proposed model is solved in less time and, unlike the MMBJS* model, all the optimal solutions were found.

Table 3. Performance of the analysed models.

Instance	MMBJS		MMBJS*		Proposed	
	Time (s)	GAP (%)	Time (s)	GAP (%)	Time (s)	GAP (%)
1	247.97	00.00	196.59	00.00	169.19	00.00
2	338.99	00.00	90.38	00.00	256.82	00.00
3	7214.19	29.90	7200.37	21.25	144.08	00.00
4	7206.98	13.25	1200.00	00.00	120.19	00.00
5	146.58	00.00	117.15	00.00	84.03	00.00
6	922.54	00.00	1188.33	00.00	407.17	00.00
7	7238.41	23.88	749.04	00.00	154.46	00.00

3.2 Sensitivity Analysis

In this section, numerical experiments were performed to analyze the results obtained by CPLEX when using the proposed model for the resolution of instances with different characteristics. Three characteristics of the base group of instances (*ID = 0*) were modified: availability interval of modules, except for the ICU modules, number of ICU postoperative modules, and number of preoperative modules. For each characteristic, two groups of instances were created. In the first group, modules were added or the availability of modules was increased in relation to the base problem configuration, making the system more flexible. In the second group, the availability of modules was reduced or modules were excluded, consequently, the system became more restricted. In each group, five different amounts of available patients were considered (5, 7, 10, 12, 15). For each combination of characteristic, group, and number of patients, 7 instances were generated, i.e, $3 \times 2 \times 5 \times 7 = 210$ instances were solved. Additionally, 35 instances (5 sets of patients × 7 instances) were generated to compose the base group.

The solver was able to achieve the optimal solution within the allowed execution time in approximately 86% of instances. In the remaining instances, the average gap was around 9%. As expected, the solutions found by the solver for more restricted instances contain results (number of scheduled patients) equal to or worse than the results of the base group. Conversely, in instances where idleness was added to the proposed model, the solver found similar or better results. This pattern was verified in the 210 instances tested. In order to illustrate the solver's behavior using the proposed model, results obtained for instances with 10 patients and different features are detailed in Table 4. In this table, the first column (*ID*) identifies the instance and its attributes (characteristic and group), while the column named GAP (%) indicates the percentage difference between the incumbent solution and the lower bound determined by CPLEX. Details about the numerical experiment described in this section can be found in [24].

Table 4. Results for an instance with 10 patients.

ID	Makespan	# scheduled patients	Unscheduled patients	GAP (%)	CPU time (s)
0	3 d 13 h 43 min	8	6, 9	0.0*	203.40
1.1	4 d 16 h 41 min	10	–	0.0*	169.19
1.2	4 d 13 h 25 min	8	6, 9	0.0*	144.30
2.1	4 d 13 h 25 min	10	–	0.0*	2369.01
2.2	3 d 14 h 52 min	6	1, 6, 9, 10	0.0	85.61
3.1	3 d 13 h 43 min	8	6, 9	0.0*	176.45
3.2	4 d 13 h 16 min	8	6, 9	0.0*	238.49

*indicates that the suboptimality tolerance level has been reached

4 Final Remarks

This research addressed the scheduling of elective surgical procedures with multiple stages and limited resources. According to the literature review, it was possible to observe that most prior studies have not considered multiple stages and resources in their mathematical models. The present work aimed to study and improve the MILP model presented in [21], called Multi-Mode Blocking Job Shop. First, the MMBJS model was analyzed and modifications were suggested to improve the addressed model, such as a reduction in the number of constraints and an effective way to evaluate the makespan of the final schedule. Then, a mixed integer linear programming model based on MMBJS was proposed. The benefits of this objective function were discussed through the analysis of an example.

Numerical experiments were performed. A comparison between the MMBJS model and the proposed model showed a significant improvement in execution time. A sensitivity analysis was then performed to interpret the model's behavior. As a suggestion for future work, heuristics and metaheuristics should be developed to solve large instances seeking to reduce the computational effort.

Acknowledgments. The authors would like to thank the careful reading of the reviewers, whose comments helped to improve the quality of the manuscript.

References

1. Demeulemeester, E., Belien, J., Cardoen, B., Samudra, M.: Operating room planning and scheduling. In: Denton, B. (ed.) Handbook of Healthcare Operations Management: Methods and Applications, pp. 121–152. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-5885-2_5
2. Samut, P., Cafri, R.: Analysis of the efficiency determinants of health systems. Soc. Ind. Res. **129**, 113–132 (2016)
3. Molina-Parente, J.; Framian, J.: Testing planning policies for solving the Elective Case Scheduling phase: a real application. In: 35th International Conference on Operational Research Applied to Health Services, pp. 1–10. ORAHS, Leuven (2009)
4. Federal Council of Medicine Homepage. http://portal.cfm.org.br/. Accessed 15 Mar 2020
5. Kaye, A., McDowell, J., Diaz, J., Buras, J., Young, A., Urman, R.: Effective strategies in improving operating room case delays and cancellations at an academic medical center. J. Med. Pract. Manage. **30**(6), 24–29 (2015)
6. Gupta, D., Denton, B.: Appointment scheduling in health care: challenges and opportunities. IIE Trans. **40**(9), 800–8019 (2008)
7. Hans, E., Vanberkel, P.: Operating theatre planning and scheduling. In: Hall, R. (ed.) Handbook of Heathcare System Scheduling, pp. 105–130. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-1734-7_5
8. Hortencio, H., Ronconi, D., Miyake, D.: Produtividade na programaç ao de centros cirúrgicos: revisão de literatura. XX Simpósio de Administração da Produção. Logística e Operações Internacionais, pp. 1–16. FGV-EAESP, São Paulo (2017)
9. Weinbroum, A., Ekstein, P., Ezri, T.: Efficiency of the operating room suite. Am. J. Surg. **185**(3), 244–250 (2003)
10. Conforti, D., Guerriero, F., Guido, R.: A multi-objective block scheduling model for the management of surgical operating rooms: new solution approaches via genetic algorithms. In: 2010 IEEE Workshop on Health Care Management, Venice, pp. 1–5. IEEE (2010)
11. Fei, H., Chu, C., Meskens, N., Artibac, A.: Solving surgical cases assignment problem by a branch-and-price approach. Int. J. Prod. Econ. **112**(1), 96–108 (2008)
12. Fei, H., Meskens, N., Combes, C., Chu, C.: The endoscopy scheduling problem: a case study with two specialised operating rooms. Int. J. Prod. Econ. **120**(2), 452–462 (2009)
13. Testi, A., Tànfani, E.: Tactical and operational decisions for operating room planning: efficiency and welfare implications. Health Care Manage. Sci. **12**(363), 363–373 (2009)
14. Cardoen, B., Demeulemeester, E., Beliën, J.: Optimizing a multiple objective surgical case sequencing problem. Int. J. Prod. Econ. **119**(2), 354–366 (2009)
15. Cardoen, B., Demeulemeester, E., Beliën, J.: Sequencing surgical cases in a day-care environment: an exact branch-and-price approach. Comput. Oper. Res. **36**(9), 2660–2669 (2009)
16. Perdomo, V., Augusto, V., Xie, X.: Operating theatre scheduling using Lagrangian relaxation. Eur. J. Ind. Eng. **2**(2), 172–189 (2008)

17. Augusto, V., Xie, X., Perdomo, V.: Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. Comput. Ind. Eng. **58**(5), 231–238 (2010)
18. Gartner, D., Kolisch, R.: Scheduling the hospital-wide flow of elective patients. Eur. J. Oper. Res. **233**(3), 689–699 (2014)
19. Riise, A., Mannini, C., Burke, E.: Modelling and solving generalised operational surgery scheduling problems. Comput. Oper. Res. **66**, 1–11 (2016)
20. T'Kindt, V., Billaut, J.: Multicriteria Scheduling - Theory, Models and Algorithms, 2nd edn. Springer, Heidelberg (2006). https://doi.org/10.1007/b106275
21. Pham, D., Klinkert, A.: Surgical case scheduling as a generalized job shop scheduling problem. Eur. J. Oper. Res. **185**(3), 1011–1025 (2008)
22. Ozkarahan, I.: Allocation of surgeries to operating rooms by goal programing. J. Med. Syst. **24**(6), 339–378 (2000)
23. Taillard, E.: Benchmarks for basic scheduling problems. Eur. J. Oper. Res. **64**(2), 278–285 (1993)
24. Hortencio, H.: Técnicas de pesquisa operacional aplicadas ao problema de programação de cirurgias eletivas. São Paulo (2019). www.teses.usp.br. Accessed 23 Apr 2020