New solution approaches for the surgical cases assignment problem: mixed integer programming vs. biased random-key genetic algorithm

Fabrício Sperandio[a], José Fernando Gonçalves[b], José Borges[a], Bernardo Almada-Lobo[a]

[a]*CEGI – INESC TEC, Faculdade de Engenharia, Universidade do Porto, Portugal*
[b]*LIAAD – INESC TEC, Faculdade de Economia, Universidade do Porto, Portugal*

Abstract

This work addresses the Surgical Cases Assignment Problem (SCAP) in large hospitals. The problem consists in generating a weekly surgery schedule that assigns operating rooms (ORs), surgery dates and starting times to the elective surgeries in the surgical waiting list. An admissible schedule is subject to the surgeons and the ORs capacity constraints, as well as to the patients' priorities and the established rules for waiting time limits. Due to long waiting lists and the scarcity of ORs, the aim is to maximize both the number of scheduled surgeries and the ORs' utilization. Two alternative solution approaches are proposed and compared, one exact and one approximate. The exact approach is based on mixed integer programming (MIP). For this model the problem is formulated as a scheduling problem with block synchronization using a continuous representation of time, which contributes to maximizing the ORs utilization. A comparison of the MIP model against a model using a discrete representation of time, proposed in a previous work, is presented. The approximate approach is based on the biased random-key genetic algorithm (BRKGA). This is a population based approach which uses a vector of random numbers to represent each individual in the population and requires a decoding procedure to translate them into valid surgery schedules. Our This approach employs an efficient heuristic which keeps track of the resource available times and is able to translate every vector into a high quality solution. The alternative methods are compared using instances based on real data from a large hospital. The experimental results show that the MIP model, using a continuous representation of time, outperforms in terms of quality of the solutions the model using a discrete representation of time in all instances. In addition, the BRKGA outperforms the MIP in terms of quality of solutions in the majority of the test instances. **(** falta aqui qq coisa **)**

Keywords: Surgery scheduling, Mixed integer programming, Genetic algorithm

1. Introduction

Healthcare expending continues to rise among OECD countries. In 2012, the overall healthcare expenditure across these countries accounted for 9.3% of GPD on average, higher than the 8.6% accounted before the global financial crisis of 2007-08 [1]. Such rise is driven by an increasing demand for healthcare services which in turn is influenced by factors like: new and more expensive technology, ageing population and lifestyle issues (e.g. obesity). In this scenario, healthcare managers face a tough challenge to improve quality and efficiency, while preserving the sustainability of healthcare organizations. This paper is a contribution to the field of operations research and to society, as it promotes the efficient utilization of a core hospital resource, the operating theater (OT).

The OT accounts for up to 40% of hospital revenues and expenses [2, 3], making it one of the hospital's most important facilities. Its expenses are driven by a high consumption of human and material resources. Many surgeries are technically complex and require a range of people to be in one place working in harmony, including one or more surgeons, one or more anaesthetists, as well as special theatre nurses, assistants and technicians [4]. Mayer et al.[5] emphasize the importance of optimizing the utilization of such an expensive resource, citing the average cost of running one OT in NHS Scotland (National Health Service for Scotland) facilities to be £ 1.1 million per week. Furthermore, its operation has a direct impact on many other upstream and downstream resources. As a result, the OT is often called the heart of the hospital.

The literature on operations research applied to healthcare includes a high volume of studies tackling operating room (OR) planning and scheduling problems. The largest portion tackles problems in the operational decision level.

The surgery scheduling problem in the operational decision level consists in selecting patients from the surgical waiting lists and assign the OR, the surgery date and starting time to each of them. Usually this task is performed over a one week planning horizon. Due to its complexity, the problem it is often decomposed into two sub-problems: the advance and the allocation scheduling problems. The advance problem consists in selecting a set of patients from the waiting list and assigning the dates for their surgeries. The allocation problem consists in sequencing the surgeries within each day. **(** Bernardo, verifica se isto verdade: The decision of the OR allocation can be left to the advance or to the allocation problem. **)** The majority of the studies tackle each of the two problems separately, although there is a trend to adopt an integrated approach. The integrated approaches have to deal with very complex, thus, in order to reduce the problem complexity to the best of our knowledge the reported approaches use a discrete representation of time.

This work proposes a new modelling approach for the integrated (advance and allocation) Surgical Cases Assignment Problem (SCAP) using a continuous representation of time, thus providing a more accurate representation of the problem and a potential higher resource utilization. The challenge is to propose an exact yet efficient mathematical formulation for the problem. A Mixed Integer Programming (MIP) model is proposed that is inspired on efficient formulations for the Travelling Salesman Problem (TSP), using an analogy between the cities of the TSP and the operating rooms a surgeon works in a given shift. In our model, a surgeon is allowed to move between ORs during the same day and working shift. Thus, the utilization rates may increase since the surgeon's turnover time, the time required for a surgeon to start a new surgery in a different OR, is, in general, lower than the required cleaning time between two consecutive surgeries in the same OR. **(** Assuming that there are more surgeons than OR ??? **)** In addition, we propose an original heuristic solution method that aims to find near optimal solutions within a reduced amount of time. The proposed approach is based on the Biased Random-Key Genetic Algorithm framework (BRKGA)[6] and on an efficient decoding procedure to translate each individual in the population into a high quality schedule.

The proposed methods are evaluated both on the required computational times and on the quality of the induced solutions. The performance of the proposed approaches is analysed from three different perspectives. First, the new MIP model is compared with a modification of the model presented in [7], which uses a discrete representation of time. The modified model is described in the appendix of this Section. **(** porquê modificado?**)** Second, the new MIP model is compared with the heuristic solution approach. Finally, the quality of solutions found by the two proposed approaches along the search progress is compared. All computational experiments were performed over instances generated with real data from a large hospital.

The remainder of this paper is organized as follows. Section 2 reviews existing approaches for the surgery scheduling problem. Section 3 describes in detail the particular problem addressed in this paper. Section 4 introduces the two proposed approaches: the exact MIP model and the heuristic genetic algorithm. Section 5 describes the computational experiments designed to compare both approaches and presents the results. Finally, the last section highlights the main contributions of this paper and indicates some areas for future work.

2. Literature Review

This section is based on a review of selected papers from both problem and solution perspectives and reports the main characteristics of these papers such as decisions, objective and constraints summarized in Table 1 and Table 2.

2.1. Problem Perspective

The management of surgical services entails several complex decision problems. These problems are often classified into three decision levels: strategic, tactical and operational. The strategic level encompasses case mix and capacity planning problems. The first consists in determining the volume and type of surgeries to be performed by each specialty in the long term (1 to 5 years). The second consists in determining the number and capacity of resources dedicated to surgical services as well as their allocation. In the tactical decision level, two main different strategies are used: open scheduling and block scheduling. In the open scheduling strategy ORs are occupied by patients of any specialty. This strategy aims to maximize ORs' utilization rates. On the other hand, the block scheduling strategy requires to solve a Master Surgery Scheduling (MSS) problem, which consists in determining the ORs reserved for each specialty in each day of the week and working shift. The resulting plan is a weekly timetable implemented in the medium term (6 to 12 months). This is the most used strategy, mainly in large hospitals where the use of a MSS is well established. Regarding the operational problem, in the open strategy it encompasses all specialties together, while in the block strategy it is subdivided among the specialties.

The problems arising in the operational decision level are classified into off-line and on-line scheduling problems. The off-line problem consists in selecting patients from the waiting list and assigning ORs, surgery dates and starting times over a short term planning horizon (typically 1-week). The on-line problem consists in scheduling daily emergency and high priority cases as well as rescheduling previous elective cases.

This study tackles the off-line surgery scheduling problem at the operational decision level, also known as surgical cases assignment problem. We focus in the deterministic version of the problem, since addressing uncertainty is out of the scope of this study. A comprehensive review of surgery planning and scheduling problems can be found in Cardoen et al. [8] and Guerriero and Guido [9]. Herein, a selected set of papers addressing the SCAP problem was reviewed from both the problem perspective and the solution perspective. Table 2.1 summarizes the review of modeling approaches and Table 2.2 the review of solution approaches.

As mentioned in the previous section, the SCAP problem can be decomposed into two sub-problems: the advance and the allocation scheduling problems. The advance problem concerns the surgeries' dates while the allocation problem concerns their starting times. In most studies the OR assignment is part of the advance problem, however, in Ozkarahan [10] it is part of the allocation problem with the advance problem consisting only in assigning a surgery date. In addition, some studies combine other decisions, such as: assigning available ORs [11, 12], assigning ORs to specialties [13, 14] and assigning surgeons to patients [15]. Most studies tackle the advance and allocation problems separately but there is a growing number of integrated approaches. **(** referłncia **)**

The optimization objectives in SCAP problems are either related with resources or patients. Regarding resources, the main objectives are: maximize the ORs occupancy rates, minimize overtime and minimize makespan. The latter objective, along with the objective of minimizing the stay in recovery after closure time, is closely related with minimizing overtime, so that one can infer that even more studies aim to minimize overtime. It is worth noting that this is planned overtime since all studies consider deterministic surgery durations. In such cases, there is often a trade-off between opening new ORs, keeping patients waiting and incurring overtime costs. Concerning human resources, Ogulata and Erol [16] aim to balance the distribution of surgeries among groups of surgeon and Meskens et al. [17] aim to maximize the affinities among members of the surgical team. The patient related objectives include: maximizing the number of scheduled patients, minimizing the patients waiting time and minimizing the costs of keeping patients in the hospital waiting to be treated. In addition, [18, 19] focus on particular patients' groups, such as: high priority patients, children, and patients with long travel distance. In spite of the aforementioned objectives, most approaches combine multiple objectives. This is most often achieved through an aggregated objective function.

With regard to the constraints, they are either related to physical resources, human resources or patients. In the first category, ORs are the main resource followed by post-anaesthesia care unit (PACU) and intensive care unit (ICU). Pham and Klinkert [20] also consider the preoperative holding unit (PHU) as a limited resource. In addition, few studies consider the availability of surgical materials, medical instruments and equipment. Also, Augusto et al. [21] consider the availability of transporters, since such resource may be a bottleneck, mainly in the beginning of the day. Regarding human resources, the main surgeon in charge is the main constraint. Most studies consider the surgeon's availability and a few consider workload and overtime limits. Typically, a surgeon is assigned to the surgical case during the waiting list registration phase. To the best of our knowledge, Vijayakumar et al. [15] is the only work that assigns surgeons to surgical cases during the scheduling phase. Also, few papers have considered the other members of the surgical team, such as anaesthetists and nurses. Regarding patients, studies have considered constraints in the patient due date and admission date.

2.2. Solution Perspective

Regarding continuous and discrete representations of time, this literature review reveals that most studies represent time as discrete intervals, being fifteen minutes is the most common value used for the size of intervals. Cardoen [18] and Cardoen et al. [19] use a smaller interval (five minutes), but these studies focus solely on the allocation problem (sequencing patients within a day), which is less complex than the integrated problem.

The majority of the approaches which use a continuous representation of time tackle the allocation problem alone or decompose the overall problem into two sub-problems, each one approached independently.

Table 1: Summary table of the literature review: modelling Approaches

	Decisions						Objectives															Constraints																	
							Resources									Patients						Physical Resources									Human Resources					Patients			
Reference	A	B	C	D	E	F	A	B	C	D	E	F	G	H	I	A	B	C	D	E	F	A	B	C	D	E	F	G	H	I	A	B	C	D	E	A	B	C	D
[13]	•	•		•			•											•					•	•								•	•				•			
[14]	•	•		•			•															•	•								•	•				•			
[11]		•		•	•				•		•												•	•						•						•		•	•	
[22]		•		•			•		•									•				•	•		•	•	•					•	•				•		
[12]		•			•		•		•		•												•	•	•		•					•							•		
[23]		•							•									•				•	•	•			•				•	•						•	•	
[24]		•							•									•				•	•	•			•				•	•						•	•	
[25]	•	•		•					•								•						•									•							•	
[18]				•									•	•					•	•	•	•	•			•		•			•								•
[19]				•									•	•					•	•	•	•	•			•		•			•								
[20]				•				•																		•														
[16]	•	•					•							•		•						•				•									•		•		
[21]				•				•	•													•								•									
[17]				•				•							•																	•				•		
[15]	•	•		•		•												•											•							•	•		
[26]				•				•																														
[27]	•	•								•												•	•															•	
[10]			•	•			•		•																													
This paper	•	•		•			•	•															•	•	•						•	•	•				•	•	

Decisions

(A) Select patients
(B) Assign OR and date
(C) Assign OR only
(D) Assign starting time
(E) Select open ORs
(F) Assign surgeons to patients

Objectives

Resources

(A) Max OR occupancy rate
(B) Min makespan
(C) Min overtime costs
(D) Min total operating cost
(E) Min costs of opening ORs
(F) Min peak number of bed spaces used in PACU units
(G) Min stay in recovery after the closure time
(H) Balance the distribution of surgeries among surgeons groups
(I) Max affinities among members of the surgical team

Patients

(A) Min patients waiting time
(B) Max number of scheduled surgeries
(C) Min cost of keeping patients in hospital waiting to be treated
(D) Min sum of the starting times of surgeries performed on children
(E) Min sum of the starting times of surgeries performed on prioritized patients
(F) Min number of patients with long travel distance scheduled below a reference time

Constraints

Physical Resources

(A) Operating Rooms (OR)
(B) OR opening hours
(C) OR overtime limits
(D) Intensive care unit (ICU)
(E) Recovery room (PACU)
(F) Equipment
(G) Medical instruments
(H) Pharmaceutical products
(I) Transporters

Human Resources

(A) Surgeons
(B) Surgeon workload limits
(C) Surgeon overtime limits
(D) Surgeon groups
(E) Surgical team

Patients

(A) Patient priority
(B) Patient due date
(C) Patient admission date
(D) Cleaning time for infected patients

Zhong et al. [26] and Ozkarahan [10] address the integrated problem considering time as a continuous variable but use simple heuristics (longest processing time and shortest processing time) to get an approximate solution. To the best of our knowledge, Pham and Klinkert [20] is the only work that presents an exact model using a continuous representation of time. The authors propose an extension of the job shop scheduling problem, called multi-mode blocking job shop, and conclude that the model can obtain (good) feasible solutions for only small to medium-sized instances. **(** verificar: The integrated problem is known to be hard to solve, resulting in long running times. Thus, in general an optimal solution is difficult to obtain and in practice is useful more as a reference solution to evaluate the quality of the solutions obtained by the heuristics. **)**

Due to the high complexity of the SCAP problem researchers are focused on developing efficient search algorithms, relaxation approaches and search heuristics. Only the studies that decompose the overall problem into more manageable sub-problems do not have to rely on approximation algorithms. Among the exact search algorithms with proof of optimality we highlight: branch and bound, column generation, DantzigWolfe decomposition, branch and price, and the Hungarian algorithm. In addition, Lagrangian relaxation is commonly used to find an approximate solution. Other solution methods applied to find approximate solutions are: iterated local search, genetic algorithms (GA) and tabu search. Finally, researchers have considered constructive and improvement heuristics.

Random-key genetic algorithms (RKGA) for solving sequencing problems were introduced by Bean [28]. The biased random-key genetic algorithm, proposed by Gonçalves and Resende [6], is a slight modification of Bean's original method, differing in the way parents are selected for mating and how mating is carried out. Gonçalves et al. [29] compared biased and unbiased versions of RKGAs and concluded that the biased variant is faster. Heuristics based on BRKGAs and RKGAs have already been applied with success on resource constrained project scheduling [30, 6], resource constrained multi-project scheduling [31] and job shop scheduling [32] problems, which are similar to the SCAP problem. A detailed description of the BRKGA is provided in section 4.2.

3. Problem Description

The problem consists in assigning a surgery date, an operating room and a starting time to a set of elective patients in the waiting list, thus integrating advance and allocation scheduling. Each surgery of a patient has a pre-assigned surgeon, a date limit for the surgery and and estimated surgery duration.

The objective is to maximize the number of scheduled surgeries and the average OR utilization rate. These are conflicting objectives. On one hand, the maximization of the number of scheduled surgeries induces short surgeries. On the other hand, the maximization of the utilization rate prefers longer surgeries in order to avoid the setup time involved in the cleaning activities that must take place between surgeries. The cleaning time consumes OR capacity but is not taken into account in the utilization rate. An admissible surgeries' schedule must obey the following constraints:

1. OR cleaning time - Time after each surgery that must be reserved for performing OR cleaning protocol activities in order to setup the OR for the next event. The next surgery in the same OR can only start after the setup operation (that occurs in between surgeries) is completed;
2. Surgeon turnover time - A surgeon is allowed to have scheduled surgeries in more than one OR in the same shift as long as is is guaranteed an offset time between two consecutive surgeries of the same surgeon. This offset is called turnover time and denotes the required time for a surgeon to change from one OR to another after finishing a surgery;
3. OR time capacity - Each OR has a predefined time capacity on each shift. Naturally, the summation of the scheduled surgeries' durations and setup times within each OR and shift must not exceed such predefined capacity. Furthermore, as overtime is not allowed, a surgery must not be scheduled to end after the OR's closing time.
4. Surgeon availability and working limits - Each surgeon may be or may not be available to operate in a given shift of a certain day. When a surgeon is not available none of his/her patients can be scheduled for that period. Moreover, surgeons are subject to constraints with respect to the number of working shifts per week since such working limits are not necessarily guaranteed by the surgeon availability.
5. Patient priority and waiting time rules - Each patient in the elective surgery waiting list has a predefined priority and a current waiting time. In some countries, there are limits for the maximum waiting time established according to the priority level. Every surgery in the national health service should respect those limits. Moreover,

Table 2: Summary table of the literature review: solution approaches

	Approach		Planning Horizon		Strategy		Formulation				Solution Approach														Time Modelling		
Reference	A	B	A	B	A	B	A	B	C	D	A	B	C	D	E	F	G	H	I	J	K	L	M	N	A	B	C
[13]		•		•	•		•		•			•	•												•		15
[14]		•		•	•			•	•				•												•		15
[11]		•		•				•		•	•	•													•		15
[22]		•		•				•		•			•													•	
[12]	•			•	•			•		•	•				•			•								•	
[23]	•			•				•		•									•						•		60
[24]	•			•		•		•		•									•						•		60
[25]		•		•		•		•		•										•					•		15
[18]	•		•			•		•		•			•												•		5
[19]			•			•		•		•					•										•		5
[20]	•	•	•			•	•		•		•					•										•	
[16]	•			•		•				•	•														NA	NA	NA
[21]		•																		•				•		10	
[17]	•		•			•	•																•		•		30
[15]		•		•		•	•				•			•											•		15
[26]		•		•		•	•								•											•	
[27]	•			•	•		•										•							•		•	
[10]	•		•			•	•															•				•	
This paper		•		•		•		•		•	•	•														•	

Approach
(A) Decomposed
(B) Integrated

Planning Horizon
(A) 1 day
(B) 1 or 2 week

Strategy
(A) Open scheduling
(B) Block scheduling

Formulation
(A) Single-objective
(B) Multi-objective
(C) Independent
(D) Aggregated

Solution Approach
(A) Linear Programming (LP)
(B) Genetic Algorithms (GA)
(C) LP + Heuristics
(D) Heuristics
(E) Column Generation
(F) Branch and Price
(G) Branch and Bound
(H) Hybrid (GA + Tabu Search)
(I) Hungarian Method
(J) Iterated Search
(K) Lagrangian Relaxation
(L) Goal Programming
(M) Constraint Programming
(N) DantzigWolfe Decomposition

Time Modelling
(A) Discrete
(B) Continuous
(C) Size of Intervals (min)

there are limits for the waiting time until the surgery is scheduled for each priority. Therefore, when a patient reaches the limits for scheduling he must be scheduled, with the surgery date subject to the constraint of the total waiting time for surgery.

4. Methodology

4.1. Exact Solution Approach: Mixed Integer Programming Model

The first approach is a mixed integer programming (MIP) model, which uses a continuous representation of time. The aim is to determine the scheduled surgeries in a given planning horizon, and the corresponding start and finishing times. A surgeon is allowed to perform one or more surgeries within a period of work, which is defined as the time between the start of the first surgery and the end of the last consecutive surgery of surgeon in a shift and OR. Once the surgeons' periods of work are determined by the model, the start and end times of each specific surgery are assigned by means of a simple heuristic. The following paragraphs describe the model in detail. The idea is to define the sequence of surgeons in each OR avoiding potential overlaps between periods of work of the same surgeon in different ORs. The model uses three groups of decision variables: the first to decide which patients to schedule in each shift and OR; the second to decide on the sequence of surgeons within each shift and OR; and the third to determine the start and end times of surgeons in each shift and OR. It is assumed that each patient is waiting for a single surgery and therefore hereafter the terms patient and surgery may be used interchangeably.

We start by introducing the necessary notation:

Sets and indices

I	set of patients (index i)
J	set of working shifts (index j)
K	set of operating rooms (index k)
K_j	set of available ORs in shift j
S	set of surgeons (index s)
I_s	set of patients of surgeon s (index i)
H	set of weeks in the planning horizon (index h)
J_h	set of working days in a given week h (index j)
$I_{maxsched}$	set of patients with maximum scheduling time within the planning horizon
$I_{maxswait}$	set of patients with maximum waiting time within the planning horizon

Parameters

d_i	estimated duration of patient's i surgery
s_i	surgeon in charge of patient's i surgery
max_i	maximum waiting time of patient's i surgery
c_{jk}	available capacity of the OR k in shift j
a_{js}	1, if surgeon s is available in shift j; 0, otherwise
day_j	day of shift j
α	weight of the number of scheduled surgeries in the objective function
β	weight of the average OR utilization rate in the objective function
γ	best number of scheduled surgeries
δ	best average OR utilization rate
ct	OR cleaning time
tt	surgeon turnover time
C	total OR capacity
ms	maximum number of shifts per week

Decision variables

$$X_{ijk} = \begin{cases} 1, \text{if patient } i \text{ is scheduled for shift } j \text{ and OR } k \\ 0, \text{otherwise} \end{cases}$$

$$Y_{jkss'} = \begin{cases} 1, \text{if surgeon } s \text{ operates after surgeon } s' \text{ in shift } j \text{ and OR } k \\ 0, \text{otherwise} \end{cases}$$

$$Z_{jks} = \begin{cases} 1, \text{if surgeon } s \text{ is the first to operate in shift } j \text{ and OR } k \\ 0, \text{otherwise} \end{cases}$$

$$W_{jks} = \begin{cases} 1, \text{if surgeon } s \text{ is the last to operate in shift } j \text{ and OR } k \\ 0, \text{otherwise} \end{cases}$$

$$V_{jkk's} = \begin{cases} 1, \text{if surgeon } s \text{ operates in OR } k' \text{ after operated in OR } k \text{ in shift } j \\ 0, \text{otherwise} \end{cases}$$

$$\mu_{jks}^{start} = \text{starting time of surgeon } s \text{ in OR } k \text{ and shift } j$$

$$\mu_{jks}^{end} = \text{end time of surgeon } s \text{ in OR } k \text{ and shift } j$$

Throughout the exposition, i denotes a patient, j denotes a shift, which is a combination of a given day in the planning horizon and a working shift (morning or afternoon), k denotes an operating room and s denotes a surgeon. The weights α and β define the search directions and are normalized, i.e. $\alpha + \beta = 1$.

Regarding the decision variables, the binary variable X pertains to the patients and is used for selecting which patients are scheduled and assigned to the respective shifts and ORs, the binary variables Y, Z, W and V relate to the surgeons and are used for designating the sequence in which the surgeons work in a given shift and OR, and real variables μ^{start} and μ^{end} keep track of the start and end times of each surgeon in a given shift and OR. Herein, it is assumed that each surgeon can work at most one period of work in each shift and OR. This assumption favours the efficiency of the model.

Objective function

Expression (1) denotes the objective function which maximizes the number of scheduled surgeries and the average ORs' utilization rate. However, as these are competing goals, the preferences for the objectives are *a priori* declared as a weighted linear scalarizing function, used to aggregate the objectives into a single one. The value of each objective is normalized based on the maximum values both can take (γ and δ), to prevent the magnitude of each measure to bias the final value of the function, yielding a non-dimensional objective function value.

The value of the objective function must be minimized since the greater the number of scheduled patients and the average OR utilization rate are the lower the function value will be.

$$min\ F = \alpha \cdot \frac{\gamma - \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} X_{ijk}}{\gamma} + \beta \cdot \frac{\delta - \dfrac{\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} X_{ijk} \cdot d_i}{C}}{\delta} \tag{1}$$

Constraints

The constraints are grouped into three categories, the first related to patients, the second to surgeons and the latter to time periods. The inequality (2) prevents a patient from being scheduled more than once. The due date of a surgery is defined according to patient's priority and the rules that regulate the admissible waiting time. **(** porquê esta frase anterior aqui? **)** Inequality (3) expresses a constraint for the shifts capacity. The summation of all surgeries durations and cleaning times in a given shift must be lower than or equal to the capacity of the OR in that shift. These constraints also ensure that each specialty uses only the ORs available to it according to the master surgery schedule. The ORs with a capacity greater than zero are considered available. Note that the model works with only a single specialty.

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} \leq 1, \forall i \in I \tag{2}$$

$$\sum_{i\in I} X_{ijk} \cdot (d_i + ct) \leq c_{jk}, \forall j \in J, \forall k \in K \tag{3}$$

Expression (4) states that surgeries with a maximum scheduling time lower than the planning horizon must be scheduled. Expression (5) states that surgeries with a maximum waiting time lower than the planning horizon must be scheduled and inequality (6) states that the surgery day must be lower than the maximum waiting time. In the model the maximum scheduling times and maximum waiting times are defined in days relative to the beginning of the planning horizon. The absolute values are defined according to the patient's priority and waiting time rules and can be find in the waiting list manual [33]. The aforementioned constraints related specifically to the patients, as the other are related to the surgeons.

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} = 1, \forall i \in I_{maxsched} \tag{4}$$

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} = 1, \forall i \in I_{maxwait} \tag{5}$$

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} \cdot day_j \leq max_i, \forall i \in I_maxwait \tag{6}$$

Thus, the following sets of constraints aims to determine the sequence of surgeons working in each shift and OR. Expressions (7) and (8) define the surgeons who are the first and the last to operate in each shift and OR, respectively. Inequality (9) states that in a given shift a surgeon must **(** must ou can? **)** be the first in at most one OR, otherwise there would be an overlap between surgeries of that particular surgeon as all OR sessions start at the beginning of the shift. In addition, the inequality (10) ensures that in every shift and OR, a surgeon is either the first to operate or will operate after another surgeon in the sequence. Similarly, according to expression (11), a surgeon is either the last one or precedes another one. These two constraints ensure that a surgeon occurs only once in the sequence of each OR and shift, i.e. he will have a single bucket of consecutive work. Expression (12) is defined to avoid circular references in the sequence of surgeons. Finally, expression (13) is the flow equation, specifying the balance of the inflow and outflow of position for each of the surgeons. **(** a frase anterior estranha **)** It guarantees the consistency between expressions (10) and (11) ensuring that there is a link (via variables Y) between all the surgeons in the sequence.

$$\sum_{s\in S} Z_{jks} = 1, \forall j \in J, \forall k \in K \tag{7}$$

$$\sum_{s\in S} W_{jks} = 1, \forall j \in J, \forall k \in K \tag{8}$$

$$\sum_{k\in K} Z_{jks} \leq 1, \forall j \in J, \forall s \in S \tag{9}$$

$$Z_{jks} + \sum_{s'\in S} Y_{jkss'} \leq 1, \forall j \in J, \forall k \in K, \forall s \in S \tag{10}$$

$$W_{jks} + \sum_{s'\in S} Y_{jks's} \leq 1, \forall j \in J, \forall k \in K, \forall s \in S \tag{11}$$

$$Y_{jkss} = 0, \forall j \in J, \forall k \in K, \forall s \in S \tag{12}$$

$$Z_{jks} + \sum_{s'\in S} Y_{jkss'} = W_{jks} + \sum_{s'\in S} Y_{jks's}, \forall j \in J, \forall k \in K, \forall s \in S \tag{13}$$

The next set of constraints aims to assign the start and end times of each surgeon in each shift and OR according to the scheduled patients (determined by variable X) and the sequence of surgeons (determined by variables Y,Z,W,V). Expression (14) enforces that the starting time of the first surgeon will take place at the beginning of the shift. Similarly, constraint (15) sets the ending time of each surgeon on each shift and OR, taking into account the starting time, the duration of all the surgeries performed by him and the cleaning times between surgeries. It is important to define the surgeons' finishing time in order to enable assessing if a given surgeon can, afterwards, start a surgery in another

OR. Moreover, expression (16) declares that all end times must be within the capacity (in time units) of the OR in that particular shift.

$$\mu^{start}_{jks} \leq c_{jk} - (c_{jk} \cdot Z_{jks}), \forall j \in J, \forall k \in K, \forall s \in S \tag{14}$$

$$\mu^{start}_{jks} + \sum_{i \in I_s} X_{ijk} \cdot d_i + (\max\{1, \sum_{i \in I_s} X_{ijk}\} - 1) \cdot ct \leq \mu^{end}_{jks}, \forall j \in J, \forall k \in K, \forall s \in S \tag{15}$$

$$\mu^{end}_{jks} \leq c_{jk}, \forall j \in J, \forall k \in K, \forall s \in S \tag{16}$$

Constraint (17) is set to eliminate any subtours in the sequence, preventing the occurrence of two or more disconnected groups of surgeons, and guaranteeing that there is a link between all surgeons. The inequality implies that if a surgeon s comes after another surgeon s' then the end time of s' (the previous), denoted by $\mu^{end}_{jks'}$, must be lower than or equal to the start time of s (the next), denoted by μ^{start}_{jks}. In other words, if one surgeon comes after another, then the previous surgeon must end before the beginning of the next. **(** o conceito de link entre surgeons não é claro **)**

$$\mu^{end}_{jks'} + (Y_{jkss'} - 1) \cdot (C + ct) \leq \mu^{start}_{jks}, \forall j \in J, \forall k \in K, \forall s \in S, \forall s' \in S \tag{17}$$

The working times of a given surgeon in different ORs must be synchronized. The following constraints aim to avoid overlaps between working periods of a given surgeon in different ORs within the same shift. Expression (18) states that if a surgeon s works in OR k after having worked in OR k' in a given shift j, denoted by $V_{jkk's}$, then the surgeon's end time in OR k' (the previous) must be lower than the surgeon's start time in OR k (the next), denoted by μ^{start}_{jks}. In contrast, the inverse must also hold. According to expression (19), if a surgeon s does not work in OR k after having worked in OR k', denoted by $V_{jkk's}$, then the surgeon's start time in OR k', denoted by $\mu^{start}_{jk's}$, must be greater than or equal to the end of the surgeon's working period in OR k, denoted by μ^{end}_{jks}. The two constraints work together, the first validates the situation in which the surgeon works in a given OR after another and the second the situation in which he does not work, to determine the working periods in parallel ORs avoiding overlaps.

$$\mu^{end}_{jk's} + tt \leq \mu^{start}_{jks} + c_{jk} \cdot (1 - V_{jkk's}), \forall j \in J, \forall k \in K, \forall k' \in K, \forall s \in S \tag{18}$$

$$\mu^{end}_{jks} \leq \mu^{start}_{jk's} + c_{jk} \cdot V_{jkk's}, \forall j \in J, \forall k \in K, \forall k' \in K, \forall s \in S \tag{19}$$

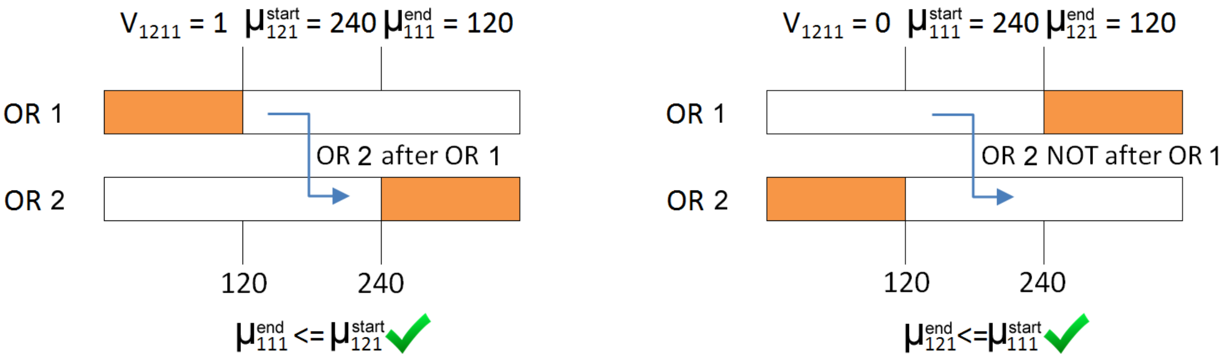
Figure 1 shows an illustrative example to support the description of the synchronization constraints. Let $J = \{1\}$ be the set of shifts, $K = \{1, 2\}$ be the set of ORs and $S = \{1\}$ be the set of surgeons. Figure 1(a) shows a case in which surgeon 1 operates in OR 2 after having operated in OR 1, therefore the surgeon's start time in OR 2 must be greater than the surgeon's end time in OR 1. Figure 1(b) shows the opposite, when surgeon 1 does not work in OR 2 after having worked in OR 1 the surgeon's start time in OR 1 must be greater than the surgeon's end time in OR 2.

The next constraints link the sequence of surgeons to the scheduled surgeries. Expression (20) states that if a patient i is scheduled in a shift j and OR k then the surgeon responsible for this operation, denoted by s_i, must be the first to operate or come after another surgeon in that shift and OR. In contrast, inequality (21) states that if a surgeon s does not have any scheduled patient in shift j and OR k then he must not appear in the sequence. In this expression, M denotes a big number, greater than the highest possible value for the summation of variables Z, W and Y for this particular surgeon.

$$X_{ijk} \leq \sum_{s' \in S} Y_{jks_is'} + Z_{jks}, \forall i \in I, \forall j \in J, \forall k \in K \tag{20}$$

$$Z_{jks} + W_{jks} + \sum_{s' \in S} (Y_{jkss'} + Y_{jks's}) \leq \sum_{i \in I_s} X_{ijk} \cdot M, \forall j \in J, \forall k \in K, \forall s \in S \tag{21}$$

The output of the model is the set of surgeries scheduled for each shift and OR, as well as the sequence of surgeons in each shift and OR and their respective start and end times. In order to determine the starting time of surgeries one must iterate over the sequence of surgeons. Algorithm 1 shows the two functions used in this process. The procedure starts by calling the function *generateSchedule* which iterates through all shifts and ORs and if the OR is available,



(a) Surgeon operates in OR 1 after OR 2

(b) Surgeon does not operate in OR 1 after OR 2

Figure 1: An illustrative example of the synchronization constraints

iterates through the set of surgeons until it finds the first in the sequence. The next step (line 8), is to print the list of patients of this surgeon, which consists in iterating through the list of patients and for each of the surgeon's patients scheduled for this particular shift and OR, print its starting time. Note that the starting times of patients are relative to the beginning of the surgeon working period, the starting time of the first patient is equal to μ_{jks}^{start} and the starting time of the following is equal to the duration of the previous plus the OR cleaning time. Next (line 16), the procedure verifies if the current surgeon is the last to operate and, if that is the case, it returns to the calling function, otherwise, it finds the next surgeon in the sequence and calls function *ListPatients* again, recursively (line 21).

Figure 2 shows a sample schedule generated by the proposed MIP model. It is a weekly schedule for the Neurosurgery specialty. This schedule has 5 days, 2 shifts per day and 2 ORs. Both ORs are closed on Saturday afternoon. In the figure, each box represents a surgery, the different colors **(** vamos usar cor? **)** represent different surgeons, and the numbers inside each box mean the surgeon Id (between parenthesis) and start and end time of each surgery **(** em minutos? **)**. The times are relative to the start of the shift. This schedule has 56 scheduled surgeries and 82.4% of average OR utilization rate taking into account an OR cleaning time of 17 minutes after each surgery and no surgeon turnover time. In this case, the turnover time, which is the required time for a surgeon to switch between ORs, is included in surgery duration. It is worth noting that there are situations in which a surgeon is able to start a new surgery in a different OR before starting the next surgery in the same OR, therefore saving time in which the surgeon would otherwise be idle. These situations are signed with a circle.

4.2. Heuristic Solution Approach: Biased Random Key Genetic Algorithm

4.2.1. General Genetic Algorithm Description

In this section, a heuristic approach based on the biased random-key genetic algorithm [6] is proposed as an alternative to the exact approach proposed in the previous section. Genetic algorithms [34, 35] are part of a group of nature inspired algorithms based on the concept of natural selection, or survival of the fittest, that are used to find near-optimal solutions for optimization problems. This algorithms are referred as population based because a set of individuals evolves over a number of generations. Each individual in a generation represents a solution for the optimization problem, in our case a surgery schedule. Moreover, each individual has an associated chromosome that encodes the corresponding solution. A chromosomes is a string of genes and the value in each gene is referred to as an allele. In general, alleles can take binary or real values.

In random-key genetic algorithms (RKGAs) [28] each chromosome is encoded as a vector of random-keys in which each allele is a random number between 0.0 and 1.0. Figure 3 shows a sample RKGA chromosome which indirectly represents a surgery schedule. In this representation, each allele is a random number corresponding to a patient in the waiting list. A decoding procedure, or simply a decoder, is required to translate a chromosome into a solution in order to compute the associated performance metrics. In our case, the performance metrics are the number of scheduled surgeries and the average OR utilization rate. It is worth mentioning that the decoder efficiency plays an important role in the overall algorithm performance as it consumes most of the computational time.

Algorithm 1: Algorithm for generating a schedule from a solution of the MIP

function GenerateSchedule() **begin**
    for all j *in* J **do**
        for all k *in* K **do**
            if $c_{jk} > 0$ **then**
                for all s *in* S **do**
                    if $Z_{jks} = 1$ **then**
                        ListPatients(j,k,s)
                        break

function ListPatients(j, k, s)
begin
    startTime $\leftarrow \mu_{jks}^{start}$
    for all i *in* I **do**
        if $s_i = s$ **and** $X_{ijk} = 1$ **then**
            PrintPatient(i,j,k,startTime)
            startTime=startTime+d_i+ct
    if $W_{jks'} = 1$ **then**
        return
    else
        for all s' *in* S **do**
            if $Y_{jks's} = 1$ **then**
                ListPatients(j,k,s')

Figure 2: Sample Neurosurgery schedule generated by the MIP model

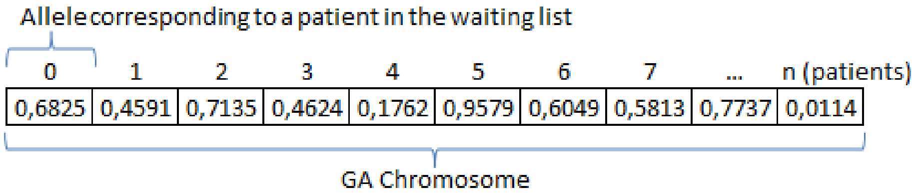


Figure 3: Sample chromosome indirectly representing a surgery schedule

As population based heuristics, GAs evolve populations of solutions through means of recombination and mutation. The recombination process consists in selecting two parents from the population and copying sequences of genes from both of them into a new individual. This procedure is called crossover. In particular, the proposed GA uses a parametrized uniform crossover [36]. **(** não se devia explicar o que é ? **)** On the other hand, mutation aims to introduce diversity into the population in order to enable the method to escape entrapments in local optima. In the case of RKGAs the mutation is achieved by generating completely new individuals, called mutants, and introducing them into the populations.

RKGAs use an elitist strategy to evolve populations of solutions over generations. In this strategy, after decoding the individuals and computing their fitness values (the associated performance metrics), the best individuals are labelled as ELITE. These individuals remain in the population for the next generation in order to preserve good genes. The biased random-key genetic algorithm differs from standard RKGA in the way individuals are selected for the recombination process. In a BRKGA, instead of randomly selecting two individuals from the entire population, each new individual is generated by combining one individual from the ELITE group and one from the NON-ELITE part**, or from the entire population (** não é claro. da non-elite ou do total?**)**. This increases the probability of good individuals passing their characteristics to future generations.

BRKGAs are based on a generic metaheuristic framework. Figure 4 shows an overview of the BRKGA optimization process. Note that this framework makes a clear distinction between the problem dependent and independent parts of the process. The problem independent part includes the initialization, selection, recombination and mutation procedures, which are similar among other optimization problems. The problem dependent part encompasses the decoding procedure. This procedure is crucial for the algorithm performance, since it consumes a large portion of the overall computational time. In this paper, we propose a procedure to decode a vector of random-keys into a valid surgery schedule based on lists of available resource time slots. **This procedure is able to generate good schedules both in terms of number of scheduled surgeries as in terms of average OR utilization rate. (** which procedure? the decoding procedure? of the full BRKGA? **)**

Figures 5(a) and 5(b) show a conceptual view of the decoding procedure. The idea is to keep track of the periods in which the resources are available. The example considers two ORs (A and B), two surgeons (1 and 2) and a one week planning horizon. In Figure 5(a) the highlighted areas represent the availability of resources over the planning horizon. It is worth noting that OR B is not available on Thursday and Friday, as well as the two surgeons have distinct available periods. These patterns are directly mapped from the master surgery schedule, which denotes the ORs assigned to each specialty over the week, and from the staff roster, which shows staff's working shifts. Moreover, Figure 5(b) shows the same availability periods represented in terms of data structures; the numbers within the cells represent the start and end time of each period in minutes. In this case, lists of time periods (start and end time) represented in minutes from the beginning of the planning horizon until the end. The decoder works using the chromosome of random-keys to determine the scheduling sequence and the lists of available periods to find a time in which both surgeon and OR are free. The following paragraphs describe the procedure in detail.

The following steps are used for decoding a vector of random-keys into a valid surgery schedule. The sequence of steps is illustrated in Figure 6.

1. **Initialize available times** - Creates the data structures to support the procedure. It creates the lists of available periods, as illustrated in Figure 5(b), for each OR and surgeon based on the pre-defined OR capacity and surgeon availability;
2. **Sort patients by random-keys** - Sorts the chromosome by ascending order of the random-key in each gene.

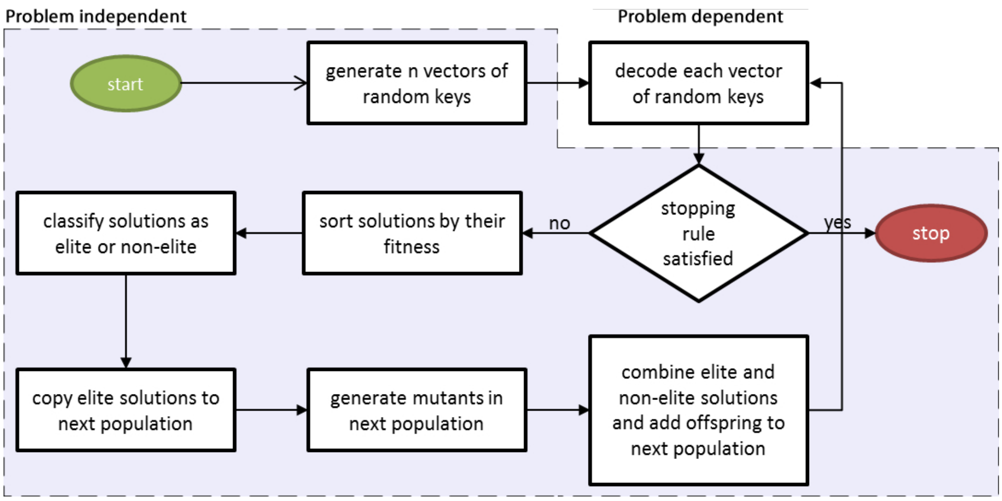
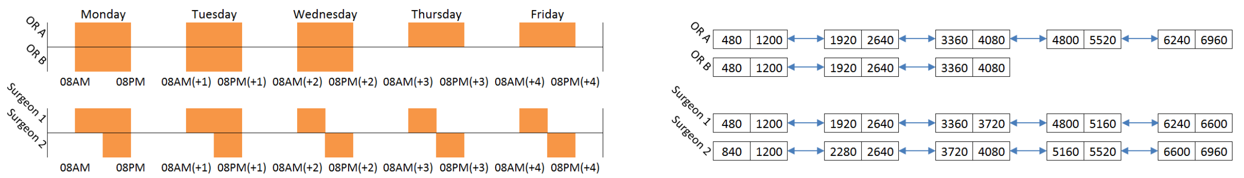


Figure 4: Flowchart of the BRKGA framework



(a) Conceptual view of resource available periods (b) Lists used to implement the concept of resource available periods

Figure 5: Resource available periods from two different perspectives: conceptual and implementation

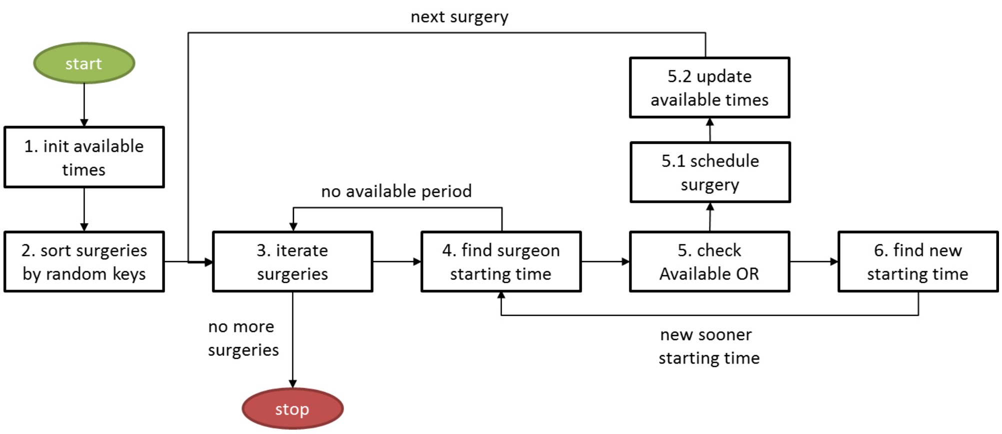


Figure 6: Flowchart of the decoding procedure

The resulting sorted vector determines the sequence in which patients are evaluated in the next step, e.g. patients with lower random-keys are evaluated first and patients with higher are evaluated last;

3. **Iterate patients** - Evaluates each patient according to the sequence encoded in the chromosome. If there are no more patients to evaluate, then the procedure ends.
4. **Find surgeon's starting time** - Searches the list of available periods of the surgeon responsible for the current surgery and finds the first period that fits the current surgery duration plus cleaning time. If an available period is found, goes to step (5) to search for an OR, otherwise returns to the step (3) to evaluate the next patient, because the current patient will not be scheduled due to a lack of time of the responsible surgeon;
5. **Check available OR** - Iterates through the list of available ORs and searches for a time period that fits the current surgery duration plus cleaning time with the surgery starting exactly at the starting time of the surgeon available period defined in the previous step. If an available time is found, the procedure goes to step (5.1) to schedule the surgery, otherwise goes to step (6) to search for an available time in the future;

   5.1. **Schedule surgery** - Updates the output surgery schedule with the current surgery, day, shift, OR and starting time;

   5.2. **Update available time** - Updates the list of available periods of the surgeon responsible for the current patient and the selected OR. In the surgeon case, the surgery duration plus the turnover time must be subtracted from the time period in which the surgery was scheduled, meaning the surgeon is not available at this time. In the case of the OR, it is the surgery duration plus the cleaning time, meaning the selected OR is not available from the beginning of the surgery until the end of the cleaning time.
6. **Find new starting time** - Finds the first OR time period that fits the surgery duration from the current surgeon available time until the end of the planning horizon and returns to step (4) to find a new surgeon available time from this point onwards.

The decoding procedure is able to translate every chromosome into a near feasible solution. The restrictions related to patients' priority and waiting time rules as well as surgeons' workload are not guaranteed. In order to tackle these issues we calculate all the metrics associated with a schedule and penalize the violations in the fitness function. Once we have a schedule as a result of the decoding process we compute the following metrics: number of scheduled surgeries, average OR occupancy rate, number of violations of surgery due date, number of violations of maximum scheduling date, total deviation from the limit number of working shifts per week. The final objective function is similar to the one used for the models in this Section, only with the additional terms for the waiting list's violations and surgeon's workload. The decoder is not able to guarantee these problem constraints are respected, therefore, we address them in the objective function. The surgery's due date is defined as a function of the patient's maximum waiting time (time between the day a patient enters the waiting list and the day the surgery is performed) according to the Portuguese legislation and the scheduling date is the maximum time a patient can be in the waiting list without be scheduled for a surgery (time between the day a patient enters the waiting list and the day a surgery date is assigned) [33].

4.2.2. Local Search and Chromosome Correction

A local search procedure is performed after the decoding procedure to further enhance the quality of solutions. It uses the decoder's supporting data structures to find available time periods in the ORs and to try to swap the surgeries scheduled immediately before and after such available periods by unscheduled surgeries with a larger size. For each available time period, the procedure evaluates all possible swaps (changing one surgery for another), ranks them by the benefit they provide (improvement of the objective function) and implement the swap that improves the objective function the most. The computational experiments show that these small changes are effective in enhancing the quality of solutions. They enable the algorithm to quickly improve the quality of the solutions in particular cases, what would require several generations in the standard evolution process.

After the local search, the chromosome associated with each solution in the population is corrected to represent the actual order in which surgeries are scheduled in the solution. Hence, it is not necessary to keep track of the local changes for the next time the chromosome has to be evaluated.

5. Computational Experiments

5.1. Test Instances

The test instances use in the experimental evaluation are based on real data provided by a large hospital. In the hospital, there are 10 different surgical specialties: ascular surgery, oral and maxillofacial surgery, neurosurgery, ophthalmology, orthopaedics, urology, otorhinolaryngology, general surgery 1, general surgery 2, general surgery 3. In total, two different sets of six instances each for each specialty were generated, summing up 120 instances. The two sets differ in the total number of available ORs. The first set contains instances having the same number of ORs as there are in use at the hospital (regular size instances) and the second set contains twice the number of ORs, in order to simulate a larger size hospital or an expansion of capacity at the hospital (large size instances). Within each set, the instances differ by the number of patients and the length of the planning horizon.

Algorithm 2 illustrates the procedure used for generating the test instances. The sets of parameters used in the procedure are the following: the specialty identifier $SP = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$; the instance size $IS = \{regular, large\}$; the planning horizon (in weeks) $H = \{1, 2\}$; and the capacity multipliers $CM = \{2, 4, 6\}$. The capacity multiplier is used to set the number of patients in the instance. The algorithm works as follows. First, the procedure iterates through the set of specialties and gets the time blocks associated with each specialty (a time block denotes a working shift in an OR). This initial set of time blocks will correspond to a standard instance (regular size and 1 week planning horizon). Second, the algorithm iterates through the set of instance types and, in case of large size instances, duplicates the number of time blocks. Third, it iterates through the set of planning horizons and, in case of more than one week, generates new time blocks for the additional weeks. Next, it iterates through the set of capacity multipliers and multiplies each value by the total capacity of the current time blocks, setting an auxiliary variable to represent the maximum value for the sum of surgery durations in this instance. Finally, it iterates through the set of patients in the waiting list, adding them to the current instance and subtracting the auxiliary variable. Line 19 shows that when the value of the auxiliary variable reaches zero, the procedure stops and the current instance is ready.

Table 3 shows the detailed characteristics of the generated regular instances and Table 4 of the large size instances. In both cases the estimated duration of a surgery is considered deterministic and set as the median value of the historical data. **The last two columns (** seems a mistake since the last two columns are not that. **)** in the tables show the number of patients in each instance whose maximum scheduling time or maximum total waiting time were reached or are within the planning horizon. It is worth mentioning that some of the instances were duplicated due to the lack of patients in the waiting list of each specialty and were excluded from the final analysis, resulting in 96 different instances.

5.2. Implementation Details

The methods were coded in C++ and compiled using g++ (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4) with "-O3" and "-fopenmp" compiler options. The exact models use the IBM ILOG CPLEX Optimization Studio V12.4 libraries

Algorithm 2: Procedure used for generating the testing instances

```
function GenerateInstances(SP,IS,H,CM)
begin
    for sp ∈ SP do /* for each specialty */
        SP_tb ⟵ GetTimeBlocks(sp)
        for is ∈ IS do /* for each set of instance types */
            if is == large then
                SP_tb ⟵ DuplicateTimeBlocks(SP_tb, is)
            for h ∈ H do /* for each planning horizon */
                if h > 1 then
                    SP_tb ⟵ GenerateNewWeeks(SP_tb, h)
                for cm ∈ CM do /* for each capacity multiplier */
                    totalCapacity ⟵ GetTotalCapacity(SP_tb) · cm
                    SP_i ⟵ GetPatientsSortedByPriorityAndWaitingTime(sp)
                    for i ∈ SP_i do /* for each patient in the waiting list */
                        if totalCapacity > 0 then
                            AddPatientToInstance(i)
                            totalCapacity ⟵ totalCapacity − (d_i + ct)
                        else
                            break
```

through the Concert Technology. Further, the GA is based on the application programming interface (API) for the algorithmic framework of biased random-key genetic algorithms, brkgaAPI, presented by Toso and Resende [37]. All computation experiments are performed on machines running Scientific Linux 6 (SL6) distribution and equipped with Intel Xeon Processor E5-2650 CPUs (2 GHz). The number of parallel threads is limited to 8 and the amount of RAM is limited to 16 GB. This configuration was chosen to represent a standard server available in a hospital by the end of 2014. Additional computational experiments showed that the MIP benefits from more memory. For instance, comparing the MIP limited to 8GB of RAM with the MIP limited to 16GB of RAM, the latter obtained better results in 38% of the instances, with a relative change of 3.9%.

5.3. Configuration of Parameters

5.3.1. General Parameters

A time limit of 1 hour was set for each computational experiment. The GA is let to restart at most 30 times and each evolution let run for 2 min. In addition, the exact models may stop before the time limit if an optimal solution is found or if the memory size limit is reached.

For all experiments the cleaning time is set to 17 min and the surgeon turnover time is set to 0 min. Also, the availability of operating rooms respects the original hospital master surgery schedule and the surgeons are set to be available at any time. For the discrete model, the time within each shift is discretized in intervals of 15 min, which is the most common value according to the literature review presented in Section 2. Finally, the constraints concerning patients' priority, the waiting time rules and the surgeons workload are disabled. This configuration makes the problem harder to solve as it expands the feasible region, helping to evidence the differences among the alternative solution methods.

5.3.2. Genetic Algorithm Parameters

The values for the BRKGA parameters were defined based on previous studies with the algorithm and on an extensive sensitivity analysis. First, sets of values for each parameter were defined based on the recommended values found in previous studies, such as Gonçalves et al. [29], Gonçalves et al. [38], Toso and Resende [37] and Gonçalves and Resende [39]. Table 5 shows the pre-defined values for each parameter. Second, every combination of these

Table 3: Regular size instances: problem instances generated based on real data

Regular Size Instances									
Specialty	Planning Horizon	Capacity Multiplier	No. Patients	No. Time Blocks	No. Surgeons	Avg. Duration	Max. Schedule Date	Max. Surgery Date	Instance Group
1. Vascular surgery									
1.1	1	2	91	9	15	53.74	74	22	1
1.2	1	4	214	9	15	43.52	77	22	1
1.3	1	6	346	9	15	39.22	77	22	3
1.4	2	2	214	18	15	43.52	102	23	1
1.5	2	4	473	18	15	37.79	102	23	3
1.6	2	6	721	18	17	36.95	102	23	3
2. Oral and maxillofacial surgery									
2.1	1	2	32	3	11	47.69	1	0	1
2.2	1	4	75	3	12	38.17	1	0	1
2.3	1	6	115	3	13	36.78	1	0	1
2.4	2	2	75	6	12	38.17	2	0	1
2.5	2	4	165	6	14	33.19	2	0	1
2.6	2	6	179	6	14	33.83	2	0	1
3. Neurosurgery									
3.1	1	2	96	18	10	118.91	39	20	1
3.2	1	4	179	18	15	124.34	39	20	1
3.3	1	6	268	18	16	123.52	39	20	1
3.4	2	2	179	36	15	124.34	46	24	2
3.5	2	4	286	36	16	122.67	46	24	4
4. Ophthalmology									
4.1	1	2	299	24	34	26.59	1	0	3
4.4	2	2	299	48	34	26.59	1	1	4
5. Orthopaedics									
5.1	1	2	143	22	27	90.76	134	133	1
5.2	1	4	282	22	29	92.34	273	272	3
5.3	1	6	416	22	33	94.19	407	380	3
5.4	2	2	282	44	29	92.34	274	272	4
5.5	2	4	558	44	33	93.45	487	393	4
5.6	2	6	861	44	35	90.38	487	393	4
6. Urology									
6.1	1	2	93	12	20	73.37	43	17	1
6.2	1	4	206	12	20	63.73	43	17	1
6.3	1	6	287	12	21	62.86	43	17	3
6.4	2	2	206	24	20	63.73	50	17	1
6.5	2	4	287	24	21	62.86	50	17	3
7. Otolaryngology									
7.1	1	2	87	9	14	56.22	16	1	1
7.2	1	4	170	9	16	57.89	16	1	1
7.3	1	6	253	9	16	58.36	16	1	1
7.4	2	2	170	18	16	57.89	20	2	1
7.5	2	4	335	18	16	58.80	20	2	3
7.6	2	6	448	18	16	58.20	20	2	3
8. General surgery 1									
8.1	1	2	59	9	9	91.17	48	46	1
8.2	1	4	140	9	11	77.19	129	127	1
8.3	1	6	204	9	13	77.10	174	136	1
8.4	2	2	140	18	11	77.19	131	127	1
8.5	2	4	275	18	15	75.91	178	141	1
8.6	2	6	329	18	16	77.34	178	141	3
9. General surgery 2									
9.1	1	2	65	8	8	70.51	31	17	1
9.2	1	4	129	8	8	69.69	31	17	1
9.3	1	6	192	8	8	70.33	31	17	1
9.4	2	2	129	16	8	69.69	41	21	1
9.5	2	4	214	16	8	69.12	41	21	1
10. General surgery 3									
10.1	1	2	64	7	11	66.31	37	19	1
10.2	1	4	123	7	11	63.10	37	19	1
10.3	1	6	163	7	12	63.17	37	19	1
10.4	2	2	123	14	11	63.10	40	24	1
10.5	2	4	163	14	12	63.17	40	24	1

Table 4: Large size instances: problem instances generated based on real data

Large Size Instances									
Specialty	Planning Horizon	Capacity Multiplier	Patients	Time Blocks	No. Surgeons	Avg. Duration	Max. Schedule Date	Max. Surgery Date	Instance Group
1. Vascular surgery									
1.7	1	2	205	18	15	44.03	69	15	1
1.8	1	4	457	18	15	37.67	69	15	3
1.9	1	6	691	18	15	37.18	69	15	3
1.10	2	2	457	36	15	37.67	69	15	4
1.11	2	4	767	36	18	36.84	69	15	4
2. Oral and maxillofacial surgery									
2.7	1	2	71	6	12	39.00	1	0	1
2.8	1	4	156	6	14	33.81	1	0	1
2.9	1	6	179	6	14	33.83	1	0	1
2.10	2	2	156	12	14	33.81	1	0	1
2.11	2	4	179	12	14	33.83	1	0	1
3. Neurosurgery									
3.7	1	2	178	36	15	121.92	35	17	2
3.8	1	4	286	36	16	122.67	35	17	4
3.10	2	2	286	72	16	122.67	35	17	4
4. Ophthalmology									
4.7	1	2	299	48	34	26.59	0	0	4
4.10	2	2	299	96	34	26.59	0	0	4
5. Orthopaedics									
5.7	1	2	277	44	29	92.35	268	267	2
5.8	1	4	551	44	33	92.81	453	367	4
5.9	1	6	847	44	35	90.17	453	367	4
5.10	2	2	551	88	33	92.81	453	367	4
5.11	2	4	1153	88	37	87.95	453	367	4
5.12	2	6	1281	88	37	86.59	453	367	4
6. Urology									
6.7	1	2	202	24	20	63.69	34	15	1
6.8	1	4	287	24	21	62.86	34	15	3
6.10	2	2	287	48	21	62.86	34	15	4
7. Otolaryngology									
7.7	1	2	165	18	16	58.13	13	1	1
7.8	1	4	327	18	16	58.73	13	1	3
7.9	1	6	448	18	16	58.20	13	1	3
7.10	2	2	327	36	16	58.73	13	1	4
7.11	2	4	448	36	16	58.20	13	1	4
8. General surgery 1									
8.7	1	2	138	18	11	73.86	126	125	1
8.8	1	4	268	18	15	76.41	168	130	1
8.9	1	6	329	18	16	77.34	168	130	3
8.10	2	2	268	36	15	76.41	168	130	2
8.11	2	4	329	36	16	77.34	168	130	4
9. General surgery 2									
9.7	1	2	126	16	8	70.01	30	13	1
9.8	1	4	214	16	8	69.12	30	13	1
9.10	2	2	214	32	8	69.12	30	13	2
10. General surgery 3									
10.7	1	2	120	14	11	63.32	32	17	1
10.8	1	4	163	14	12	63.17	32	17	1
10.10	2	2	163	28	12	63.17	32	17	2

Table 5: Ranges of each tested GA parameter

Parameter	Tested Sets
Population Size Multiplier	10, 20, 30, 40
Percentage of Elite Population	0.1, 0.15, 0.2, 0.25
Percentage of Mutants	0.15, 0.2, 0.25, 0.3
Probability of Crossover	0.7, 0.75, 0.8, 0.85
No. of Independent Populations	1, 2
No. of Generations until Exchange Best Individuals	50, 100
No. of Generations without improving until Restart	100, 200

Table 6: Characteristics of instance groups and best combination of values

Instance Group	Percentage of Instances	No. of Patients	No. of Time Blocks	Best Combination of Parameter Values
1	46%	≤ 283	≤ 27	10, 0.25, 0.15, 0.7, 1, 100, 100
2	9%	≤ 283	> 27	20, 0.1, 0.3, 0.8, 2, 50, 200
3	17%	> 283	≤ 27	40 ,0.15, 0.15, 0.85, 1, 100, 200
4	28%	> 283	> 27	10, 0.2, 0.2, 0.85, 2, 100, 200

values was tested on four pilot instances. These instances represent four different groups, based on the number of patients and on the number of time blocks. The criteria used for distinguishing the groups were the average number of patients and the average number of time blocks.

Table 6 shows the characteristics of these four different instance groups alongside with the best combination of parameter values for each group of instances. The last column lists the parameter values in the order they appear in Table 5. The best configuration for each instance size was the one that, among all combinations, minimized the objective function value and the running time. It is worth noting that the population size is defined as a multiple of the total number of patients. This approach showed good results in the aforementioned studies enabling the algorithm to adjust according to the instance size. The GA computation time can be adjusted using the number of restarts parameter. A number of restarts higher than 1 make the algorithm restart with a different random seed after a given stopping criteria (i.e. number of generations without improvement) is reached. In the computational tests, the value 30 for the number of restarts parameter produced the best results.

5.4. Experimental Results

5.4.1. Continuous MIP model vs. Discrete IP model

Table 7 compares for the regular size instances the performance of the proposed MIP model using a continuous representation of time with an IP model using a discrete representation of time. The discrete IP model was able to find an optimal solution for 56% of the instances with an average gap of 0.5% compared to a 12% of optimal solutions and a 4.5% average gap obtained by the continuous model. However, the analysis of the quality of the solutions reveals that the continuous model produces solutions with a lower objective function value (better) in all the cases.

On average, the value of the objective function for the MIP model are 53% lower than the corresponding solutions for the IP model.

The last column in Table 7 gives the relative change **(** deviation? alterar tabela? **)** obtained by dividing the difference between the objective function values of the continuous model and the discrete model by the objective function values of the discrete model, used as reference. Even optimal solutions of the discrete model reveal inferior quality when compared to the corresponding solution given by the continuous model, showing that in fact the discrete model is just an approximation of the real problem. It is worth nothing that the quality of solutions increases with the number of patients in each instance. This was expected because the optimization procedures have more options to find better solutions.

Table 8 gives a similar analysis for the large instances. In these instances the IP model was able to find an optimal solution in 35% of the cases, with an average gap of 1.8%. In comparison, the MIP model found an optimal solution in 5% of the cases, with an average gap of 11%, which is higher than the 4.5% gap obtained for the regular size instances.

Regular Size Instances														
	IP Model - discrete time representation						MIP Model - continuous time representation							
Instance	Objective Function Value	N. of Scheduled Surgeries	Average OR Occupancy Rate (%)	Status	Gap (%)	Running Time (s)	Objective Function Value	N. of Scheduled Surgeries	Average OR Occupancy Rate (%)	Status	Gap (%)	Running Time (s)	Continuous better than discrete?	Relative Change (%)
1. Vascular surgery														
1.1	0.38814	35	64.0	Feasible	0.4	MAX	0.19077	61	64.6	Feasible	3.6	MAX	Yes	-50.8
1.2	0.38533	35	64.5	Feasible	0.4	MAX	0.16672	65	63.5	Feasible	6.7	2528	Yes	-56.7
1.3	0.38304	35	64.9	Feasible	0.3	MAX	0.16443	65	63.9	Feasible	5.4	MAX	Yes	-57.1
1.4	0.38290	57	69.7	Optimal	0.0	258	0.15572	126	63.4	Feasible	4.4	MAX	Yes	-59.3
1.5	0.38107	57	70.0	Optimal	0.0	301	0.15435	126	63.7	Feasible	13.6	1387	Yes	-59.5
1.6	0.37537	57	71.0	Optimal	0.0	3264	0.15279	126	63.9	Feasible	12.8	1571	Yes	-59.3
2. Oral and maxillofacial surgery														
2.1	0.52517	11	61.0	Optimal	0.0	0	0.33127	22	61.9	Optimal	0.0	1	Yes	-36.9
2.2	0.51288	13	56.3	Optimal	0.0	0	0.30268	25	57.0	Optimal	0.0	6	Yes	-41.0
2.3	0.51216	14	52.7	Optimal	0.0	0	0.28425	27	53.5	Optimal	0.0	2	Yes	-44.5
2.4	0.47219	22	61.9	Optimal	0.0	0	0.24813	49	57.9	Optimal	0.0	58	Yes	-47.5
2.5	0.46177	26	57.0	Optimal	0.0	5	0.22513	54	53.6	Feasible	2.0	2559	Yes	-51.2
2.6	0.46070	26	57.2	Optimal	0.0	19	0.22513	54	53.6	Feasible	1.8	1791	Yes	-51.1
3. Neurosurgery														
3.1	0.29099	45	72.3	Feasible	1.4	MAX	0.13730	58	83.5	Feasible	6.5	1805	Yes	-52.8
3.2	0.24645	52	71.0	Feasible	1.4	MAX	0.10848	63	81.9	Feasible	11.6	1340	Yes	-56.0
3.3	0.23277	55	69.4	Optimal	0.0	889	0.09471	65	81.8	Feasible	9.8	1399	Yes	-59.3
3.4	0.25809	88	73.2	Feasible	2.9	MAX	0.09775	114	83.7	Feasible	11.8	MAX	Yes	-62.1
3.5	0.23524	93	73.7	Feasible	5.1	MAX	0.07831	120	82.8	Feasible	18.6	MAX	Yes	-66.7
4. Ophthalmology														
4.1	0.30701	126	51.9	Optimal	0.0	383	0.07795	206	57.5	Feasible	11.4	MAX	Yes	-74.6
4.4	0.10772	262	43.5	Feasible	4.3	MAX	0.00000	299	47.8	Optimal	0.0	551	Yes	-100.0
5. Orthopaedics														
5.1	0.43150	69	69.3	Optimal	0.0	148	0.31872	89	79.5	Feasible	1.5	MAX	Yes	-26.1
5.2	0.42051	87	61.3	Optimal	0.0	184	0.26792	117	73.5	Feasible	1.0	MAX	Yes	-36.3
5.3	0.41547	89	61.2	Optimal	0.0	387	0.24855	128	71.1	Feasible	1.1	MAX	Yes	-40.2
5.4	0.40204	145	67.8	Feasible	0.0	MAX	0.29367	177	78.3	Feasible	5.3	MAX	Yes	-27.0
5.5	0.39059	177	60.3	Optimal	0.0	2185	0.25298	221	72.7	Feasible	9.8	MAX	Yes	-35.2
5.6	0.37826	195	57.2	Feasible	0.3	MAX	0.22763	261	65.4	Feasible	18.4	MAX	Yes	-39.8
6. Urology														
6.1	0.40588	47	61.6	Optimal	0.0	8	0.25025	64	72.7	Feasible	2.0	1393	Yes	-38.3
6.2	0.37313	62	51.7	Optimal	0.0	25	0.18908	81	66.0	Feasible	2.8	MAX	Yes	-49.3
6.3	0.36107	65	50.7	Feasible	0.2	MAX	0.16863	87	63.4	Feasible	0.7	2461	Yes	-53.3
6.4	0.33289	109	55.6	Feasible	0.0	MAX	0.14872	140	70.4	Feasible	1.7	MAX	Yes	-55.3
6.5	0.31485	117	54.2	Feasible	0.0	MAX	0.12318	152	68.0	Feasible	3.2	MAX	Yes	-60.9
7. Otolaryngology														
7.1	0.39939	33	65.0	Feasible	0.0	MAX	0.25039	47	73.9	Feasible	2.1	705	Yes	-37.3
7.2	0.39892	32	66.3	Optimal	0.0	105	0.23113	52	71.1	Feasible	1.1	846	Yes	-42.1
7.3	0.39837	32	66.4	Optimal	0.0	1965	0.20610	58	68.1	Feasible	1.9	731	Yes	-48.3
7.4	0.33700	72	61.0	Optimal	0.0	4	0.17629	91	74.7	Feasible	5.4	1233	Yes	-47.7
7.5	0.33672	72	61.1	Optimal	0.0	9	0.13880	105	71.0	Feasible	2.1	769	Yes	-58.8
7.6	0.33510	72	61.3	Optimal	0.0	22	0.10984	116	68.0	Feasible	3.9	MAX	Yes	-67.2
8. General surgery 1														
8.1	0.44153	27	68.8	Optimal	0.0	34	0.29224	42	76.4	Feasible	1.2	1552	Yes	-33.8
8.2	0.41806	35	62.2	Optimal	0.0	13	0.20228	62	66.0	Feasible	0.5	MAX	Yes	-51.6
8.3	0.40811	36	62.7	Feasible	0.0	MAX	0.18951	65	64.3	Feasible	1.4	2103	Yes	-53.6
8.4	0.42140	75	55.1	Feasible	0.0	MAX	0.23319	100	72.5	Feasible	0.5	814	Yes	-44.7
8.5	0.39923	70	63.0	Optimal	0.0	533	0.17410	124	66.1	Feasible	1.1	1294	Yes	-56.4
8.6	0.39288	73	62.0	Optimal	0.0	103	0.16420	128	65.0	Feasible	1.4	1394	Yes	-58.2
9. General surgery 2														
9.1	0.37361	27	67.1	Feasible	0.2	MAX	0.19961	40	75.1	Feasible	0.9	2312	Yes	-46.6
9.2	0.34518	32	62.8	Optimal	0.0	2	0.17277	44	72.4	Feasible	0.7	857	Yes	-49.9
9.3	0.33784	33	62.3	Optimal	0.0	8	0.17162	44	72.6	Feasible	6.7	1465	Yes	-49.2
9.4	0.31254	55	67.1	Feasible	0.0	MAX	0.13241	79	75.3	Feasible	3.7	954	Yes	-57.6
9.5	0.28666	66	60.9	Feasible	0.0	MAX	0.09825	88	72.5	Feasible	5.9	1076	Yes	-65.7
10. General surgery 3														
10.1	0.32482	27	59.3	Optimal	0.0	392	0.15395	33	76.1	Feasible	5.1	2707	Yes	-52.6
10.2	0.29064	30	58.7	Feasible	0.3	MAX	0.11723	37	73.8	Optimal	0.0	43	Yes	-59.7
10.3	0.28811	31	57.0	Feasible	0.2	MAX	0.10030	39	72.4	Feasible	1.5	MAX	Yes	-65.2
10.4	0.25573	54	60.1	Feasible	0.2	MAX	0.07230	66	76.4	Feasible	12.0	849	Yes	-71.7
10.5	0.23182	57	60.6	Feasible	0.2	MAX	0.05246	70	75.1	Feasible	5.0	2984	Yes	-77.4

Table 7: Regular Instances: Comparison of LP models using discrete and continuous representation of time - Best solutions considering the N. of Scheduled Surgeries and Average OR Occupancy Rate

Large Size Instances														
	IP Model - discrete time representation						MIP Model - continuous time representation							
Instance	Objective Function Value	N. of Scheduled Surgeries	Average OR Occupancy Rate (%)	Status	Gap (%)	Running Time (s)	Objective Function Value	N. of Scheduled Surgeries	Average OR Occupancy Rate (%)	Status	Gap (%)	Running Time (s)	Continuous better than discrete?	Relative Change (%)
1. Vascular surgery														
1.7	0.38290	57	69.7	Optimal	0.0	51	0.15572	126	63.4	Feasible	5.2	MAX	Yes	-59.3
1.8	0.38107	57	70.0	Optimal	0.0	3521	0.15279	126	63.9	Feasible	12.7	2072	Yes	-59.9
1.9	0.37537	57	71.0	Feasible	0.0	MAX	0.15435	126	63.7	Feasible	13.6	1068	Yes	-58.9
1.10	0.36373	138	56.1	Feasible	1.7	MAX	0.12122	241	62.7	Feasible	30.9	MAX	Yes	-66.7
1.11	0.34830	126	62.0	Feasible	0.3	MAX	0.10740	244	63.9	Feasible	29.0	MAX	Yes	-69.2
2. Oral and maxillofacial surgery														
2.7	0.47219	22	61.9	Optimal	0.0	0	0.24813	49	57.9	Feasible	0.1	MAX	Yes	-47.5
2.8	0.46177	26	57.0	Optimal	0.0	7	0.22513	54	53.6	Feasible	1.8	MAX	Yes	-51.2
2.9	0.46070	26	57.2	Optimal	0.0	9	0.22513	54	53.6	Feasible	1.9	2141	Yes	-51.1
2.10	0.41248	44	61.7	Feasible	0.2	MAX	0.17225	104	55.3	Feasible	0.4	MAX	Yes	-58.2
2.11	0.41154	44	61.9	Optimal	0.0	23	0.16837	106	54.4	Feasible	1.4	MAX	Yes	-59.1
3. Neurosurgery														
3.7	0.26299	86	73.8	Feasible	5.1	MAX	0.10144	113	83.7	Feasible	15.0	MAX	Yes	-61.4
3.8	0.23953	92	73.6	Feasible	7.1	MAX	0.08393	118	83.2	Feasible	24.1	MAX	Yes	-65.0
3.10	0.24075	163	71.3	Feasible	3.0	MAX	0.10989	202	79.1	Feasible	67.7	MAX	Yes	-54.4
4. Ophthalmology														
4.7	0.10472	263	43.6	Feasible	1.5	MAX	0.00000	299	47.8	Optimal	0.0	259	Yes	-100.0
4.10	0.00000	299	23.9	Optimal	0.0	83	0.00000	299	23.9	Optimal	0.0	204	No	-100.0
5. Orthopaedics														
5.7	0.40208	145	67.7	Feasible	0.0	MAX	0.30693	174	76.7	Feasible	9.4	MAX	Yes	-23.7
5.8	0.39290	177	59.8	Feasible	0.6	MAX	0.24725	222	73.4	Feasible	7.7	MAX	Yes	-37.1
5.9	0.37768	196	57.0	Feasible	0.2	MAX	0.20615	262	69.1	Feasible	9.9	MAX	Yes	-45.4
5.10	0.40409	227	68.1	Feasible	14.8	MAX	0.28901	323	71.9	Feasible	27.4	MAX	Yes	-28.5
5.11	0.40409	227	68.1	Feasible	21.7	MAX	1.00000	nfs	nfs	nfs	nfs	MAX	No	147.5
5.12	0.40409	227	68.1	Feasible	23.2	3547	1.00000	nfs	nfs	nfs	nfs	MAX	No	147.5
6. Urology														
6.7	0.33289	109	55.6	Feasible	0.0	MAX	0.15253	139	70.3	Feasible	4.1	MAX	Yes	-54.2
6.8	0.31485	117	54.2	Feasible	0.0	MAX	0.12318	152	68.0	Feasible	3.2	MAX	Yes	-60.9
6.10	0.24980	201	57.5	Feasible	1.6	MAX	0.09049	237	72.0	Feasible	34.0	MAX	Yes	-63.8
7. Otolaryngology														
7.7	0.33700	72	61.0	Optimal	0.0	5	0.17809	91	74.4	Feasible	6.4	851	Yes	-47.2
7.8	0.33672	72	61.1	Optimal	0.0	12	0.14022	105	70.7	Feasible	3.1	1299	Yes	-58.4
7.9	0.33510	72	61.3	Optimal	0.0	22	0.10908	116	68.1	Feasible	3.3	MAX	Yes	-67.4
7.10	0.27136	150	58.8	Feasible	0.0	MAX	0.08965	186	74.0	Feasible	8.6	MAX	Yes	-67.0
7.11	0.26852	157	56.4	Feasible	0.0	MAX	0.07446	196	72.4	Feasible	15.7	MAX	Yes	-72.3
8. General surgery 1														
8.7	0.42140	75	55.1	Feasible	0.0	MAX	0.23445	100	72.3	Feasible	1.0	1554	Yes	-44.4
8.8	0.39923	70	63.0	Optimal	0.0	132	0.17535	124	65.8	Feasible	1.8	1953	Yes	-56.1
8.9	0.39288	73	62.0	Optimal	0.0	247	0.16412	128	65.0	Feasible	1.4	1869	Yes	-58.2
8.10	0.34611	152	54.6	Feasible	0.0	MAX	0.14376	195	73.2	Feasible	2.4	MAX	Yes	-58.5
8.11	0.33959	156	54.1	Optimal	0.0	3524	0.11882	211	70.9	Feasible	2.8	MAX	Yes	-65.0
9. General surgery 2														
9.7	0.31254	55	67.1	Feasible	0.1	MAX	0.13241	79	75.3	Feasible	3.7	778	Yes	-57.6
9.8	0.28666	66	60.9	Feasible	0.0	MAX	0.09856	88	72.5	Feasible	6.2	861	Yes	-65.6
9.10	0.25254	115	64.3	Optimal	0.0	2475	0.06204	153	75.9	Feasible	8.1	1050	Yes	-75.4
10. General surgery 3														
10.7	0.25573	54	60.1	Feasible	0.1	MAX	0.07268	66	76.4	Feasible	12.5	1125	Yes	-71.6
10.8	0.23182	57	60.6	Feasible	0.1	MAX	0.05799	69	75.3	Feasible	14.1	1165	Yes	-75.0
10.10	0.23188	101	61.5	Feasible	1.3	MAX	0.03823	127	76.7	Feasible	32.6	3168	Yes	-83.5

nfs = no feasible solution until the time limit

Table 8: Large Instances: Comparison of LP models using discrete and continuous representation of time - Best solutions considering the N. of Scheduled Surgeries and Average OR Occupancy Rate

Also, the continuous model failed to obtain a feasible solution within the established time limit for 3 instances of Orthopaedics (#5.10, #5.11, #5.12). **(** na tabel 5.10 pare ser feasible **)** For these instances the continuous model requires more than 16GB of RAM to find a feasible solution in 1 hour. However, for the other instances, on average, the objective function values of the MIP model are 61% lower than those of the IP model. It means that, compared to the IP model using a discrete representation of time, the proposed MIP model using a continuous representation of time is able to find much better solutions even for large instances.

5.4.2. Continuous MIP model vs. BRKGA heuristic

Table 9 compares for the regular size instances the results obtained with the MIP model to the results obtained using the BRKGA presented in Section 4.2. The MIP model's results are repeated in this table to make the comparison easier. The GA was able to find a solution with lower objective function value under the specified stopping criteria in 62% of the instances with 29% of the continuous MIP model. However, the differences in quality of solutions between the two proposed approaches are very small. In the instances in which the GA is better, the relative improvement was only 1.1% **(** on average?? **)** against 2.6% for the instances in which the continuous model was better. The GA has a better performance on instances that require more memory, such as Orthopaedics #5.6. On the other hand, the highest difference in favour of the exact model is in instances Neurosurgery #3.5 and General surgery 3 #9.5. These instances are characterized by a high number of parallel ORs, suggesting that the exact model is able to deal with such issue better than the BRKGA. The BRKGA decoder is able to prevent overlaps but may leave some idle time between the surgeries, what is difficult to improve only through crossover and mutation. A local search procedure is required to eliminate the idle time.

The GA computation time can be adjusted using the number of restarts parameter. In the computational tests, a value of 30 was used for the number of restarts for producing the best results under the specified time limit. The last column in Table 9 shows that many times the algorithm did not improve after the first restarts. In this situation one can reduce the number of restarts and save computational time.

Table 10 provides a similar comparison for the large size instances. In this instance set, both approaches found 45% of solutions with lower objective function values. However, among the solutions in which the GA obtained a better value, the average improvement was 13% compared to 6% of the exact model. **(** parece estar ao contrário na tabela 11 ? **)** The MIP was able to obtain better values for instances that require more memory, such as the largest instances of Orthopaedics in which the model did not obtain any feasible solution. **(** esta frase parece estar errada. MIP é que não encontrou sol. **)** In its turn, the GA lost more comparisons among medium size instances, like the ones of General Surgery. These instances are characterized by a relative low number of surgeons and larger surgery durations, which increases the chance of occurring overlaps. In this case, the GA would benefit from a local search procedure to make small improvements in the quality of solutions that are difficult to promote with the GA alone.

Table 11 summarizes the results of the computational experiments based on the percentage of better solutions that each alternative approach obtained on each algorithm comparison and on the relative change between the objective function values.

The continuous MIP model shows to be clearly better than the discrete IP model as it was able to find better solutions for all the instances. Those solutions are substantially better, 53% in regular size instances and 61% on large size instances. In its turn, the BRKGA was able to find better solutions than the MIP model for 62% of the regular size instances and for 45% of the large size instances. Surprisingly, the BRKGA was able to find a higher proportion of better solutions among smaller size instances. Table 12 shows the proportion of better solutions obtained by the MIP model and the BRKGA in each group of instances. The proportions are balanced, except the better result of the BRKGA heuristic in smaller instances.

6. Discussion and Future Work

This paper proposed two alternative solution methods for the integrated SCAP: (1) an exact and (2) a heuristic. Both methods were testes with two sets of instances generated from real data, a set of regular size instances and a set of large instances.

In the first case, our contribution is a new formulation for the problem using a continuous representation of time. Compared to a model using a discrete representation of time, which is an adaptation of the model presented in [7], this new formulation found better solutions for all instances. In comparison with the heuristic proposed in this paper it found better quality solutions for instances with a high number of parallel ORs. This results show that the exact formulation is very effective in synchronizing the utilization of parallel resources. Its downside is the required amount of memory.

In the second case, the contribution is an approximation method based on the biased random-key genetic algorithm featuring an original decoding procedure as well as additional local search procedures. The heuristic was able to obtain better solutions than the continuous model in 62% of the regular size instances and 45% of the large size instances. Surprisingly, the results are better in the smaller instances. The reason that justifies this is that the GA is not able to make certain small changes to enhance the quality of solutions. Local search procedures were implemented, which helped to improve most part of the solutions. However, such procedures have a limited number of movements and are able to increase the utilization of ORs but not the number of scheduled surgeries.

In future work, the authors intend to enhance the performance of the heuristic with the addition of new local search procedures. The incorporated local search procedures provide good results, enabling the GA to find better quality solutions in almost all instances. However, only two simple movements that swap one surgery by another

Regular Size Instances													
	MIP Model - continuous time				Genetic Algorithm							Comparison	
Instance	Objective Function Value	N. of Scheduled Patients	Avg. OR Occupancy Rate (%)	Running Time (s)	Objective Function Value	N. of Scheduled Patients	Avg. OR Occupancy Rate (%)	N. of Restarts	N. of Improve-ments	Last Improve-ment	Running Time (s)	GA better than or equal to MIP?	Relative Change (%)
1. Vascular surgery													
1.1	0.19077	61	64.6	MAX	0.19042	61	64.6	12	1	3	MAX	Yes	-0.2
1.2	0.16672	65	63.5	2528	0.16637	65	63.5	12	1	3	MAX	Yes	-0.2
1.3	0.16443	65	63.9	MAX	0.16421	65	63.9	12	1	3	MAX	Yes	-0.1
1.4	0.15572	126	63.4	MAX	0.15710	126	63.1	12	1	3	MAX	No	0.9
1.5	0.15435	126	63.7	1387	0.15591	126	63.3	12	1	3	MAX	No	1.0
1.6	0.15279	126	63.9	1571	0.15435	126	63.6	12	2	11	MAX	No	1.0
2. Oral and maxillofacial surgery													
2.1	0.33127	22	61.9	1	0.33111	22	61.9	12	1	3	MAX	Yes	0.0
2.2	0.30268	25	57.0	6	0.30262	25	57.0	12	1	3	MAX	Yes	0.0
2.3	0.28425	27	53.5	2	0.28419	27	53.5	12	1	3	MAX	Yes	0.0
2.4	0.24813	49	57.9	58	0.24810	49	57.9	12	1	3	MAX	Yes	0.0
2.5	0.22513	54	53.6	2559	0.22513	54	53.6	12	1	3	MAX	Yes	0.0
2.6	0.22513	54	53.6	1791	0.22513	54	53.6	12	1	3	MAX	Yes	0.0
3. Neurosurgery													
3.1	0.13730	58	83.5	1805	0.13760	58	83.4	12	3	7	MAX	No	0.2
3.2	0.10848	63	81.9	1340	0.11417	62	82.2	12	3	7	MAX	No	5.2
3.3	0.09471	65	81.8	1399	0.10046	64	82.1	12	3	12	MAX	No	6.1
3.4	0.09775	114	83.7	MAX	0.09570	115	83.3	12	4	6	MAX	Yes	-2.1
3.5	0.07831	120	82.8	MAX	0.08464	118	83.1	12	3	12	MAX	No	8.1
4. Ophthalmology													
4.1	0.07795	206	57.5	MAX	0.07781	210	56.3	12	2	4	MAX	Yes	-0.2
4.4	0.00000	299	47.8	551	0.00000	299	47.8	12	1	3	MAX	Yes	0.0
5. Orthopaedics													
5.1	0.31872	89	79.5	MAX	0.32105	89	79.0	12	3	10	MAX	No	0.7
5.2	0.26792	117	73.5	MAX	0.27023	117	73.1	12	2	9	MAX	No	0.9
5.3	0.24855	128	71.1	MAX	0.24996	128	70.8	12	3	8	MAX	No	0.6
5.4	0.29367	177	78.3	MAX	0.29913	177	77.3	12	1	3	MAX	No	1.9
5.5	0.25298	221	72.7	MAX	0.24983	225	72.0	12	3	9	MAX	Yes	-1.2
5.6	0.22763	261	65.4	MAX	0.20960	267	66.9	12	3	11	MAX	Yes	-7.9
6. Urology													
6.1	0.25025	64	72.7	1393	0.24798	64	73.1	12	2	8	MAX	Yes	-0.9
6.2	0.18908	81	66.0	MAX	0.18574	82	65.6	12	1	3	MAX	Yes	-1.8
6.3	0.16863	87	63.4	2461	0.16837	87	63.5	12	1	3	MAX	Yes	-0.2
6.4	0.14872	140	70.4	MAX	0.15258	139	70.3	12	1	3	MAX	No	2.6
6.5	0.12318	152	68.0	MAX	0.12251	153	67.6	12	4	7	MAX	Yes	-0.5
7. Otolaryngology													
7.1	0.25039	47	73.9	705	0.24946	47	74.1	12	3	7	MAX	Yes	-0.4
7.2	0.23113	52	71.1	846	0.23032	52	71.2	12	3	9	MAX	Yes	-0.4
7.3	0.20610	58	68.1	731	0.20608	58	68.1	12	2	4	MAX	Yes	0.0
7.4	0.17629	91	74.7	1233	0.17605	91	74.7	12	2	7	MAX	Yes	-0.1
7.5	0.13880	105	71.0	769	0.13830	105	71.1	12	3	10	MAX	Yes	-0.4
7.6	0.10984	116	68.0	MAX	0.10718	117	67.7	12	3	12	MAX	Yes	-2.4
8. General surgery 1													
8.1	0.29224	42	76.4	1552	0.28979	42	76.9	12	1	3	MAX	Yes	-0.8
8.2	0.20228	62	66.0	MAX	0.20188	62	66.1	12	2	4	MAX	Yes	-0.2
8.3	0.18951	65	64.3	2103	0.19267	64	65.1	12	2	5	MAX	No	1.7
8.4	0.23319	100	72.5	814	0.23241	100	72.7	12	4	11	MAX	Yes	-0.3
8.5	0.17410	124	66.1	1294	0.17440	124	66.0	12	3	10	MAX	No	0.2
8.6	0.16420	128	65.0	1394	0.16434	128	65.0	12	6	12	MAX	No	0.1
9. General surgery 2													
9.1	0.19961	40	75.1	2312	0.19977	40	75.0	12	3	6	MAX	No	0.1
9.2	0.17277	44	72.4	857	0.17911	43	73.1	12	3	12	MAX	No	3.7
9.3	0.17162	44	72.6	1465	0.17162	44	72.6	12	2	11	MAX	Yes	0.0
9.4	0.13241	79	75.3	954	0.13249	79	75.3	12	1	3	MAX	No	0.1
9.5	0.09825	88	72.5	1076	0.09525	89	72.1	12	1	3	MAX	Yes	-3.0
10. General surgery 3													
10.1	0.15395	33	76.1	2707	0.15374	33	76.1	12	1	3	MAX	Yes	-0.1
10.2	0.11723	37	73.8	43	0.11744	37	73.8	12	2	5	MAX	No	0.2
10.3	0.10030	39	72.4	MAX	0.10897	38	73.0	12	1	3	MAX	No	8.6
10.4	0.07230	66	76.4	849	0.06981	67	75.7	12	3	12	MAX	Yes	-3.4
10.5	0.05246	70	75.1	2984	0.05768	69	75.4	12	1	3	MAX	No	9.9

Table 9: Regular instances: MIP Model vs. GA Heuristic - Best solutions considering N. of Scheduled Surgeries and Average OR Occupancy Rate

Large Size Instances													
	MIP Model - continuous time				Genetic Algorithm							Comparison	
Instance	Objective Function Value	N. of Scheduled Patients	Avg. OR Occupancy Rate (%)	Running Time (s)	Objective Function Value	N. of Scheduled Patients	Avg. OR Occupancy Rate (%)	N. of Restarts	N. of Improve- ments	Last Improve- ment	Running Time (s)	GA better than or equal to MIP?	Relative Change (%)
1. Vascular surgery													
1.7	0.15572	126	63.4	MAX	0.15710	126	63.1	12	4	7	MAX	No	0.9
1.8	0.15279	126	63.9	2072	0.15591	126	63.3	12	1	3	MAX	No	2.0
1.9	0.15435	126	63.7	1068	0.15591	126	63.3	12	1	3	MAX	No	1.0
1.10	0.12122	241	62.7	MAX	0.11734	243	62.6	12	3	10	MAX	Yes	-3.2
1.11	0.10740	244	63.9	MAX	0.12169	241	62.6	12	3	10	MAX	No	13.3
2. Oral and maxillofacial surgery													
2.7	0.24813	49	57.9	MAX	0.24810	49	57.9	12	1	3	MAX	Yes	0.0
2.8	0.22513	54	53.6	MAX	0.22513	54	53.6	12	1	3	MAX	No	0.0
2.9	0.22513	54	53.6	2141	0.22513	54	53.6	12	1	3	MAX	No	0.0
2.10	0.17225	104	55.3	MAX	0.17208	104	55.3	12	2	6	MAX	Yes	-0.1
2.11	0.16837	106	54.4	MAX	0.16866	106	54.3	12	3	7	MAX	No	0.2
3. Neurosurgery													
3.7	0.10144	113	83.7	MAX	0.10352	114	82.5	12	3	5	MAX	No	2.0
3.8	0.08393	118	83.2	MAX	0.08709	118	82.6	12	4	7	MAX	No	3.8
3.10	0.10989	202	79.1	MAX	0.08713	209	80.3	12	2	4	MAX	Yes	-20.7
4. Ophthalmology													
4.7	0.00000	299	47.8	259	0.00000	299	47.8	12	1	3	MAX	No	0.0
4.10	0.00000	299	23.9	204	0.00000	299	23.9	12	1	3	MAX	No	0.0
5. Orthopaedics													
5.7	0.30693	174	76.7	MAX	0.30394	177	76.3	12	3	6	MAX	Yes	-1.0
5.8	0.24725	222	73.4	MAX	0.25842	223	71.0	12	3	8	MAX	No	4.5
5.9	0.20615	262	69.1	MAX	0.21954	263	66.2	12	2	4	MAX	No	6.5
5.10	0.28901	323	71.9	MAX	0.24426	345	76.1	12	2	9	MAX	Yes	-15.5
5.11	1.00000	nfs	nfs	MAX	0.16479	467	68.8	12	4	6	MAX	Yes	-83.5
5.12	1.00000	nfs	nfs	MAX	0.14881	488	68.0	12	2	6	MAX	Yes	-85.1
6. Urology													
6.7	0.15253	139	70.3	MAX	0.15302	139	70.2	12	3	10	MAX	No	0.3
6.8	0.12318	152	68.0	MAX	0.12565	152	67.6	12	4	11	MAX	No	2.0
6.10	0.09049	237	72.0	MAX	0.08705	240	71.5	12	3	10	MAX	Yes	-3.8
7. Otolaryngology													
7.7	0.17809	91	74.4	851	0.17634	91	74.7	12	2	6	MAX	Yes	-1.0
7.8	0.14022	105	70.7	1299	0.13894	105	70.9	12	2	4	MAX	Yes	-0.9
7.9	0.10908	116	68.1	MAX	0.11038	116	67.9	12	3	9	MAX	No	1.2
7.10	0.08965	186	74.0	MAX	0.10113	185	72.5	12	3	5	MAX	No	12.8
7.11	0.07446	196	72.4	MAX	0.07438	198	71.6	12	4	10	MAX	Yes	-0.1
8. General surgery 1													
8.7	0.23445	100	72.3	1554	0.23539	100	72.1	12	3	8	MAX	No	0.4
8.8	0.17535	124	65.8	1953	0.17901	123	65.8	12	4	8	MAX	No	2.1
8.9	0.16412	128	65.0	1869	0.16976	127	64.6	12	4	11	MAX	No	3.4
8.10	0.14376	195	73.2	MAX	0.16538	192	70.5	12	7	12	MAX	No	15.0
8.11	0.11882	211	70.9	MAX	0.14216	207	68.3	12	5	12	MAX	No	19.6
9. General surgery 2													
9.7	0.13241	79	75.3	778	0.13327	79	75.2	12	4	9	MAX	No	0.7
9.8	0.09856	88	72.5	861	0.10011	88	72.2	12	2	11	MAX	No	1.6
9.10	0.06204	153	75.9	1050	0.06893	152	75.2	12	6	11	MAX	No	11.1
10. General surgery 3													
10.7	0.07268	66	76.4	1125	0.07082	67	75.5	12	3	7	MAX	Yes	-2.6
10.8	0.05799	69	75.3	1165	0.05901	69	75.1	12	2	9	MAX	No	1.8
10.10	0.03823	127	76.7	3168	0.05932	124	75.0	12	3	9	MAX	No	55.2

nfs = no feasible solution until the time limit

Table 10: Large instances: MIP Model vs. GA Heuristic - Best solutions considering N. of Scheduled Surgeries and Average OR Occupancy Rate

Table 11: Summary of the computational experiments

Regular size instances

	Discrete exact model	Continuous exact model	Continuous exact model	BRKGA heuristic
Percentage of better solutions	0%	100%	29%	62%
Avg. relative difference in better instances	-	53%	2.6%	1.1%

Large size instances

	Discrete exact model	Continuous exact model	Continuous exact model	BRKGA heuristic
Percentage of better solutions	0%	100%	45%	45%
Avg. relative difference in better instances	-	61%	13%	6%

Table 12: Percentage of better solutions by solution method in each instance group

Instance Group	Continuous exact model (%)	BRKGA heuristic (%)
1	22	65
2	50	50
3	59	41
4	45	40

were allowed. New movements should be implemented to exchange one scheduled surgery by multiple unscheduled ones. In addition, the problem of finding the best combination of parameters should be addressed to allow the GA to have a more uniform performance across different instances. Furthermore, in what concerns different problem settings, we intend to evaluate the performance of the proposed approaches in a rolling horizon framework. In this case, additional constraints are required to minimize the rescheduling of previously scheduled patients as well as to minimize situations in which the sequence of the waiting list, determined by priority and waiting time rules, is broken. This framework would allow to compare the performance of alternative objective functions in the long term, to better understand the impact of prioritizing the number of scheduled patients or the average OR utilization rate.

Acknowledgments

This work has been partially supported by projects PTDC/EGE-GES/ 117692/2010 and NORTE-07-0124-FEDER-000057 funded by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF) and the Programme COMPETE, and by national funds, through the Portuguese funding agency, Fundac£o para a Ciłncia e a Tecnologia (FCT).

Appendix

7. Exact Model Using a Discrete Representation of Time

This section briefly describes the discrete model compared with the proposed continuous model. The discrete model is described in detail in [7]. Similar models are also presented in Marques et al. [13] and Guinet and Chaabane [23].

7.1. Sets and Indices

The sets and indices are equal to the ones presented in Section 4.1, except for the introduction of a new set L to denote the intervals in which surgeries are allowed to start in each working shift. These discrete intervals are a result of the discretization of time and their size usually ranges from 10 min to 1 hour, with the most used value being 15 min. Among the parameters the only new entry is parameter n to denote the number of intervals in each shift. This value is determined by dividing the capacity of ORs by the selected size of interval, e.g. $360/15 = 24$.

I set of patients (index i)
J set of working shifts in the planning horizon (index j)
K set of operating rooms (index k)
K_j set of available ORs in shifts j
S set of surgeons (index s)
I_s set of patients of surgeon s (index i)
H set of weeks in the planning horizon (index h)
J_h set of days in a given week h (index j)
L set of intervals in each shift j (index l)
$I_{maxsched}$ set of patients with maximum scheduling time within the planning horizon
$I_{maxswait}$ set of patients with maximum waiting time within the planning horizon

7.2. Parameters

d_i	estimated duration in minutes of patient's i surgery
s_i	surgeon in charge of patient's i surgery
max_i	maximum waiting time of patient's i surgery
c_{jk}	available capacity in shift j of OR k
a_{js}	availability in shift j of surgeon s
day_j	day of shift j
α	weight of the number of scheduled surgeries in the objective function
β	weight of the average OR utilization rate in the objective function
γ	best number of scheduled surgeries
δ	best average OR utilization rate
ct	OR cleaning time
tt	surgeon turnover time
C	total OR capacity
ms	maximum number of shifts per week
n	number of intervals per shift

The discrete model has only one decision variable to represent the scheduled patients. Variable X_{ijkl} represents all at once the selected patient, day, shift, OR and starting time. The objective function (23) is very similar to the objective function of the continuous model. The only difference is that it has one more cycle, through the set L, to determine the scheduled patients. For a detailed description of the objective function used in the continuous model see Section 4.1.

7.3. Decision Variables

$$X_{ijkl} = \begin{cases} 1, \text{if patient } i \text{ is scheduled for shift } j, \text{OR } k \text{ and period } l \\ 0, \text{otherwise} \end{cases}$$

7.4. Objective Function

$$min\ F = \alpha \cdot \frac{\gamma - \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L} X_{ijkl}}{\gamma} + \beta \cdot \frac{\delta - \frac{\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L} X_{ijkl}\cdot d_i}{C}}{\delta} \tag{22}$$

7.5. Constraints

The first set of constraints provide the basic structure of the model. Inequality (23) prevents a patient from being scheduled more than once, expression (24) restricts the scheduling of patients to the capacity of available shifts and ORs, and constraint (25) prevents surgeries from having a scheduled end time greater than the surgical suite closing time.

$$\sum_{j\in J}\sum_{k\in K}\sum_{l\in L} X_{ijkl} \leq 1, \forall i \in I \tag{23}$$

$$\sum_{l\in L}\sum_{i\in I} X_{ijkl}\cdot(d_i + ct) \leq c_{jk}, \forall j \in J, \forall k \in K \tag{24}$$

$$\sum_{l\in L|l+d_i+ct\leq n}\ \sum_{i\in I} X_{ijkl} \leq c_{jk}, \forall j \in J, \forall k \in K \tag{25}$$

$$\tag{26}$$

Expression (27) states that surgeries with a maximum scheduling time lower than the planning horizon must be scheduled. Expression (28) states that surgeries with a maximum waiting time lower than the planning horizon must be scheduled and inequality (29) states that the surgery day must be lower than the maximum waiting time. These constraints are equal to the ones used in the continuous model and are designed to respect patients' priority and waiting time rules.

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} = 1, \forall i \in I_{maxsched} \tag{27}$$

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} = 1, \forall i \in I_{maxwait} \tag{28}$$

$$\sum_{j\in J}\sum_{k\in K} X_{ijk} \cdot day_j \leq max_i, \forall i \in I_m axwait \tag{29}$$

The next group prevents overlap of patients in the same room and the overlap of patients from the same surgeon in different rooms. It is worth mentioning that these constraints are not required in the proposed continuous model. Thus, it is one of the main differences between the models. Constraint (30) prevents the overlap of surgeries in the same shift and OR also ensuring the cleaning time after each surgery, while constraint (31) avoids the overlap of patients of the same surgeon in different ORs in the same shift observing surgeons' turnover time.

$$\sum_{i\in I}\sum_{j'\in J|j'\geq j-d_i+1-ct \text{ and } j'\leq j} X_{ij'kl} \leq 1, \forall j \in J, \forall k \in K, \forall l \in L \tag{30}$$

$$\sum_{i\in I_s}\sum_{k\in K}\sum_{l'\in L|l'\geq 0 \text{ and } l'\geq l-di+tt+1 \text{ and } l'\leq l \text{ and } l'<n} X_{ijkl'} \leq 1, \forall s \in S, \forall j \in J \tag{31}$$

Finally, the last set concerns surgeon availability and workload. Constraint (32) restricts the scheduling of patients for a given surgeon to his/her availability and constraint (33) constrains the surgeon's workload in terms of number of working shifts per week.

$$\min\{1, \sum_{k\in K}\sum_{l\in L}\sum_{i\in I_s} X_{ijkl}\} \leq a_{js}, \forall j \in J, \forall s \in S \tag{32}$$

$$\sum_{j\in J_h}\min\{1, \sum_{k\in K}\sum_{l\in L}\sum_{i\in I_s} X_{ijkl}\} \leq ms, \forall s \in S, \forall h \in H \tag{33}$$

References

[1] OECD, Society at a glance 2014doi:http://dx.doi.org/10.1787/soc-glance-2014-en.
URL `/content/book/soc-glance-2014-en`

[2] T. Gordon, S. Paul, A. Lyles, J. Fountain, Surgical unit time utilization review: Resource utilization and management implications, Journal of Medical Systems 12 (3) (1988) 169–179.

[3] HFM, Achieving operating room efficiency through process integration., Healthcare financial management : journal of the Healthcare Financial Management Association 57 (3) (2003) suppl 1–7 following 112.
URL `http://www.ncbi.nlm.nih.gov/pubmed/12793445`

[4] A. Commission, Operating theatres — Review of national findings, Tech. rep., National Health Service in England & Wales, London (2003).

[5] E. Mayer, O. Faiz, T. Athanasiou, C. Vincent, Measuring and enhancing elective service performance in NHS operating theatres: an overview, Journal of the Royal Society of Medicine 101 (6) (2008) 273–277.

[6] J. F. Gonçalves, M. G. C. Resende, Biased random-key genetic algorithms for combinatorial optimization, Journal of Heuristics 17 (5) (2011) 487–525.

[7] F. Sperandio, C. Gomes, J. Borges, A. Carvalho Brito, B. Almada-Lobo, An Intelligent Decision Support System for the Operating Theater: A Case Study, IEEE Transactions on Automation Science and Engineering 11 (1) (2014) 265–273. doi:10.1109/TASE.2012.2225047.

[8] B. Cardoen, E. Demeulemeester, J. Beliën, Operating room planning and scheduling: A literature review, European Journal of Operational Research 201 (3) (2010) 921–932. doi:10.1016/j.ejor.2009.04.011.
URL `http://linkinghub.elsevier.com/retrieve/pii/S0377221709002616`

[9] F. Guerriero, R. Guido, Operational research in the management of the operating theatre: a survey, Health Care Management Science 14 (1) (2011) 89–114. doi:10.1007/s10729-010-9143-6.
URL `http://dx.doi.org/10.1007/s10729-010-9143-6`

[10] I. Ozkarahan, Allocation of surgical procedures to operating rooms., Journal of medical systems 19 (4) (1995) 333–52.
URL `http://www.ncbi.nlm.nih.gov/pubmed/8522909`

[11] B. Roland, C. Di Martinelly, F. Riane, Y. Pochet, Scheduling an operating theatre under human resource constraints, Computers & Industrial Engineering 58 (2) (2010) 212–220. doi:10.1016/j.cie.2009.01.005.
URL `http://linkinghub.elsevier.com/retrieve/pii/S0360835209000084`

[12] H. Fei, N. Meskens, C. Chu, A planning and scheduling problem for an operating theatre using an open scheduling strategy, Computers & Industrial Engineering 58 (2) (2010) 221–230. doi:10.1016/j.cie.2009.02.012.
URL http://linkinghub.elsevier.com/retrieve/pii/S0360835209000801
[13] I. Marques, M. E. Captivo, M. V. Pato, An integer programming approach to elective surgery scheduling, OR Spectrum 34 (2) (2012) 407–427. doi:10.1007/s00291-011-0279-7.
URL http://dx.doi.org/10.1007/s00291-011-0279-7
[14] I. Marques, M. E. Captivo, M. V. Pato, Scheduling elective surgeries in a portuguese hospital using a genetic heuristic, Operations Research for Health Care 3 (2) (2014) 59 – 72, special Issue of the 2012 conference of the {EURO} working group Operational Research Applied To Health Services (ORAHS). doi:http://dx.doi.org/10.1016/j.orhc.2013.12.001.
URL http://www.sciencedirect.com/science/article/pii/S2211692313000398
[15] B. Vijayakumar, P. J. Parikh, R. Scott, A. Barnes, J. Gallimore, A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital, European Journal of Operational Research 224 (3) (2013) 583–591.
[16] S. Ogulata, R. Erol, A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals, Journal of Medical Systems 27 (3).
[17] N. Meskens, D. Duvivier, A. Hanset, Multi-objective operating room scheduling considering desiderata of the surgical team, Decision Support Systems 55 (2) (2013) 650–659.
[18] B. Cardoen, Operating room planning and scheduling: solving a surgical case sequencing problem, 4or 8 (1) (2009) 101–104. doi:10.1007/s10288-009-0106-z.
URL http://www.springerlink.com/index/10.1007/s10288-009-0106-z
[19] B. Cardoen, E. Demeulemeester, J. Beliën, Sequencing surgical cases in a day-care environment: An exact branch-and-price approach, Computers & Operations Research 36 (9) (2009) 2660–2669. doi:10.1016/j.cor.2008.11.012.
URL http://linkinghub.elsevier.com/retrieve/pii/S0305054808002384
[20] D.-N. Pham, A. Klinkert, Surgical case scheduling as a generalized job shop scheduling problem, European Journal Of Operational Research 185 (3) (2008) 1011–1025. doi:10.1016/j.ejor.2006.03.059.
URL http://linkinghub.elsevier.com/retrieve/pii/S0377221706005820
[21] V. Augusto, X. Xie, V. Perdomo, Operating theatre scheduling with patient recovery in both operating rooms and recovery beds, Computers & Industrial Engineering 58 (2) (2010) 231–238. doi:10.1016/j.cie.2009.04.019.
URL http://linkinghub.elsevier.com/retrieve/pii/S0360835209001193
[22] A. Jebali, A. B. Hadj Alouane, P. Ladet, Operating rooms scheduling, International Journal of Production Economics 99 (1-2) (2006) 52–62. doi:10.1016/j.ijpe.2004.12.006.
URL http://linkinghub.elsevier.com/retrieve/pii/S0925527304004517
[23] A. Guinet, S. Chaabane, Operating theatre planning, International Journal of Production Economics 85 (1) (2003) 69–81. doi:10.1016/S0925-5273(03)00087-2.
URL http://linkinghub.elsevier.com/retrieve/pii/S0925527303000872
[24] S. Chaabane, N. Meskens, A. Guinet, M. Laurent, Comparison of two methods of operating theatre planning: Application in belgian hospital, Journal of Systems Science and Systems Engineering 17 (2) (2008) 171–186.
[25] A. Riise, E. Burke, Local search for the surgery admission planning problem, Journal of Heuristics 17 (4) (2011) 389–414. doi:10.1007/s10732-010-9139-x.
URL http://dx.doi.org/10.1007/s10732-010-9139-x
[26] L. Zhong, S. Luo, L. Wu, L. Xu, J. Yang, G. Tang, A two-stage approach for surgery scheduling, Journal of Combinatorial Optimization 27 (3) (2014) 545–556.
[27] H. Fei, C. Chu, N. Meskens, A. Artiba, Solving surgical cases assignment problem by a branch-and-price approach, International Journal of Production Economics 112 (1) (2008) 96–108. doi:10.1016/j.ijpe.2006.08.030.
URL http://linkinghub.elsevier.com/retrieve/pii/S0925527307001296
[28] J. C. Bean, Genetic algorithms and random keys for sequencing and optimization, ORSA Journal on Computing 6 (2) (1994) 154–160.
[29] J. F. Gonçalves, M. G. C. Resende, R. F. Toso, An experimental comparison of biased and unbiased random-key genetic algorithms, Pesquisa Operacional 34 (2014) 143 – 164.
[30] J. J. M. Mendes, J. F. Gonçalves, M. G. C. Resende, A random key based genetic algorithm for the resource constrained project scheduling problem, Computers & Operations Research 36 (1) (2009) 92–109.
[31] J. F. Gonçalves, J. J. M. Mendes, M. G. C. Resende, A genetic algorithm for the resource constrained multi-project scheduling problem, European Journal of Operational Research 189 (3) (2008) 1171 – 1190.
[32] J. F. Gonçalves, J. J. M. Mendes, M. G. C. Resende, A hybrid genetic algorithm for the job shop scheduling problem, European Journal of Operational Research 167 (1) (2005) 77 – 95.
[33] Ministrio da Sade, Manual do SIGIC - MGIC 2011 - Volume I Princípios Gerais, Tech. rep., Unidade Central de Gestão de Inscritos para Cirurgia, Administração Central do Sistema de Saúde, IP, Lisboa, Portugal (February 2011).
[34] J. H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975.
[35] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[36] V. M. Spears, K. A. D. Jong, On the virtues of parameterized uniform crossover, in: In Proceedings of the Fourth International Conference on Genetic Algorithms, 1991, pp. 230–236.
[37] R. F. Toso, M. G. C. Resende, A c++ application programming interface for biased random-key genetic algorithms, Optimization Methods and Software published online 13 March 2014.
[38] J. F. Gonçalves, M. G. C. Resende, M. D. Costa, A biased random-key genetic algorithm for the minimization of open stacks problem, International Transactions in Operational Research published online 2 July 2014.
[39] J. F. Gonçalves, M. G. C. Resende, A biased random-key genetic algorithm for a 2d and 3d bin packing problem, International Journal of

Production Economics 145 (2013) 500–510.