Operations Research for Health Care 3 (2014) 59–72

Contents lists available at ScienceDirect

Operations Research for Health Care

journal homepage: www.elsevier.com/locate/orhc

Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic

Inês Marques [a,b,c,∗], M. Eugénia Captivo [a,b], Margarida Vaz Pato [a,d]

[a] *Centro de Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Portugal*

[b] *Universidade de Lisboa, Faculdade de Ciências, DEIO, Edifício C6 - Piso 4, Campo Grande, 1749-016 Lisboa, Portugal*

[c] *Universidade Lusófona de Humanidades e Tecnologias, ECEO, Campo Grande, 376, 1749-024 Lisboa, Portugal*

[d] *Universidade de Lisboa, Instituto Superior de Economia e Gestão, Dept. Matemática, Rua do Quelhas, 6, 1200-781 Lisboa, Portugal*

A R T I C L E   I N F O

Article history:
Received 31 October 2012
Accepted 2 December 2013
Available online 14 December 2013

Keywords:
Health care
Operating rooms
Elective case scheduling
Genetic heuristics

A B S T R A C T

The Portuguese National Health Plan outlines two main guidelines for hospital units: improve the efficient use of the available resources and reduce the waiting list for surgery. The aim of this work is to provide a contribution in the field of operations research to achieve these guidelines. The operating theater is a hospital unit that represents a great proportion of the hospital budget. Furthermore, it is a central service with connections and implications in the service of many other hospital units. Therefore, this work is dedicated to a case study of an elective surgery scheduling problem arising in a Portuguese public hospital. The problem consists of assigning an intervention date, an operating room and a starting time for elective surgeries that remain in the hospital waiting list, thus combining simultaneously advance and allocation scheduling. Two conflicting optimization criteria are independently considered: maximize the surgical suite occupation and maximize the number of surgeries scheduled. Two versions of a single objective genetic heuristic are developed and applied to real data from the studied hospital. The results show that this approach improves the quality of the hospital surgical plans in light of the objectives considered, requiring much fewer resources to construct the surgical plans. Real instances with 508–2306 elective surgeries are successfully solved in less than 240 s. These are better results than authors' previous approaches to the same problem.

1. Introduction

The health care sector has undergone major changes over the last decade. The increase in life expectancy coupled with a great technological development has boosted the demand for health care (e.g. on external consultations, treatments and surgery applications). This increased demand forces an effective provision of health services. Nevertheless, similar to other economic sectors, the health sector has been suffering from successively more restrictive budgets, forcing a resource rationalization practice among health care providers and also a more efficient usage of these resources. Thus, public hospitals have been enduring increasing difficulties in fulfilling their mission and contractual responsibilities with the successive budget cuts that have been occurring in Portugal.

The operating theater is a hospital unit with extremely high costs (mainly staff and equipment costs) representing a great proportion of the hospital's budget [1,2]. In addition, the surgical suite is a central service with connections and direct implications in many other hospital units, such as wards and recovery units. These factors require the development of operating room planning practices that enable an efficient use of the operating theater. Moreover, improving the efficient use of a hospital's available resources as well as reducing the waiting list for surgery are two main guidelines which are outlined in the Portuguese National Health Plan [3]. Waiting lists for surgery result from a mismatch between the demand for surgery and the health system supply capacity. The latter is defined by the number of installed and available surgical suites and by the efficiency of the organization providing health care. Therefore, optimizing the operating room's installed capacity, as well as the human and technological resources available, also reflects on the balance of the waiting list for surgery.

This work presents a case study of an elective surgery scheduling problem arising in a Portuguese hospital. The aim is to contribute to increase the efficiency of the hospital surgical supply's installed capacity and to reduce the hospital's waiting list for

∗ Corresponding author at: Centro de Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Portugal. Tel.: +351 21 7500142; fax: +351 21 7500022.

E-mail addresses: ines.marques@fc.ul.pt (I. Marques), mecaptivo@fc.ul.pt (M.E. Captivo), mpato@iseg.utl.pt (M. Vaz Pato).

2211-6923/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.orhc.2013.12.001

surgery. At the beginning of this study, the surgical suite of this hospital had a regular time occupancy rate of about 42% with a waiting list reduction rate of nearly 5.5%. These indicators reveal a clearly inefficient use of the operating theater representing a significant opportunity cost, both social and economic.

The elective surgery planning problem tackled in this paper consists of scheduling elective surgeries for a day, an operating room and a starting time, within a weekly planning horizon. This problem arises at an operational level of decisions for operating room planning, combining advance and allocation scheduling within the same problem. Marques [4] proved that this problem is NP-hard. Few papers in the literature consider a combination of these two components within an unique problem and all of these papers deal with short planning horizons, e.g. a day or a week (see e.g. Roland et al. [1] and Riise and Burke [5]). Due to the high complexity of these two components, the problem is very difficult to solve for real size instances, as pointed out in Riise and Burke [5]. The computational difficulty inherent to the surgeries' scheduling problem has encouraged the development of heuristic approaches (see e.g. Hans et al. [6], Fei et al. [2], Roland et al. [1], Liu et al. [7], Riise and Burke [5] and Marques et al. [8]). Notwithstanding, in the literature there are some integer programming approaches to the problem (see e.g. Guinet and Chaabane [9], Velásquez and Melo [10], Cardoen et al. [11,12] and Marques et al. [13]). The specificities of the different cases under study as well as the different national realities are the most relevant factors contributing to the diversity of the work in this area. This makes difficult the comparisons between the various approaches and also the application of existing approaches in distinct realities. Recently, Cardoen et al. [14] and Guerriero and Guido [15] presented literature reviews on the contribution of operations research to operating room planning and scheduling. Cardoen et al. [16] proposed a classification scheme of the research in this domain.

In previous approaches to the same problem, the authors presented an integer linear programming model [13], and specially designed constructive and improvement heuristics [8]. The model was solved with CPLEX 11.0 and we obtained good quality surgical plans for the smaller real instances. However, this exact methodology has consumed much computational time and was unable to build a single feasible solution for high dimension instances. The heuristics presented in [8] were adapted to be embedded in the genetic algorithm (GA) that is now presented. As we will see in Section 4, the results obtained with this approach are very competitive (in terms of solutions quality and computational time) even in the larger real instances.

Roland et al. [1] also present a GA to the elective case scheduling problem. However, the problem studied is different in its specifications and in the objective considered. In addition, the characteristics of the designed GA are rather different and the codification structure does not guarantee feasible solutions. When considering small instances (only one operating day and 19 elective surgeries), the GA computing times never exceeded 322 s, but on a real life weekly instance (composed of 80 surgeries) the computational time increased significantly (more than 1.6 h if considering a population containing 20 individuals and 5000 generations, as emphasized by the authors, or a population with 100 individuals and 500 generations).

This paper proceeds in Section 2 with a description of the elective case scheduling problem for the hospital under study and some hospital specifications. Section 3 presents the GA developed for this problem and Section 4 reports on the results of the computational experiments performed using real data obtained from the hospital. Finally, conclusions and future research directions are outlined in Section 5.

2. Problem description and case study

The present case study focuses on a general, central and university hospital located in Lisbon, incorporated within the Portuguese National Health Service (public hospital). The hospital performs nearly 5000 surgeries per year and the waiting list for surgery had about 2200 surgeries at the time of the beginning of this study. It has no emergency service so the problem is exclusively dedicated to elective surgeries. Elective surgeries can be either *conventional* (inpatient surgeries) or *ambulatory* (outpatient surgeries). For an ambulatory surgery both the hospital admission and the discharge of the patient occur on the same day and therefore the patient is not in hospital overnight. According to Portuguese legislation [17], elective surgeries are classified in four levels of priority defining the due date in which they must be performed: *deferred urgency* surgeries must be completed in three days; *high priority* surgeries within 15 days; *priority* surgeries must be completed within two months; and *normal* surgeries in one year.

The hospital has one central surgical suite with six identical operating rooms. One of these operating rooms is exclusively devoted to ambulatory surgeries while the remaining operating rooms are reserved for conventional surgeries. The surgical suite is open in regular time between 8.30 am and 8 pm, from Monday to Friday. The surgery schedule should respect this regular time and no surgery should be planned using extra time. Five surgical specialties compete for the operating rooms' time. The practice of this hospital is to assign rooms to surgical specialties throughout the day. Two main reasons are pointed out for this: the fragile nature of the mobile specialized equipment usually specific to each specialty; and the required downtime of about an hour to exchange a surgical specialty in a room during the day which would represent a major inefficiency in the surgical service. Nevertheless, scheduling is performed in open scheduling strategy and there is not a defined pre-allocation of operating room time to specialties in a master surgery schedule. Between two surgeries performed in the same room, cleaning and disinfecting protocols, performed by auxiliary staff and taking about 30 m, must take place. Each operating room is staffed with fixed and permanent nursing teams throughout the surgical suite's regular time. Each patient is assigned to a surgeon at waiting list booking time and, therefore, when planning, patients and surgeons are already assigned, which thus constitutes an input for the planning problem. This is a cultural practice commonly used in Portugal. Surgeons can operate at any time of the day within the operating theater's regular time and also in any operating room provided that the surgical specialties match. Practical experience in this hospital shows that other resources do not limit the activity of the surgical suite, namely beds (in the recovery units and wards), staff (nurses and auxiliary staff), equipment and materials.

In the hospital under study, surgical planning is weekly and is finalized on Friday for the following week. It is a manual procedure that requires the contribution of several human resources. Each surgical ward collects the proposed surgeries for the following week and sends the corresponding planning maps to the head nurse of the surgical suite. The head nurse confers and verifies the possibility of combining the planning maps of all surgical wards and finalizes the surgical plan. This task requires too much staff time and the result achieved is far from contributing to the efficient use of the operating theater. It is still possible to make online changes to the surgical plan during the course of the week until noon of the day before the surgery. This turns to a re-scheduling problem that does not fall in the scope of this work.

As a result, scheduling is performed in open strategy but it must assign up to one surgical specialty to each room and day.

Surgeries must be scheduled in order to meet the deadline set by the associated priority level. As such, once the planning is finalized on Friday, deferred urgency surgeries must be scheduled for Monday in order to fulfill the deadline. Despite the deadline of 15 days for the high priority surgeries, they should be scheduled throughout the planning week. This option allows one to anticipate the completion of this type of surgeries so as not to press the schedule in the following planning period. The remaining surgeries (classified as priority or normal) may be scheduled or not during the planning week. Note that it is not guaranteed that the operating room time available is sufficient to schedule all the deferred urgency and high priority surgeries in accordance. However, practical experience, based on the number of surgeries of this type usually found in the waiting list for surgery and their distribution by the surgical specialties, shows that such constraints are not sufficient to render the problem unfeasible.

The elective surgery scheduling problem schedules elective surgeries for a day, a room and a starting time for a weekly planning horizon. Surgical plans must respect the surgical suite's regular working hours. The scheduling problem must also take into consideration structural constraints such as no overlapping of surgeries in the same operating room and of surgeons between operating rooms in the same time period and day. Time for room cleaning and disinfecting protocols (30 min) must be considered between two surgeries scheduled. The problem also considers daily and weekly operating time limits for each surgeon.

The aim of this work (as stated in Section 1) suggests two main optimization criteria to the scheduling problem: to maximize the surgical suite occupation and to maximize the number of surgeries scheduled. These objectives have clearly a conflictuous nature. In fact, due to the need of having 30 min idle for room cleaning at the end of each surgery, when we intend to maximize surgical suite occupation, longer surgeries are preferable to short surgeries thus avoiding the inactivity inherent to cleanliness. But when the objective is to maximize the number of surgeries scheduled, short surgeries are preferable to long surgeries. These two objectives are independently considered in the developing and testing of the algorithm.

Time has a discrete representation in periods of 15 min, thus generating 46 daily time periods in regular working time. The choice of the time period duration results from a trade-off between precision in the duration of surgeries and its impact on the dimension of the instances. Reducing the dimension of the time periods to, e.g., 10 min would increase the number of periods in regular working hours to 69. Surgery durations are assumed deterministic. Induction and waking up time are included in the surgery duration since these procedures are also performed in the operating room. Nevertheless, cleaning room time is not incorporated in the surgeries' duration. In this hospital, surgeons may exchange between operating rooms. Hence, when a surgeon operates two consecutive surgeries, excluding cleaning time in the surgeries' duration allows the surgeon to perform the second surgery in another operating room soon after completing the first surgery, thus avoiding the idle time for cleaning the room.

The mathematical models designed for this elective surgery scheduling problem are described in Appendix A. Due to the high complexity of the problem and the large dimension of real instances, in the authors' previous approaches referred to in Section 1 the problem was decomposed into two hierarchical phases according to the nature of the surgeries: conventional surgeries are scheduled in the first phase and ambulatory surgeries are scheduled in the second phase. Surgeons represent the connection between the two phases. This decomposition allows the reduction of the overall problem dimension, however it limits the solution space. The results achieved with real instances suggested the development of a heuristic approach, which is presented in Section 3. This approach tackles the overall problem, not requiring the division into hierarchical phases as happened with the previous approaches. Moreover, new results obtained with the mathematical model for the overall problem are reported in Section 4.

$$(\underbrace{\boxed{1\,3\,2\,1}\;1\;\underbrace{1}_{Amb}}_{Monday}/\underbrace{1\,2\,4\,1\,5\;\underbrace{1}_{Amb}}_{Tuesday}/\underbrace{4\,2\,1\,4\,5\;\underbrace{1}_{Amb}}_{Wednesday}/\underbrace{2\,4\,4\,5\,1\;\underbrace{1}_{Amb}}_{Thursday}/\underbrace{1\,5\,4\,4\,2\;\underbrace{1}_{Amb}}_{Friday})$$

Fig. 1. Chromosome encoding.

3. Genetic algorithm

Algorithm 1 summarizes the GA specially developed for the elective surgery scheduling problem arising in the hospital under study. This GA is based on the biased random-key genetic algorithms (BRKGA) for combinatorial optimization problems introduced by Gonçalves and Resende [18]. Other versions were developed and tested. A brief report on these versions is presented in Section 3.2. A detailed description about GA can be found in Davis [19].

The solutions of the problem correspond to the individuals of the population within the GA, each one associated with one chromosome. Each chromosome is a *Days* $\times$ *Rooms* matrix, composed of integer numbers. Each integer represents the surgical specialty that is assigned to the corresponding operating room and day. Numbers 1 to 5 correspond, respectively, to the following surgical specialties: digestive and general surgery (1), thorax surgery (2), angiology and vascular surgery (3), otorhinolaryngology (4), and urology (5). In each planning week, some operating rooms are needed to perform deferred urgency surgeries on Monday. The corresponding genes are then reserved for these surgical specialties and therefore remain equal for all chromosomes throughout the execution of the algorithm. Fig. 1 shows an example of a chromosome. In the figure, rooms A to *D* are dedicated to operate deferred urgency surgeries from digestive and general surgery (1), thorax surgery (2), and angiology and vascular surgery (3) specialties and are therefore reserved for these surgical specialties on Mondays in all the solutions. The last gene in each planning day represents the operating room devoted to the ambulatory surgeries. There are no ambulatory thorax surgeries (specialty number 2). An indirect encoding is used and therefore a decoding technique must be applied to generate a solution from a chromosome. This decoding technique is detailed ahead in Section 3.1. Fig. 2 shows the surgical plan (solution) represented by the chromosome of Fig. 1.

The algorithm starts by initializing a population of `nPOP` individuals (step 1). An elitist chromosome is included in the initial population. This chromosome corresponds to the best solution obtained by applying constructive and improvement heuristics based on the procedures presented in [8]. These heuristics are similar to

Algorithm 1 Genetic algorithm scheme

1: Initialize a population of `nPOP` chromosomes. Decode each chromosome and evaluate it.
2: Sort chromosomes by their fitness. Classify the best `nPOPe` individuals as elite and the remaining individuals as non-elite.
3: Copy the elite individuals to the next population. Generate `nPOPm` mutants to next population.
4: Recombine elite and non-elite individuals and generate `nPOP` − `nPOPe` − `nPOPm` new chromosomes for the next population.
5: Create the new population. Decode each chromosome and evaluate it.
6: Stop, if a maximum number of generations (`nGER`) is met; otherwise, return to step 2.

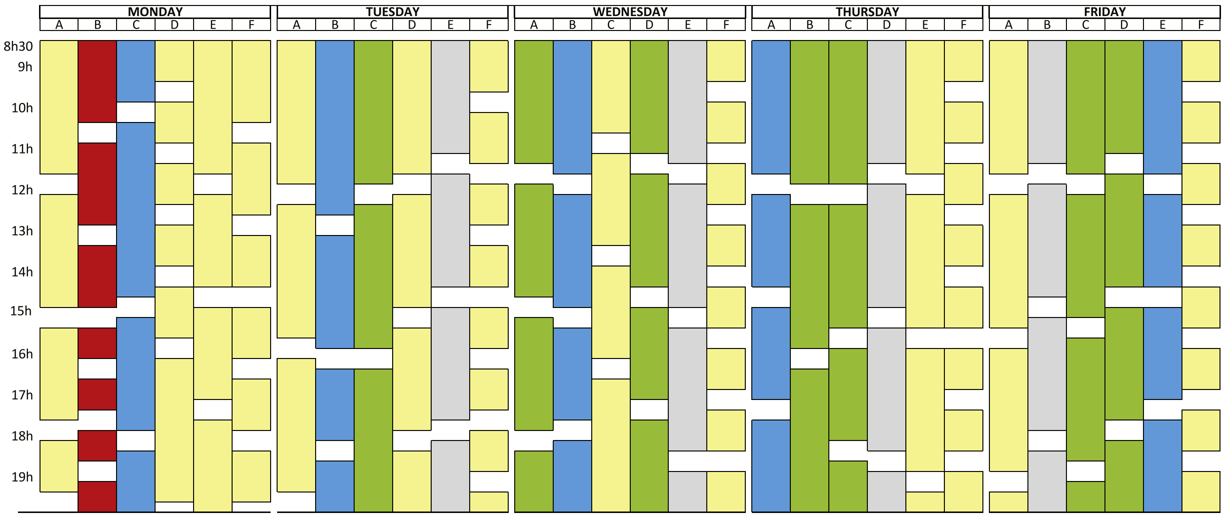

Fig. 2. Solution resulting from chromosome decoding.

Elite parent:	($\boxed{1\ 3\ 2\ 1}$ 1 1 / 1 2 4 1 5 1 / 4 2 1 4 5 1 / 2 4 4 5 1 1 / 1 5 4 4 2 1)
Non-elite parent:	($\boxed{1\ 3\ 2\ 1}$ 4 3 / 1 4 3 5 4 1 / 1 5 4 4 3 4 / 4 1 2 3 4 1 / 5 3 2 1 5 1)
Random number:	- - - - .18 .85 .09 .93 .97 .10 .62 .37 .14 .92 .03 .70 .90 .34 .52 .78 .92 .50 .48 .86 .66 .70 .37 .16 .69 .47
Child:	($\boxed{1\ 3\ 2\ 1}$ 1 3 / 1 4 3 1 4 1 / 4 5 1 4 3 1 / 2 1 2 5 1 1 / 5 3 4 4 5 1)

Fig. 3. Recombination operator: parametrized uniform crossover ($\rho_e = 0.6$).

the procedures used as decoder and thus the respective description is left for Section 3.1. Introducing this high-quality individual in the initial population helps the algorithm to find better solutions more quickly than starting from an exclusively random population [20] (see Table 17 in Appendix B). The remaining initial population is randomly generated according to the distribution of the surgical specialties in the waiting list for surgery for each type of surgery (conventional and ambulatory) at the planning time. The genes (operating rooms) reserved for the surgical specialties corresponding to deferred urgency surgeries remain equal in all chromosomes. After creating the starting population, the quality of each chromosome is evaluated by computing its fitness value, i.e., the objective function value for the corresponding solution. The decoder is then required to obtain the solution associated to each chromosome of the population being evaluated (Section 3.1).

At the beginning of each iteration, after evaluating each individual, the population is sorted and partitioned into two groups of individuals (step 2): the `nPOPe` best fitted individuals compose the set of elite solutions and the remaining `nPOP`–`nPOPe` individuals make up the set of non-elite solutions. As it happens in random-key genetic algorithms, this GA uses an elitist strategy and thus the `nPOPe` elite individuals are copied to the next generation (step 3). Mutation is used in order to introduce some diversity in the new population and prevent the algorithm from getting stuck in a local optimum. In this algorithm mutation is applied by introducing, in each generation, `nPOPm` randomly generated individuals (*mutants*) [18]. Mutants are generated in the same way as the individuals of the initial population. The remaining `nPOP`–`nPOPe`–`nPOPm` individuals are produced by applying a crossover procedure where one parent is selected from the set of elite solutions and the other one from the set of non-elite solutions (being thus a *biased* GA) (step 4). The elite parent is uniformly selected from the set of elite individuals and the non-elite parent is uniformly selected among the non-elite individuals. Each pair of selected parents recombines to produce one child within this uniform crossover. Each gene of the children inherits the corresponding gene of the elite parent with probability ρ_e, i.e. for $i = 1, \ldots, Days \times Rooms$, children's gene i is set equal to elite parent's gene i with probability ρ_e and equal to non-elite parent's gene i with probability $1 - \rho_e$. Fig. 3 illustrates this recombination operator. If the selected parents can be decoded into feasible solutions, the resulting children also encode feasible solutions to the planning problem. This procedure is repeated until `nPOP`–`nPOPe`–`nPOPm` children are obtained. These new chromosomes join the elite individuals and the mutants thus creating a new generation (step 5). This new generation is then decoded and evaluated and all the process is repeated (from step 2) until a maximum number of generations (`nGER`) is reached. The best surgical plan is the solution corresponding to the best chromosome of the last population of the algorithm.

This GA deals with a population of solutions that evolves from an initial set of individuals by the action of the traditional genetic operators (selection, crossover and mutation). The crossover takes profit of the best characteristics already presented in the population whereas mutation brings new diversity. An important feature of this genetic heuristic is introduced by elitism, allowing not to loose the good solutions created during the evolutionary process as well as including a high-quality solution in the initial population. Two versions of the GA were developed according to the optimization criterion considered for the problem: maximizing the surgical suite occupation or maximizing the number of surgeries scheduled. The differences arise from the fitness function and from the decoder used to obtain a solution associated to a given chromosome. This GA can easily be adapted to include other constraints. For instance, the possibility to consider surgeons' or patients' unavailability in specific time periods only requires small adjustments in the constructive and improvement heuristics used as decoder. Table 1 summarizes the parameters required for this GA.

Table 1
GA parameters.

nGER	Number of generations
nPOP	Size of the population
pPOPe	Percentage of elite individuals in the population ($\mathtt{pPOPe} = \frac{\mathtt{nPOPe}}{\mathtt{nPOP}}$)
pPOPm	Percentage of mutants in the population ($\mathtt{pPOPm} = \frac{\mathtt{nPOPm}}{\mathtt{nPOP}}$)
ρ_e	Probability of inheriting a gene from the elite parent

Algorithm 2 Constructive decoder phase

1: Schedule all deferred urgency surgeries on Monday.
2: Schedule all high priority surgeries on the first day and room that can be scheduled.
3: Try to schedule the remaining (priority and normal) surgeries on the first day and room that can be scheduled.

3.1. Chromosome decoder

The chromosome decoder is based on the heuristics proposed by the authors in [8]. These heuristic procedures were developed to produce a surgical plan taking into account the elective surgery scheduling problem characteristics and constraints as defined in Section 2 and the assignment of operating rooms to surgical specialties stated by the corresponding chromosome. Therefore, the decoder is composed of two sequential parts: the *constructive* phase provides an initial solution and the *improvement* phase tries to improve the quality of this solution. In both phases, priority and normal surgeries are scheduled in order to accomplish the objective that is being optimized. When maximizing the surgical suite occupation, these surgeries are scheduled in descending order of estimated duration and the opposite happens if the objective is to maximize the number of surgeries scheduled. The constructive phase of the decoder is described in Algorithm 2. The improvement phase, specific to each criteria, is detailed in Algorithms 3 (surgical suite occupation) and 4 (waiting list reduction).

Step 1 of Algorithms 3 and 4 does not change directly the corresponding objective function value although it rearranges the surgeries scheduled in order to provide space for improving the solution in the following steps. Step 2, also common to both algorithms, allows the improvement of both objective functions, respectively the estimated duration and the number of surgeries scheduled. Steps 3 and 4 of Algorithm 3 attempt to recover the idle time between surgeries whilst steps 5 and 6 take profit of the idle time at the end of each day, all ensuring the improvement of the surgical suite occupation for each exchange performed. In Algorithm 4, steps 3 and 4 enable changing longer surgeries by two or three smaller ones thus increasing, respectively, by one or by two, the number of surgeries scheduled per each move undertaken. Step 5 has the same effect in the objective function value by taking profit of the idle time at the end of each day. Step 7 of Algorithm 3 and step 6 of Algorithm 4 extend the scope of the local search by enabling the exchange of the surgical specialty assigned to an operating room although ensuring that the move is performed to a solution which is better regarding the criterion being optimized.

This decoding technique is deterministic. The encoding and decoding specifically designed for this GA always produced feasible solutions from the chromosomes, for all the instances tested. Hence, the evaluation function measures up the objective function value of the associated solution. If, for some particular instance, an unfeasible solution is obtained, then the corresponding chromosome's fitness should be penalized.

The procedure to obtain the initial elitist chromosome is similar to the one used as decoder. The only difference results from the fact that the decoder has the surgical specialties previously assigned to the operating rooms given by the chromosome encoding whereas generating the initial elitist chromosome simultaneously allocates one surgical specialty to each operating room per day. Note that, a pre-processing is always required to define the genes on Monday reserved for the surgical specialties that perform deferred urgency surgeries. In practice, this pre-processing is part of the constructive and improvement heuristics used to obtain the initial elitist chromosome. Therefore, by running the procedure for all the operating rooms, days and surgeries, we obtain the elitist chromosome for the initial population without significantly penalizing the algorithm's execution time.

Algorithm 3 Improvement decoder phase: maximize the surgical suite occupation

1: Re-schedule surgeries as early as possible in the day, while retaining the same order.
2: Try to schedule unscheduled surgeries in the time available at the end of each day.
3: Try to exchange two consecutive scheduled surgeries, with priority or normal level of priority, for one unscheduled surgery with the longest estimated duration such that it is greater than the sum of the estimated durations of the two surgeries to unschedule and no greater than the sum of the estimated durations of the two surgeries to unschedule plus the idle time between each pair.
4: Try to exchange three consecutive scheduled surgeries, with priority or normal level of priority, for one unscheduled surgery with the longest estimated duration such that it is greater than the sum of the estimated durations of the three surgeries to unschedule and no greater than the sum of the estimated durations of the three surgeries to unschedule plus the idle time between each pair.
5: If the last scheduled surgery of the day is classified as priority or normal, try to exchange it for one unscheduled surgery with longer estimated duration, in order to occupy the remainder of the regular time in the day.
6: If the last scheduled surgery of the day is classified as priority or normal, try to exchange it for two unscheduled surgeries occupying the remainder of the regular time in the day and such that the sum of the estimated durations of the two surgeries to be scheduled is greater than the estimated duration of the surgery to unschedule.
7: Try to fill the idle time periods that remain in the room, by exchanging or transferring scheduled surgeries between operating rooms and exchanging the surgical specialty in an operating room. The exchange of the surgical specialty allocated to an operating room can only be executed if it originates a solution with more time periods scheduled. If at least one surgical specialty exchange occurs, return to step 3; otherwise, stop.

3.2. Other tested versions

Straightforward versions of the GA – v1 and v2 – were also developed for this elective surgery scheduling problem. Tests reported on Appendix B led to the selection of the version described above. Nevertheless a brief summary of these versions is presented in this Subsection.

Algorithm 5 describes the overall structure of v1 and v2. The differences rely mainly in the genetic operators. In each generation, nPOP parents are selected (step 2) from the population of chromosomes to form new individuals. The parent selection operator performs according to the uniform distribution (version v1) or using a roulette wheel selection (version v2). The tests performed did not provide clear preference for one type of selection operator although the roulette wheel selection operator provided slightly

$$\text{Parent 1:}\quad (\boxed{1\;3\;2\;1}\,1 \mid 1 \;/\; 1 \mid 2\;4\;1\;5\;1 \;/\; 4\;2\;1\;4\;5 \mid 1 \;/\; 2\;4\;4\;5 \mid 1\;1 \;/\; 1\;5\;4 \mid 4\;2\;1 \;)$$
$$\text{Parent 2:}\quad (\boxed{1\;3\;2\;1}\,4 \mid 3 \;/\; 1 \mid 4\;3\;5\;4\;1 \;/\; 1\;5\;4\;4\;3 \mid 4 \;/\; 4\;1\;2\;3 \mid 4\;1 \;/\; 5\;3\;2 \mid 1\;5\;1 \;)$$
$$c_1 \qquad c_2 \qquad c_3 \qquad c_4 \qquad c_5$$
$$\text{Child 1:}\quad (\boxed{1\;3\;2\;1}\,1 \mid 3 \;/\; 1 \mid 4\;3\;5\;4\;1 \;/\; 4\;2\;1\;4\;5 \mid 4 \;/\; 2\;4\;4\;5 \mid 4\;1 \;/\; 1\;5\;4 \mid 1\;5\;1 \;)$$
$$\text{Child 2:}\quad (\boxed{1\;3\;2\;1}\,4 \mid 1 \;/\; 1 \mid 2\;4\;1\;5\;1 \;/\; 1\;5\;4\;4\;3 \mid 1 \;/\; 4\;1\;2\;3 \mid 1\;1 \;/\; 5\;3\;2 \mid 4\;2\;1 \;)$$

Fig. 4. Recombination operator: multi-point crossover.

Algorithm 4 Improvement decoder phase: maximize the number of surgeries scheduled

1: Re-schedule surgeries as early as possible in the day, while retaining the same order.
2: Try to schedule unscheduled surgeries in the time available at the end of each day.
3: Try to exchange one scheduled surgery, with priority or normal level of priority, for two unscheduled surgeries whose total estimated duration (including the cleaning time between each pair) is no greater than the duration of the surgery to unschedule.
4: Try to exchange one scheduled surgery, with priority or normal level of priority, for three unscheduled surgeries whose total estimated duration (including the cleaning time between each pair) is no greater than the duration of the surgery to unschedule.
5: If the last surgery scheduled to the end of the day is classified as priority or normal, try to exchange it for two or three unscheduled surgery, possibly occupying the remainder of the regular time in the day.
6: Try to fill the idle time periods that remain in the room, by exchanging or transferring scheduled surgeries between operating rooms and exchanging the surgical specialty in an operating room. The exchange of the surgical specialty allocated to an operating room can be executed only if it originates a solution with more surgeries scheduled. If at least one surgical specialty exchange occurs, return to step 3; otherwise, stop.

better average results (see Table 16 on Appendix B). Each pair of selected parents recombines to produce two children (step 3). Thus, the recombination operator generates a number of new individuals equal to the size of the population (nPOP). This recombination uses a multi-point crossover operator. Reeves [20] suggests that increasing the number of crossover points (from the one-point crossover operator) improves the performance of a GA. Following this suggestion, in these versions of the algorithm, five crossover points, one per planning day, are randomly chosen according to the uniform distribution. The children chromosomes are constructed by cutting the parents at the crossover points and exchanging parental genes after these points, for each planning day. Therefore, for each day, the first child chromosome gets all genes equal to the first parent until the crossover point and the remaining genes equal to the second parent. The opposite happens with the second

Algorithm 5 Genetic algorithm scheme (v1 and v2)

1: Initialize a population of nPOP chromosomes and evaluate.
2: Select nPOP parents from the population of chromosomes.
3: Produce new children by recombining parents and apply mutation.
4: Create the new population and evaluate the respective chromosomes.
5: Stop, if a maximum number of generations (nGER) is met; otherwise, return to step 2.

child. This procedure results, in fact, in nine crossover points: five are randomly generated and the remaining four are fixed and correspond to the end of each planning day. This operator originates two new children chromosomes that, by the action of the decoder, generate two solutions for the elective surgery scheduling problem. The recombination operator is illustrated in Fig. 4. After applying the crossover operator, nPOP new individuals are then available to suffer mutation (yet in step 3). The mutation operator randomly changes the genes of the new chromosomes depending on two constant probability rates. First, it selects the chromosomes to suffer mutation (chromosome level) with probability $\mathtt{pMUT}_c$. The operator then chooses the genes to mutate with probability $\mathtt{pMUT}_g$ (gene level) from the chromosomes selected in the previous level. The genes reserved for the surgical specialties corresponding to deferred urgency surgeries are not subject to mutation. Genes chosen to mutate are exchanged by new genes which are randomly generated according to the distribution of the surgical specialties in the waiting list for surgery for the corresponding type of surgery (conventional and ambulatory). Different mutation techniques were also tested but produced worse results in the computational tests. One of those versions used a variable mutation rate proposed by Beasley and Chu [21] for the set covering problem. Details about this version are reported in [22]. Mutation using one level constant probability rate (simple mutation technique) was also tested, but it also produced worse average results. The reproduction technique used is generational replacement (step 4). Therefore the old population is fully replaced by the new individuals thus forming a new generation. The quality of each chromosome of the new generation is then evaluated. In order to avoid loosing the best fitted chromosome, an elitist strategy replaces the worst chromosome of each generation by the best chromosome found so far. Theoretical results indicate that the use of elitism increases the GA performance (see e.g. Davis [19]). Steps 2–4 are iteratively repeated until a maximum number of generations (nGER) is reached.

4. Computational experiments

The hospital under study provided a historical record with information on all the surgeries performed from 1 January 2004 to 28 December 2007, containing 21,050 elective surgeries. Details on this historical record are available in [13]. The duration for each surgery in the waiting list is set equal to the average value for the same surgical procedure obtained on the historical record. If one specific surgical procedure is not available in the historical record, the surgery duration gets the average value for the respective surgical specialty of the corresponding surgery type (conventional or ambulatory). In addition, the hospital surgical plan and the respective hospital record were provided for seven specific weeks along with the waiting list for surgery at the moment when the planning occurred (the Friday before each planning week). These waiting lists had about 2200 elective surgeries on average: 87.5% conventional surgeries and 12.5% ambulatory surgeries; deferred urgency surgeries and high priority surgeries correspond to less than 1% of each type of surgery. Five real instances were tested for each planning week. The one with higher dimension contains

Table 2
Instances' characteristics.

Operating rooms	6
Planning days	4–5
Time periods	46
Duration of a time period	15 min
Surgical specialties	5
Surgeries	508–2306
Surgeries estimated duration	1–18 time periods
Surgery priority levels	4
Surgeons	45–61
Surgeon daily operating time limit	24 time periods
Surgeon weekly operating time limit	100 time periods

all the surgeries in the waiting list. The remaining four instances consider a subset of 250, 300, 500 and 1000 conventional surgeries in the waiting list. This provides different sized instances to test the performance of the GA. The selection of conventional surgeries from the waiting list was made in order to contain all the surgeries included in the hospital surgical plan for the corresponding week, to enable a comparison, followed by a selection of surgeries by order of priority until attaining the required number of conventional surgeries. Thus, the instances of higher dimension include all the surgeries of the instances of smaller dimension for the same planning week. Nevertheless, instances of higher dimension give greater flexibility to the surgery scheduling and are closer to the real situation. All five instances for the same planning week have the same number of ambulatory surgeries (275 on average). Therefore, instances contain between 508 and 2306 elective surgeries. In the authors' previous approaches to the same problem [13,8], only the four instances with a subset of conventional surgeries were tested (including 508 to 1295 elective surgeries). Note that, in these studies, the problem was decomposed in two hierarchical phases while in the current approach the decomposition was not revealed to be necessary.

The number of surgeons ranges from 45 to 61. All instances refer to a surgical suite with 6 operating rooms, 4 or 5 planning days (the last planning week under study has a holiday on Friday), 5 surgical specialties and 46 daily time periods (of 15 min) in regular working time. The duration of conventional surgeries varies from 2 to 18 time periods, with a mean value of 5.3 time periods and a standard deviation of 2.8; and the duration of ambulatory surgeries ranges from 1 to 8 time periods, with a mean value of 2.9 time periods and a standard deviation of 0.8. Table 2 summarizes the characteristics of the real instances used in the computational experiments.

The GA was coded in C++ language and ran on an Intel Core i7-3610QM Processor, 2.30 GHz computer with 8 GB of RAM. The computational experiments were performed with the above described 35 real instances. The GA was repeated 25 times for each instance tested. Different GA versions were previously tested as well as several values for the parameters used in each version as mentioned in Section 3.2. In this section only the results obtained with the GA's actual version are presented along with the parameter values that gave rise to the better average results (Table 3). Appendix B reports on the methodology used to tune the algorithm parameters and to select the best version among those tested. Some preliminary results are also presented in Appendix B.

The computational results are compiled in Tables 4–8. The performance of the GA is summarized in Tables 4 and 5. These tables show the average results obtained over the 25 runs of the GA for each tested instance with the objectives of maximizing, respectively, the surgical suite occupation or the number of surgeries scheduled. Table 6 provides details on the effect of the initial elitist chromosome over the quality of the solutions obtained by the GA. Table 7 compares the performance of the GA with an integer linear programming approach for the same problem thus allowing to evaluate the quality of the solutions obtained with the genetic heuristic. Finally, Table 8 gives some surgical suite

Table 3
Best values for GA parameters.

nGER	500	pPOPe	25%	ρ_e	60%
nPOP	100	pPOPm	20%		

productivity measures associated to the surgical plans resulting from the GA solutions.

Tables 4 and 5 are similar in structure. The instance is shown in column 1. Column 2 refers to the GA average running time. The objective value of the solution corresponding to the initial elitist chromosome is included in column 3 (*Elitist chrom. fitness*). Columns 4–6 refer to the fitness value of the best individual in the last generation, respectively the lowest, the mean and the highest value obtained in the 25 runs of the algorithm. The average percentage of improvement from the initial elitist chromosome is shown in column 7 (*Improv. gap*[1]). Column 8 (*Best gen.*) presents the generation that first originated the best chromosome, on average. The average percentage of chromosomes in the last generation with fitness equal to the best individual is shown in column 9 (*Equal fitted last gen.*). Column 10 (*# Runs equal to best*) refers to the number of runs that reached the best solution found (column 6) over the 25 runs of the algorithm for the same instance. The number of runs that failed to improve the initial elitist chromosome is given in column 11 (*# Runs without improv.*). Table 6 shows the running time and the (average) gap[2] for the heuristics used to obtain the initial elitist chromosome (*H*: columns 2–3 and 8–9), for the GA without this elitist chromosome (*GA without*: columns 6–7 and 12–13) and for the actual GA version (*GA with*: columns 4–5 and 10–11), respectively with the objectives of maximizing the surgical suite occupation and maximizing the number of surgeries scheduled.

The GA consumes low computational time (between 22 and 240 s) and improves the solution corresponding to the initial elitist chromosome in virtually all the tests performed. An exception occurred for all the 25 runs of instance I1_2042 when maximizing the number of surgeries scheduled (column 11 in Table 5). The genetic heuristic also shows good performance on convergence (column 9) and on the number of runs that originated a solution equal to the best found in the respective instance (column 10). On average, when maximizing the number of surgeries scheduled (Table 5), the GA needs smaller computational times (with the singular exception of the 7th planning week—I7), obtains earlier the best chromosome, has greater improvement over the initial elitist chromosome and ends with fewer chromosomes equal to the best individual, when compared with the results obtained considering the maximization of the surgical suite occupation (Table 4). When maximizing the surgical suite occupation (Table 4), the GA consumes more computational time for the higher instances, although the running time is less dependent on the instances dimension when maximizing the number of surgeries scheduled (Table 5). The results in columns 8 show that, on average, the best chromosome is obtained relatively early in the evolutive process. In fact, even though performing 500 generations, the mean value (and the standard deviation) for the generation that first obtained the best individual is 86.1 (91.8) when maximizing the surgical suite occupation and 45.1 (80.7) when maximizing the number of surgeries scheduled. The best chromosome was obtained in a generation above 250 for nearly 8% and 4% of the runs, respectively, for the first and the second objective function, and the latest ones were found in generation 494 and 489, respectively. Results in columns 9 show that this GA performs

1 Improvement gap $= \frac{\text{Best chromosome mean fitness (5)} - \text{Elitist chromosome fitness (3)}}{\text{Elitist chromosome fitness (3)}}$.

2 Gap $= \frac{\text{upper bound} - \text{(average) solution value}}{\text{(average) solution value}}$.

Table 4
GA average results over 25 runs: maximize the surgical suite occupation.

(1) Instance	(2) Time (s)	(3) Elitist chrom. fitness	(4) Best chrom. fitness	(5)	(6)	(7) Improv. gap	(8) Best gen.	(9) Best fitted chrom. last gen.	(10) # Runs equal to best	(11) # Runs without improv.
			Min	Mean	Max					
I1_250	50.6	1071	1074	1074.9	1075	0.4%	87.8	37.0%	23	0
I1_300	51.5	1083	1093	1093	1093	0.9%	82.2	36.6%	25	0
I1_500	61.8	1114	1122	1122.9	1123	0.8%	45.8	35.3%	23	0
I1_1000	81.5	1132	1138	1138	1138	0.5%	18.9	31.8%	25	0
I1_2042	148.9	1137	1141	1141	1141	0.4%	30.8	32.0%	25	0
I2_250	51.8	1072	1082	1082.6	1083	1.0%	40.7	39.6%	15	0
I2_300	56.8	1077	1091	1091.8	1092	1.4%	120.2	38.2%	20	0
I2_500	62.4	1108	1115	1115.9	1116	0.7%	66.1	36.4%	22	0
I2_1000	97.7	1122	1129	1129	1129	0.6%	32.5	32.3%	25	0
I2_1982	123.5	1123	1128	1128	1128	0.4%	30.4	33.4%	25	0
I3_250	56.3	1073	1080	1080.6	1081	0.7%	97.6	32.4%	15	0
I3_300	62.3	1088	1096	1096.8	1098	0.8%	130.0	38.8%	1	0
I3_500	72.0	1115	1123	1124.5	1126	0.9%	123.8	42.4%	1	0
I3_1000	113.1	1134	1139	1140.0	1141	0.5%	82.2	39.2%	2	0
I3_1944	189.7	1139	1145	1145.8	1146	0.6%	90.5	38.9%	19	0
I4_250	69.0	1082	1091	1092.3	1094	1.0%	125.5	37.0%	1	0
I4_300	65.4	1097	1105	1106.1	1109	0.8%	135.4	37.0%	3	0
I4_500	70.4	1120	1127	1128.4	1130	0.8%	128.8	38.8%	1	0
I4_1000	116.3	1144	1149	1150.5	1152	0.6%	140.0	40.6%	2	0
I4_1899	193.7	1149	1152	1153.8	1154	0.4%	90.6	40.0%	21	0
I5_250	60.2	1084	1096	1097.0	1097	1.2%	69.8	39.4%	24	0
I5_300	68.6	1095	1106	1106.8	1108	1.1%	125.8	40.2%	1	0
I5_500	75.8	1126	1129	1132.2	1133	0.6%	113.2	41.0%	12	0
I5_1000	109.2	1138	1148	1150.5	1153	1.1%	116.4	37.9%	1	0
I5_1866	207.8	1146	1152	1153.4	1154	0.6%	98.6	38.8%	13	0
I6_250	50.0	1084	1099	1099.9	1101	1.5%	88.8	40.2%	1	0
I6_300	46.8	1090	1109	1109.9	1110	1.8%	116.7	39.4%	22	0
I6_500	71.6	1114	1128	1130.5	1133	1.5%	147.1	38.7%	1	0
I6_1000	106.9	1129	1138	1140.0	1141	1.0%	71.0	38.0%	2	0
I6_1887	139.2	1130	1142	1143.8	1144	1.2%	45.3	37.8%	23	0
I7_250	35.5	882	892	893.0	894	1.3%	67.1	39.0%	2	0
I7_300	38.6	885	894	895.5	896	1.2%	103.9	40.0%	14	0
I7_500	47.4	897	906	906.9	907	1.1%	55.9	38.4%	23	0
I7_1000	73.4	909	916	916.8	917	0.9%	66.4	38.2%	20	0
I7_1897	103.5	909	920	920	920	1.2%	26.0	33.3%	25	0
Average						0.9%	86.1	37.7%	13.7	0

a diversified search throughout all the generations, avoiding a premature convergence to a non-optimal solution. The percentage of improvement over the initial elitist chromosome (column 7) tends to be higher for the smaller instances, particularly when maximizing the number of surgeries scheduled (Table 5). Although this percentage is revealed to be small, the GA improves the quality of the solutions obtained by the constructive and improvement heuristics (represented by the initial elitist chromosome) (see also Table 6) getting closer (or even improving) the quality of the solutions given by the integer linear programming approach (Table 7) with a small computational time. The heuristic procedure needs to be partially run in order to fix the operating rooms reserved to deferred urgency surgeries. Table 6 shows that the entire heuristic procedure uses negligible computational time and has no significant effect on the average quality of the solutions obtained by the GA as well as on the corresponding running time.

The integer linear programming models presented in Appendix A were solved by CPLEX 12.4 [23] using the same 35 test instances. Time limit to run CPLEX was set to 48 h (172,800 s). To allow a comparison of the genetic heuristic approach, here the model considers the overall problem without decomposing it into two hierarchical phases as happened in the authors' previous study [13]. The results are presented in Table 7. This table shows the running time and the best gap for the integer linear programming approach (columns 2–3 and 6–7) and for the genetic heuristic approach (columns 4–5 and 8–9), with the first and the second objective functions (respectively, maximizing the surgical suite occupation and maximizing the number of surgeries scheduled).

The GA obtained a better average gap with the objective function of maximizing the surgical suite occupation whereas for the second objective function the IL approach produced a higher average gap. Nevertheless, the exact approach gets slightly tighter gaps if considering the overall results (IL: 2.42%; GA: 2.69%). However, the running time is clearly the most significant difference between these two approaches. While CPLEX ran for more than 29 h (on average), the AG approach needed less than 0.07% of the CPU time (67 s, on average). The average running time for the exact approach is less than 48 h (time limit) because in 50 (71.4%) instances CPLEX exited with out of memory status. Since the surgical plan has implications on the daily lives of many people (mainly in the patient's life and in the surgeon working schedule), it becomes unsustainable to take more than one day to get a surgical plan which will be held in the following week and can even suffer online changes during the course of the week.

Table 8 presents operating room productivity measures associated to the surgical plans obtained with the GA for each of the objective functions: the operating room occupation rate without cleaning time (columns 3 and 6) and the waiting list reduction rate (columns 4 and 7). The results presented in this table correspond to the average values over the seven planning weeks for each instance dimension (column 1). These indicators assess the surgical plans in light of the objectives stated for this work. The surgical plans

Table 5
GA average results over 25 runs: maximize the number of surgeries scheduled.

(1) Instance	(2) Time (s)	(3) Elitist chrom. fitness	(4) Best chrom. fitness Min	(5) Mean	(6) Max	(7) Improv. gap	(8) Best gen.	(9) Best fitted chrom. last gen.	(10) # Runs equal to best	(11) # Runs without improv.
I1_250	38.0	233	241	241.6	242	3.7%	124.5	30.6%	16	0
I1_300	38.2	250	255	255	255	2.0%	56.6	35.0%	25	0
I1_500	28.8	268	270	270	270	0.7%	8.2	34.8%	25	0
I1_1000	36.4	286	287	287	287	0.3%	52.8	40.7%	25	0
I1_2042	43.6	300	300	300	300	0.0%	1.0	42.1%	25	25
I2_250	33.1	225	234	234.1	235	4.0%	52.7	36.9%	2	0
I2_300	33.0	232	245	245.8	246	5.9%	81.0	33.2%	20	0
I2_500	30.7	263	268	268.0	269	1.9%	13.7	35.6%	1	0
I2_1000	32.9	276	285	285	285	3.3%	38.8	38.7%	25	0
I2_1982	48.2	290	297	297.3	298	2.5%	78.7	41.5%	8	0
I3_250	32.1	241	248	248	248	2.9%	18.9	34.0%	25	0
I3_300	34.3	244	252	252	252	3.3%	22.8	33.4%	25	0
I3_500	31.7	266	271	271	271	1.9%	15.9	30.3%	25	0
I3_1000	31.4	283	287	287	287	1.4%	5.7	39.6%	25	0
I3_1944	44.8	297	300	300	300	1.0%	53.8	41.7%	25	0
I4_250	37.4	239	243	244.2	245	2.2%	90.5	34.8%	7	0
I4_300	38.1	250	252	253.1	254	1.2%	148.0	33.2%	5	0
I4_500	26.6	269	272	272	272	1.1%	6.6	32.9%	25	0
I4_1000	42.2	286	289	289.2	290	1.1%	32.2	38.1%	6	0
I4_1899	35.4	300	302	302	302	0.7%	8.9	44.5%	25	0
I5_250	42.7	239	246	247.4	248	3.5%	163.3	34.1%	10	0
I5_300	34.7	253	258	258.2	259	2.0%	43.7	35.2%	4	0
I5_500	34.5	268	275	275	275	2.6%	29.8	29.4%	25	0
I5_1000	36.5	286	289	289	289	1.0%	4.9	43.8%	25	0
I5_1866	44.3	300	302	302	302	0.7%	6.6	33.2%	25	0
I6_250	37.8	234	240	240	240	2.6%	15.8	31.7%	25	0
I6_300	35.7	247	250	250.9	251	1.6%	62.8	37.5%	22	0
I6_500	31.5	263	266	266	266	1.1%	5.6	35.7%	25	0
I6_1000	38.2	281	283	283	283	0.7%	2.9	36.7%	25	0
I6_1887	49.2	297	298	298	298	0.3%	8.0	33.2%	25	0
I7_250	65.7	202	204	204.6	205	1.3%	110.7	31.3%	15	0
I7_300	68.8	207	211	212.0	213	2.4%	116.2	38.2%	1	0
I7_500	109.0	223	226	226	226	1.3%	95.4	38.2%	25	0
I7_1000	150.8	238	239	239	239	0.4%	1.0	41.6%	25	0
I7_1897	159.2	253	255	255	255	0.8%	1.1	34.2%	25	0
Average						1.8%	45.1	36.2%	19.1	0.7

resulting from the GA represent good scheduling performance levels and improve the hospital productivity measures with respect to both criteria. As mentioned in Section 1, at the beginning of our study, the hospital was performing a regular time occupancy rate of the surgical suite of about 42% with a waiting list reduction rate of 5.5%. Table 8 confirms the conflictuous nature of both objective functions and shows the increasing of productivity levels for the higher dimension instances with respect to the indicator that reflects the optimized objective function.

5. Conclusions

This paper contributes with a population based approach to solve an elective surgery scheduling problem applied to real case instances and with the specifications of the Portuguese hospital under study. The problem combines simultaneously advance and allocation scheduling, which is rarely tackled in the literature. Two optimization criteria are considered: maximizing the surgical suite occupation and maximizing the number of surgeries scheduled. Due to the high complexity of the problem and the large dimension attained in real instances, in the authors' previous approaches the problem was decomposed into two hierarchical phases according to the nature of the surgeries. The approach developed in this paper gets closer to the real situation as it tackles the overall problem. Moreover, this approach yields better results with respect to the quality of the solutions and the computational time than previous approaches to the same problem, thus providing the opportunity to test instances with the entire waiting list for surgery for the first time. Instances with 508–2306 elective surgeries were successfully solved in 22–240 s using the genetic heuristic. The number of surgical cases considered for scheduling is higher than most of published results while time to get a good surgical plan is reduced. This procedure thus allows considering a large number of surgeries giving greater flexibility in planning and enabling to achieve better surgical plans. Furthermore, this approach requires much less staff working time than the current manual procedure. Therefore this work contributes to increase the efficiency of the hospital surgical suite and to reduce the hospital's waiting list for surgery, with a significative impact on the hospital management and in the quality of life of those waiting for a surgery at this hospital and thus fulfilling the aim initially proposed for the project.

This methodology for scheduling elective surgeries reflects the hospital under study. It can be easily adapted to include other constraints. Until now, beds in the recovery units and wards do not limit the activity of the surgical suite. Whether this statement remains valid when facing a great improvement in surgical productivity is an open question for future analysis. In the future the authors also intend to approach the problem as a bicriteria one considering both objective functions, independently optimized in this paper. In fact, the genetic algorithm, being a population based procedure, is well suited to deal with more than one criterion and will be therefore extended accordingly.

Table 6
Details on the effect of the initial elitist chromosome.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Instance	Max. surgical suite occupation						Max. nbr surgeries scheduled					
	H		GA with		GA without		H		GA with		GA without	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
I1_250	0	3.17	50.6	2.80	46.1	2.79	0	8.58	38.0	4.70	35.3	4.72
I1_300	0	3.79	51.5	2.84	46.7	2.84	0	5.20	38.2	3.14	35.7	3.15
I1_500	0	3.41	61.8	2.59	56.2	2.59	0	4.85	28.8	4.07	26.5	4.07
I1_1000	0	3.00	81.5	2.46	75.7	2.46	0.02	5.24	36.4	4.88	33.6	4.89
I1_2042	0	2.99	148.9	2.63	134.9	2.63	0	3.67	43.6	3.67	53.2	3.67
I2_250	0	3.73	51.8	2.72	47.0	2.70	0.02	9.78	33.1	5.52	32.6	5.50
I2_300	0	3.99	56.8	2.58	51.4	2.57	0	11.64	33.0	5.37	33.4	5.37
I2_500	0	3.79	62.4	3.06	56.9	3.07	0	6.46	30.7	4.46	31.8	4.46
I2_1000	0	3.39	97.7	2.75	87.9	2.75	0	7.97	32.9	4.56	34.0	4.56
I2_1982	0	3.74	123.5	3.28	123.6	3.27	0	6.21	48.2	3.59	47.1	3.62
I3_250	0	2.05	56.3	1.33	51.2	1.34	0	5.39	32.1	2.42	32.6	2.42
I3_300	0	2.21	62.3	1.39	57.6	1.39	0	6.56	34.3	3.17	34.4	3.17
I3_500	0.02	2.42	72.0	1.55	66.9	1.59	0	6.02	31.7	4.06	31.1	4.06
I3_1000	0	1.94	113.1	1.40	108.6	1.40	0	6.36	31.4	4.88	32.8	4.88
I3_1944	0	2.02	189.7	1.42	173.0	1.42	0	4.71	44.8	3.67	46.1	3.67
I4_250	0.01	3.05	69.0	2.08	66.4	2.06	0	5.44	37.4	3.19	38.0	3.11
I4_300	0	2.37	65.4	1.53	62.7	1.49	0	4.00	38.1	2.73	37.7	2.73
I4_500	0	2.59	70.4	1.82	65.8	1.84	0	4.83	26.6	3.68	27.6	3.68
I4_1000	0	1.84	116.3	1.26	106.6	1.31	0	5.59	42.2	4.41	35.4	4.45
I4_1899	0	1.91	193.7	1.49	175.7	1.58	0	4.00	35.4	3.31	35.5	3.31
I5_250	0	2.40	60.2	1.19	56.3	1.19	0	6.28	42.7	2.68	41.5	2.68
I5_300	0	2.19	68.6	1.10	62.7	1.11	0	4.35	34.7	2.26	34.6	2.26
I5_500	0	1.95	75.8	1.39	71.6	1.38	0.02	5.97	34.5	3.27	35.2	3.27
I5_1000	0	2.55	109.2	1.43	100.0	1.43	0	5.59	36.5	4.50	37.3	4.48
I5_1866	0	2.09	207.8	1.44	200.7	1.46	0	4.00	44.3	3.31	45.9	3.31
I6_250	0	3.51	50.0	2.01	46.3	2.01	0	4.27	37.8	1.67	37.5	1.67
I6_300	0	3.67	46.8	1.81	44.6	1.82	0	4.05	35.7	2.44	37.8	2.52
I6_500	0	3.68	71.6	2.17	65.2	2.18	0	4.56	31.5	3.38	32.0	3.38
I6_1000	0	3.10	106.9	2.11	98.3	2.12	0	5.34	38.2	4.59	38.2	4.59
I6_1887	0	3.27	139.2	2.02	124.9	2.03	0	3.37	49.2	3.02	49.7	3.02
I7_250	0	3.29	35.5	2.01	34.4	2.01	0	3.47	65.7	2.15	66.7	2.21
I7_300	0.01	3.16	38.6	1.95	35.8	1.96	0	4.83	68.8	2.38	68.7	2.36
I7_500	0	3.68	47.4	2.54	43.8	2.55	0.02	4.48	109.0	3.10	109.6	3.12
I7_1000	0	3.19	73.4	2.31	65.8	2.32	0.02	3.78	150.8	3.35	151.6	3.35
I7_1897	0	3.41	103.5	2.17	95.7	2.16	0	2.77	159.2	1.96	160.7	1.96
Average	0.0	2.93	86.5	2.02	80.2	2.02	0.0	5.42	47.3	3.53	47.5	3.53

Table 7
Comparison of integer linear approach and genetic heuristic approach.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Instance	Max. surgical suite occupation				Max. nbr surgeries scheduled			
	IL approach		GA approach		IL approach		GA approach	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
I1_250	172 809.2	1.94	50.6	2.79	54 062.9	4.98	38.0	4.55
I1_300	66 849.4	2.09	51.5	2.84	75 134.3	3.14	38.2	3.14
I1_500	72 183.9	2.22	61.8	2.58	93 813.5	3.69	28.8	4.07
I1_1000	173 959.2	2.55	81.5	2.46	153 275.7	3.44	36.4	4.88
I1_2042	16 883.1	8.23	148.9	2.63	177 922.5	6.14	43.6	3.67
I2_250	132 200.6	2.11	51.8	2.68	80 400.0	2.07	33.1	5.11
I2_300	62 172.5	2.85	56.8	2.56	80 427.3	1.97	33.0	5.28
I2_500	66 395.3	2.40	62.4	3.05	105 689.0	2.19	30.7	4.09
I2_1000	75 831.5	2.29	97.7	2.75	151 956.9	2.76	32.9	4.56
I2_1982	173 361.9	7.18	123.5	3.28	174 623.9	5.12	48.2	3.36
I3_250	50 049.9	1.20	56.3	1.30	108 002.5	2.42	32.1	2.42
I3_300	172 805.7	0.63	62.3	1.28	107 796.5	2.36	34.3	3.17

(*continued on next page*)

Acknowledgments

The authors would like to thank Dr. Manuel Delgado for his enthusiastic interest in this work, Dra. Margarida Baltazar for her patient provision of all necessary data and head nurse Fátima Menezes for the friendly and patient description of the entire process. This research is supported by Portuguese National Funding from FCT—Fundação para a Ciência e a Tecnologia, under the project: PEst-OE/MAT/UI0152.

Appendix A. Mathematical models

The integer linear programming models involve the parameters presented in Table 9 and are formulated as follows:

Table 7 (*continued*)

(1) Instance	(2) Max. surgical suite occupation	(3)	(4)	(5)	(6) Max. nbr surgeries scheduled	(7)	(8)	(9)
	IL approach		GA approach		IL approach		GA approach	
	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %	Time (s)	Gap %
I3_500	51 275.7	0.97	72.0	1.42	118 338.3	2.92	31.7	4.06
I3_1000	74 322.8	2.48	113.1	1.31	173 667.2	3.08	31.4	4.88
I3_1944	173 239.0	5.25	189.7	1.40	177 465.6	2.98	44.8	3.67
I4_250	127 190.4	0.72	69.0	1.92	93 764.9	1.61	37.4	2.86
I4_300	44 704.1	1.63	65.4	1.26	111 707.7	1.17	38.1	2.36
I4_500	58 041.3	1.06	70.4	1.68	129 524.0	2.92	26.6	3.68
I4_1000	54 871.9	2.46	116.3	1.13	173 030.9	3.42	42.2	4.14
I4_1899	173 631.4	5.12	193.7	1.47	174 101.8	4.00	35.4	3.31
I5_250	39 438.3	1.00	60.2	1.19	63 426.2	2.01	42.7	2.42
I5_300	40 213.7	1.08	68.6	0.99	81 898.5	1.15	34.7	1.93
I5_500	56 291.5	1.06	75.8	1.32	172 832.3	2.16	34.5	3.27
I5_1000	51 804.9	1.21	109.2	1.21	163 924.4	2.37	36.5	4.50
I5_1866	178 831.0	1.47	207.8	1.39	177 357.9	2.97	44.3	3.31
I6_250	40 932.7	1.63	50.0	1.91	70 115.4	1.24	37.8	1.67
I6_300	59 109.9	2.63	46.8	1.80	54 132.8	1.18	35.7	2.39
I6_500	67 688.7	1.49	71.6	1.94	89 835.8	1.48	31.5	3.38
I6_1000	66 287.5	2.65	106.9	2.02	104 638.4	1.72	38.2	4.59
I6_1887	179 118.3	3.73	139.2	2.01	178 843.2	1.66	49.2	3.02
I7_250	49 024.2	1.11	35.5	1.90	67 046.0	1.46	65.7	1.95
I7_300	71 521.4	1.44	38.6	1.90	112 575.9	1.40	68.8	1.88
I7_500	86 523.6	2.31	47.4	2.54	40 017.7	2.19	109.0	3.10
I7_1000	172 956.8	2.07	73.4	2.29	73 693.1	0.82	150.8	3.35
I7_1897	173 598.4	2.08	103.5	2.17	174 555.3	0.78	159.2	1.96
Average	95 032.0	2.35	86.5	1.95	118 274.2	2.48	47.3	3.43

Table 8
Average surgical suite productivity measures.

(1) Instance size	(2) Max. surgical suite occupation	(3)	(4)	(5) Max. nbr surgeries scheduled	(6)	(7)
	Time (s)	OR occup. rate (%)	WL reduct. rate (%)	Time (s)	OR occup. rate (%)	WL reduct. rate (%)
I_250	53.4	79.1	7.3	41.0	65.0	10.7
I_300	55.7	80.0	7.1	40.4	64.0	11.2
I_500	65.9	81.7	6.6	41.8	61.0	12.0
I_1000	99.7	82.8	6.4	52.6	58.3	12.7
I_complete	158.0	83.0	6.2	60.7	57.4	13.3

Decision variables

$$x_{crtd} = \begin{cases} 1, & \text{if surgery } c \text{ starts at the beginning of period } t \\ & \text{on day } d \text{ in room } r \\ 0, & \text{otherwise } (c \in C,\ r \in R,\ t \in T_c,\ d \in D). \end{cases}$$

Additional variables

$$y_{jrd} = \begin{cases} 1, & \text{if a surgery of specialty } j \text{ starts in} \\ & \text{room } r \text{ on day } d \\ 0, & \text{otherwise } (j \in E,\ r \in R,\ d \in D). \end{cases}$$

Objective functions

- Maximize the surgical suite occupation

$$\max\ z_1 = \sum_{c\in CS}\sum_{r\in R^{CS}}\sum_{t\in T_c}\sum_{d\in D} p_c\, x_{crtd} + \sum_{c\in AS}\sum_{r\in R^{AS}}\sum_{t\in T_c}\sum_{d\in D} p_c\, x_{crtd}.$$

- Maximize the number of surgeries scheduled

$$\max\ z_2 = \sum_{c\in CS}\sum_{r\in R^{CS}}\sum_{t\in T_c}\sum_{d\in D} x_{crtd} + \sum_{c\in AS}\sum_{r\in R^{AS}}\sum_{t\in T_c}\sum_{d\in D} x_{crtd}.$$

Table 9
Parameters of the mathematical models.

C	Set of elective surgeries selected from the waiting list
CS	Set of conventional surgeries selected from the waiting list
AS	Set of ambulatory surgeries selected from the waiting list
E	Set of surgical specialties
P	Set of surgery priority levels
CS_j^{SP}	Set of conventional surgeries of specialty $j \in E$
AS_j^{SP}	Set of ambulatory surgeries of specialty $j \in E$
CS_i^{PR}	Set of conventional surgeries classified with priority level $i \in P$
AS_i^{PR}	Set of ambulatory surgeries classified with priority level $i \in P$
R	Set of operating rooms available for scheduling
R^{CS}	Set of operating rooms dedicated to conventional surgeries
R^{AS}	Set of operating rooms dedicated to ambulatory surgeries
D	Set of days in the planning horizon
T	Set of daily time periods available for scheduling
h_c	Surgeon assigned to surgery $c \in C$
p_c	Number of time periods required to perform surgery $c \in C$
T_c	Set of possible time periods to schedule surgery $c \in C$ in order to be completed within the surgical suite regular working hours
$T_{hd}^{MAX_D}$	Operating time limit of surgeon $h \in H$ in day $d \in D$
$T_h^{MAX_W}$	Weekly operating time limit of surgeon $h \in H$

Constraints

$$\sum_{r\in R^{CS}}\sum_{t\in T_c} x_{crt1} = 1, \quad \forall\, c \in CS_1^{PR} \tag{c1C}$$

$$\sum_{r\in R^{AS}}\sum_{t\in T_c} x_{crt1} = 1, \quad \forall\, c \in AS_1^{PR} \tag{c1A}$$

$$\sum_{r\in R^{CS}}\sum_{t\in T_c}\sum_{d\in D} x_{crtd} = 1, \quad \forall\, c \in CS_2^{PR} \tag{c2C}$$

$$\sum_{r\in R^{AS}}\sum_{t\in T_c}\sum_{d\in D} x_{crtd} = 1, \quad \forall\, c \in AS_2^{PR} \tag{c2A}$$

$$\sum_{r\in R^{CS}}\sum_{t\in T_c}\sum_{d\in D} x_{crtd} \le 1, \quad \forall\, c \in CS \setminus (CS_1^{PR} \cup CS_2^{PR}) \tag{c3C}$$

Table 10
nGER: 200/400/500.

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness	(4)	(5)	(6) Improv. gap	(7) Average deviation
		Min	Mean	Max		
nGER = 200						
z_1	37.16	1141.90	1143.54	1144.74	0.798%	1.457
z_2	10.93	298.37	298.73	298.99	0.583%	0.269
nGER = 400						
z_1	74.39	1142.45	1143.99	**1144.98**	0.838%	1.012
z_2	22.59	298.43	298.81	298.99	0.609%	0.191
nGER = **500**						
z_1	91.92	**1142.51**	**1144.11**	1144.96	**0.848**%	**0.895**
z_2	28.94	**298.50**	**298.82**	**299.00**	**0.614**%	**0.177**

$$\sum_{r\in R^{AS}}\sum_{t\in T_c}\sum_{d\in D} x_{crtd} \le 1, \quad \forall\, c \in AS \setminus (AS_1^{PR} \cup AS_2^{PR}) \tag{c3A}$$

$$\sum_{c\in CS}\sum_{\substack{t'=t-p_c+1-\gamma\\ t'\in T_c}}^{t} x_{crt'd} \le 1, \quad \forall\, r \in R^{CS},\ t \in T,\ d \in D \tag{c4C}$$

$$\sum_{c\in AS}\sum_{\substack{t'=t-p_c+1-\gamma\\ t'\in T_c}}^{t} x_{crt'd} \le 1, \quad \forall\, r \in R^{AS},\ t \in T,\ d \in D \tag{c4A}$$

$$\sum_{j\in E} y_{jrd} \le 1, \quad \forall\, r \in R,\ d \in D \tag{c5}$$

$$\sum_{c\in CS_j^{SP}}\sum_{t\in T_c} x_{crtd} \le y_{jrd}|T|, \quad \forall\, j \in E,\ r \in R^{CS},\ d \in D \tag{c6C}$$

$$\sum_{c\in AS_j^{SP}}\sum_{t\in T_c} x_{crtd} \le y_{jrd}|T|, \quad \forall\, j \in E,\ r \in R^{AS},\ d \in D \tag{c6A}$$

$$\sum_{\substack{c\in CS:\\ h_c=h}}\sum_{r\in R^{CS}}\sum_{\substack{t'=t-p_c+1\\ t'\in T_c}}^{t} x_{crt'd} + \sum_{\substack{c\in AS:\\ h_c=h}}\sum_{r\in R^{AS}}\sum_{\substack{t'=t-p_c+1\\ t'\in T_c}}^{t} x_{crt'd} \le 1, \quad \forall\, h \in H,\ t \in T,\ d \in D \tag{c7}$$

$$\sum_{\substack{c\in CS:\\ h_c=h}}\sum_{r\in R^{CS}}\sum_{t\in T_c} p_c x_{crtd} + \sum_{\substack{c\in AS:\\ h_c=h}}\sum_{r\in R^{AS}}\sum_{t\in T_c} p_c x_{crtd} \le T_{hd}^{\mathrm{MAX}_D}, \quad \forall h \in H,\ d \in D \tag{c8}$$

Table 12
pMUT_g : 5%/10%/15%/20% (nGER = 500, nPOP = 100).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness	(4)	(5)	(6) Improv. gap	(7) Average deviation
		Min	Mean	Max		
$\mathrm{pMUT}_g = 5\%$						
z_1	151.99	1141.75	1144.00	1145.00	0.839%	1.000
z_2	43.87	298.50	**298.93**	299.00	**0.648**%	**0.075**
pMUT_g = **10%**						
z_1	147.72	**1142.25**	**1144.03**	1145.00	**0.841**%	**0.975**
z_2	45.67	298.50	**298.93**	299.00	**0.648**%	**0.075**
$\mathrm{pMUT}_g = 15\%$						
z_1	149.74	1142.00	1143.80	1145.00	0.821%	1.200
z_2	47.93	298.50	298.88	299.00	0.631%	0.125
$\mathrm{pMUT}_g = 20\%$						
z_1	150.47	1142.00	1143.53	1144.50	0.797%	1.475
z_2	49.31	298.50	**298.93**	299.00	**0.648**%	**0.075**

$$\sum_{\substack{c\in CS:\\ h_c=h}}\sum_{r\in R^{CS}}\sum_{t\in T_c}\sum_{d\in D} p_c x_{crtd} + \sum_{\substack{c\in AS:\\ h_c=h}}\sum_{r\in R^{AS}}\sum_{t\in T_c}\sum_{d\in D} p_c x_{crtd} \le T_h^{\mathrm{MAX}_W}, \quad \forall\, h \in H \tag{c9}$$

$$x_{crtd} \in \{0, 1\}, \quad \forall\, c \in CS,\ r \in R^{CS},\ t \in T_c,\ d \in D$$
$$x_{crtd} \in \{0, 1\}, \quad \forall\, c \in AS,\ r \in R^{AS},\ t \in T_c,\ d \in D$$
$$y_{jrd} \in \{0, 1\}, \quad \forall\, j \in E,\ r \in R,\ d \in D.$$

Some sets of constraints must be splitted for conventional and ambulatory surgeries because they use different operating rooms (R^{CS} and R^{AS}). The sets of constraints in this situation are identified with the same number and a C or an A is added in order to distinguish, respectively, between conventional and ambulatory surgeries.

Constraint sets (c1) force deferred urgency level priority surgeries to be scheduled on Monday in order to meet the 72 h deadline for their completion. Constraint sets (c2) impose high priority surgeries to be scheduled during the planning week. Constraints (c3) state that the remaining surgeries, classified as priority or normal, may be scheduled or not during the planning week. Constraints (c4) guarantee that surgeries do not overlap in the same room. These constraints also impose γ empty periods for room cleaning at the end of each surgery (based on the definition of the lower sum limit). Constraint sets (c5) prevent assignment of more than one surgery specialty to each room and day. Constraints (c6) are

Table 11
nPOP: 20/50/70/100 (nGER = 500).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness	(4)	(5)	(6) Improv. gap	(7) Average deviation	(8*) # Runs equal best
		Min	Mean	Max			
nPOP = 20							
z_1	31.81	1142.07	1143.83	1144.95	0.824%	1.166	3.41
z_2	10.24	298.22	298.63	299.00	0.549%	0.370	6.30
nPOP = 50							
z_1	77.31	1142.43	1144.21	**1145.00**	0.857%	0.793	5.02
z_2	24.51	298.53	298.83	299.00	0.617%	0.167	8.33
nPOP = 70							
z_1	106.98	**1142.83**	**1144.22**	1144.96	**0.858**%	**0.781**	5.10
z_2	33.56	298.60	298.89	299.00	0.637%	0.107	8.93
nPOP = **100**							
z_1	151.59	1142.72	1144.16	1144.92	0.853%	0.840	**5.32**
z_2	47.47	**298.62**	**298.94**	299.00	**0.652**%	**0.063**	**9.37**

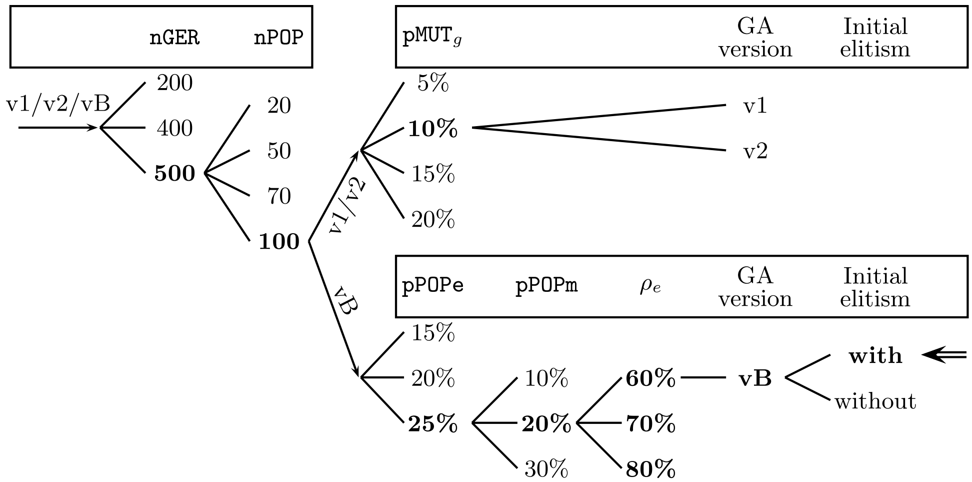

Fig. 5. Scheme used to tune algorithm parameters and to select GA version.

Table 13
pPOPe: 15%/20%/25% (nGER = 500, nPOP = 100).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness Min	(4) Mean	(5) Max	(6) Improv. gap	(7) Average deviation
pPOPe = 15%						
z_1	154.66	1144.06	1144.76	1145.00	0.906%	0.239
z_2	50.40	298.83	298.98	299.00	0.668%	0.017
pPOPe = 20%						
z_1	153.32	1144.17	**1144.83**	1145.00	0.912%	**0.172**
z_2	49.64	**298.89**	**298.99**	299.00	0.670%	**0.011**
pPOPe = **25%**						
z_1	156.42	**1144.22**	**1144.83**	1145.00	0.912%	**0.172**
z_2	47.05	**298.89**	**298.99**	299.00	0.670%	**0.011**

Table 14
pPOPm: 10%/20%/30% (nGER = 500, nPOP = 100, pPOPe = 25%).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness Min	(4) Mean	(5) Max	(6) Improv. gap	(7) Average deviation
pPOPm = 10%						
z_1	155.58	1144.00	1144.80	1145.00	0.909%	0.200
z_2	50.32	298.83	298.98	299.00	0.668%	0.017
pPOPm = **20%**						
z_1	157.22	**1144.50**	**1144.85**	1145.00	**0.914**%	**0.150**
z_2	45.21	**299.00**	**299.00**	299.00	**0.673**%	**0.000**
pPOPm = 30%						
z_1	156.47	1144.17	1144.83	1145.00	0.912%	0.167
z_2	45.60	298.83	298.98	299.00	0.668%	0.017

Table 15
ρ_e : 60%/70%/80% (nGER = 500, nPOP = 100, pPOPe = 25%, pPOPm = 20%).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness Min	(4) Mean	(5) Max	(6) Improv. gap	(7) Average deviation
ρ_e = 60%						
z_1	154.53	1144.50	1144.85	1145.00	0.914%	0.150
z_2	43.92	299.00	299.00	299.00	0.673%	0.000
ρ_e = 70%						
z_1	155.94	1144.50	1144.85	1145.00	0.914%	0.150
z_2	46.74	299.00	299.00	299.00	0.673%	0.000
ρ_e = 80%						
z_1	161.19	1144.50	1144.85	1145.00	0.914%	0.150
z_2	44.97	299.00	299.00	299.00	0.673%	0.000

Table 16
Version: v1/v2/vB (nGER = 500, nPOP = 100).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness Min	(4) Mean	(5) Max	(6) Improv. gap	(7) Average deviation
v1: $\mathtt{pMUT}_g$ = 10%						
z_1	146.24	1141.50	1143.95	1145.00	0.834%	1.050
z_2	46.21	298.50	298.90	299.00	0.640%	0.100
v2: $\mathtt{pMUT}_g$ = 10%						
z_1	149.21	1143.00	1144.10	1145.00	0.847%	0.900
z_2	45.12	298.50	298.95	299.00	0.657%	0.050
vB: pPOPe = 25%, pPOPm = 20%, ρ_e = 60%						
z_1	154.53	**1144.50**	**1144.85**	1145.00	**0.914**%	**0.150**
z_2	43.92	**299.00**	**299.00**	299.00	**0.673**%	**0.000**

the linking constraints for variables x and y. Constraints (c7) ensure that surgeons do not overlap between rooms in the same time period and day. Constraint sets (c8) and (c9) impose a daily and weekly operating time limit on each surgeon. The remaining constraints express the variables' domain.

Appendix B. Other results

This Appendix reports on the methodology used to tune the algorithm parameters and to select the best GA version among the three tested (vB—actual version, v1 and v2). Fig. 5 shows the scheme used which is supported by the results in Tables 10–17.

Several parameters were successively tested for each version. The choices were made based on average results over 10 runs using two instances (I3_1944 and I6_1887). For each parameter, a corresponding table, from Tables 10 to 17, presents the average results and the value that obtained the best average results is fixed for the following tests (highlighted in Fig. 5). The number of generations (nGER = 200, 400, 500) and the dimension of the population (nPOP = 20, 50, 70, 100), common for all the versions, were first tested. After fixing global parameters, tests focused on the parameters specific to each version. Remember that versions v1 and v2 only differ on the selection operator (respectively, uniform distribution and roulette wheel). The probability of a gene

Table 17
Initial elitism: with/without (nGER = 500, nPOP = 100, pPOPe = 25%, pPOPm = 20%, ρ_e = 60%).

(1) Obj. function	(2) Time (s)	(3) Best chrom. fitness	(4)	(5)	(6) Improv. gap	(7) Average deviation	(8*) # Runs equal best	(9*) Best gen.
		Min	Mean	Max				
With initial elitism								
z_1	154.53	**1144.50**	**1144.85**	1145.00	0.914%	**0.150**	**8.50**	**56.15**
z_2	43.92	299.00	299.00	299.00	0.673%	0.000	10	**33.95**
Without initial elitism								
z_1	152.22	1143.50	1144.65	1145.00	–	0.350	7	111.90
z_2	45.14	299.00	299.00	299.00	–	0.000	10	43.45

being selected to suffer mutation is usually small. Therefore, the gene level mutation rate ($\mathtt{pMUT}_g$) was tested with small values (5%, 10%, 15%, 20%) for versions v1 and v2. The values tested for parameters specific to version vB (pPOPe, pPOPm and ρ_e) followed Gonçalves and Resende's recommendations stated in [18]. After defining the best value for each parameter, the three versions were compared and the GA version proposed in this paper (version vB) is the one that presents slightly better average results. The results for this version were finally compared with results obtained for the same version without considering the initial elitist chromosome. Table 17 shows that although the quality of the solutions do not differ significantly, introducing the high-quality individual in the initial population helps the algorithm to find earlier the best chromosome (column 9*) and produces more consistent results (columns 7 and 8*).

Comparisons were made based on columns 3–7. Columns 3–5 refer to the fitness value of the best individual in the last generation. The average percentage of improvement from the initial elitist chromosome is shown in column 6 (*Improv. gap*). Column 7 (*Average deviation*) presents the average deviation from the best solution found in all the versions tested. In some cases, columns were added in order to get more insight on the results. These columns are identified with an asterisk. Column 8* (*# Runs equal best*) gives the average number of runs that reached the best solution found over the 10 runs of the algorithm. Column 9* (*Best gen.*) refers to the average generation that first originated the best chromosome. Results were analyzed for each objective function (column 1), z_1 and z_2 corresponding, respectively, to the surgical suite occupation and to the number of surgeries scheduled.

References

[1] B. Roland, C. Di Martinelly, F. Riane, Y. Pochet, Scheduling an operating theatre under human resource constraints, Comput. Ind. Eng. 58 (2) (2010) 212–220.
[2] H. Fei, N. Meskens, C. Chu, A planning and scheduling problem for an operating theatre using an open scheduling strategy, Comput. Ind. Eng. 58 (2) (2010) 221–230.
[3] General Direction of Health, (Ed.), Plano Nacional de Saúde 2004–2010: Mais Saúde Para Todos. General Direction of Health, Lisbon, 2004 (in Portuguese).
[4] I. Marques, Planeamento de cirurgias electivas—abordagens em programação inteira, Ph.D. Thesis, Faculdade de Ciências, Universidade de Lisboa, 2010 (in Portuguese).
[5] A. Riise, E.K. Burke, Local search for the surgery admission planning problem, J. Heuristics 17 (4) (2011) 389–414.
[6] E. Hans, G. Wullink, M. van Houdenhoven, G. Kazemier, Robust surgery loading, European J. Oper. Res. 185 (3) (2008) 1038–1050.
[7] Y. Liu, C. Chu, K. Wang, A new heuristic algorithm for the operating room scheduling problem, Comput. Ind. Eng. 61 (3) (2011) 865–871.
[8] I. Marques, M.E. Captivo, M.V. Pato, Exact and heuristic approaches for elective surgery scheduling, in: Proceedings of the CLAIO/SBPO 2012, Rio de Janeiro, 2012, pp. 3729–3738.
[9] A. Guinet, S. Chaabane, Operating theatre planning, Int. J. Prod. Econ. 85 (1) (2003) 69–81.
[10] R. Velásquez, M.T. Melo, A set packing approach for scheduling elective surgical procedures, in: Operations Research Proceedings 2005, Springer, Berlin, Heidelberg, 2006, pp. 425–430.
[11] B. Cardoen, E. Demeulemeester, J. Beliën, Optimizing a multiple objective surgical case sequencing problem, Int. J. Prod. Econ. 119 (2) (2009) 354–366.
[12] B. Cardoen, E. Demeulemeester, J. Beliën, Sequencing surgical cases in a day-care environment: an exact branch-and-price approach, Comput. Oper. Res. 36 (9) (2009) 2660–2669.
[13] I. Marques, M.E. Captivo, M.V. Pato, An integer programming approach to elective surgery scheduling, OR Spectrum 34 (2) (2012) 407–427.
[14] B. Cardoen, E. Demeulemeester, J. Beliën, Operating room planning and scheduling: a literature review, European J. Oper. Res. 201 (3) (2010) 921–932.
[15] F. Guerriero, R. Guido, Operational research in the management of the operating theatre: a survey, Healthc. Manage. Sci. 14 (1) (2011) 89–114.
[16] B. Cardoen, E. Demeulemeester, J. Beliën, Operating room planning and scheduling problems: a classification scheme, Int. J. Health Manag. Inform. 1 (1) (2010) 71–83.
[17] Ministry of Health. Administrative regulation (''Portaria'') nb 45/2008, in: Diário da República, Number 10 in 1st Series, INCM, 2008, pp. 526–536 (in Portuguese).
[18] J.F. Gonçalves, M.G.C. Resende, Biased random-key genetic algorithms for combinatorial optimization, J. Heuristics 17 (5) (2011) 487–525.
[19] L. Davis, Handbook of Genetic Algorithms, International Thompson Computer Press, London, 1996.
[20] C.R. Reeves, Genetic algorithms, in: C.R. Reeves (Ed.), Modern Heuristic Techniques for Combinatorial Problems, in: Advanced Topics in Computer Science, Halsted Press, New York, 1993, pp. 151–196 (Chapter 4).
[21] J.E. Beasley, P.C. Chu, A genetic algorithm for the set covering problem, European J. Oper. Res. 94 (2) (1996) 392–404.
[22] I. Marques, M.E. Captivo, M.V. Pato, Planning elective surgeries in a Portuguese hospital: study of different mutation rules for a genetic heuristic, Lect. Notes Manag. Sci. 4 (2012) 238–243.
[23] IBM. IBM ILOG CPLEX Optimization Studio. Getting Started with CPLEX. Version 12, Release 4, 2011.