



## Discrete Optimization

## Surgical scheduling with simultaneous employment of specialised human resources

Thiago A. O. Silva<sup>a,b</sup>, Mauricio C. de Souza<sup>c,\*</sup>, Rodney R. Saldanha<sup>d</sup>, Edmund K. Burke<sup>e</sup><sup>a</sup> Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, cep: 31270-901, Belo Horizonte, MG, Brazil<sup>b</sup> Departamento de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, MG, Brazil<sup>c</sup> Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, cep: 31270-901, Belo Horizonte, MG, Brazil<sup>d</sup> Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, cep: 31270-901, Belo Horizonte, MG, Brazil<sup>e</sup> Computing Science and Mathematics, School of Natural Sciences, University of Stirling, Stirling FK9 4LA, United Kingdom

## ARTICLE INFO

## Article history:

Received 1 March 2014

Accepted 6 April 2015

Available online 11 April 2015

## Keywords:

Surgical scheduling

Mathematical modelling

Relax-and-fix heuristics

## ABSTRACT

Surgical scheduling is a challenging problem faced by hospital managers. It is subject to a wide range of constraints depending upon the particular situation within any given hospital. We deal with the simultaneous employment of specialised human resources, which must be assigned to surgeries according to their skills as well as the time windows of the staff. A particular feature is that they can be assigned to two surgeries simultaneously if the rooms are compatible. The objective is to maximise the use of the operating rooms. We propose an integer model and integer programming based heuristics to address the problem. Computational experiments were conducted on a number of scenarios inspired by real data to cover different practical problem solving situations. Numerical results show that relaxations provide tight upper bounds, and relax-and-fix heuristics are successful in finding optimal or near optimal solutions.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

## 1. Introduction

Operating room management has been recognised as a main source of income for hospitals. It is important to improve performance by using surgical resources as efficiently and effectively as possible. It has been reported in the literature that the operating theatre is one of the most critical and costly functional areas in a hospital (Guinet & Chaabane, 2003). It represents a bottleneck in many hospitals and it consumes a significant amount of a hospital's annual budget because it typically utilises the most expensive resources. See, for instance, the recent surveys by Cardoen, Demeulemeester, and Beliën (2010), Guerriero and Guido (2011), May, Spangler, Strum, and Vargas (2011) and Rais and Viana (2011). As in other service sectors, the decision process comprises different decision levels – from capacity and facility planning to detailed scheduling.

In this paper, we concentrate on the detailed surgical scheduling of elective patients on a daily basis. Elective patients are those for whom the surgery is not completely unexpected and can be planned in advance. Broadly speaking, surgical scheduling consists of the

selection of surgical interventions to be performed in each available operating room along with dates, starting times and the allocation of required resources. Surgical scheduling problems are very hard to solve, and different optimisation modelling approaches have been proposed in the literature to face different situations. Meskens, Duvivier, and Hanset (2013) observed from visits to several hospitals that problems are specific to each institution. Indeed, as also noted by other authors, for instance, Cardoen et al. (2010); Riise and Burke (2011), each hospital has its own established practices reflecting specific constraints related to both human and material facets. These specific constraints may vary from medical staff availabilities (e.g., regular working hours or surgeon preferences) and patient priorities (e.g., children or diabetics) to operating room versatilities and the limited capacity of human and material resources (e.g., number of nurses, auxiliary staff, medical instruments, recovery beds or places in intensive care unities).

We deal with surgical scheduling in a hospital in Brazil. The hospital has 14 non-identical operating rooms dedicated to elective interventions – some are multifunctional, some have particular medical devices installed that cannot be moved, and some are restricted in size. Each surgeon has a set of patients, so that each operation has to be carried out by a specific surgeon. However, decisions assigning surgeries to operating rooms is part of the problem, and scheduling starting times are subject to surgeries and surgeon's time windows,

\* Corresponding author. Tel.: +55 31 34 09 48 93.

E-mail addresses: [thiago@deen.ufop.br](mailto:thiago@deen.ufop.br) (T.A.O. Silva), [prof.mauriciodesouza@gmail.com](mailto:prof.mauriciodesouza@gmail.com), [mauricio.souza@pq.cnpq.br](mailto:mauricio.souza@pq.cnpq.br) (M.C. de Souza), [rodney@cpdee.ufmg.br](mailto:rodney@cpdee.ufmg.br) (R.R. Saldanha), [e.k.burke@stir.ac.uk](mailto:e.k.burke@stir.ac.uk) (E.K. Burke).

<http://dx.doi.org/10.1016/j.ejor.2015.04.008>

0377-2217/© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

among other constraints. Time windows may capture patient priorities in terms of time of the day that interventions must occur, e.g., children should be operated upon early in the morning (Riise & Burke, 2011) or there should be later starts for out-of-town patients (Vijayakumar, Parikh, Scott, Barnes, & Gallimore, 2013). There are also medical staff preferences, which can be viewed as an attempt to improve working conditions (Roland, di Martinelly, Riane, & Pochet, 2010). The problem is also constrained by limited shared resources. We may have surgery requirements on renewable and/or nonrenewable resources, a characteristic already observed in the case studied by Roland et al. (2010). Renewable resources (human or material) are those allocated to a surgery strictly for its duration, like nurses or auxiliary staff and medical equipment. On the other hand, nonrenewable resources are those allocated to a surgery for the entire day, in our case places in the intensive care unit.

A particular feature of our case study is the simultaneous employment of specialised human resources. Anaesthetists, for instance, are identified due to specialities, training skills and availabilities. An anaesthetist has his/her own time windows. Each surgery is associated with a subset of anaesthetists able to perform that intervention. An anaesthetist can perform more than one intervention simultaneously depending upon the room in which the surgeries are taking place. Thus, the simultaneous employment of specialised human resources is constrained by room compatibilities. This leads to another decision to be made jointly with assigning and scheduling surgeries, as anaesthetists must be assigned and scheduled according to surgery requirements and subject to room compatibilities and to their own time windows. The objective is to maximise the use of the operating rooms. In fact, recent surveys list a number of objectives studied in the literature including minimisation of costs or makespan, and maximisation of throughput or utilisation (Cardoen et al., 2010; Guerriero & Guido, 2011; May et al., 2011). In our case, the manager of the hospital where the study was conducted suggested the maximisation of the utilisation of the operating rooms.

We propose an integer programming formulation to model the particular surgical scheduling problem in hand. Preliminary computational experiments have shown that the linear relaxation of the proposed model could provide good upper bounds to the optimal solution values. Based on these early experiments, we decided to develop integer programming-based heuristics to obtain high quality feasible solutions with a posteriori performance guarantees. The heuristics in the first step draw from relaxation candidate assignments of surgeries to rooms and periods, and then in the second step they address a reduced model to obtain feasibility. We test the proposed approaches on a variety of scenarios inspired from real data. Computational results show that it is possible to find optimal or near optimal solutions for most of the instances.

The paper is structured as follows. In the next section, we present a literature review which is focused on deterministic models for surgical scheduling. In Section 3, we formulate the integer programming model. We describe, in Section 4, the integer programming-based heuristics. In Section 5, we report on computational experiments on real-based instances, and in the last section we draw concluding remarks.

## 2. Literature review

Many modelling studies aimed at using surgical rooms efficiently have been motivated by practical considerations and goals. Ozkarahan (2000) proposed a goal programming model to assign surgeries to operating rooms under a policy of reserving blocks of operating room time for surgical specialties or authorised surgeons. Marcon, Kharraja, and Simonnet (2003) adapted the multiple knapsack model to assign surgeries to operating rooms optimising measures related to the risk of nonrealisation. Some authors have investigated applications based on bin-packing to select surgeries that make efficient use of

available operating room time, see Dexter, Macario, and Traub (1999), Dexter and Traub (2002), Dexter, Traub, and Macario (2003), Hans, Wullink, van Houdenhoven, and Kazemier (2008), van Houdenhoven, van Oostrum, Hans, Wullink, and Kazemier (2007). Guinet and Chaabane (2003) proposed a model to assign surgeries to operating rooms over a horizon of one or two weeks subject to time and capacity constraints. Santibáñez, Begen, and Atkins (2007) also dealt with time and constraints to manage a set of hospitals as a single system.

Jebali, Hadj Alouane, and Ladet (2006) addressed the daily scheduling of a surgical centre with an assignment model of surgeries to operating rooms, taking into account capacity constraints such as opening duration and overtime of operating rooms, the working time of surgeons, and the number of beds in the intensive care unit. The objective is to minimise the costs associated to keeping patients waiting, and to minimise undertime and overtime. Then, sequencing is modelled as a two-stage hybrid flow shop where the first stage is represented by surgical rooms and the second stage by recovering beds. Analogies to scheduling problems more often found in industrial applications were also exploited by Pham and Klinkert (2008). The authors introduced an extension of the job shop with blocking to model the patient flow comprising preoperative, perioperative and postoperative stages.

Hierarchical approaches in three stages have been proposed (Ogulata & Erol, 2003; Testi, Tanfani, & Torre, 2007). Ogulata and Erol (2003) developed models to select, in the first stage, patients from a candidate list. In the second stage, they are assigned to surgical groups, and, in the final stage, operating rooms are determined independently for each group. Testi et al. (2007) developed models to distribute operating room time among surgical groups, and then to build a cyclic timetable that determines the surgical unit associated with each block of operating room time. The last stage is performed by simulation. Testi and Tanfani (2009) proposed a model to assign both patient and surgical sub-speciality to operating rooms and days within a planning horizon.

Augusto, Xie, and Perdomo (2008) developed a Lagrangian relaxation to a patient flow problem considering the transfer from the ward to the operating room, the surgery, and the transfer to the recovery bed. The model defines starting times for the different tasks in order to minimise completion times subject to capacity constraints. The same authors (Augusto, Xie, & Perdomo, 2010) addressed the impact of allowing patient recovery in the operating room when no recovery bed is available. Marques, Captivo, and Pato (2012) proposed a model to schedule elective surgeries on a weekly time horizon with the objective of maximising the operating rooms occupancy. The authors considered different surgery priority levels, surgery time windows, and operating time limits.

In recent years, some authors have developed branch-and-price approaches (Cardoen, Demeulemeester, & Beliën, 2009b; Fei, Chu, Meskens, & Artiba, 2008). Fei et al. (2008) focused on surgical cases assignment to operating rooms. The model takes into consideration the room's opening time and surgical deadlines to plan one week minimising undertime and overtime costs. The master problem partitions surgeries into days, whereas the subproblem deals with opening duration constraints to find an improving column. This approach was extended by Fei, Meskens, and Chu (2010) to consider daily scheduling performed by a hybrid genetic algorithm. Cardoen et al. (2009b) developed a branch-and-price approach over a multi-objective model proposed by the same authors (Cardoen, Demeulemeester, & Beliën, 2009a). The model decides which surgeries to start in each slot of each period, restricted to operating rooms and starting ranges that have been previously defined. Other constraints include the availabilities of medical instruments and recovering areas and additional cleaning due to particular infection concerns. A column represents all surgeries sequenced for a specific surgeon, and the pricing problem is addressed by dynamic programming.

Roland et al. (2010) proposed a model dealing with different aspects of human and material resource management. Resource

**Table 1**

Constraints typically modelled in the literature: Cap – capacity; T W – time windows; PI H – planning horizon divided in days; Ud-Ov/Fx – under-overtime and/or fixed costs; M-S – multi-stage; Ad Cl – additional cleaning; Sp H R – specialised human resources; Si Emp – simultaneous employment of specialised human resources.

Paper	Constraints							
	Cap	T W	PI H	Ud-Ov/Fx	M-S	Ad Cl	Sp H R	Si Emp
Guinet and Chaabane (2003)	✓	✓	✓	✓				
Jebali et al. (2006)	✓	✓		✓	✓			
Pham and Klinkert (2008)	✓	✓			✓		✓	
Augusto et al. (2008)	✓				✓			
Fei et al. (2008)	✓	✓	✓	✓				
Cardoen et al. (2009a)	✓	✓				✓		
Roland et al. (2010)	✓	✓	✓	✓				
Riise and Burke (2011)	✓	✓	✓	✓				
Marques et al. (2012)	✓	✓	✓					
Ghazalbash et al. (2012)	✓	✓				✓		
Meskens et al. (2013)	✓	✓		✓			✓	✓
Vijayakumar et al. (2013)	✓	✓	✓				✓	
This paper	✓	✓					✓	✓

requirements are distinguished between renewable (e.g., medical staff) and nonrenewable (e.g., pharmaceutical products). Surgeons express preferences in terms of availabilities, and some medical staff (e.g., anaesthetists) do not participate during the whole surgery to allow a broader coverage of all operating rooms. Riise and Burke (2011) addressed both the intervention assignment (i.e., the assignment of interventions to rooms and days) and the intervention scheduling (i.e., the sequencing of interventions within a day and room). The objective function comprises three objectives: patient waiting time, surgeon overtime, and waiting time for children. The model can incorporate a master surgery scheduling plan where operating room hours are divided into blocks and allocated to specialities, in addition to an existing plan in which surgeries are classified into un-served, planned but not fixed, or fixed. Ghazalbash, Sepehri, Shadpour, and Atighehchian (2012) dealt with the distribution of surgeries in order to ensure that residents and fellows have equal opportunities to acquire experience in a training hospital. Jeang and Chiang (2012) have taken into account surgeon availabilities and unfavourable surgery times to minimise the deviation between the total operation time and the total available time in operating rooms. Vijayakumar et al. (2013) aimed to maximise a weighted sum of the number of surgeries to be performed on a day-of-week and time-of-day basis. The authors modelled their surgical scheduling case as a multi-bin, unequal-sized, multi-dimensional dual bin-packing problem. Constraints are related to resource availabilities, patient priorities, and surgical times and the specialities of surgeons. Meskens et al. (2013) proposed a constraint programming approach to deal with resource availabilities, staff preferences, and affinities among staff members. The model has a multi-objective function by minimising makespan and overtime, and maximising affinities among members of the surgical team.

In Table 1, we summarise the main blocks of constraints typically found in the literature on detailed surgery scheduling. This provides a broad overview but it is important to note that different papers often treat specific constraints differently within each block. The block of constraints that is common to all modelling approaches is the group of capacity constraints. These constraints deal with the operating rooms opening hours, medical staff working hours, or resources availability. Time windows are often found in terms of the periods of the day when a surgeon is available to start a surgery. Most papers model the daily scheduling with time-indexed formulations, and some authors use an extra index for the day to cover a planning horizon. Under and over time have been considered when minimising operational costs. Roland et al. (2010) also considered a fixed cost to open an operating room during a day. Some authors have modelled the flow of patients from wards to operating rooms and then to recovery beds, characterising the problem as a multi-stage scheduling problem. Cardoen

et al. (2009a) introduced a constraint to model additional cleaning in order to avoid scheduling a surgery of a non-infected patient after the surgery of an infected patient. In terms of modelling, the main contributions of our research are the treatment of specialised human resources and the simultaneous employment of them. We have a set of types of specialised human resources, and, on one hand, a subset of such specialities that are needed to perform a surgery. On the other hand, we have for each of these specialities a set of individuals. For each speciality, we also specify a subset, along with their time windows, which are suitable to perform the given surgery. We propose a model that assigns, conjointly with the surgery scheduling, suitable individuals of each speciality, satisfying the given time windows constraints. Moreover, an individual of a given speciality can perform more than one surgery simultaneously, as far as the distance between rooms where surgeries are taking place permits. Thus, our model takes into account the simultaneous employment of specialised human resources constrained by room compatibilities. Our modelling treatment of specialised human resources and the feature of simultaneous employment, to the best of our knowledge, has not been studied so far in the literature.

### 3. Mathematical modelling

As discussed in the previous section, one can find approaches in the literature modelling particular cases which are found in different countries. This includes, for examples, Belgium (Cardoen et al., 2009a, 2009b; Fei et al., 2010; Roland et al., 2010), Canada (Santibáñez et al., 2007), France (Augusto et al., 2008; 2010; Guinet & Chaabane, 2003; Marcon et al., 2003), Iran (Ghazalbash et al., 2012), Italy (Testi & Tanfani, 2009; Testi et al., 2007), Netherlands (Hans et al., 2008; van Houdenhoven et al., 2007), Norway (Riise & Burke, 2011), Portugal (Marques et al., 2012), Turkey (Ogulata & Erol, 2003; Ozkarahan, 2000), and USA (Dexter et al., 1999; Dexter & Traub, 2002; Dexter et al., 2003; Vijayakumar et al., 2013). In this paper, we model a daily surgical scheduling problem found at a hospital in Brazil. We next describe the details of the modelling approach. Table 2 gives an overview of the required data to run the proposed model.

The day is divided in periods, and  $T$  is the set of periods. The set of surgeons is denoted by  $C$ , and  $T_c \subseteq T$  is the subset of periods a surgeon  $c$  can start a surgery, i.e., set  $T_c$  accounts for the surgeon  $c$  time windows. The set of surgeries is denoted by  $S$ , and  $S_c \subseteq S$  is the subset of surgeries to be performed by surgeon  $c$ . Given a surgeon  $c \in C$  and a surgery  $s \in S_c$ ,  $T_s \subseteq T_c$  is the subset of periods a surgery can start, i.e., the surgery  $s$  time windows. The set of operating rooms is denoted by  $R$ , and  $R_s \subseteq R$  is the subset of operating rooms where a surgery  $s$  can take place. A room is in use with a surgery  $s$  during  $d_s$  periods,

**Table 2**  
Summary of required data to run the integer programming model.

Sets	Description
$S$	Surgeries
$C$	Surgeons
$S_c \subseteq S$	Surgeries performed by surgeon $c \in C$
$R$	Operating rooms
$R_s \subseteq R$	Operating rooms surgery $s \in S$ can take place
$T$	Periods
$T_c \subseteq T$	Periods surgeon $c \in C$ can start a surgery
$T_s \subseteq T_c$	Periods surgery $s \in S_c$ can be started
$K_1$	Renewable resources
$R_k \subseteq R$	Operating rooms where resource $k \in K_1$ is installed
$K_2$	Nonrenewable resources
$E$	Types of specialised human resources
$E_s \subseteq E$	Types specialised human resources required to perform surgery $s \in S$
$A_e$	Individuals treated as specialised resources of type $e \in E$
$A_e^s \subseteq A_e$	Individuals of specialised resource type $e \in E_s$ suitable to perform surgery $s \in S$
$T_e^a \subseteq T$	Periods individual $a \in A_e$ can start a surgery
Parameters	Description
$d_s$	Duration in periods of surgery $s \in S$ (room cleaning included)
$\bar{d}_s$	Duration in periods of only the intervention of surgery $s \in S$ (room cleaning excluded)
$g_{sk}^1$	Amount of renewable resource $k \in K_1$ required to perform surgery $s \in S$
$q_k^1$	Mobile units of renewable resource $k \in K_1$
$g_{sk}^2$	Amount of nonrenewable resource $k \in K_2$ required to perform surgery $s \in S$
$q_k^2$	Units of nonrenewable resource $k \in K_2$
$q_a$	Number of surgeries individual $a \in A_e (e \in E_s)$ can perform simultaneously
$b_{r\bar{r}}$	Whether or not a specialised resource can be employed simultaneously in rooms $r, \bar{r} \in R$

which includes the time for cleaning. However, surgeons and other medical staff as well are free to start another surgery when the intervention itself is finished. The parameter  $\bar{d}_s$  accounts for the duration of intervention only.

The resources are classified into sets  $K_1$  and  $K_2$  of renewable and nonrenewable resources, respectively. Given  $k \in K_1$  (resp.  $k \in K_2$ ),  $g_{sk}^1$  (resp.  $g_{sk}^2$ ) is the amount of renewable (resp. nonrenewable) resource  $k$  required to perform surgery  $s$ . A renewable resource  $k \in K_1$  may have a certain number  $q_k^1$  of mobile units, and may also be already installed on a subset  $R_k \subseteq R$  of operating rooms. So, a surgery requiring renewable resource  $k$  assigned to an operating room  $r \in R_k$  does not consume any of the  $q_k^1$  mobile units. On the other hand, when a surgery which does not require renewable resource  $k$  is assigned to an operating room  $r \in R_k$ , the resource installed in that room stays idle as it cannot be moved elsewhere. A nonrenewable resource  $k \in K_2$  has  $q_k^2$  units. An amount of  $g_{sk}^2$  must be reserved during the whole day for a surgery  $s$  requiring resource  $k$  if  $s$  is to be performed, e.g., a resource in an intensive care unit which is occupied for several hours.

The specialised human resources are treated differently because, in our case, the employment of anaesthetists can be undertaken simultaneously, depending upon the rooms in which surgeries are taking place. Although motivated by a particular case, our description enables the handling of different types of specialised resources. The set of types of specialised resources is denoted by  $E$ , and  $E_s \subseteq E$  is the subset required to perform surgery  $s \in S$ . Suppose, as an illustrative example,  $E = \{\alpha, \beta\}$ , i.e., the hospital counts specialised resources of types  $\alpha$  and  $\beta$ ,  $E_s = \{\alpha\}$  and  $E_{\hat{s}} = \{\alpha, \beta\}$  for surgeries  $s$  and  $\hat{s}$ , respectively. Given  $e \in E$ ,  $A_e$  is the set of individuals treated as specialised resources of type  $e$ . For example,  $A_\alpha = \{\alpha_1, \alpha_2, \alpha_3\}$  and  $A_\beta = \{\beta_1, \beta_2\}$ . It may happen, given  $s \in S$  and  $e \in E_s$ , that not all the individuals that belong to  $A_e$  are suitable for performing surgery  $s$ . Thus,  $A_e^s \subseteq A_e$  is the subset of individuals of specialised resource type  $e$  suitable to perform surgery  $s$ . Suppose, in our example,  $A_\alpha^s = \{\alpha_1, \alpha_2\}$ ,  $A_\alpha^{\hat{s}} = \{\alpha_2, \alpha_3\}$ ,  $A_\beta^{\hat{s}} = \{\beta_1\}$ . An individual  $a \in A_e$  can perform  $q_a$  sur-

geries simultaneously, as far as the distance between rooms  $r$  and  $\bar{r}$  where surgeries are taking place permits. A parameter  $b_{r\bar{r}}$  is set to 1 if a specialised resource can be employed simultaneously in rooms  $r$  and  $\bar{r}$ , and to 0 otherwise. Moreover, each individual  $a \in A_e$  of specialised resource type  $e \in E$  may have his/her own time windows. So,  $T_e^a \subseteq T$  is the set of periods in which individual  $a$  can start a surgery.

We define two binary variables. The first,  $y_{st}^r$ , has value 1 if surgery  $s \in S$  is assigned to room  $r \in R_s$  and starts in period  $t \in T_s$ , and 0 otherwise. The second,  $w_{st}^a$ , has value 1 if surgery  $s \in S$  is assigned to individual  $a \in A_e^s$  of specialised resource type  $e \in E_s$  and starts in period  $t \in T_e^a \cap T_s$ , and 0 otherwise. The model can be written as follows:

$$\max \sum_{s \in S} \sum_{r \in R_s} \sum_{t \in T_s} d_s y_{st}^r \quad (1)$$

$$\sum_{r \in R_s} \sum_{t \in T_s} y_{st}^r \leq 1 \quad \forall s \in S \quad (2)$$

$$\sum_{s \in S_c} \sum_{r \in R_s} \sum_{l=t-\bar{d}_s+1|l \in T_s}^t y_{sl}^r \leq 1 \quad \forall c \in C, \forall t \in T_c \quad (3)$$

$$\sum_{s \in S} \sum_{r \in R_s} \sum_{l=t-\bar{d}_s+1|l \in T_s}^t y_{sl}^r \leq 1 \quad \forall r \in R, \forall t \in T \quad (4)$$

$$\sum_{s \in S} \sum_{r \in R_s \setminus R_k} \sum_{l=t-\bar{d}_s+1|l \in T_s}^t g_{sk}^1 y_{sl}^r \leq q_k^1 \quad \forall k \in K_1, \forall t \in T \quad (5)$$

$$\sum_{s \in S} \sum_{r \in R_s} \sum_{t \in T_s} g_{sk}^2 y_{st}^r \leq q_k^2 \quad \forall k \in K_2 \quad (6)$$

$$\sum_{r \in R_s} y_{st}^r - \sum_{a \in A_e^s | t \in T_e^a} w_{st}^a \leq 0 \quad \forall s \in S, \forall e \in E_s, \forall t \in T_s \quad (7)$$

$$\sum_{t \in T_s} \sum_{a \in A_e^s | t \in T_e^a} w_{st}^a \leq 1 \quad \forall s \in S, \forall e \in E_s \quad (8)$$

$$\sum_{s \in S} \sum_{e \in E_s} \sum_{l=t-\bar{d}_s+1|l \in T_s \cap T_e^a} w_{sl}^a \leq q_a \quad \forall e \in E, \forall a \in A_e, \forall t \in T_e^a \quad (9)$$

$$y_{st}^r + w_{st}^a + \sum_{l=t-\bar{d}_s+1|l \in T_s \cap T_{\bar{s}}^a} (y_{sl}^{\bar{r}} + w_{sl}^a) \leq 3 \quad \forall s, \bar{s} \in S, \forall e \in E_s \cap E_{\bar{s}}, \forall a \in A_e^s \cap A_{\bar{e}}^{\bar{s}}, \forall r \in R_s, \forall \bar{r} \in R_{\bar{s}}, b_{r\bar{r}} = 0, \forall t \in T_e^a \cap T_{\bar{e}}^{\bar{a}} \quad (10)$$

$$y_{st}^r \in \{0, 1\} \quad \forall s \in S, \forall r \in R_s, \forall t \in T_s \quad (11)$$

$$w_{st}^a \in \{0, 1\} \quad \forall e \in E, \forall a \in A_e, \forall s \in S, \forall r \in R_s, \forall t \in T_e^a \cap T_s \quad (12)$$

The objective function (1) maximises the occupation of the operating rooms. Constraint (2) ensures that a surgery is not performed more than once. An important issue when maximising the utilisation in a daily basis could be whether an operation would be carried from day to day without ever being scheduled. Thus, when using the model on a daily basis, it might be useful to introduce a slack variable in constraint (2) to impose a surgery that has not been scheduled by setting the slack variable associated to that surgery to zero. Overlapping surgeries are avoided by constraints (3) and (4). The former

prevents a surgeon from starting a surgery without having finished an earlier one. The latter prevents a room from being in use with two surgeries at the same time. Note that a surgeon is free to start another surgery as soon as intervention duration  $\bar{d}_s$  ends, while a room is occupied with a surgery for  $d_s$  time periods. Capacity constraints (5) and (6) are associated with renewable and nonrenewable resources, respectively. Constraints (7) and (8) associate specialised human resources with surgeries. Whenever a surgery  $s$  is scheduled to start in a period  $t \in T_s$ , an individual  $a \in A_s^e$  of each type of specialised resource  $e \in E_s$  must be assigned as long as  $t$  belongs to his/her time windows  $T_s^e$ . Constraint (9) bounds the number of surgeries an individual can perform simultaneously. As already mentioned, the distance between two rooms  $r$  and  $\bar{r}$  may refrain an individual from performing two surgeries simultaneously. In this case, i.e.,  $b_{r\bar{r}} = 0$ , constraint (10) ensures that although an individual may be suitable to perform two surgeries  $s$  and  $\bar{s}$  occurring simultaneously in rooms  $r$  and  $\bar{r}$ , respectively, he/she can only be assigned to one of them. Finally, constraints (11) and (12) define the domain of the variables.

**4. Integer programming-based heuristics**

In many practical settings, integer programming formulations have been used to generate very good, but not necessarily optimal, solutions, e.g., see Ball (2011) and Wolsey (1998). Preliminary computational experiments have shown that the linear relaxation of (11) and (12) can produce very good upper bounds to the optimal solution values. For this reason, we use relaxations to propose two integer programming-based heuristics, which can be implemented with a mixed integer programming solver and we can assess the quality of solutions found. The heuristics consist of two main steps: first draw from relaxation candidate assignments of surgeries to rooms and periods, then solve a reduced model to obtain feasibility.

**4.1. Relax-and-fix heuristic**

We propose a relax-and-fix heuristic in which only the integrality of  $w$  is dropped. In the first step we solve formulation (1)–(11) with the LP relaxation of (12) to produce an upper bound and to construct a set of candidate surgery assignments. Let  $\bar{y}$  be an optimal (integer)  $y$ -vector to the relaxation. We denote by  $\Upsilon$  the set of candidate surgery assignments drew from the relaxation, i.e.,  $\Upsilon$  contains each surgery  $s$  for which there exists a room  $\eta_s$  and a period  $\tau_s$  such that  $\bar{y}_{s\tau_s}^{\eta_s} = 1$ . Although the surgery assignments selected in  $\Upsilon$  are feasible in terms of overlapping and renewable/nonrenewable resources, i.e., feasible to constraints (2)–(6), there may be no assignment of specialised resources which are feasible for constraints (7)–(10).

In the second step, we solve a reduced mixed integer programming model to obtain feasibility. The idea is for every  $s \in \Upsilon$  either to successfully assign a specialised resource or to not perform  $s$ . Thus, the binary variable  $w_s^a$  takes value 1 if surgery  $s \in \Upsilon$  is assigned to individual  $a \in A_s^e$  of specialised resource type  $e \in E_s$ , and 0 otherwise. Note now that variable  $w$  has a reduced dimension with respect to the original model, since  $s$  starts in  $\tau_s$  if it is to be performed. The continuous variable  $z_s$  naturally takes value 0 or 1, 1 if surgery  $s \in \Upsilon$  is not to be performed, and 0 otherwise.

We define a potential conflict set  $\Lambda_s, s \in \Upsilon$ , which contains every surgery  $s' \in \Upsilon - \{s\}$  that if started in  $\tau_{s'}$  will not be finished in  $\tau_s$ , i.e.  $\Lambda_s = \{s' \in \Upsilon - \{s\} | \tau_s - d_{s'} + 1 \leq \tau_{s'} \leq \tau_s\}$ . The set  $\bar{\Lambda}_s$  is defined analogously, except that only duration  $d_s$  of the intervention itself is considered. The set  $\Gamma_r, r \in R$ , contains candidate surgeries assigned to room  $r$ , i.e.,  $\Gamma_r = \{s \in \Upsilon | \eta_s = r\}$ . The reduced model can be written as follows:

$$\min \sum_{s \in \Upsilon} d_s z_s \tag{13}$$

$$w_s^a + \sum_{s' \in \bar{\Lambda}_s | a \in A_{s'}^e, \tau_{s'} \in T_s^e} w_{s'}^a \leq q_a \quad \forall s \in \Upsilon, \forall e \in E_s, \forall a \in A_s^e, \tau_s \in T_s^e \tag{14}$$

$$w_s^a + w_{s'}^a \leq 1 \quad \forall s \in \Upsilon, \forall s' \in \bar{\Lambda}_s, b_{\eta_s \eta_{s'}} = 0, \forall e \in E_s \cap E_{s'}, \forall a \in A_s^e \cap A_{s'}^e, \tau_s \wedge \tau_{s'} \in T_s^e, q_a > 1 \tag{15}$$

$$\sum_{a \in A_s^e | \tau_s \in T_s^e} w_s^a + z_s = 1 \quad \forall s \in \Upsilon, \forall e \in E_s \tag{16}$$

$$0 \leq z_s \leq 1 \quad \forall s \in \Upsilon \tag{17}$$

$$w_s^a \in \{0, 1\} \quad \forall s \in \Upsilon, \forall e \in E_s, \forall a \in A_s^e, \tau_s \in T_s^e \tag{18}$$

The objective function (13) minimises occupation losses due to candidate surgery assignments that cannot be performed. Constraint (14) bounds the number of surgeries an individual of a specialised resource type can perform simultaneously. Constraint (15) avoids the situation where an individual of a specialised resource type is assigned to two overlapping surgeries taking place in non compatible rooms. Constraint (16) ensures that either a candidate surgery has a specialised resource assigned to or it cannot be performed. Finally, constraints (17) and (18) define the domain of the variables. Since  $w$  is a binary variable, it follows from (16) that  $z$  assumes either 0 or 1.

At the end of the second step, the heuristic returns a feasible solution with an occupancy of  $\sum_{s \in \Upsilon} d_s (1 - \bar{z}_s)$ , where  $\bar{z}$  is an optimal  $z$ -vector to formulation (13)–(18).

One may not solve formulation (1)–(11) with the LP relaxation of (12) to optimality in order to obtain good feasible solutions to the original problem. Indeed, we note, for instance, that high quality solutions can be found when the formulation in the first step is solved to an optimality gap of 5 percent.

**4.2. LP-based heuristic**

We try to construct a larger set of candidate surgery assignments by solving the LP relaxation of formulation (1)–(12). Let  $\bar{y}$  be an optimal (not necessarily integer)  $y$ -vector to the LP relaxation. Given a surgery  $s \in S$ , let  $\eta_s$  be a room and  $\tau_s$  be a period yielding the greatest  $\bar{y}$  value, i.e.,  $(\eta_s, \tau_s) = \arg \max_{(r,t) \in R_s \times T_s} \{\bar{y}_{st}^r\}$ . The set  $\Upsilon = \{s \in S | \bar{y}_{s\tau_s}^{\eta_s} \geq \rho\}$  now contains each surgery  $s$  for which the assignment variable to  $\eta_s$  and  $\tau_s$  in the LP optimal solution is greater than or equal to a threshold value  $\rho$ . In addition to constraints (7)–(10), this set of candidate surgery assignments may not be feasible with respect to the overlapping constraints (3) and (4), and to the resource constraints (5) and (6) as well.

In the second step, we solve a reduced mixed integer programming model consisting of constraints (13)–(18) and constraints (19)–(22) given below.

$$z_s + z_{s'} \geq 1 \quad \forall c \in C, \forall s \in S_c \cap \Upsilon, \forall s' \in S_c \cap \bar{\Lambda}_s \tag{19}$$

$$z_s + z_{s'} \geq 1 \quad \forall r \in R, \forall s \in \Gamma_r, \forall s' \in \Gamma_r \cap \Lambda_s \tag{20}$$

$$g_{sk}^1 (1 - z_s) + \sum_{s' \in \bar{\Lambda}_s | \eta_{s'} \notin R_k} g_{s'k}^1 (1 - z_{s'}) \leq q_k^1 \quad \forall k \in K_1, \forall s \in \Upsilon, \eta_s \notin R_k \tag{21}$$

$$\sum_{s \in \Upsilon} g_{sk}^2 (1 - z_s) \leq q_k^2 \quad \forall k \in K_2 \tag{22}$$

Constraints (19) and (20) force one surgery to not be performed for every pair of two overlapping surgeries having the same surgeon

and/or room in common. The former prevents a surgeon from starting a surgery without having finished one that was started earlier. The latter prevents a room from being in use with two surgeries at the same time. Constraints (21) and (22) ensure that candidate surgery assignments which exceed renewable and nonrenewable resource capacities (respectively) are avoided.

## 5. Computational experiments

We generated instances based on real data from a Brazilian hospital to analyse the performance of the proposed methods on a variety of potential practical situations. However, we first discuss a real practical example to characterise a concrete case study.

### 5.1. A real example

The case study consists of 64 surgeries, which in practice means two days of scheduling. In total, 43 surgeons are involved with this set of surgeries. We have 24 periods of 30 minutes each. The sum of the duration of all surgeries is 367 periods, and cleaning takes 1 period after performing a surgery. There are two renewable resources: 3 mobile units of a microscope, and 5 mobile units of an image intensifier, which has also another unit installed on an operating room. The nonrenewable resource is an intensive care unit with 7 beds. The hospital has 14 rooms, divided into two blocks, for elective surgeries. Non-elective surgeries are undertaken in a third block of rooms. One main block has 11 rooms, and an auxiliary block has 3 rooms. Most of the surgeries, in this case 57 out of 64, have to be performed on the main block because they require renewable and/or nonrenewable resources. Cardiac surgeries have to be performed on a specific room. The same constraint applies to neurological surgeries, which have to be performed on another specific room. However, these specific rooms are not exclusive for cardiac and neurological surgeries, as other kinds of surgeries can be performed there as well. The specialised human resources are the 12 anaesthetists, which 6 of them work the whole day, 2 only in the morning, and 4 only in the afternoon. The set of surgeries each anaesthetist can perform is defined based on professional skills and personal preferences. They may perform at most 2 surgeries simultaneously, as long as they occur in the same block.

The integer model solved this case in 416 seconds on a Intel Core i3, 2.6 gigahertz with 4 gigabyte of RAM memory. In the optimal solution found, a total of 49 surgeries were scheduled summing up 292 periods of occupation. It corresponds to an utilisation of 87 percent. In the

practice, the surgical scheduling is done manually, and utilisation rates are about 45–50 percent. The hospital's goal is to have more than 60 percent of utilisation.

### 5.2. Experimental platform

We conducted computational experiments using a variety of scenarios built with data presented in the previous section. All experiments were carried out on an Intel Xeon X5690 @ 3.47 gigahertz with 24-CPU and 132 gigabyte of RAM memory running Linux. We conducted experiments with the integer model, the LP-based heuristic, the relax-and-fix heuristic, and the relax-and-fix heuristic with first step solved to an optimality gap of 5 percent (denoted relax-and-fix 5 percent). All the proposed models and methods were coded in AMPL and solved by CPLEX version 12.4 with parameters `presolve = 1`, `parallelmode = 1` to enable deterministic parallel search mode, and `timelimit = 21,600` seconds. Constraint (10) was treated with parameter `lazy` when running the integer model and the first step of the relax-and-fix heuristics. Parameter `mipemphasis` was set to 1 in the first step of the relax-and-fix 5 percent to emphasise feasibility over optimality, and set to 0 otherwise.

In total 144 instances were generated. All instances are available upon request. Several features, like the number of rooms and resources, are fixed to those found in practice; while others, like the number of surgeries and the number of surgeries anaesthetists can perform simultaneously, were generated from values consistent with the hospital's practice to cover diverse situations. The following data is fixed for all instances. The hospital has 14 rooms for elective surgeries. There are two renewable resources, i.e.,  $|K_1| = 2$ , and one nonrenewable resource, i.e.,  $|K_2| = 1$ . There are 6 surgeons, i.e.,  $|C| = 6$ . A day is divided in 24 periods, i.e.,  $|T| = 24$ . Different scenarios containing 3 instances for each value of  $|S| = 40, 50, 60, 70, 80, 90$  were generated by varying configurations from the basic scenario as described in the next section.

We report computational results got with the proposed models and methods on Tables 3–10. The tables have all the same structure. Objective function values are given in periods of occupation of the operating rooms, and running times are given in CPU seconds with CPLEX parameter `clocktype = 1`. The first two columns present the identification and the number of surgeries of each instance. Then, the next three columns present results obtained running the integer model (1)–(12). We report LB as the best feasible solution obtained,  $g(\text{percent})$  as the optimality gap returned by CPLEX, and the user CPU time as the sum of `AMPL_solve_user_time` and `_ampl_user_time`.

**Table 3**

Instances HC, similar to the real case. UB – upper bound, LB – lower bound,  $g(\text{percent})$  – percentage gap given by  $\frac{UB-LB}{LB}$ ,  $t$  – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent				
		LB	$g(\text{percent})$	$t$	UB	$t$	LB	$t$	$g(\text{percent})$	$t$	UB	$t$	LB	$t$	$g(\text{percent})$	$t$	LB	$g(\text{percent})$	$t$
HC1	40	76	0.0	12	77	1	72	0	6.9	2	76	57	76	0	0.0	58	76	1.3	2
HC2	40	115	0.0	146	117	12	91	0	28.6	19	115	5	115	0	0.0	22	114	2.6	21
HC3	40	82	0.0	28	83	3	74	0	12.2	6	82	3	82	0	0.0	17	80	3.8	15
HC4	50	98	0.0	17,932	102	6	83	0	22.9	11	98	90	98	0	0.0	109	98	3.1	24
HC5	50	117	0.0	4113	120	17	95	0	26.3	25	117	34	114	0	2.6	63	111	5.4	46
HC6	50	77	0.0	34	79	2	70	2	12.9	9	77	2	77	0	0.0	20	74	4.1	19
HC7	60	95	0.0	62	97	6	87	0	11.5	11	95	2	95	0	0.0	27	95	1.1	26
HC8	60	107	0.0	14,090	110	10	94	0	17.0	17	107	282	107	0	0.0	314	105	2.9	32
HC9	60	89	0.0	34	90	3	83	0	23.3	6	89	2	89	0	0.0	18	86	3.5	17
HC10	70	81	0.0	60	81	7	72	0	12.5	14	81	3	81	0	0.0	36	79	2.5	31
HC11	70	99	0.0	65	100	5	86	0	16.3	14	99	3	99	0	0.0	49	96	3.1	43
HC12	70	100	0.0	49	103	5	89	0	15.7	10	100	10	100	0	0.0	39	97	4.1	29
HC13	80	101	0.0	118	103	12	92	0	12.0	25	101	4	101	0	0.0	67	101	0.0	61
HC14	80	121	0.0	96	123	15	99	0	24.2	24	121	8	121	0	0.0	58	120	2.5	48
HC15	80	112	0.0	260	113	24	90	0	25.6	37	112	6	112	0	0.0	74	109	2.8	68
HC16	90	102	0.0	138	102	14	97	0	5.2	28	102	5	102	0	0.0	74	99	3.0	68
HC17	90	100	0.0	98	101	7	82	0	23.2	15	100	3	100	0	0.0	53	100	0.0	51
HC18	90	101	0.0	128	102	7	93	0	9.7	18	101	3	101	0	0.0	59	98	4.1	56

**Table 4**

Instances HB, compatible with adjacent room only. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ , t – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent				
		LB	g(percent)	t	UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t	LB	g(percent)	t
HB1	40	76	0.0	31	77	2	71	0	8.5	14	76	1	76	0	0.0	14	75	1.3	16
HB2	40	115	0.0	398	117	15	103	0	13.6	59	115	82	115	0	0.0	122	113	1.8	81
HB3	40	82	0.0	122	83	7	79	0	5.1	33	82	6	82	0	0.0	32	82	1.2	33
HB4	50	98	2.0	21,816	102	15	82	0	24.4	54	98	90	98	0	0.0	126	97	3.1	55
HB5	50	112	4.5	22,140	120	25	92	0	30.4	89	117	170	117	0	0.0	229	113	3.5	392
HB6	50	77	0.0	82	79	7	66	0	19.7	45	77	3	77	0	0.0	48	77	0.0	48
HB7	60	95	0.0	922	97	13	87	0	11.5	58	95	4	95	0	0.0	59	94	2.1	58
HB8	60	107	0.0	4654	110	29	94	0	17.0	82	107	6	107	0	0.0	69	105	1.9	67
HB9	60	89	0.0	64	90	6	74	0	21.6	30	89	30	89	0	0.0	61	88	1.1	52
HB10	70	81	0.0	150	81	17	71	0	14.1	77	81	4	81	0	0.0	74	81	0.0	72
HB11	70	99	0.0	142	100	16	95	0	5.3	96	99	10	99	0	0.0	103	99	0.0	102
HB12	70	100	0.0	146	103	11	88	0	17.0	62	100	5	100	0	0.0	63	100	0.0	74
HB13	80	101	0.0	309	103	66	91	0	13.2	183	101	237	101	0	0.0	370	101	2.0	148
HB14	80	121	0.0	198	123	40	94	0	30.9	129	121	190	121	0	0.0	291	121	1.7	119
HB15	80	112	0.0	1331	113	33	94	0	20.2	182	112	22	112	0	0.0	163	112	0.9	173
HB16	90	102	0.0	324	102	27	94	0	8.5	164	102	8	102	0	0.0	162	102	0.0	151
HB17	90	100	0.0	180	101	19	79	0	27.8	115	100	6	100	0	0.0	118	100	0.0	111
HB18	90	101	1.0	21,749	102	18	94	0	8.5	133	101	156	101	0	0.0	285	101	1.0	161

**Table 5**

Instances HA, with 40 percent less individuals considered as specialised resources. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ , t – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent				
		LB	g(percent)	t	UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t	LB	g(percent)	t
HA1	40	75	0.0	72	75	0	57	0	31.6	5	75	6	75	0	0.0	12	65	15.4	3
HA2	40	115	0.0	896	117	5	95	0	23.2	19	115	9872	115	0	0.0	9889	114	2.6	26
HA3	40	82	0.0	696	83	1	75	0	10.7	12	82	1	82	0	0.0	13	81	2.5	9
HA4	50	98	0.0	12,068	102	4	81	0	25.9	19	98	104	98	0	0.0	122	96	4.2	14
HA5	50	117	0.0	2399	120	11	94	0	27.7	30	117	10	107	1	9.3	33	108	8.3	44
HA6	50	77	0.0	15	79	1	71	0	11.3	12	77	1	77	0	0.0	14	75	4.0	11
HA7	60	95	0.0	54	97	10	86	0	12.8	25	95	2	95	0	0.0	27	92	3.3	20
HA8	60	103	0.0	229	108	6	77	3	40.3	23	103	95	91	1	13.2	117	97	8.2	162
HA9	60	65	0.0	11	65	1	52	0	25.0	7	65	1	65	0	0.0	12	62	4.8	10
HA10	70	81	0.0	30	81	4	63	0	28.6	19	81	2	81	0	0.0	21	79	2.5	17
HA11	70	99	0.0	36	100	4	89	0	12.4	26	99	5	99	0	0.0	35	96	4.2	26
HA12	70	100	0.0	30	103	4	87	0	18.4	21	100	15	100	0	0.0	37	100	0.0	24
HA13	80	101	0.0	78	103	14	87	0	18.4	50	101	3	101	0	0.0	49	98	4.1	40
HA14	80	121	0.0	43	123	6	105	0	17.1	28	121	3	121	0	0.0	34	120	0.8	35
HA15	80	112	0.0	283	113	13	100	0	13.0	38	112	5	112	0	0.0	39	108	3.7	42
HA16	90	102	0.0	91	102	10	84	0	21.4	54	102	3	102	0	0.0	55	100	2.0	49
HA17	90	100	0.0	149	101	7	88	0	14.8	42	100	7	100	0	0.0	51	100	1.0	56
HA18	90	101	0.0	144	102	10	86	0	18.6	42	101	93	101	0	0.0	135	98	4.1	45

**Table 6**

Instances HE, with two types of specialised resources, i.e., |E| = 2. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ , t – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent				
		LB	g(percent)	t	UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t	LB	g(percent)	t
HE1	40	76	0.0	49	77	5	67	0	14.9	27	76	2	76	0	0.0	6	74	4.1	6
HE2	40	115	0.0	1121	117	27	99	0	18.2	78	115	8	115	0	0.0	45	115	0.9	44
HE3	40	82	0.0	72	83	8	73	0	13.7	39	82	5	82	0	0.0	34	81	2.5	29
HE4	50	98	0.0	347	102	720	81	0	25.9	763	98	21	98	0	0.0	66	95	4.2	44
HE5	50	117	0.0	1457	120	2134	93	0	29.0	2206	117	511	112	0	4.5	580	111	5.4	255
HE6	50	77	0.0	79	79	7	70	0	12.9	48	77	3	77	0	0.0	39	75	2.7	43
HE7	60	95	0.0	212	97	1545	83	0	16.9	1600	95	11	95	0	0.0	70	95	1.1	112
HE8	60	107	0.9	21,765	110	553	93	0	18.3	626	107	1717	107	0	0.0	1795	105	3.8	71
HE9	60	89	0.0	209	90	12	73	0	23.3	49	89	8	89	0	0.0	53	84	6.0	46
HE10	70	81	0.0	163	81	28	66	0	22.7	98	81	5	81	0	0.0	79	81	0.0	80
HE11	70	99	0.0	546	100	20	84	0	19.0	105	99	5	99	0	0.0	97	98	1.0	89
HE12	70	100	0.0	724	103	41	80	0	28.8	118	100	319	100	0	0.0	400	98	3.1	231
HE13	80	101	0.0	368	103	2708	94	0	9.6	2822	101	7	101	0	0.0	139	101	0.0	133
HE14	80	121	0.0	432	123	80	104	0	18.3	194	121	276	121	0	0.0	414	121	0.8	160
HE15	80	112	0.0	2052	113	2491	87	0	29.9	2610	112	71	112	0	0.0	216	110	2.7	350
HE16	90	102	0.0	297	102	1070	89	0	14.6	1190	102	8	102	0	0.0	154	102	0.0	141
HE17	90	100	0.0	193	101	375	82	0	23.2	474	100	12	100	0	0.0	122	100	1.0	118
HE18	90	101	0.0	359	102	55	87	0	17.2	182	101	48	101	0	0.0	183	101	0.0	155

**Table 7**  
Instances HL, with time windows only for the specialised resources. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ , t – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent							
		LB	g(percent)	t	UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t	LB	g(percent)	t			
HL1	40	120	0.0		423	122	21	101	0	20.8		63	120	8	120	0	0.0		56	117	2.6	60
HL2	40	132	0.0		1284	134	26	117	0	14.5		77	132	415	132	0	0.0		476	130	3.1	68
HL3	40	114	0.0		136	114	11	112	0	1.8		63	114	10	114	0	0.0		76	110	3.6	70
HL4	50	165	1.2	21,825	168	97	138	0	21.7		185	166	9488	166	0	0.0		9596	165	1.8	263	
HL5	50	153	0.7	21,792	154	63	122	0	26.2		129	153	5347	153	0	0.0		5414	153	0.7	295	
HL6	50	132	0.0	1276	134	41	126	0	6.3		141	132	3317	132	0	0.0		3445	130	3.1	191	
HL7	60	173	0.6	21,860	174	95	155	0	12.3		169	173	1435	173	0	0.0		1530	169	3.0	105	
HL8	60	163	9.8	21,744	181	3607	137	0	32.1		3698	178	2160	178	0	0.0		2257	178	1.7	545	
HL9	60	156	1.9	21,761	161	91	126	0	27.8		170	159	15,466	159	0	0.0		15,569	159	0.6	714	
HL10	70	153	0.7	22,366	156	95	117	0	33.3		179	154	2566	154	0	0.0		2663	149	4.7	301	
HL11	70	177	0.0	6703	179	139	137	0	30.7		308	179	21,611	177	0	1.1		21,823	177	1.1	374	
HL12	70	173	1.7	21,778	177	111	121	0	46.3		194	174	3987	174	0	0.0		4079	170	4.1	254	
HL13	80	191	1.0	21,799	193	5802	148	0	30.4		5990	193	21,616	191	0	1.0		21,822	187	3.2	2521	
HL14	80	186	0.0	18,997	187	148	157	0	19.1		267	186	2355	186	0	0.0		2508	185	1.1	835	
HL15	80	184	0.0	18,944	186	297	141	0	31.9		492	184	3873	184	0	0.0		4086	183	1.6	445	
HL16	90	180	0.0	8212	181	327	148	0	22.3		543	180	4194	180	0	0.0		4470	180	0.6	830	
HL17	90	-	-	21,768	192	3551	138	0	39.1		3710	192	21,607	188	0	2.1		21,782	188	2.1	11,669	
HL18	90	189	0.5	21,835	190	318	139	0	36.7		536	190	21,610	189	0	0.5		21,845	188	1.1	450	

**Table 8**  
Instances HS, with no time windows. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ , t – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent							
		LB	g(percent)	t	UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t	LB	g(percent)	t			
HS1	40	120	0.0		312	122	22	104	0	17.3		73	120	550	120	0	0.0		610	118	3.4	64
HS2	40	132	0.0		458	134	14	116	0	15.5		71	132	234	132	0	0.0		291	131	2.3	76
HS3	40	114	0.0		199	114	11	110	0	3.6		70	114	6	114	0	0.0		77	114	0.0	68
HS4	50	165	1.2	21,875	168	67	144	0	16.7		165	167	21,605	166	0	0.6		21,720	162	3.7	165	
HS5	50	153	0.7	22,020	154	82	141	1	9.2		177	153	365	153	0	0.0		475	153	0.7	118	
HS6	50	132	0.0	1283	134	41	126	0	6.3		152	132	3378	132	0	0.0		3485	130	3.1	197	
HS7	60	173	0.0	5048	174	82	158	0	10.1		170	173	1406	173	0	0.0		1515	170	2.4	116	
HS8	60	177	0.6	21,694	181	129	131	0	38.2		243	178	1846	178	0	0.0		1979	177	2.3	350	
HS9	60	160	0.6	21,734	163	94	124	0	31.5		228	160	991	160	0	0.0		1125	159	1.9	315	
HS10	70	154	1.3	21,961	156	160	120	0	30.0		335	154	905	154	0	0.0		1080	154	1.3	287	
HS11	70	177	0.0	7961	179	181	137	0	30.7		353	177	1218	177	0	0.0		1396	175	2.3	233	
HS12	70	174	0.6	21,919	177	105	136	0	30.1		215	174	5598	174	0	0.0		5710	174	1.7	327	
HS13	80	189	2.1	21,846	193	285	146	0	32.2		474	191	3778	191	0	0.0		3987	186	3.8	413	
HS14	80	186	0.0	3675	187	92	149	0	25.5		244	186	77	186	0	0.0		271	186	3.8	217	
HS15	80	184	0.5	21,822	186	281	152	0	22.4		529	184	5320	184	0	0.0		5610	182	2.2	636	
HS16	90	180	0.0	11,031	181	246	148	0	22.3		544	180	3630	180	0	0.0		3940	180	0.6	582	
HS17	90	185	3.2	21,897	192	197	136	0	41.2		411	191	21,609	188	0	1.6		21,877	185	3.8	2182	
HS18	90	188	1.1	21,852	190	308	149	0	27.5		564	189	1880	189	0	0.0		2186	188	1.1	533	

**Table 9**  
Instances HLA, with 40 percent less individuals as specialised resources and time windows for them. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ , t – computational time in seconds.

I	S	IP model			LP-based heuristic					Relax-and-Fix heuristic					Relax-and-Fix 5 percent						
		LB	g(percent)	t	UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t	LB	g(percent)	t		
HLA1	40	120	1.7	21,640	122	9	90	0	35.6		27	120	7	120	0	0.0		10	118	3.4	28
HLA2	40	131	0.8	21,778	134	9	112	0	19.6		37	132	664	132	0	0.0		682	131	2.3	42
HLA3	40	114	0.0	155	114	8	108	0	5.6		37	114	16	114	0	0.0		38	112	1.8	43
HLA4	50	165	1.2	21,674	168	1577	124	0	35.5		1621	168	21,604	166	0	1.2		21,640	165	1.8	627
HLA5	50	150	2.7	21,643	154	31	139	0	10.8		64	154	21,643	151	0	2.0		21,671	144	6.9	563
HLA6	50	132	0.0	1643	134	12	125	0	7.2		59	132	358	132	0	0.0		398	129	3.9	57
HLA7	60	173	0.6	21,728	174	93	136	0	27.9		152	173	126	173	0	0.0		171	168	3.6	83
HLA8	60	124	30.6	21,651	162	2717	116	0	39.7		2764	162	1384	160	0	1.3		1422	160	1.2	986
HLA9	60	136	1.5	21,663	138	58	94	0	46.8		98	138	12	138	0	0.0		46	136	1.5	55
HLA10	70	152	2.6	22,622	156	83	114	0	36.8		128	156	21,605	154	0	1.3		21,644	151	3.3	75
HLA11	70	177	0.0	12,288	179	130	129	0	38.8		227	179	21,745	177	0	1.1		21,847	172	4.1	114
HLA12	70	174	0.6	22,282	177	79	127	0	39.4		139	174	7434	174	0	0.0		7478	171	2.3	76
HLA13	80	-	-	22,288	193	2660	134	0	44.0		2761	193	21,606	187	2	3.2		21,700	185	4.3	157
HLA14	80	185	1.1	21,730	187	934	160	0	16.9		986	186	292	186	0	0.0		352	181	3.3	91
HLA15	80	184	0.0	5047	186	171	136	0	36.8		238	186	21,619	184	0	1.1		21,723	184	1.1	116
HLA16	90	175	3.4	21,779	181	2980	134	0	35.0		3112	181	21,694	177	0	2.3		21,820	173	4.6	178
HLA17	90	104	83.6	21,722	191	4944	126	0	51.6		5081	191	21,609	187	0	2.1		21,739	184	3.8	15,657
HLA18	90	149	27.5	21,770	190	262	152	0	25.0		398	190	21,607	189	0	0.5		21,749	186	2.1	3218



**Table 10**

Instances HLB, with adjacent room compatibility only and time windows for the specialised resources. UB – upper bound, LB – lower bound, g(percent) – percentage gap given by  $\frac{UB-LB}{LB}$ ,  $t$  – computational time in seconds.

I	S	IP model				LP-based heuristic				Relax-and-Fix heuristic				Relax-and-Fix 5 percent						
		LB	g(percent)	t		UB	t	LB	t	g(percent)	t	UB	t	LB	t	g(percent)	t			
HLB1	40	120	0.0		9069	122	37	94	0	29.8	119	120	194	120	0	0.0	210	120	0.0	35
HLB2	40	131	1.5	21,969	134	39	122	0	9.8	137	132	330	132	0	0.0	388	132	1.5	62	
HLB3	40	114	0.0	485	114	57	102	0	11.8	163	114	12	114	0	0.0	76	114	0.0	71	
HLB4	50	164	1.8	21,802	168	2858	134	0	25.4	2999	167	21,612	166	0	0.6	21,724	162	3.7	310	
HLB5	50	74	108.1	21,763	154	96	128	0	20.3	230	153	16,308	149	0	2.7	16,383	145	6.2	1604	
HLB6	50	132	0.0	4405	134	126	127	0	5.5	351	132	337	132	0	0.0	483	132	0.8	269	
HLB7	60	173	0.0	4715	174	105	154	0	13.0	263	173	1708	173	0	0.0	1809	171	1.7	125	
HLB8	60	108	67.6	21,817	181	2729	131	0	38.2	2925	178	7572	175	0	1.7	7711	178	0.7	7637	
HLB9	60	153	3.9	21,792	161	197	126	0	27.8	342	160	21,608	154	0	3.9	21,729	153	4.6	158	
HLB10	70	150	4.0	21,950	156	217	115	2	35.6	406	154	2895	154	0	0.0	2937	152	2.0	1921	
HLB11	70	177	0.6	22,071	179	287	157	0	14.0	682	179	21,619	177	0	1.1	21,711	177	1.1	797	
HLB12	70	173	1.2	21,870	177	219	118	0	50.0	398	174	13,475	174	0	0.0	13,514	168	4.8	661	
HLB13	80	189	2.1	22,123	193	1651	159	0	21.4	2082	193	21,716	191	0	1.0	21,837	187	3.2	1155	
HLB14	80	185	0.5	21,968	187	790	154	0	21.4	1026	186	650	184	0	1.1	712	176	6.2	1323	
HLB15	80	184	0.0	21,038	186	13,682	158	0	17.7	14,201	186	21,626	184	0	1.1	21,879	178	4.5	2029	
HLB16	90	180	0.6	22,217	181	9412	129	0	40.3	9831	181	21,687	180	0	0.6	21,798	180	0.7	4147	
HLB17	90	-	-	22,887	192	21,659	142	0	35.2	22,061	191	21,624	175	0	9.1	21,823	151	26.5	21,975	
HLB18	90	185	2.7	21,819	190	6847	143	0	32.9	7383	189	21,694	188	0	0.5	21,966	186	2.2	4040	

In the columns related to the LP-based heuristic (cf. Section 4.2) we report UB as the upper bound due to LP-relaxation, the user CPU time returned by `AMPL_solve_user_time` to solve LP-relaxation (i.e., the first step of the heuristic), LB as the best feasible solution obtained with the reduced model (13)–(22) (i.e., the second step of the heuristic), the user CPU time returned by `AMPL_solve_user_time` to solve the reduced model, g(percent) as the percentage gap between UB and LB given by  $\frac{UB-LB}{LB}$ , and the user CPU time as the sum of `AMPL_total_solve_user_time` and `AMPL_solve_user_time`. The columns related to the Relax-and-Fix heuristic are analogous to those related to the LP-based heuristic, except that we report UB as the upper bound obtained with the relaxation of (12) and LB as the best feasible solution obtained with the reduced model (13)–(18). In the last three columns, which are related to the Relax-and-Fix 5 percent, we report LB as the best feasible solution obtained with the reduced model (13)–(22), g(percent) as the percentage gap computed with respect to the upper bound returned by CPLEX when the procedure stops at the first step (not necessarily with an optimal solution to the relaxation of (12)), and the total user CPU time as the sum of `AMPL_total_solve_user_time` and `AMPL_solve_user_time` (including the two steps of the heuristic, given that the time needed to solve the reduced model is negligible).

Note that each column g(percent) is computed solely with information obtained by the respective method, and not with respect to optima or best known upper bounds. As such, each column g(percent) gives the posteriori performance guarantees obtained with each method. Average gaps with respect to optimal or best known solutions are presented in an overview at the end of this section. In the LP-based heuristic, the threshold value  $\rho$  can be tuned to size the problem to be solved in the second step of the heuristic. Preliminary computational results have shown that the reduced model (13)–(22) is solved very quickly. For this reason we set  $\rho$  to 0.01, and take almost all surgeries with an active  $y$  variable in the LP-relaxation of the integer model (1)–(12) to build set  $\Upsilon$  (cf., Section 4.2). Thus, we give more options to the reduced model without significantly increasing total computational times as the second step of the heuristic is solved very quickly.

### 5.3. Basic scenario

The parameter values for these basic instances, denoted by HC, were generated from guidelines provided by the staff charged with

the surgical scheduling at the hospital. The rooms are divided into two groups, one with 3 rooms and another with 11 rooms. The rooms within each group are compatible with the simultaneous use of specialised resources, but no simultaneous use can be undertaken with rooms belonging to different groups. There is just one type of specialised human resource, i.e.,  $|E| = 1$ , with 10 individuals, i.e.,  $|A| = 10$ . For each instance, between 30 percent and 50 percent of the individuals belonging to  $A$  are able to perform 2 surgeries simultaneously, i.e.,  $q_a = 2$ , while the others are unable to perform surgeries simultaneously, i.e.,  $q_a = 1$ . The duration of surgeries were generated from a discrete uniform distribution between 1 and 11 periods. The distribution of surgeries among surgeons to form each set  $S_c$  was done at random. Each surgery  $s$  has a probability of 90 percent to require a specialised human resource. Individuals from  $A$  were chosen at random to form  $A^s$ . Also, for each surgery  $s$ , rooms from  $R$  were chosen at random to form  $R_s$ . Each surgery  $s$  has a probability between 10 percent and 20 percent to require renewable resources, and between 0 percent and 15 percent to require the nonrenewable ones. Resource capacities were generated from a discrete uniform distribution between 1 and 6 for renewable resources, and between 3 and 6 for the nonrenewable resources. A room has a probability of 2 percent to have a renewable resource installed. Resources are used at the rate of one per surgery when needed, i.e.,  $g_{sk}^1$  and  $g_s^2$  are either 0 or 1. The medical staff work in morning shifts, in afternoon shifts, or for the whole day. So, time windows intervals were randomly chosen as being the first 12 periods, the last 12 periods, or all the 24 periods.

Table 3 presents results obtained for the HC instances. We were able to solve to optimality all these instances, although computational times to run the integer model are considerably high for some instances. The proposed model provides tight LP-relaxation bounds, as can be seen on column UB of the LP-based heuristic. However, LP-relaxation solution values were not useful to derive good feasible solutions, as one might have expected. Indeed, the LP-based heuristic did not have a good performance returning gaps of more than 10% for almost all instances. The relax-and-fix has been shown to be very efficient on these instances which are similar to the real case. Computational times are moderate, even on the instances which the model has taken a long time to run, and it was able to find optimal solutions for all instances except HC5. For this instance, the upper bound obtained in the first step is optimal, but the set of surgeries selected are not feasible with respect to the assignment of specialised resources. The reduced model (13)–(18) runs extremely quickly. The

**Table 11**  
Overview on the performance of each method.

	IP model gap(percent)				LP-based heuristic gap(percent)				Relax-and-Fix heuristic gap(percent)				Relax-and-Fix 5 percent gap(percent)				Open
	Max	Av	Best	Av-t	Max	Av	Best	Av-t	Max	Av	Best	Av-t	Max	Av	Best	Av-t	
HC	0.0	0.0	18	2081	26.4	14.2	0	16	2.6	0.1	17	64	5.4	2.0	5	36	0
HB	4.5	0.2	15	4153	28.7	14.6	0	89	0.0	0.0	18	133	3.5	0.7	11	106	0
HA	0.0	0.0	18	962	33.8	18.6	0	26	13.2	1.3	16	594	15.4	3.5	2	35	0
HE	0.0	0.0	18	1691	28.7	17.8	0	735	4.5	0.2	17	249	6.0	1.6	8	117	0
HL	Inf	–	12	15,250	43.8	23.3	0	940	0.0	0.0	18	7972	3.6	1.2	6	1110	3
HS	1.6	0.2	13	13,810	38.2	21.4	0	279	0.0	0.0	18	4219	2.7	0.9	6	382	2
HLA	Inf	–	7	18,061	48.4	29.0	0	996	0.0	0.0	18	11,452	4.9	1.8	2	1231	9
HLB	Inf	–	8	18,098	47.5	22.6	0	3644	1.7	0.1	16	12,149	15.9	2.2	7	2684	9

relax-and-fix 5 percent seems to be a fast alternative to obtain good results, because within a minute the heuristic returned solutions with a posteriori performance guarantee of no more than 5.4 percent and it obtained 5 out of 18 optima.

#### 5.4. Scenarios varying specialised resources

We generated three different scenarios, from the basic one, by parameters associated to specialised resources were varied. Each of the instances denoted HB, HA and HE was generated from its counterpart of HC by changing one parameter at a time: compatibility between rooms, the cardinality of set  $A$ , and the cardinality of set  $E$ . In HB instances, rooms are no longer divided into groups, and parameter  $b_{r\bar{r}}$ , cf., constraint (10), assumes 1 only if rooms  $r$  and  $\bar{r}$  are adjacent to each other. HA instances have 40 percent less individuals considered as specialised human resources than HC instances. In HE instances, we consider two types of specialised resources, i.e.,  $E = \{1, 2\}$ . Type 1 is the type of specialised resource already present in HC instances. Type 2 was generated with 7 individuals, i.e.,  $|A_2| = 7$ . For each instance, between 30 percent and 50 percent of the individuals belonging to  $A_2$  are able to perform 2 surgeries simultaneously, each surgery  $s$  has a probability of 90 percent to require a specialised human resource of type 2, and individuals from  $A_2$  were chosen at random to form  $A_2^s$ . Note that a surgery may require both specialised resources, just one of them, or none of them.

Tables 4–6 present results for the HB, HA and HE instances, respectively. The parameter of compatibility between rooms and the cardinality of  $E$  impacts upon the performance of solving the problem with the integer model, as for 4 instances the model exceeded the time limit without closing the optimality gap. The relax-and-fix performed well to the varying of these parameters. In fact, it obtained all optima for instances HB and HE, except for HE5. It did this in reasonable computational times, except for HE8 which took longer than 10 minutes. The relax-and-fix 5 percent also performed well on instances HB and HE, presenting small gaps and reasonable computational times. It obtained optimal solutions for 11 and 8 instances, respectively. On the other hand, the HA instances were easier for the integer model to solve due to the reduced number of variables. They were also more challenging to the relax-and-fix heuristics which present larger gaps than observed in the previous instances, due to fewer possibilities to make solutions feasible in the second step. In terms of LP-relaxation, although the model still provides tight LP-relaxation upper bounds with the modified parameters, the LP-based heuristic had a poor performance, with some very high gaps. We remark that, for some instances, the LP-based heuristic takes longer to run than the Relax-and-Fix heuristics because in the latter cases constraint (10) was treated with parameter  $\lambda_{azy}$  when running their first step. Indeed, it was observed that with a simple experiment of removing constraint (10) to run the linear relaxation of the integer model the computational times of the LP-based heuristic were significantly reduced.

#### 5.5. Scenarios varying time windows

We generated scenarios varying time windows, also taking the HC instances as a starting point. The HL instances have time windows only for the specialised resources, i.e.,  $T_c = T_s = T$  and  $T_c^a \subseteq T$ , while the HS instances have no time windows at all. Tables 7 and 8 present results for instances HL and HS, respectively. Because time windows allow us to eliminate variables in time indexed models, these instances were much harder to solve due to the increasing number of variables. We were not able, by running the integer model, to close the optimality gaps for 10 instances of each scenario within the time limit, and even to obtain a feasible solution for instance HL17. On the other hand, with relax-and-fix, we were able to close the optimality gaps for most of the instances. Nevertheless, these results were obtained with high computational times, and five instances remain open: HL13, HL17, HL18, HS5, HS17. The relax-and-fix 5 percent has been shown to be an efficient alternative to tackle these harder instances, as it obtained results with a posteriori performance guarantee of less than 5 percent (including 5 of HL and 6 of HS optima) in reasonable computational times, except for instance HL17 which took longer than the others.

We then generated critical scenarios in terms of specialised resources. Surgeon and surgery time windows have been removed from instances HA and HB to generate instances HLA and HLB, respectively. So, in these instances, in addition to only specialised resources being subject to time windows, HLA has 40 percent less individuals, and HLB has only adjacent rooms which are compatible with each other. Tables 9 and 10 present results for instances HLA and HLB, respectively. These seem to be the hardest instances in our suite of experiments. We were not able, neither with the integer model nor with the relax-and-fix, to close the optimality gaps for half of the instances of each scenario. The relax-and-fix 5 percent again has been shown to be the best alternative to tackle these harder instances, as it obtained results with a posteriori performance guarantee of less than 7 percent, except for instance HLB17. In most of the cases it did so with reasonable computational times.

#### 5.6. Overview

Table 11 gives an overview on the performance of each method in terms of solution quality and computational time. The first column identifies the scenario, and then, for each method, we present the maximum and the average percentage gaps with respect to optima or to best known solution values, the number of best solutions obtained, and the average computational time in seconds. The last column reports how many instances out of 18 for each scenario remain open after considering best upper and lower bounds obtained in the experiment. The gap is computed as  $\frac{(BLB-LB)}{LB}$ , where BLB is the optimum or the best know lower bound and LB is the lower bound obtained by the method. The maximum gap is indicated with “inf” when the integer model was not able to find a feasible solution for at least one instance within the time limit of 21,600 seconds, and in this case the

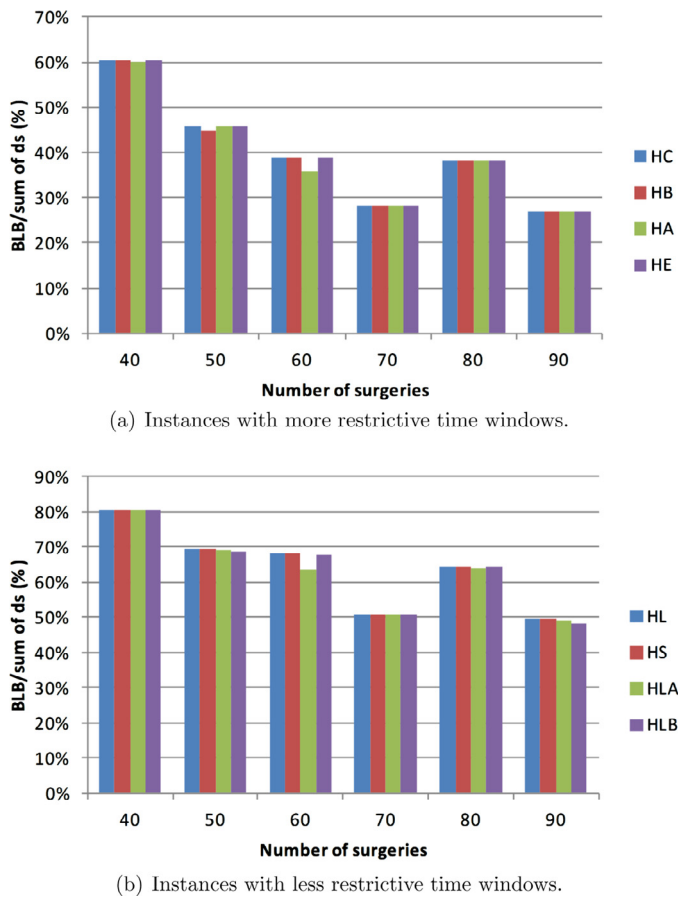


Fig. 1. Percentage of surgeries performed with respect to the total of surgeries.

average gap is not computed. This happened for some instances with less restrictive time windows, which seem to be harder. For such scenarios, namely HL, HS, HLA, and HLB, the integer model needed high computational times, very often reaching the time limit without proving optimality. As has already been observed, the LP-based heuristic fails in obtaining high quality feasible solutions although the linear relaxation provides good upper bounds. The relax-and-fix strategy has been shown to be quite successful as it obtains optimal or near optimal solutions for all scenarios. The Relax-and-Fix heuristic obtained consistently the best solutions, although it needed in average about 3 hours of computational time for the harder instances such as HLA and HLB. The Relax-and-Fix 5 percent has been shown to be an effective alternative, obtaining small average gaps to optima or best known lower bounds in acceptable average computational times (less than 1 hour).

Fig. 1 shows, in percentage, optimal solution values (or best known solution values when optima are not available) with respect to the trivial upper bound if all surgeries could be performed, i.e.,  $\sum_{s \in S} d_s$ . Part (a) refers to instances with more restrictive time windows. A bar refers to the average over the three instances for each cardinality of  $S$  in each scenario HC, HB, HA, and HE. We see that instances are not “loose” since solution values are at most 60 percent of the trivial bound, and for larger instances these values are not even 40 percent. Part (b) refers to scenarios HL, HS, HLA, and HLB with less restrictive time windows. The less restrictive time windows have a great impact on the number of surgeries that can be actually scheduled, and solution values are at least about 50 percent of the trivial bound. This second group of instances have an increasing number of variables and they seem to be harder since 23 out of 72 instances remain open.

## 6. Concluding remarks

We have addressed a surgery scheduling problem from a hospital in Brazil. In particular, we have modelled the simultaneous employment of specialised human resources, which in our case were anaesthetists. This enlarged the decision space since the assignment of anaesthetists to surgeries must be carried out jointly with the scheduling and assignment of surgeries to rooms. We developed integer programming based heuristics decoupling decisions into two steps because the integer model has been shown to provide tight bounds. We generated instances inspired by real data in order to conduct computational experiments on a number of scenarios covering different practical situations. The proposed relax-and-fix heuristic has been successful in finding optimal or near optimal solutions, and a variant that stops with a gap of 5 percent when assigning surgeries to rooms has been shown to be an effective alternative finding good quality solutions in less than 1 hour on average.

It could be observed that sometimes the Relax-and-Fix heuristic has selected (in its first step) surgeries leading to an optimal objective function value, but which were not feasible in terms of specialised resources assignment. Thus, there may have many combinations of surgeries leading to optimal values, some of them feasible in terms of specialised resources assignment and some of them not. Based on this observation, a line of future research is to introduce ideas of local branching (Fischetti & Lodi, 2003) in the second step of the heuristic. The reduced model would not only decide on assignment of specialised resources, but also decide on changing the allocation of some surgeries to rooms and periods within neighbourhoods obtained through the introduction of local branching cuts. The purpose of local branching neighbourhoods would be to rearrange a subset of surgeries among rooms and periods to increase the number of surgeries that can actually be scheduled with a successful assignment of specialised resources. Another line of research is to develop a meta-heuristic that first uses heuristic rules to select surgeries, and then applies the reduced model to assign specialised resources. In a VNS (Hansen, Mladenović, & Moreno Pérez, 2010) like manner a shaking phase could make changes rearranging and/or increasing the set of surgeries proposed to be scheduled, whereas the reduced model would impose feasibility.

## Acknowledgements

The authors wish to thank Heloísa Sampaio for useful discussions. The authors wish to thank the two anonymous referees for helpful suggestions in improving this paper. This research was partially supported by Brazilian funding agencies CAPES, CNPq, and FAPEMIG.

## References

- Augusto, V., Xie, X., & Perdomo, V. (2008). Operating theatre scheduling using Lagrangian relaxation. *European Journal of Industrial Engineering*, 2, 172–189.
- Augusto, V., Xie, X., & Perdomo, V. (2010). Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers & Industrial Engineering*, 58, 231–238.
- Ball, M. O. (2011). Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16, 21–38.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2009a). Optimizing a multiple objective surgical case sequencing problem. *International Journal of Production Economics*, 119, 354–366.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2009b). Sequencing surgical cases in a day-care environment: An exact branch-and-price approach. *Computers & Operations Research*, 36, 2660–2669.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201, 921–932.
- Dexter, F., Macario, A., & Traub, R. D. (1999). Which algorithm for scheduling add-on elective cases maximizes operating room utilization? Use of bin packing algorithms and fuzzy constraints in operating room management. *Anesthesiology*, 91, 1491–1500.
- Dexter, F., & Traub, R. D. (2002). How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia & Analgesia*, 94, 933–942.

- Dexter, F., Traub, R. D., & Macario, A. (2003). How to release allocated operating room time to increase efficiency: Predicting which surgical service will have the most under-utilized operating room time. *Anesthesia & Analgesia*, 96, 507–512.
- Fei, H., Chu, C., Meskens, N., & Artiba, A. (2008). Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics*, 112, 96–108.
- Fei, H., Meskens, N., & Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58, 221–230.
- Fischetti, M., & Lodi, A. (2003). Local branching. *Mathematical Programming*, 98, 23–47.
- Ghazalbash, S., Sepehri, M. M., Shadpour, P., & Atighehchian, A. (2012). Operating room scheduling in teaching hospitals. *Advances in Operations Research*, 2012, 548493.
- Guerriero, F., & Guido, R. (2011). Operational research in the management of the operating theatre: A survey. *Health Care Management Science*, 14, 89–114.
- Guinet, A., & Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, 85, 69–81.
- Hans, E. W., Wullink, G., van Houdenhoven, M., & Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, 185, 1038–1050.
- Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research*, 175, 367–407.
- Jeang, A., & Chiang, A.-J. (2012). Economic and quality scheduling for effective utilization of operating rooms. *Journal of Medical Systems*, 36, 1205–1222.
- Jebali, A., Hadj Alouane, A. B., & Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99, 52–62.
- Marcon, E., Kharraja, S., & Simonnet, G. (2003). The operating theatre planning by the follow-up of the risk of no realization. *International Journal of Production Economics*, 85, 83–90.
- Marques, I., Captivo, M. E., & Pato, M. V. (2012). An integer programming approach to elective surgery scheduling. *OR Spectrum*, 34, 407–427.
- May, J. H., Spangler, W. E., Strum, D. P., & Vargas, L. G. (2011). The surgical scheduling problem: Current research and future opportunities. *Production and Operations Management*, 20, 392–405.
- Meskens, N., Duvivier, D., & Hanset, A. (2013). Multi-objective operating room scheduling considering desiderata of the surgical team. *Decision Support Systems*, 55, 650–659.
- Ogulata, S. N., & Erol, R. (2003). A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals. *Journal of Medical Systems*, 27, 259–270.
- Ozkarahan, I. (2000). Allocation of surgeries to operating rooms by goal programming. *Journal of Medical Systems*, 24, 339–378.
- Pham, D.-N., & Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, 185, 1011–1025.
- Rais, A., & Viana, A. (2011). Operations research in healthcare: A survey. *International Transactions in Operational Research*, 18, 1–31.
- Riise, A., & Burke, E. K. (2011). Local search for the surgery admission planning problem. *Journal of Heuristics*, 17, 389–414.
- Roland, B., di Martinelly, C., Riane, F., & Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering*, 58, 212–220.
- Santibáñez, P., Begen, M., & Atkins, D. (2007). Surgical block scheduling in a system of hospitals: An application to resource and wait list management in a British Columbia health authority. *Health Care Management Science*, 10, 269–282.
- Testi, A., & Tanfani, E. (2009). Tactical and operational decisions for operating room planning: Efficiency and welfare implications. *Health Care Management Science*, 12, 363–373.
- Testi, A., Tanfani, E., & Torre, G. (2007). A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10, 163–172.
- van Houdenhoven, M., van Oostrum, J. M., Hans, E. W., Wullink, G., & Kazemier, G. (2007). Improving operating room efficiency by applying bin-packing and portfolio techniques to surgical case scheduling. *Anesthesia & Analgesia*, 105, 707–714.
- Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., & Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224, 583–591.
- Wolsey, L. A. (1998). *Integer programming*. John Wiley and Sons.