

A decision support system for cyclic master surgery scheduling with multiple objectives

Jeroen Beliën · Erik Demeulemeester · Brecht Cardoen

Received: 12 March 2007 / Accepted: 16 July 2008 / Published online: 15 August 2008
© Springer Science+Business Media, LLC 2008

Abstract This paper presents a decision support system for cyclic master surgery scheduling and describes the results of an extensive case study applied in a medium-sized Belgian hospital. Three objectives are taken into account when building the master surgery schedule. First of all, the resulting bed occupancy at the hospitalization units should be leveled as much as possible. Second, a particular operating room is best allocated exclusively to one group of surgeons having the same speciality; i.e., operating rooms should be shared as little as possible between different surgeon groups. Third, the master surgery schedule is preferred to be as simple and repetitive as possible, with few changes from week to week. The system relies on mixed integer programming techniques involving the solution of multi-objective linear and quadratic optimization problems, and on a simulated annealing metaheuristic.

Keywords Master surgery scheduling · Decision support system · Mixed integer programming · Simulated annealing · Case study

1 Introduction

Due to the ageing of society and the continuously growing demands, health care is becoming very expensive. According to the 2005 report of RIZIV, the Belgian national expenses for health care amounted to 15.38 billion Euro in 2003. Five years earlier, in 1998, we spent no more than 11.29 billion Euro. In other words, the total health care expenses have increased by 36% in only five years. The annual figures indicate an average growth of 6.3% per year with a strong acceleration of 8.2% in 2003 (RIVIZ 2005). A possible way to keep the expenses at an acceptable level is to introduce more responsibility into the system. Principally, the Belgian health care system is free at the point of delivery, and therefore neither the patients nor the care providers directly feel the real cost-price of health care. The problem of health care finance is not that the incomes are too low, but mainly that the expenses grow too fast. Policy makers and health care providers must determine how to provide the most effective health care to citizens using the limited resources that are available. Therefore, they need effective methods for planning, prioritization, and decision making.

A critical resource in each hospital is the operating room. As pointed out by Litvak and Long (2000), the operating room can be seen as the engine of the hospital. Indeed, the activities inside the operating room have a dramatic impact on many other activities within hospitals. For instance, operated patients are expected to recover during a number of days, called the length of stay (LOS), during which they oc-

J. Beliën as postdoctoral researcher supported by Fonds Wetenschappelijk Onderzoek—Vlaanderen.

J. Beliën (✉)
Research Center for Modelling and Simulation, Campus Economische Hogeschool, Hogeschool-Universiteit Brussel (HUB), Stormstraat 2, 1000 Brussels, Belgium
e-mail: jeroen.belien@hubrussel.be

J. Beliën · E. Demeulemeester · B. Cardoen
Department of Decision Sciences and Information Management,
Faculty of Business and Economics, Katholieke Universiteit
Leuven, Naamsestraat 69, 3000 Leuven, Belgium

E. Demeulemeester
e-mail: erik.demeulemeester@econ.kuleuven.be

B. Cardoen
e-mail: brecht.cardoen@econ.kuleuven.be

cupy a bed in a recovery department, also called a ward or a hospitalization unit.

This paper presents a decision support system for assisting in the development of master surgery schedules. The master surgery schedule is a cyclic schedule that defines the number and type of operating rooms available, the hours that rooms will be open, and the surgeons who are to be given priority for the operating room time (Blake et al. 2002). A cyclic schedule is a schedule that is repeated after a predetermined time, called the cycle time. The cycle time is usually one week. A new master schedule is created whenever the total amount of operating room time changes. We define a block as the amount of time during which operating room time in a specific room can be allocated to a specific surgeon.

The system aims at three objectives. A first important objective is the visualization and *optimization* of the resulting bed occupancy. An *optimal* resulting bed occupancy, in this respect, is one that is as leveled as possible. To this purpose, the model presented in Beliën and Demeulemeester (2007) is adopted and embedded in the decision support system. Second, surgeons prefer to share an operating room with their direct colleagues, i.e., with surgeons belonging to the same surgeon group. We define a surgeon group as all surgeons having the same specialty, for instance, ophthalmology or oncology. Finally, in the case that the cycle time is two or more weeks, the schedule should be as repetitive as possible within the individual weeks of the cycle. In other words, the changes in the schedule from week to week should be minimized.

Although not necessarily, the three objectives often conflict with each other, meaning that building an optimal schedule with respect to one objective goes at the cost of the two other objectives. This requires a multi-objective approach in which the performance with respect to the three objectives is quantified in order to measure the schedule quality. To this purpose, our model has a weighted penalty function in which the weights represent the importance of each objective. The relative magnitudes of these weights are of course a subjective matter and can be set (and adapted) by the human scheduler.

The system presented in this paper only takes into account elective cases. In contrast to non-elective (emergency) cases, elective cases are surgical interventions that are planned beforehand. Obviously, non-elective cases contribute dramatically to the huge amount of variability in the bed occupancy, however, an important part of the variance can be controlled by applying well-thought-out scheduling policies to the elective cases. More specifically, if the bed occupancy resulting from the elective cases is nicely leveled, there is at any time instance room left to absorb unexpected peaks in bed requirements from non-elective cases.

2 Literature review

This literature review focuses on work in which operations research (OR) and artificial intelligence (AI) techniques have been applied in real-life health care settings. Despite the rich literature on effective methods and efficient algorithms for health care scheduling problems, relatively few papers present results for real-life data. Case studies where the algorithms are implemented in some kind of software system and applied in practice are even harder to find. An early bibliographic overview that focuses on OR health care applications can be found in Fries (1976). Wiers (1997) gives a more general review on the applicability of scheduling techniques in practice that also includes some interesting health care applications. A more recent collection of OR applications in health care is provided by Brandeau et al. (2004). Besides operating room scheduling, successful applications of OR/MS techniques in health care include benchmarking using DEA (e.g., Coppola et al. 2003), location of health care facilities (e.g., Adenso-Díaz and Rodríguez 1997; Brotcorne et al. 2003), organ allocation (e.g., Pritsker 1998), disease control and vaccination (e.g., Sewell and Jacobson 2003), radiotherapy treatment planning (e.g., Romeijn et al. 2006), and many others.

A number of interesting case studies and applications on health care scheduling deal with the development of nurse rosters. A literature review on nurse rostering including an interesting classification with respect to the applicability of the approach has been provided by Burke et al. (2004). Kellogg and Walczak (2007) present an exploratory research study in which they examine the models that academics have produced and the models that have been actually used in practice. According to their study only 30% of the models presented in the literature have been implemented. One remarkable result was that the research-application gap is much larger in the US (more publications, less implementations) than in Europe. Aickelin and Dowsland (2000) apply a genetic algorithm approach on real life data from wards of up to 30 nurses in a major UK hospital. More recently, Aickelin and White (2004) experiment with Bayesian optimization and classifier techniques to similar rostering problems. Burke et al. (1999, 2004) describe the successful application of a hybrid tabu search metaheuristic. Their method has been implemented in software that has been used to create nurse rosters in over forty Belgian hospitals. Further research led to the development of variable neighborhood search techniques that also have been applied on highly constrained real world nurse rostering data (Burke et al. 2003). This approach was hybridized with heuristic ordering (Burke et al. 2008) to produce a methodology that significantly outperforms a commercially implemented genetic algorithm approach on real-life data.

In the hierarchical framework for hospital production and control by Vissers et al. (2001) master surgery scheduling

could be positioned on the tactical level, somewhere between the Resource Planning & Control level and the Patient Group Planning & Control level. An interesting paper that also concerns the development of master surgery schedules has been written by Blake et al. (2002). They propose an integer programming model that minimizes the weighted average undersupply of operating room hours, that is allocating to each surgical group a number of operating room hours as close as possible to its target operating room hours (see also Blake and Donald 2002). The master surgery schedule is preferred to be as simple and repetitive as possible which entails as few changes as possible from week to week. Blake et al. (2002) use a two-step approach that successively deals with both objectives. In the first step, they assume a short cycle time (one week) and use integer programming to find the cyclic schedule with minimal undersupply of target operating room hours. In the second step, a post-improvement heuristic is run that tries to further improve this objective by introducing some changes in the schedule from week to week. The model has been successfully applied in a large Canadian hospital.

Van Oostrum et al. (2006) model the problem of constructing master surgical schedules as a Mathematical Program containing probabilistic constraints. Due to computational intractability the authors propose a two-phase column generation approach that first maximizes the OR utilization and subsequently levels the demand for succeeding hospital departments, such as wards and intensive care units. The approach was tested using data from the Erasmus Medical Center, Rotterdam, the Netherlands. Hans et al. (2008) propose several constructive and local search heuristics for the robust surgery loading problem. The objective is to assign the surgeries by the specialties in such a way, that the risk of working in overtime is minimized, no surgeries are canceled, and at the same time the operating room capacity utilization can be improved. The approach has also been developed in collaboration with the Erasmus Medical Center and tested on historical data.

Santibanez et al. (2007) present a system-wide optimization model for block scheduling that enables managers to explore trade-offs between operating room availability, booking privileges by surgeons, bed capacity and waiting lists for patients.

Kusters and Groot (1996) present a decision support system for admission planning based on a series of resource availability models. The main support of the computer model is in predicting the effects of decisions on the availability of scarce resources like beds, operating theater facilities, and nursing staff. It enables the human decision maker to choose the *right* patients from the waiting list in order to better balance the daily demand of hospital facilities against the availability of these resources. The paper

contains an empirical study validating the statistical models and reports on the problems encountered when implementing the system in practice. The results obtained with the model show that such an approach based on statistical data provides sufficiently accurate results to be useful.

Lapierre et al. (1999) propose guidelines in order to set up a measurement system to improve on-time performance of first health care services of the day. Their main finding is that, even if surgeons are the main cause of delay, efforts are likely better aimed at improving hospital workers' on-time performance than on improving surgeons' on-time performance. If the on-time performance of other hospital departments is improved, then anesthesiologists will improve their performance and surgeons will eventually improve theirs, with a lag. These findings are illustrated by analyzing one hospital organization's case.

Everett (2002) describes the design of a simulation model to provide decision support for the management and scheduling of patients waiting for elective surgery in the public hospital system. The model can also be useful in monitoring the performance of the system and exploring the relative effectiveness of alternative policies in coping with historical or statistically generated patient load.

Hsu et al. (2003) present a deterministic approach to schedule patients in an ambulatory surgical center such that the number of postanesthesia care unit nurses at the center is minimized. Their heuristic has been tested on a set of real data from a university hospital's ambulatory surgical center.

Marcon and Dexter (2006) use discrete event simulation to analyze the impact of seven different sequencing rules on over-utilized operating room time, delays in phase I post anesthesia care unit (PACU) admission, the PACU completion time, and PACU nurse staffing.

The ability to cope with uncertainty is considered to be an essential part of modern health care scheduling. Kim et al. (2000) describe a flexible bed allocation scheme that reserves one or more beds for the exclusive use of elective-surgery patients to enhance the operations of the intensive care unit. Kim and Horowitz (2002) elaborate on this work and show through a simulation model that the combination of this flexible bed allocation scheme and a quota system for elective surgery greatly reduces the number of canceled surgeries.

The operating room scheduling problems described in the literature often contain several, sometimes conflicting objectives. Cardoen et al. (2006) present a multi-objective optimization model for scheduling individual cases in the surgical day-care center of a large Belgian hospital. The authors introduce a so-called room for improvement measure to trade-off between six different objectives (children as early as possible, prioritized patients as early as possible, patients having a large travel distance as much as possible before a particular hour, minimizing overtime in recovery and leveling bed occupancy in both recovery phase 1 and

recovery phase 2). The model is solved using integer programming and branch-and-bound algorithms. In a more recent work, Cardoen et al. (2007) also present a branch-and-price approach to deal with this multi-objective optimization problem. Ogulata and Erol (2003) develop a set of hierarchical multiple criteria mathematical programming models to generate weekly operating room schedules. The objectives considered in this study are maximum utilization of operating room capacity, balanced distribution of operations among surgeon groups and minimization of patient waiting times. Pérez et al. (2005) propose a so-called possibilistic linear multi-objective programming model as an information system in order to analyze the internal coherency of the different goals expressed by Spanish Health Service in relation to the maximum stay on a waiting list. Using a Multicriteria Decision technique they intend to assign and manage, in an optimal way, the real performance of the surgical services of a medium-sized hospital in Spain.

The remainder of this paper is structured as follows. Section 3 describes the mathematical model on which the decision support system is built. Section 4 presents the solution procedures that were developed starting from this mathematical model. Section 5 gives some more information on the case study. Section 6 contains a presentation of the graphical user interface that was built on top of the algorithms to visualize the operation and performance of the system. Section 7 discusses the results obtained by applying the different algorithms while Sect. 8 draws conclusions and lists some topics for future research.

3 Mathematical model

As already mentioned in the introduction, our decision support model aims at three objectives. With respect to the first objective, leveling the resulting bed occupancy, the system uses the model presented in Beliën and Demeulemeester (2007). This model assumes multinomial distribution functions for both the number of patients per operating room block and the length of stay of each operated patient. Using this information, the model is capable of constructing a master surgery schedule with leveled resulting bed occupancy. In addition, performance measures such as the daily expected bed occupancy, the variance on this occupancy, the expected bed shortage, and the probability of a shortage on each day can be calculated. Leveling is achieved using mixed integer programming (linear as well as quadratic) and a simulated annealing metaheuristic.

The theoretic model described in Beliën and Demeulemeester (2007) has been slightly modified in order to deal with some practical issues. First, in the theoretic model, all patients are assumed to recover in one hospitalization unit. In practice, of course, a hospital has more than one hospitalization unit at which predetermined groups of patients can

recover from surgery. This is a fairly straightforward extension, that can be achieved by considering different probability distributions for each surgeon–hospitalization-unit combination. A second limit of the theoretic model is the fact that block sizes are assumed to be fixed. In practice, however, the time for which an operating room can be allocated to a particular surgeon is not necessarily fixed. For instance, one surgeon can be allocated to a block of 4 hours while another surgeon gets a block of 6 hours. This extension has some consequences for the simulated annealing approach (described in Beliën and Demeulemeester 2007) for which we have added a corresponding neighborhood move. Finally, the system described in this paper allows for the allocation of operating room time to individual surgeons instead of surgeon groups. This extension requires the addition of an extra constraint that prevents individual surgeons from being scheduled in different rooms at the same time.

We now state the mixed integer programming models that are used to develop the master surgery schedules. We start with the model that only aims at a leveled resulting bed occupancy. Starting from this model we present the modifications needed in order to take into account the other two objectives. The notation used in these models is as follows:

The indices and sets are:

i, j, d, d_1, d_2 : days in the cycle.

s : surgeons.

r : rooms.

h : hospitalization units.

D : set of days in the cycle.

A : set of days on which surgery takes place (=active days) (usually all days, except for the weekends).

S : set of surgeons.

R : set of rooms.

H : set of hospitalization units.

The decision variable is:

$$x_{isr} = \begin{cases} 1, & \text{if surgeon } s \text{ obtains an operating room block} \\ & \text{in room } r \text{ on day } i; \\ 0, & \text{otherwise.} \end{cases}$$

The help variables are:

$mean_{hi}$ = the mean bed occupancy in hospitalization unit h on day i .

\overline{mean}_h = the peak mean bed occupancy in hospitalization unit h over all days in the cycle.

var_{hi} = the variance of the bed occupancy in hospitalization unit h on day i .

\overline{var}_h = the peak variance of the bed occupancy in hospitalization unit h over all days in the cycle.

The data parameters are:

req_s : the number of blocks required by surgeon s .

cap_{ir} : the total capacity (in hours) of room r on day i .

- dur_s : the duration (in hours) of a block allocated to surgeon s .
- mb_{sijh} : the contribution of allocating a block to surgeon s on day j to the mean bed occupancy on day i in hospitalization unit h .
- vb_{sijh} : the contribution of allocating a block to surgeon s on day j to the variance of the bed occupancy on day i in hospitalization unit h .
- w_{mean_h} : the relative importance of leveling the mean occupancy in hospitalization unit h .
- w_{var_h} : the relative importance of leveling the variance of the occupancy in hospitalization unit h .

We start with the linear min–max model that aims at minimizing the weighted peaks in the expected bed occupancy and/or the variance in bed occupancy:

$$\text{Minimize } \sum_{h \in H} (w_{mean_h} \overline{mean}_h + w_{var_h} \overline{var}_h) \tag{1}$$

$$\text{subject to: } \sum_{i \in A} \sum_{r \in R} x_{isr} = req_s \quad \forall s \in S; \tag{2}$$

$$\sum_{s \in S} dur_s x_{isr} \leq cap_{ir} \quad \forall i \in A \text{ and } \forall r \in R; \tag{3}$$

$$\sum_{r \in R} x_{isr} \leq 1 \quad \forall i \in A \text{ and } \forall s \in S; \tag{4}$$

$$\begin{aligned} mean_{hi} &= \sum_{s \in S} \sum_{j \in A} \sum_{r \in R} mb_{sijh} x_{jsr} \\ &\forall h \in H \text{ and } \forall i \in D; \end{aligned} \tag{5}$$

$$\begin{aligned} var_{hi} &= \sum_{s \in S} \sum_{j \in A} \sum_{r \in R} vb_{sijh} x_{jsr} \\ &\forall h \in H \text{ and } \forall i \in D; \end{aligned} \tag{6}$$

$$mean_{hi} \leq \overline{mean}_h \quad \forall h \in H \text{ and } \forall i \in D; \tag{7}$$

$$var_{hi} \leq \overline{var}_h \quad \forall h \in H \text{ and } \forall i \in D; \tag{8}$$

$$mean_{hi}, var_{hi} \geq 0 \quad \forall h \in H \text{ and } \forall i \in D; \tag{9}$$

$$\overline{mean}_h, \overline{var}_h \geq 0 \quad \forall h \in H; \tag{10}$$

$$x_{isr} \in \{0, 1\} \quad \forall i \in A, \forall s \in S \text{ and } \forall r \in R. \tag{11}$$

The objective function (1) minimizes the weighted sum of peaks in the bed occupancy and variance over all hospitalization units. Constraint set (2) ensures that every surgeon obtains the right number of blocks. The number of hours preserved for each surgeon is decided on a higher level and is a consequence of the hospital’s strategic decision for which ailments capacity will be preserved (case mix planning). Constraint set (3) makes sure that the total operating time assigned on each day in each room does not exceed the available operating room time. Constraint set (4) prevents a surgeon to be scheduled simultaneously in two different rooms. Constraint set (5) calculates the expected bed

occupancy in each hospitalization unit as a function of the operating room schedule, while constraint set (6) calculates the variance on this occupancy. The reader is referred to Beliën and Demeulemeester (2007) for more details on how the values mb_{sijh} and vb_{sijh} are calculated. Constraint set (7) provides the link with the objective function by imposing for each hospitalization unit that the expected bed occupancy on each day cannot exceed the peak expected bed occupancy. Constraint set (8) does the same for the variance. Finally, constraint sets (9) and (10) define the mean and variance to be nonnegative and constraint set (11) defines x_{isr} as a binary decision variable.

The quadratic MIP model is identical to the linear min–max MIP model except for the objective function that represents an explicit (weighted) leveling of the mean and variance of the occupancy in the different hospitalization units:

$$\text{Minimize } \sum_{h \in H} \left(w_{mean_h} \sum_{i \in A} mean_{hi}^2 + w_{var_h} \sum_{i \in A} var_{hi}^2 \right). \tag{12}$$

Obviously, constraints (7), (8), and (10) can now be removed as these are no longer required.

In order to take into account the second objective, that is, try to concentrate surgeons that belong to the same surgeon group as much as possible in the same room, we need to add an extra penalty term in the objective function and two extra constraints *triggering* this penalty. Define G as the set of all surgeon groups g . In the ideal case all surgeons of one group are scheduled in one and the same operating room. Let RO_g be an integer decision variable that represents the number of extra operating rooms allocated to surgeon group g . For instance, if a group is allocated to two different rooms, then RO_g equals 1, if a group is allocated to three different rooms, then RO_g equals 2, etc. Now, we can add the following term to the objective functions (1) and (12):

$$\sum_{g \in G} w_{room_g} RO_g \tag{13}$$

with w_{room_g} the relative importance (compared to the other objectives) of the room concentrating objective with respect to group g . To impose that RO_g obtains the correct value, two constraints must be added to the model. Let b_{gr} be a binary decision variable that equals 1 if at least one surgeon of surgeon group $g \in G$ obtains an operating room block in room r . Let S_g be the set containing all surgeons s that belong to group g and let $|S_g|$ and $|A|$ be the respective numbers of elements in the sets S_g and A . The extra constraints are:

$$\sum_{i \in A} \sum_{s \in S_g} x_{isr} \leq |A| |S_g| b_{gr} \quad \forall g \in G \text{ and } \forall r \in R; \tag{14}$$

$$\sum_{r \in R} b_{gr} \leq 1 + RO_g \quad \forall g \in G. \tag{15}$$

In order to take into account the third objective, that is, making the schedules as repetitive as possible from week to week if the cycle time is longer than one week, we need some extra decision variables. Since the cycle time in the studied hospital is either one or two weeks, we will only consider this case. Let O_{isr} be a binary decision variable that equals 1 if surgeon s obtains an operating room block at day i in room r only during the *odd* weeks and 0 otherwise, while E_{isr} represents the same for the *even* weeks. Whether the cycle time is one week or two weeks, the set of active days A remains the same containing 5 days (from Monday to Friday). The set of days D , however, contains all days in the cycle time (hence, 14 days in case of a cycle time of two weeks). Note that if x_{isr} is set to 1, then surgeon s obtains a block on day i in the odd weeks and on day i in the even weeks, whereas, if O_{isr} (E_{isr}) is set to 1, this surgeon only gets a block in the odd (even) weeks. The following term is added to the objective functions (1) and (12):

$$\sum_{i \in A} \sum_{s \in S} \sum_{r \in R} w_{oddeven_s} (O_{isr} + E_{isr}) \tag{16}$$

with $w_{oddeven_s}$ the relative importance of a repetitive schedule for surgeon s . The constraints (2)–(6) are modified as follows:

$$\sum_{i \in A} \sum_{r \in R} (2x_{isr} + O_{isr} + E_{isr}) = 2req_s \quad \forall s \in S; \tag{17}$$

$$\sum_{s \in S} dur_s (x_{isr} + O_{isr} + E_{isr}) \leq cap_{ir} \quad \forall i \in A \text{ and } \forall r \in R; \tag{18}$$

$$\sum_{r \in R} (x_{isr} + O_{isr} + E_{isr}) \leq 1 \quad \forall i \in A \text{ and } \forall s \in S; \tag{19}$$

$$mean_{hi} = \sum_{s \in S} \sum_{j \in A} \sum_{r \in R} (mb_{sijh} x_{jsr} + omb_{sijh} O_{jsr} + emb_{sijh} E_{jsr}) \quad \forall h \in H \text{ and } \forall i \in D; \tag{20}$$

$$var_{hi} = \sum_{s \in S} \sum_{j \in A} \sum_{r \in R} (vb_{sijh} x_{jsr} + ovb_{sijh} O_{jsr} + evb_{sijh} E_{jsr}) \quad \forall h \in H \text{ and } \forall i \in D, \tag{21}$$

with omb_{sijh} (emb_{sijh}) the contribution to the mean bed occupancy on day i in hospitalization unit h of allocating a block to surgeon s on day j of the odd (even) weeks and ovb_{sijh} (evb_{sijh}) the contributions to the respective bed occupancy variances.

4 Solution procedures

To solve the model outlined above, the application can call the CPLEX MIP solver (ILOG 2002) for linear and quadratic optimization. Unfortunately, incorporating the three objectives simultaneously leads to such a large mixed integer program that the problem becomes computationally intractable (see further). As the system is designed to support the decision making process, it is crucial that users get quick answers to what-if questions. Therefore, alternative heuristic solution procedures have been developed. A first heuristic is a modified version of the simulated annealing procedure described in Beliën and Demeulemeester (2007) that also evaluates the second and third objective.

Simulated annealing (SA) is a technique to find a good solution to an optimization problem by trying random variations of the current solution. A worse variation is accepted as the new solution with a probability that decreases as the computation proceeds. The slower the cooling schedule, or rate of decrease, the more likely the algorithm is to find an optimal or near-optimal solution. The algorithm is based upon that of Metropolis et al. (1958), which was originally proposed as a means of finding the equilibrium configuration of a collection of atoms at a given temperature. The connection between this algorithm and mathematical minimization was first noted by Pincus (1970), but it was Kirkpatrick et al. (1983) who proposed that it forms the basis of a search technique for combinatorial (and other) problems. Good theoretic expositions on simulated annealing can also be found in Huang et al. (1986) and Van Laarhoven and Aarts (1988).

We implemented a basic SA implementation in which the neighborhood is defined as all solutions which could be obtained after swapping two surgery blocks from the current solution. The first block is chosen randomly. The second block is the first encountered block for which a swap results in an improvement (decrease) of the objective value. If no such block can be found, the block leading to the smallest increase is chosen. In order to deal with variable block sizes we added the restriction that surgeon block allocations can only be swapped if they have the same duration. To be able to explore a larger neighborhood, also swaps between sets of surgeon block allocations could be performed, provided that these sets have the same total duration, making sure that available operating room time is not exceeded in one of the rooms after a swap. In order to decide whether or not to accept a worse solution, a standard Boltzman function is evaluated. Let T denote the temperature and Δf the decrease in objective function. For swaps with negative Δf the probability of acceptance is given by $e^{\frac{\Delta f}{T}}$. Of course, the best found schedule is saved. For the test results reported below, the temperature T was initialized at 500 and decreased with 5% after each 10 iterations.

A second attempt to obtain good solutions in small computation times resulted in the development of hierarchical

goal programming models. These models first concentrate on one objective after which the next objective is optimized. A first important simplification concerns the cycle time. Even when changes from the first week to the second are, to a small extent, allowed, and hence the cycle time is in theory two weeks, we will start with developing a one-week schedule. This schedule is then copied to the second week, after which some modifications can be made in a post-optimization step (see further).

Observation 1 The resulting bed occupancy remains unchanged when a surgeon block is shifted to a different operating room on the same day. Indeed, only the time of surgery determines when patients enter the hospital and when they are expected to occupy a bed; the operating room where surgery takes place has no impact on the bed occupancy.

This first observation leads to a first hierarchical goal programming approach consisting of two goal programming models that are solved successively (HIER-GOAL-1):

1. Solve the MIP that only aims at leveling the bed occupancy, that is, solve the linear MIP (1)–(11) or the quadratic MIP with objective function (12).
2. Solve a MIP that only aims at the room objective, in which surgeon blocks cannot be shifted to different days but only to different rooms.

The decision variables in the second MIP are selected based on the solution of the first MIP. If $x_{i,SR}$ equals one in the solution of the first MIP, then all $x_{i',SR}$ with $i' \neq i$ will not be present in the second MIP, ensuring that surgeons are not shifted to another day. This two-step approach makes the model much easier to solve.

Observation 2 The number of different operating rooms used by a surgeon group remains unchanged when a surgeon block is shifted to a different day in the same operating room.

This second observation leads to a second hierarchical goal programming approach consisting of two goal programming models that are solved successively (HIER-GOAL-2):

1. Solve the MIP that only aims at the room objective, that is, solve the linear MIP with objective (13) and constraints (2)–(4).
2. Solve a MIP that only aims at the leveling objective, in which surgeon blocks cannot be shifted to different rooms but only to different days.

Similarly, the decision variables in the second MIP are selected based on the solution of the first MIP. If $x_{i,SR}$ equals one in the solution of the first MIP, then all $x_{i,SR'}$ with $r' \neq r$

will not be present in the second MIP, ensuring that surgeons are not shifted to another room. This two-step approach makes the model again much easier to solve.

Improvement step As already said, even when the cycle time is two weeks, both hierarchical approaches outlined above aim at a one-week schedule. Afterwards the developed schedule is copied to the second week. Since the schedule of the first week is exactly the same as the schedule of the second week, we incur no penalty cost for the difference between odd and even weeks (no odd–even penalty cost). Although leading to an odd–even penalty cost, some modifications to the schedule could be beneficial with respect to the resulting bed occupancy. Therefore, a post-optimization procedure can be applied that searches for the surgeon swap that results in the best improvement with respect to a chosen objective (leveling, minimizing expected shortage, probability of shortage). The user can control the odd–even penalty cost by limiting the maximal number of swaps in the post-optimization procedure.

An extra asset is that the different procedures of the hierarchical goal programming models can also be used separately in order to solve what-if questions. One can start from a given schedule, for instance the schedule that is currently in practice, and execute the second MIP optimization for which either the days or the rooms of surgery are fixed. If, for instance, the room allocation is fixed, one can find an answer to the question: To what extent the bed occupancy can be leveled without changing the room allocations? Alternatively, if the days are fixed, one can obtain an answer to the question: To what extent the room allocations can be rearranged such that the same specialties share the same operating room without changing the expected bed occupancy?

5 Case study

The case study presented entails the Virga Jesse Hospital, situated in Hasselt, Belgium. Virga Jesse's central operating room complex consists of 9 rooms in which a total of 46 surgeons have been assigned operating room time. These surgeons are classified into 14 different surgeon groups with respect to the specialism. Each operating room is open from Monday to Friday for 8.5 hours. Up to now, no elective surgery takes place during the weekends. The hospital has 25 different hospitalization units, however, only 10 units have served more than 100 elective cases in 2004. The models applied in this study involve the development of a (cyclic) master surgery schedule with leveled bed occupancy in these 10 major hospitalization units.

For the leveling objective we need as input for each surgeon–hospitalization-unit combination the probability distributions of the number of patients per block and the

Table 1 Snapshot of the input file containing detailed information on all surgical interventions in 2004

OR_NR	SURGEON	ROOM	HOSP. UNIT	DATE_IN	DATE_OUT
23005838	PUTE	Operatiezaal 04	3200	2/01/2004 8:00	2/01/2004 17:00
23116828	DTRG	Operatiezaal 09	3200	2/01/2004 8:00	2/01/2004 17:00
23408780	VDVG	Operatiezaal 03	2150	2/01/2004 8:00	5/01/2004 15:00
23409553	BOES	Operatiezaal 05	2160	2/01/2004 8:00	5/01/2004 15:19
23382108	PUTE	Operatiezaal 04	3200	2/01/2004 8:05	2/01/2004 17:00
23383582	LENH	Operatiezaal 08	3200	2/01/2004 8:05	2/01/2004 17:00
23409151	PUTE	Operatiezaal 04	3200	2/01/2004 8:10	2/01/2004 17:00
23408550	PUTE	Operatiezaal 04	3200	2/01/2004 8:15	2/01/2004 17:00
23382105	PUTE	Operatiezaal 04	3200	2/01/2004 8:20	2/01/2004 17:00
23408576	VDKJ	Operatiezaal 06	3200	2/01/2004 8:20	2/01/2004 17:00
...					

LOS for each operated patient. The theoretical models assume multinomial distributions, often referred to as empirical discrete probability distributions. These general probability distributions can easily be constructed from a database containing the detailed information on all surgical interventions that have been performed in a reasonably long time period (e.g., one year). Table 1 contains a snapshot of the (relevant) fields of the input file.

Only elective cases are taken into account. The reason why the non-elective (emergency) cases are not retained is twofold. First, the occurrence as well as the recovery period of non-elective, emergency cases is, by definition, highly unpredictable, and hence it would make little or no sense to fit a probability distribution to them. Second, since non-elective (emergency) cases occur, by definition, unexpectedly, this surgery often takes place on days during which the surgeon has no block allocated. Hence, taking them into account would lead to a biased distribution for the number of patients per operating room block.

Table 2 shows an example of the derived probability distributions, for one particular surgeon. It must be clear at this point that the LOS distributions are specific for each surgeon–hospitalization-unit combination. This is a very realistic basic assumption, since the patient recovery time is usually strongly related to this unique combination as patients operated by the same surgeon and recovering in the same hospitalization unit often suffer from similar ailments. Of course, surgeons can perform different surgical treatments in one block, but the proportions of these treatments are often reasonably constant.

Our system can derive these probability distributions automatically. Before deriving the number of patients and LOS distributions, the surgeons and the existing schedule have to be read in manually. The existing schedule is needed to determine whether the case is elective or non-elective. If the intervention takes place on a day during which a block is

preserved for the surgeon, it is considered to be an elective case. Otherwise, it is considered to be a non-elective case.

A problem arises when a surgeon is assigned to blocks of different durations in the current schedule. In this case, the distribution functions are derived for each different block duration. For instance, consider a surgeon who has a block of 8.5 hours on Monday and a block of 4 hours on Tuesday. In our approach, distributions will be derived for the Monday block as well as for the Tuesday block. This implies that hours cannot be exchanged between blocks. Only shifting of total blocks will be allowed. The choice for this approach is justified as follows. First of all, a *block* is probably the best unit for deriving the probability distributions. A smaller unit (e.g., an hour) is in our view less effective to fit the real distributions. Second, a block that extends twice as long as another block, assigned to the same surgeon, does not necessarily include twice the number of patients. Hence, deriving one number of patients distribution from two blocks with different duration would lead to a biased probability distribution. Third, limiting the model to shifting entire blocks entails some interesting computational features. Also considering the exchange of hours between the different blocks of a surgeon would dramatically complicate the problem. Fourth, the graphical user interface is kept extremely simple as block allocations and swaps can easily be done by dragging and dropping. Finally, from a practical point of view, most of the surgeons have currently been allocated to blocks of the same duration, and hence relatively few extra distribution functions need to be derived.

6 Graphical user interface

In this section the graphical user interface (GUI) is presented. The application was programmed in Visual C++. NET. The GUI visualizes the surgery schedule and the resulting bed usage occupancy distributions for a given sched-

Table 2 Example of number of patient and LOS distributions for three hospitalization units for surgeon DUPA

SURGEON	HOSP. UNIT	NUMBER OF PATIENTS		LOS			
		NUMBER OF PATIENTS	PROB.	NUMBER OF DAYS	PROB.		
DUPA	2160	0	0.20	3	0.20		
		1	0.38	4	0.02		
		2	0.34	5	0.02		
		3	0.06	6	0.03		
		4	0.02	7	0.28		
				8	0.21		
				9	0.21		
				10	0.03		
				12	0.02		
			2601	0	0.56	4	0.03
				1	0.34	7	0.04
				2	0.06	8	0.41
	3	0.04		9	0.45		
				10	0.07		
	3200	0	0.16	1	1		
		1	0.10				
		2	0.22				
		3	0.30				
		4	0.12				
		5	0.08				
		6	0.02				

ule. Moreover, it allows the user to modify an existing schedule and to view the impact of a change in the schedule on the bed occupancy. Data like the schedule properties, the surgeon properties and the hospitalization properties can easily be read in and modified. Automation features include the deduction of the probability distributions for patient numbers and lengths of stay from a database (as described in Sect. 5) and the optimization of the schedule with respect to certain objective measures. Figure 1 shows the current master surgery schedule with resulting bed occupancy (only four hospitalization units are shown). The main window consists of two panes. In the left pane the master surgery schedule is shown. The columns in the grid represent the days in the week (in this case the cycle time is one week). The nine rows represent the nine operating rooms. The surgeons are shown above the grid in the color legend (only a subset is visible). The schedule could be built from scratch by dragging and dropping the surgeons to the cells of the grid. Of course, a room can also be allocated for a limited number of hours instead of the full 8.5 hours. Each block allocation introduces a patient flow in the system, which is reflected by an increase in the bed occupancy of one or more hospitalization units on one or more days. This is represented in the right pane. Only four hospitalization units are shown (2110, 2120, 2130, and 2140; each row corresponds to one unit).

The small T-ending bars on top of each colored occupancy box indicate the standard deviations of the bed occupancy distributions on the corresponding days at the corresponding hospitalization units.

A simulation run could be done in order to validate the theoretical basic assumptions of the model. To this purpose it can be verified whether the predicted bed occupancies (and shortages) obtained by calculation are similar to the ones obtained by simulation.

7 Results

It is difficult to objectively compare the quality of the generated schedules, as there is no once-and-for-all objective measure to make this comparison. To build a quality schedule or at least to improve the current schedule, one has to study the current practices and determine the most appropriate objective function and automation procedure. For instance, if capacity problems always occur at the same hospitalization unit, a linear or quadratic MIP procedure that focuses on this unit will probably render the best results. The visualization of the bed occupancies can of course assist in determining the appropriate model. However, it might

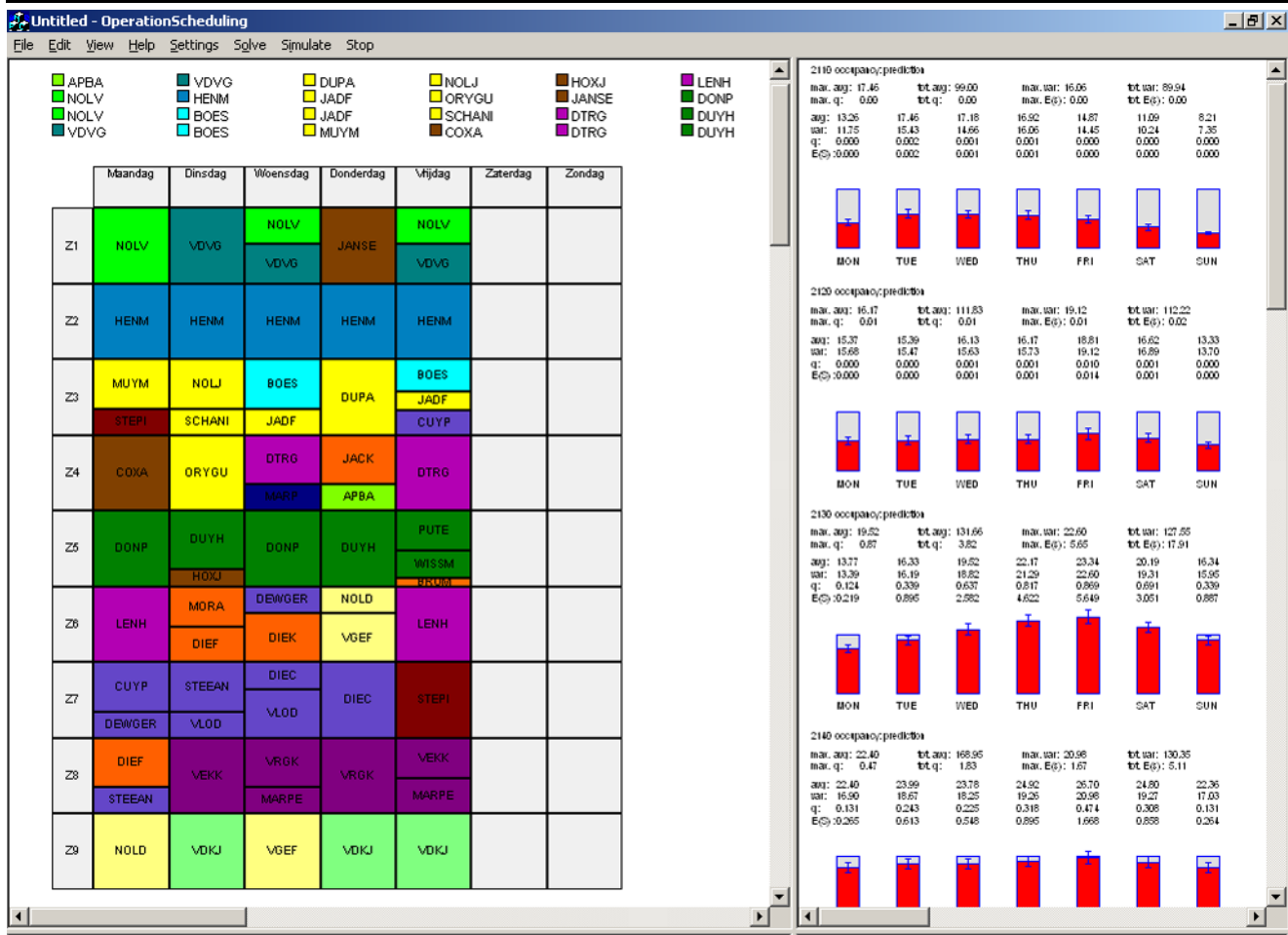


Fig. 1 Overview of the GUI

be not so easy to find the surgeon swaps that lead to a better schedule. Therefore, the automation procedures can be useful.

Rather than trying to find the overall best master surgery schedule for the Virga Jesse Hospital, which is a subjective matter after all and hence makes little sense anyway, we present the current schedule and compare it with the results of a number of different algorithm runs. Figure 1 shows the current schedule and the resulting bed occupancy in the first 4 hospitalization units. The total expected bed shortage in the current schedule is 23.49, which means that, over all hospitalization units, more than 3 beds per day are lacking in the assigned hospitalization unit and hence have to be found in another hospitalization unit.

As can be seen in Fig. 1, some problems may arise at hospitalization unit 2130 (third unit), where there is a high peak occupancy on Friday leading to a positive expected bed shortage. Also the fourth hospitalization unit (2140) suffers from large differences in the bed occupancy peaks. This asks for a scheduling procedure that simultaneously focuses on the leveling of the bed occupancy distributions in units 2130

and 2140. To this purpose, a linear MIP procedure that minimizes the weighted maximum peak of the bed occupancies in units 2130 and 2140 could be applied. It should be clear that the bed occupancy in these units is now much more leveled over the week.

The drawback of this new schedule is that many surgeons have to share an operating room with colleagues belonging to different groups. If we account for a penalty of 1 for each extra room used by a surgeon group ($w_{room_g} = 1$), we obtain a total room penalty of 34, while in the current schedule this is only 14. This means that in the current schedule the 14 surgeon groups use two rooms per group (14 extra rooms), while in the new schedule, they use more than 3 rooms per group. It would be interesting to ask the system for a schedule that takes into account both room restrictions and leveling issues. It could also be investigated to what extent the bed occupancy can be further leveled by a prolongation of the cycle time from one week to two weeks (and, hence, by allowing a limit number of differences between the odd and the even weeks).

Table 3 Results of different algorithm runs; cycle time = 1 week; comp. time = 120 seconds

Procedure	Total exp. shortage	Room penalty
Linear MIP MIN–MAX mean all units	20.77	36
Linear MIP MIN–MAX mean 2130 and 2140	21.05	37
Linear MIP MIN–MAX mean + var. 2130 and 2140	20.78	35
Quadratic MIP mean all units	20.51	33
Quadratic MIP mean 2130 and 2140	20.45	37
Quadratic MIP mean + var. 2130 and 2140	20.63	38
Simulated annealing	18.95	35

Table 4 Results of different algorithm runs taking into account room penalty; cycle time = 1 week; comp. time = 120 seconds

Procedure	Total exp. shortage	Room penalty
Linear MIP MIN–MAX mean all units	–	–
Linear MIP MIN–MAX mean 2130 and 2140	21.26	12
Linear MIP MIN–MAX mean + var. 2130 and 2140	21.05	13
Quadratic MIP mean all units	–	–
Quadratic MIP mean 2130 and 2140	–	–
Quadratic MIP mean + var. 2130 and 2140	–	–
Simulated annealing	21.91	16
HIER-GOAL-1	20.77	15
HIER-GOAL-2	20.81	6

Table 5 Results of different algorithm runs; cycle time = 2 weeks; comp. time = 120 seconds

Procedure	Total exp. shortage	Odd–even penalty
Linear MIP MIN–MAX mean all units	45.36	0
Linear MIP MIN–MAX mean 2130 and 2140	45.54	0
Linear MIP MIN–MAX mean + var. 2130 and 2140	45.05	0
Quadratic MIP mean all units	45.50	12
Quadratic MIP mean 2130 and 2140	44.42	0
Quadratic MIP mean + var. 2130 and 2140	44.18	0
Simulated annealing	46.98	0

feasible solution. A second important difference is that the simulated annealing procedure is now outperformed by the other procedures (at least the ones that find a feasible solution). This can be explained by the fact that, in contrast with the leveling objective, the room objective is a direct part of the objective function of the MIP models and, hence, simulated annealing loses its advantage of exclusively focusing on the real objective. The second hierarchical goal programming procedure performs remarkably well.

Table 5 contains again the results for the procedures of Table 3 (hence, no room penalty) but now considering a cycle time of two weeks, thus with added term (16) ($w_{oddeven_s} = 1 \forall s \in S$) in the objective function and modified constraints (17)–(21). Apparently, this extension is less complicated than the room penalty extension as all approaches again succeed in finding at least a feasible solution. Similar to the case with added room penalty, the MIP based

procedures perform better than the simulated annealing procedure. As a matter of fact, the simulated annealing procedure does not manage to improve the start solution (current schedule with two identical weeks) because the first surgeon swap results in an increase of the odd–even penalty by at least 4 (the two swapped surgeons each incur an odd–even penalty of 1 in each week). Even when this swap passes the simulated annealing temperature test, the algorithm fails to improve the solution again to one that is better than the start schedule. Only for the quadratic MIP model that focuses on all units, the found schedule differs from week 1 to week 2. This is a quite logical result since the odd–even penalty of 12 is rather small with respect to the (quadratic) remainder of the objective function (4557.24, not shown in the table).

Table 6 combines Tables 4 and 5 by considering both the room extension and the cycle time prolongation. The last two procedures are the earlier described hierarchical

Table 6 Results of different algorithm runs taking into account room constraints; cycle time = 2 weeks; comp. time = 120 seconds

Procedure	Total exp. shortage	Room penalty	Odd–even penalty
Linear MIP MIN–MAX mean all units	–	–	–
Linear MIP MIN–MAX mean 2130 and 2140	45.70	14	0
Linear MIP MIN–MAX mean + var. 2130 and 2140	45.01	12	0
Quadratic MIP mean all units	–	–	–
Quadratic MIP mean 2130 and 2140	–	–	–
Quadratic MIP mean + var. 2130 and 2140	–	–	–
Simulated annealing	46.98	15	0
HIER-GOAL-1 + improvement step	39.55	12	22
HIER-GOAL-2 + improvement step	39.76	7	18

goal programming approaches followed by an improvement step, that searches for the three most improving swaps (subsequently) with respect to the minimal total expected bed shortage objective at the cost of introducing differences between the odd and the even weeks and, hence, incurring an odd–even penalty cost. As can be seen in Table 6, these procedures manage to improve both the leveling objective and the room objective at the cost of a positive odd–even penalty cost.

Finally, Table 7 presents the solution quality, measured by the so-called gap, as a function of the computation time. For the MIP based solution procedures the gap is defined as the difference between the found solution and the lower bound, expressed as a percentage of the found solution. Remark that these gaps are calculated with respect to the objective values of the MIP models (and not with respect of the minimal total expected bed shortage objective). The reason is that the MIP optimizer provides us with a quality lower bound on the MIP objective while we lack a lower bound calculation on the real objective. For the hierarchical goal programming procedures the reported gaps apply on the solution of the second MIP. For the simulated annealing procedure the gap is defined as the difference between the found solution in the given time limit and the final solution found after 120 seconds, again expressed as a percentage of the first. Hence, the gap of the simulated annealing procedures is by definition 0 after 120 seconds of computation time. Also for the hierarchical goal programming procedures followed by an improvement step the gap was calculated in this way, since the last step is a steepest descent heuristic. Recall that ‘–’ indicates that no feasible solution was found in the given computation time. ‘N.A.’ indicates that the procedure is not applicable; e.g., the hierarchical goal programming procedures are not applicable to problems without a room penalty. Similarly, the hierarchical goal programming procedures followed by an improvement step are only applicable for problems with a 2-week cycle time and a room penalty. From Table 7 it can be concluded that the MIP models, at

least in the case of no room penalty, find near optimal solutions in the first 30 seconds which are hardly improved in the remaining computation time. On the contrary, the simulated annealing procedure keeps significantly improving the solution during the whole computation run.

8 Conclusions and future research

This paper has presented a decision support system to develop master surgery schedules. The system is built on different optimization procedures that aim at leveling the resulting bed occupancy, concentrating surgeons of the same group in the same rooms, and keeping the schedules consistent from week to week. Depending on the hospital’s situation and, in particular, on the problems it is facing, a procedure can be chosen to build a new master surgery schedule. Additionally, the system can provide managers with important insights into the opportunities or limits with respect to the master operating room schedule. The system was presented by means of a case study in a medium-sized Belgian hospital. To this purpose, the required input data, namely the distribution functions for the number of operated patients as well as for the length of stays, have been derived from the hospital’s central database containing detailed information on all surgical cases during a 1-year period.

It must be clear that the software does not provide an overall best solution. Different algorithm runs lead to different solutions and it is up to the manager to decide on the best schedule. The real power of the software lies in the visualization of the schedule and the resulting bed occupancy, the ease with which schedules can be built and the capability it provides to carry out an in-depth analysis of the existing system. Using the software, managers can find answers to questions like “what is the most leveled bed occupancy possible at hospitalization unit X?”, “which schedule simultaneously levels the bed occupancy in units X and Y?”, “to what extent surgeons can be assigned operating rooms with colleagues of the same group?”, or “could it be beneficial with

Table 7 Results of different algorithm runs: gaps for different computation times

Procedure	Cycle time = 1 week						Cycle time = 2 weeks									
	No room penalty			Room penalty			No room penalty			Room penalty						
	30 s	60 s	90 s	120 s	30 s	60 s	90 s	120 s	30 s	60 s	90 s	120 s				
Linear MIP MIN-MAX mean all units	3.5	3.5	3.5	3.5	-	-	-	-	-	7.9	2.5	2.5	-	-	-	-
Linear MIP MIN-MAX mean 2130 and 2140	1.4	1.3	1.3	1.3	-	30.7	30.7	23.6	2.6	1.5	1.5	1.5	-	18.5	17.8	17.8
Linear MIP MIN-MAX mean + var. 2130 and 2140	2.3	2.3	1.8	1.6	-	41.9	41.1	41.1	2.6	2.6	2.6	2.6	-	-	20.6	20.6
Quadratic MIP mean all units	-	-	-	0.05	-	-	-	-	-	4.6	3.9	3.9	-	-	-	-
Quadratic MIP mean 2130 and 2140	0.04	0.04	0.02	0.02	-	-	-	-	-	-	20.7	20.7	-	-	-	-
Quadratic MIP mean + var. 2130 and 2140	-	0.03	0.03	0.03	-	-	-	-	-	-	34.2	34.2	-	-	-	-
Simulated annealing	5.6	4.3	0.3	0	10.7	10.7	10.7	0	8.3	4.7	0	0	7.8	6.3	0.7	0
HIER-GOAL-1	N.A.	N.A.	N.A.	N.A.	27.3	27.3	27.3	27.3	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
HIER-GOAL-2	N.A.	N.A.	N.A.	N.A.	9.3	9.2	8.9	8.9	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
HIER-GOAL-1 + improvement step	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	-	3.5	0
HIER-GOAL-2 + improvement step	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	-	2.1	0

respect to the resulting bed occupancy distribution to introduce some differences between the schedules of the odd and the even weeks?”.

The computational results show that the built-in algorithms generally succeed well in generating schedules with leveled resulting bed occupancy. However, when the room objective is added, the single MIP approaches, in particular the quadratic ones, face computational difficulties. In this case, the hierarchical goal programming approaches provide a valuable alternative.

In real life, it is often the case that a surgeon admits or rejects patients as a function of the remaining bed capacity at the relevant hospitalization unit at that moment. In other words, an important part of the variability in the bed occupancy can be taken care of by appropriate admission of elective cases during the final stage of the surgery scheduling process which involves the detailed planning of the individual elective cases taking into account the block allocation from the master surgery schedule. Obviously, in the concern of both patient and surgeon, the postponement of surgery is best avoided as much as possible. Therefore, methods for a careful design of the master surgery schedule, as presented in this study, are still valuable. For future research, however, it would be interesting to extend the software with a module that deals with scheduling the individual elective cases. In addition, a statistical module that enables the user to make comparisons between the occupancy observed in real life on the one hand and the occupancy predicted by the system on the other hand, with a feedback mechanism to improve the predictions, would definitely increase the power of the software.

Acknowledgements We acknowledge the support given to this project by the Fonds voor Wetenschappelijk Onderzoek (FWO)—Vlaanderen, Belgium under contract number G.0463.04. We are grateful to all members of the Operating Room staff of the Virga Jesse Hospital in Hasselt, Belgium, and in particular to Els Nelissen for her support to this project and to Geert Moechars for providing the case study data. Thanks are also due to the editor and three anonymous referees for their valuable comments.

References

Adenso-Díaz, B., & Rodríguez, F. (1997). A simple search heuristic for the MCLP: Application to the location of ambulance bases in a rural region. *Omega, International Journal of Management Science*, 25, 181–187.

Aickelin, U., & Dowsland, K. A. (2000). Exploiting problem structure in a genetic algorithms approach to a nurse rostering problem. *Journal of Scheduling*, 31, 139–153.

Aickelin, U., & White, P. (2004). Building better nurse scheduling algorithms. *Annals of Operations Research*, 128, 159–177.

Beliën, J., & Demeulemeester, E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, 176, 1185–1204.

Blake, J. T., & Donald, J. (2002). Mount Sinai hospital uses integer programming to allocate operating room time. *Interfaces*, 32, 63–73.

- Blake, J. T., Dexter, F., & Donald, J. (2002). Operating room manager's use of integer programming for assigning block time to surgical groups: A case study. *Anesthesia and Analgesia*, *94*, 143–148.
- Brandeau, M. L., Sainfort, F., & Pierskalla, W. P. (2004). *Operations research and health care: A handbook of methods and applications*. Dordrecht: Kluwer Academic.
- Brotcorne, L., Laporte, G., & Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, *147*, 451–463.
- Burke, E. K., De Causmaecker, P., & Vanden Berghe, G. (1999). A hybrid tabu search algorithm for the nurse rostering problem. In *Simulated evolution and learning* (Vol. 1585, pp. 187–194). Berlin: Springer.
- Burke, E. K., De Causmaecker, P., Petrovic, S., & Vanden Berghe, G. (2003). METAHEURISTICS: computer decision-making. In M. G. C. Resende (Ed.), *Combinatorial Optimization Book Series. Variable neighborhood search for nurse rostering problems* (pp. 153–172). Dordrecht: Kluwer.
- Burke, E. K., De Causmaecker, P., & Vanden Berghe, G. (2004). Novel metaheuristic approaches to nurse rostering problems in Belgian hospitals. In J. Leung (Ed.), *Handbook of scheduling: algorithms, models and performance analysis* (Vol. 18, pp. 1–44). Boca Raton: CRC Press.
- Burke, E. K., De Causmaecker, P., Vanden Berghe, G., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling*, *7*, 441–499.
- Burke, E., Curtois, T., Post, G., Qu, R., & Veltman, B. (2008). A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, *188*(2), 330–341.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2006). *Optimizing a multiple objective surgical case scheduling problem* (Working paper KBI 0625). Katholieke Universiteit Leuven, Belgium.
- Cardoen, B., Demeulemeester, E., & Beliën, J. (2007). *Scheduling surgical cases in a day-care environment: a branch-and-price approach* (Working paper KBI 0724). Katholieke Universiteit Leuven, Belgium.
- Coppola, M., Ozcan, Y., & Bogacki, R. (2003). Evaluation of performance of dental providers on posterior restorations: Does experience matter? A data envelopment analysis (DEA) approach. *Journal of Medical Systems*, *27*, 447–458.
- Everett, J. (2002). A decision support simulation model for the management of an elective surgery waiting system. *Health Care Management Science*, *5*, 89–95.
- Fries, B. (1976). Bibliography of operations research in health-care systems. *Operations Research*, *24*, 801–804.
- Hans, E. W., Wullink, G., van Houdenhoven, M., & Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, *185*(3), 1038–1050.
- Hsu, V., de Matta, R., & Lee, C.-Y. (2003). Scheduling patients in an ambulatory surgical center. *Naval Research Logistics*, *50*, 218–238.
- Huang, M. D., Romeo, F., & Sangiovanni-Vincentelli, A. (1986). An efficient general cooling schedule for simulated annealing. In *IEEE international conference on computer-aided design* (pp. 381–384).
- ILOG (2002) *ILOG CPLEX 8.1 user's manual*.
- Kellogg, D. L., & Walczak, S. (2007). Nurse scheduling: From academia to implementation or not? *Interfaces*, *37*(4), 355–369.
- Kim, S.-C., & Horowitz, I. (2002). Scheduling hospital services: The efficacy of elective-surgery quotas. *Omega—The International Journal of Management Science*, *30*, 335–346.
- Kim, S.-C., Horowitz, I., & Buckley, T. A. (2000). Flexible bed allocation and performance in the intensive care unit. *Journal of Operations Management*, *18*, 427–443.
- Kirkpatrick, S., Gelatt, C. D. J., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.
- Kusters, R., & Groot, P. (1996). Modelling resource availability in general hospitals: Design and implementation of a decision support model. *European Journal of Operational Research*, *88*, 428–445.
- Lapierre, S., Batson, C., & McCaskey, S. (1999). Improving on-time performance in health care organizations: a case study. *Health Care Management Science*, *2*, 27–34.
- Litvak, E., & Long, M. C. (2000). Cost and quality under managed care: Irreconcilable differences? *The American Journal of Managed Care*, *6*, 305–312.
- Marcon, E., & Dexter, F. (2006). Impact of surgical sequencing on post anesthesia care unit staffing. *Health Care Management Science*, *9*, 87–98.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1958). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*, 1087–1092.
- Ogulata, S., & Erol, R. (2003). A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals. *Journal of Medical Systems*, *27*(3), 259–270.
- Pérez, B., Arenas, M., Bilbao, A., & Rodriguez, M. (2005). Management of surgical waiting lists through a possibilistic linear multi-objective programming problem. *Applied Mathematics and Computation*, *167*, 477–495.
- Pincus, M. (1970). A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, *18*, 1225–1228.
- Pritsker, A. (1998). Life and death decisions: Organ transplantation allocation policy analysis. *ORMS Today*, *25*, 22–28.
- RIZIV (2005). *Evolutie van de uitgaven voor geneeskundige verzorging*. www.riziv.be.
- Romeijn, H. E., Ahuja, R. K., Dempsey, J. F., & Kumar, A. (2006). A new linear programming approach to radiation therapy treatment planning problems. *Operations Research*, *54*(2), 201–216.
- Santibanez, P., Begon, M., & Atkins, D. (2007). Managing surgical waitlists for a British Columbia health authority. *Health Care Management Science*, *10*, 269–282.
- Sewell, E., & Jacobson, S. (2003). Using an integer programming model to determine the price of combination vaccines for childhood immunization. *Annals of Operations Research*, *119*, 261–284.
- Van Laarhoven, P. J. M., & Aarts, E. H. L. (1988). *Simulated annealing: Theory and applications*. Dordrecht: Kluwer.
- van Oostrum, J., Van Houdenhoven, M., Hurink, J., Hans, E., Wullink, G., & Kazemier, G. (2006). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spektrum*, *30*(2), 355–374.
- Vissers, J. M. H., Bertrand, J., & de Vries, G. (2001). A framework for production control in healthcare organisations. *Production Planning and Control*, *12*(6), 591–604.
- Wiers, V. C. (1997). A review of the applicability of OR and AI scheduling techniques in practice. *Omega—The International Journal of Management Science*, *25*(2), 145–153.