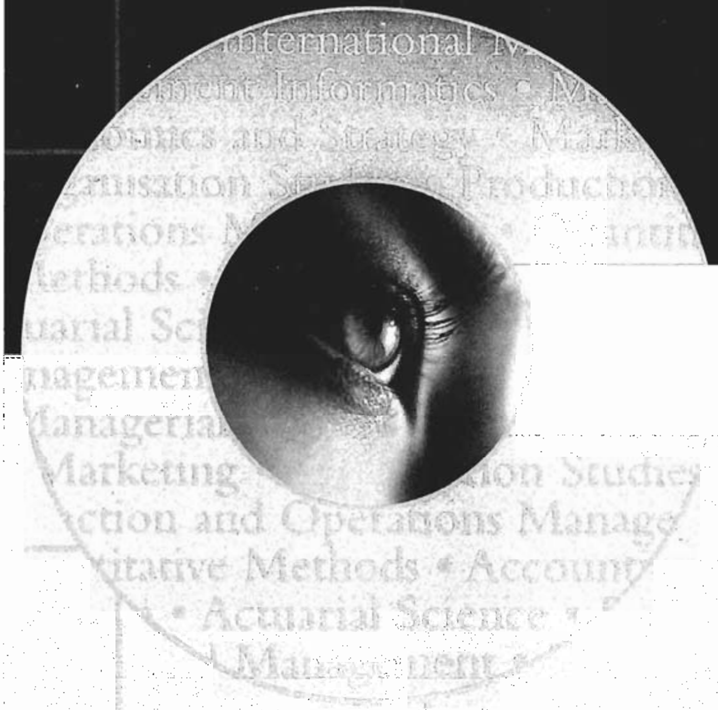
DEPARTMENT OF APPLIED ECONOMIC



RESEARCH REPORT

INTEGER PROGRAMMING FOR BUILDING ROBUST SURGERY SCHEDULES

JEROEN BELIËN • ERIK DEMEULEMEESTER

OR 0446

Integer programming for building robust surgery schedules

Jeroen Beliën
Erik Demeulemeester

Katholieke Universiteit Leuven
Naamsestraat 69, 3000 Leuven
e-mail: jeroen.belien@econ.kuleuven.be

Abstract

This paper proposes and evaluates a number of models for building robust cyclic surgery schedules. The developed models involve two types of constraints. Demand constraints ensure that each surgeon (or surgical group) obtains a specific number of operating room (OR) blocks. Capacity constraints limit the available OR blocks on each day. Furthermore, the number of operated patients per block and the length of stay (LOS) of each operated patient are dependent on the type of surgery. Both are considered stochastic, following a multinomial distribution. We develop a number of MIP-based heuristics and a metaheuristic to minimize the expected total bed shortage and present computational results.

Keywords: Surgery scheduling, resource leveling, integer programming, heuristics.

1 Introduction

Developing operating room (OR) schedules can be seen as a three stage process. In a first stage the available OR time is divided over the different surgeons (or surgical groups). This can be done based on total hours of cases per allocated block (i.e. utilization), hospital costs and gains per allocated block, hospital funding, previous number of allocated hours, demand for services, political issues, etc. This first phase is also referred to as case mix planning, since it determines for which pathologies capacity will be preserved. Case mix planning has a large impact on the quality of service. For instance, if the share of a certain pathology in the total case mix decreases, patients suffering from this pathology will be confronted with longer waiting times. Hughes and Soliman (1985) propose a linear programming model to solve case mix planning problems. Dexter and Macario (2002) argue that OR time should be allocated to maximize OR efficiency instead of "fixed hours" blocks based on historical utilization data. Blake and Carter (2002) propose a methodology that uses two linear goal programming models. One model sets case mix and volume for physicians, while holding service costs fixed; the other translates case mix decisions into a commensurate set of practical changes for physicians.

Once the OR time allocated to each surgical group has been chosen, the second stage involves the development of a master surgery schedule. The master surgery schedule is a cyclic timetable that defines the number and type of operating rooms available, the hours that rooms will be open, and the surgical groups or surgeons who are to be given priority for the operating room time. A new master schedule is created whenever the total amount of OR time changes. This can occur not only as a response to a long term change in the gross number of staffed OR hours, but also in response to seasonal fluctuations in demand (Blake et al., 2002). Compared to case mix planning (first stage) and elective case scheduling (third stage), the literature on master surgery scheduling is rather scant. Blake et al. (2002) propose an integer programming model that minimizes the weighted average undersupply of OR hours (i.e. allocating to each surgical group a number of OR hours as close as possible to its target OR hours). The produced master surgical schedule with a one week horizon is then extended to cover all weeks of the considered time horizon. Further decreases in the weighted average undersupply can often be achieved by modifying the schedule slightly from week to week. To this purpose the authors present an enumerative exchange-based heuristic procedure.

After the development of the master surgery schedule, elective cases can be scheduled. This third stage occurs on a daily base and involves detailed planning of each intervention. Each patient needs a particular surgical procedure, which defines the human (surgeon) and material (equipment) resources to use and the intervention duration. Most of the time, this is done on a first come first serve principle, regarding patient satisfaction. Without this hypothesis a decision problem is encountered. Guinet and Chaabane (2003) define this problem as a general assignment problem and propose a primal-dual heuristic to solve it. Weiss (1990) deals with the problem of determining the case orderings and presents both analytical and simulation results.

When building surgery schedules, several objectives could be taken into account. Much research has focussed on the maximization of operating room utilization for which many algorithms have been studied ranging from simple heuristics (e.g. earliest start time first, largest duration first etc...) to more complex bin packing algorithms (see e.g. Dexter and Traub, 2002 or Dexter et al., 1999). A strongly related objective is to minimize the OR staffing costs. Dexter et al. (2000) present a number of computer simulations of the effects of scheduling strategies on OR labor costs per patient. An objective that receives more and more attention nowadays is the management of uncertainty. Many studies have focussed on the increase of the punctuality of the schedule realization. Marcon et al. (2003) propose an operating theatre planning procedure that aims at mastering the risk of no realization of the tentative plan while stabilizing the operating rooms' utilization time. Dexter et al. (2001) study the effect of scheduling a delay between different surgeons' cases in order to improve the likelihood that each surgeon will start on time. Obviously, managing uncertainty requires insight in a number of aspects of the interaction of the planned (elective) and the emergency (non-elective) cases. Gerchak et al. (1996) provide a stochastic dynamic programming model for the advance scheduling of elective surgery under uncertain demand for emergency surgery. The problem is to determine how many of the requests for elective surgery to assign for each day. Their objective is to maximize a profit function which consists of a fixed profit per elective case, a fixed penalty per unit time exceeding the day's capacity and a fixed penalty for the postponement of a case. Bowers and Mould (2004) propose a policy of including planned, elective patients within the trauma (non-elective) session and show by means of simulation how substantially greater throughputs can be achieved, if patients are willing to accept a possibility of their treatment being canceled. Other proposals have been explored, including the option of concentrating health services such that one larger hospital serves a greater population (Bowers and Mould, 2002). The algorithms described in this paper also aim at an increase of the robustness of the surgery schedules. However, whereas most of the cited papers concentrate on the punctuality of the schedule realization, this work focusses on the control of the available capacity as a function of the master surgery schedule. As pointed out by Litvak and Long (2000), not only the non-elective cases contribute to the huge amount of variability in hospital environments. On the contrary, an important part of the variance can be controlled by applying well-thought-out scheduling policies of the elective cases. In what follows we briefly summarize their point.

The operation room is in fact the engine that drives the hospital. The activities inside the operation room have a dramatic impact on many other activities within hospitals. For instance, patients undergoing an operation are expected to recover during a number of days. Consequently, bed capacity and nursing staff requirements are dependent on the operation room schedule. By well-thought-out scheduling of the operation room, the expected variability in resource demand can be minimized. Variability has a very negative impact on productivity and reducing it is one of the major concerns of health care management. Litvak and Long (2000) distinguish between two types of variability: natural variability and artificial variability. Natural variability is inherent to the uncertain world of health care. This variability arises from uncertainty in patient show-ups, uncertainty in recovery time, uncertainty in the successfulness of therapies etc.... Artificial variability originates from poor scheduling policies. A poor operation room schedule could for instance directly be responsible for a shortage in beds each Wednesday, whereas there is overcapacity on all other days of the week. Exact and/or heuristic algorithms can assist in minimizing artificial variability. This is exactly the objective that our model will focus on.

2 Problem Statement

The problem addressed in this paper involves the construction of the master surgery schedule. The main objective is to minimize the expected shortage of one resource, namely beds. To make things clear, we will start from a simple example. On the one hand we have a surgery schedule divided in a number of time blocks. On the other hand we have a number of surgeons. Let us for simplicity suppose that each surgeon only performs one type of surgery. Furthermore, we assume that the number of patients operated per time block depends on the type of surgery and that this number is deterministic (this assumption will be relaxed in Section 5.1) and fixed for each surgeon. Whereas perfect knowledge is assumed concerning the number of patients undergoing surgery, there is however uncertainty concerning the length of stay (LOS) of each recovering patient. The LOS is assumed to follow a multinomial distribution with parameters which depend on the type of surgery. For instance, a patient recovering from appendix surgery leaves the hospital after 2 days with probability 20%, after 3 days with probability 50% and, finally, after 4 days with probability 30%. If a patient leaves after d days, (s)he occupies one bed for d days starting with the day of surgery. We are concerned with building a cyclic surgery schedule for which the expected total bed shortage (ETBS) is minimized. Cyclic schedules are schedules that are repeated after a certain time period (referred to as the cycle time). During such a cycle time there might be a number of time periods during which surgery cannot take place. These periods are referred to as the inactive periods, the other are active. Typically, cycle times are multitudes of weeks in which the weekends are inactive periods. To start we will state the problem mathematically. Let x_{is} ($\forall i \in A$ and $s \in S$) be the number of blocks assigned to surgeon s on day i. Here A represents the set of active periods and S the set of surgeons. Let r_s be the number of blocks required by each surgeon s. Let b_i be the maximal number of blocks available on day i. Then, our problem could be stated as follows (P1):

$$\text{Minimize } ETBS \tag{2.1}$$

subject to:

$$\sum_{i \in A} x_{is} = r_s \qquad \forall s \in S \tag{2.2}$$

$$\sum_{s \in S} x_{is} \leq b_i \qquad \forall i \in A \tag{2.3}$$

$$x_{is} \in \{0, 1, 2, \ldots, \min(r_s, b_i)\} \qquad \forall s \in S \text{ and } \forall i \in A \tag{2.4}$$

The objective function (2.1) minimizes the expected total bed shortage. Constraint set (2.2) implies that each surgeon obtains the number of required blocks. Constraint set (2.3) ensures that the number of blocks assigned does not exceed the available number of blocks on each day. Finally, constraint set (2.4) defines x_{is} to be integer. Let l be the length of the cycle time. The expected total bed shortage ($ETBS$)

equals the sum of the expected bed shortages on each day of the cycle time:

$$TEBS = \sum_{i=1}^{l} EBS_i \tag{2.5}$$

with EBS_i the expected bed shortage on day i. Let U_{ijs} be a stochastic variable representing the number of occupied beds on day i resulting from surgery on day j performed by surgeon s. It can easily be shown that U_{ijs} follows a binomial probability distribution, referred to as $f(U_{ijs})$. Now, let Z_i be a stochastic variable representing the total number of occupied beds on day i. Hence,

$$Z_i = \sum_{s \in S} \sum_{j \in A} U_{ijs} \tag{2.6}$$

The probability distribution of Z_i is given by:

$$f(Z_i = z_i) = \sum_{h \in H^{z_i}} (\prod_{U_{ijs} \in h} f(U_{ijs})) \tag{2.7}$$

with H^{z_i} the set of all combinations h of U_{ijs}'s summing up to z_i. Let c_i be the capacity of beds on day i. The expected shortage on day i is then as follows:

$$EBS_i = E[f(z_i | z_i > c_i)] = \sum_{z_i = c_i + 1}^{\infty} (z_i - c_i) f(z_i) \tag{2.8}$$

Given a certain schedule, we can calculate this expected value. If the total number of combinations leading to a shortage is not too large, we could apply complete enumeration. If complete enumeration is too time consuming, we could calculate approximated values based on the central limit theorem which states that the sum of many independent random variables is approximately normally distributed.
Since EBS_i is not linearly dependent on the decision variables, we cannot find the optimal solution using a mixed integer program (MIP) solver. Therefore, we will try to substitute EBS_i by an expression that is linear in the decision variables, such that it becomes solvable with commercial MIP packages. Of course, we want the new objective to be as equivalent as possible with the real objective.

3 Linearization of the problem

3.1 Mean

First, instead of dealing with the distribution functions $f(Z_i = z_i)$, we work with their mean values μ_i. Our assumption is that the larger the difference between c_i and μ_i, the smaller EBS_i, the expected bed shortage on day i. Without loss of generalization, we assume the bed capacity c_i to be constant for all days i, i.e.

$c_i = c$, $\forall i = 1..l$. Our objective is now to minimize the maximal μ_i. This hopefully results in a flat distribution of the expected bed occupation over all days of the week. In other words, the aim is to level the daily bed resource consumption as much as possible. In order to state our MIP, we first show that μ_i is linear with the decision variables x_{is}. Let D_{sd} be a stochastic variable representing the number of patients staying in the hospital exactly d days after one block of surgery by surgeon s. We obtain:

$$\begin{aligned}
\mu_i &= E(Z_i) && (3.1)\\
&= E\Big(\sum_{s\in S}\sum_{j\in A} U_{ijs}\Big) && (3.2)\\
&= \sum_{s\in S}\sum_{j\in A} E(U_{ijs}) && (3.3)\\
&= \sum_{s\in S}\sum_{j\in A}\Big(\sum_{d=dist(i,j)}^{m_s} E(D_{sd})\lceil d/l\rceil\Big)x_{js} && (3.4)\\
&= \sum_{s\in S}\sum_{j\in A}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}n_s\lceil d/l\rceil\Big)x_{js} && (3.5)
\end{aligned}$$

with $dist(i,j)$ the distance between day i and day j in the week, defined as $i-j+1$ if day j precedes day i and $l+i-j+1$ otherwise, m_s the maximal number of days a patient can stay in the hospital after surgery by surgeon s, p_{sd} the probability a patient stays d days in the hospital after surgery by surgeon s and n_s the number of patients surgeon s can operate in one time block. Expression (3.5) looks far more complicated than it is. We first note that the mean number of patients of surgeon s staying exactly d days in the hospital equals $p_{sd}n_s$ (mean of a binomial distribution with probability of 'success' p_{sd} and n_s trials). Obviously, a patient that leaves the hospital after d days occupies a bed from day 0 to day $d-1$. Hence, if we consider a particular day i after the day of surgery j we have to sum these expected values starting from the first LOS value reaching day i. This LOS value is given by $dist(i,j)$. For instance, if $i = 3$ (Wednesday) and $j = 1$ (Monday) we have $dist(i,j) = 3-1+1 = 3$. So, all patients staying 3 days (Monday, Tuesday and Wednesday) or more make up the expected number of patients on Wednesday resulting from surgery on Monday. Obviously, when the LOS exceeds the cycle time l, the corresponding expected number of patients has to be added twice (or more), which explains the factor $\lceil d/l\rceil$.

Since $\sum_{d=dist(i,j)}^{m_s} p_{sd}n_s\lceil d/l\rceil$ is a constant, the new objective is linear in the decision variables. Let μ be the maximal μ_i. We then have the following MIP (MIP1):

$$\text{Minimize } \mu \qquad (3.6)$$

subject to:

$$\sum_{i \in A} x_{is} = r_s \qquad \forall s \in S \qquad (3.7)$$

$$\sum_{s \in S} x_{is} \leq b_i \qquad \forall i \in A \qquad (3.8)$$

$$\mu_i = \sum_{s \in S} \sum_{j \in A} \Big(\sum_{d=dist(i,j)}^{m_s} p_{sd} n_s \lceil d/l \rceil \Big) x_{js} \qquad \forall i = 1..l \qquad (3.9)$$

$$\mu_i \leq \mu \qquad \forall i = 1..l \qquad (3.10)$$

$$x_{is} \in \{0, 1, 2, \ldots, \min(r_s, b_i)\} \qquad \forall s \in S \text{ and } \forall i \in A \qquad (3.11)$$

$$\mu_i \in \Re_0^+ \qquad \forall i = 1..l \qquad (3.12)$$

$$\mu \in \Re_0^+ \qquad (3.13)$$

Constraint set (3.9) defines the expected number of occupied beds on each day i. Constraint set (3.10) implies that μ exceeds each μ_i which ensures that the objective minimizes the maximal expected bed occupation μ.

3.2 Variance

MIP1 aims at a schedule for which the maximal expected bed occupation is reduced as much as possible over the week. We could however increase the effectiveness of our model by also taking into account the variances of the Z_i variables. Indeed, a schedule resulting from solving MIP1 may exhibit huge differences in the variances of the Z_i's. Figure 1 illustrates this point.

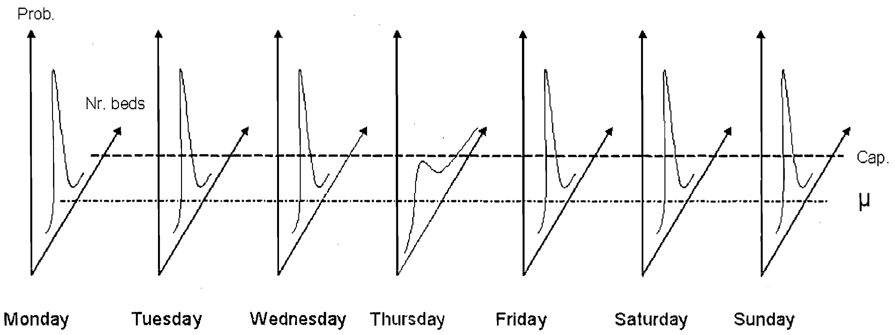


Figure 1: Role of variance

In this example we consider a cycle time of 1 week. The expected bed occupation is distributed quite level over all days of the week. However, the variance of the bed occupation is much larger on Thursday than on all other days. Consequently, there is a fair chance of running out of beds each Thursday. The question thus arises if it would be possible to include the variance in the objective function of our MIP.

Therefore, the variance of the Z_i's must be linear in the decision variables. In the derivation that follows, the next two rules are frequently applied:

$$var(a_0 + \sum_{i=1}^{n} a_i x_i) = \sum_{i=1}^{n} a_i^2 var(x_i) + \sum_{i=1}^{n} \sum_{j=1}^{i-1} 2a_i a_j cov(x_i; x_j) \tag{3.14}$$

$$\text{If } x_i \text{ and } x_j \text{ are independent, then } cov(x_i; x_j) = 0 \tag{3.15}$$

For the derivation it is important to keep in mind that the number of patients staying on the same day in the hospital but having 'entered' it via different blocks are completely independent of each other. There is only dependency between patient numbers coming from one and the same block in one and the same cycle.

Let us now start the derivation:

$$var(Z_i) = var(\sum_{s \in S} \sum_{j \in A} U_{ijs}) \tag{3.16}$$

Applying (3.14) and knowing that the covariances between the different U_{ijs}'s are all zero (the number of patients occupying a bed operated in different OR blocks are independent of each other) gives:

$$var(Z_i) = \sum_{s \in S} \sum_{j \in A} var(U_{ijs}) \tag{3.17}$$

$$= \sum_{s \in S} \sum_{j \in A} var\Big(\sum_{f=0}^{\lfloor \frac{m_s - dist(i,j)}{l} \rfloor} \sum_{g=1}^{x_{js}} \sum_{d=dist(i,j)+fl}^{m_s} D_{sd} \Big) \tag{3.18}$$

Recall that D_{sd} is a stochastic variable that stands for the number of patients who stay exactly d days in the hospital after one block of surgery by surgeon s. The first and third summations divide the D_{sd} variables into their cycles, i.e. the number of patients staying in the hospital on day i after surgery by surgeon s on day j can be divided according to the cycle in which they entered the system. For $f = 0$ all patients entered in the current cycle (= 0) are added, for $f = 1$ all patients entered in the previous cycle are added, etc.... The second summation indicates the number of blocks (x_{js}) for which patients are added.
Writing this in full gives:

$$\begin{aligned} var(Z_i) = \sum_{s \in S} \sum_{j \in A} var\Big(& \sum_{d=dist(i,j)}^{m_s} D_{sd} + \sum_{d=dist(i,j)}^{m_s} D_{sd} + \sum_{d=dist(i,j)}^{m_s} D_{sd} + \dots \\ & + \sum_{d=dist(i,j)+l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+l}^{m_s} D_{sd} + \dots \\ & + \sum_{d=dist(i,j)+2l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+2l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+2l}^{m_s} D_{sd} + \dots \\ & + \dots \Big) \end{aligned} \tag{3.19}$$

The first line indicates all patients entered in the current cycle. The different terms int this line indicate the different blocks that "produce" patients. The second line indicates the numbers entered in the previous cycle, etc....

The number of patients occupying a bed on a particular day i having undergone surgery more than 1 cycle ago is of course completely independent of the new patients entered in the current cycle. In general, the number of patients operated in the same block, but in different cycles, are independent of each other. Hence, application of again (3.14) and (3.15) gives:

$$\begin{aligned} var(Z_i) = \sum_{s\in S}\sum_{j\in A}\Big[& var\Big(\sum_{d=dist(i,j)}^{m_s} D_{sd} + \sum_{d=dist(i,j)}^{m_s} D_{sd} + \sum_{d=dist(i,j)}^{m_s} D_{sd} + \ldots\Big) \\ & + var\Big(\sum_{d=dist(i,j)+l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+l}^{m_s} D_{sd} + \ldots\Big) \\ & + var\Big(\sum_{d=dist(i,j)+2l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+2l}^{m_s} D_{sd} + \sum_{d=dist(i,j)+2l}^{m_s} D_{sd} + \ldots\Big) \\ & + \ldots\Big] \end{aligned} \tag{3.20}$$

Within each cycle the number of patients coming from one block on a particular day assigned to surgeon s is independent from the number coming from another block on the same day assigned to the same surgeon s. Applying again (3.14) and (3.15) gives:

$$\begin{aligned} var(Z_i) = \sum_{s\in S}\sum_{j\in A}\Big[& var(\sum_{d=dist(i,j)}^{m_s} D_{sd}) + var(\sum_{d=dist(i,j)}^{m_s} D_{sd}) + var(\sum_{d=dist(i,j)}^{m_s} D_{sd}) + \ldots \\ & + var(\sum_{d=dist(i,j)+l}^{m_s} D_{sd}) + var(\sum_{d=dist(i,j)+l}^{m_s} D_{sd}) + var(\sum_{d=dist(i,j)+l}^{m_s} D_{sd}) + \ldots \\ & + var(\sum_{d=dist(i,j)+2l}^{m_s} D_{sd}) + var(\sum_{d=dist(i,j)+2l}^{m_s} D_{sd}) + var(\sum_{d=dist(i,j)+2l}^{m_s} D_{sd}) + \ldots \\ & + \ldots\Big] \end{aligned} \tag{3.21}$$

Rewriting it in the shorter summation notation:

$$var(Z_i) = \sum_{s\in S}\sum_{j\in A}\sum_{f=0}^{\lfloor \frac{m_s - dist(i,j)}{l} \rfloor}\sum_{g=1}^{x_{js}} var(\sum_{d=dist(i,j)+fl}^{m_s} D_{sd}) \tag{3.22}$$

Applying (3.14) gives:

$$\begin{aligned} var(Z_i) = & \sum_{s \in S} \sum_{j \in A} \sum_{f=0}^{\lfloor \frac{m_s - dist(i,j)}{l} \rfloor} \sum_{g=1}^{x_{js}} \Big(\sum_{d=dist(i,j)+fl}^{m_s} var(D_{sd}) \\ & + \sum_{d_1=dist(i,j)+fl}^{m_s} \sum_{d_2=dist(i,j)+fl}^{d_1-1} 2cov(D_{sd_1}; D_{sd_2}) \Big) \end{aligned} \tag{3.23}$$

The covariances between the D_{sd} variables coming from the same block are of course not zero, but negative. Intuitively this can be seen as follows. The more patients that stay e.g. exactly 1 day, the less patients will stay exactly 2, 3, etc... days and vice versa. The total is always n_s. The variance and covariance formulas for the individual variables of a multinomial distribution are as follows:

$$var(D_{sd}) = p_{sd}(1 - p_{sd})n_s \tag{3.24}$$

$$cov(D_{sd_1}; D_{sd_2}) = -p_{sd_1} p_{sd_2} n_s \tag{3.25}$$

Alternatively, these formulas could be obtained by observing that the individual variables of a multinomial distribution are binomial processes with probability of 'success' p_{sd} and n_s trials. Applying these formulas gives:

$$\begin{aligned} var(Z_i) = & \sum_{s \in S} \sum_{j \in A} \sum_{f=0}^{\lfloor \frac{m_s - dist(i,j)}{l} \rfloor} \sum_{g=1}^{x_{js}} \Big(\sum_{d=dist(i,j)+fl}^{m_s} p_{sd}(1 - p_{sd})n_s \\ & - \sum_{d_1=dist(i,j)+fl}^{m_s} \sum_{d_2=dist(i,j)+fl}^{d_1-1} 2p_{sd_1} p_{sd_2} n_s \Big) \end{aligned} \tag{3.26}$$

Since g is merely a summation index and hence does not influence the calculation, summing up from 1 to x_{js} is the same as multiplying by x_{js}:

$$\begin{aligned} var(Z_i) = & \sum_{s \in S} \sum_{j \in A} \sum_{f=0}^{\lfloor \frac{m_s - dist(i,j)}{l} \rfloor} \Big(\sum_{d=dist(i,j)+fl}^{m_s} p_{sd}(1 - p_{sd})n_s \\ & - \sum_{d_1=dist(i,j)+fl}^{m_s} \sum_{d_2=dist(i,j)+fl}^{d_1-1} 2p_{sd_1} p_{sd_2} n_s \Big) x_{js} \end{aligned} \tag{3.27}$$

This expression can be further simplified by observing that also the summation over f can be turned into a multiplication. The summation is replaced by respectively the factor $\lceil d/l \rceil$ and $\lceil d_2/l \rceil$ indicating how many cycle times the D_{sd} variables contribute to respectively the variance and the covariance:

$$\begin{aligned} var(Z_i) = & \sum_{s \in S} \sum_{j \in A} \Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}(1 - p_{sd})n_s \lceil d/l \rceil \\ & - \sum_{d_1=dist(i,j)}^{m_s} \sum_{d_2=dist(i,j)}^{d_1-1} 2p_{sd_1} p_{sd_2} n_s \lceil d_2/l \rceil \Big) x_{js} \end{aligned} \tag{3.28}$$

In conclusion, the variance of each Z_i varies linearly with the decision variables.

Let us illustrate this with a simple example. Consider the following distribution of the LOS for each patient of surgeon s:

Table 1: LOS distribution for example 1

LOS (Nr. of days)	2	3	4	10	11
probability	0.2	0.3	0.1	0.3	0.1

Assume a cycle time of 1 week. For illustrative purposes, we opted for a LOS distribution having a limited number of outcomes and a 'tail' exceeding the cycle time. Although this example may not seem to be very realistic at first sight, it could represent a scenario in which the operated patients can be divided into two groups: the first group having no complications and leaving the hospital within 4 days and the second group having complications and staying much longer. Assume it is known that this surgeon can operate 10 patients per block. Now, suppose we assign one block on Monday to this surgeon. We illustrate the calculation of $E(U_{3,1,s})$ and $var(U_{3,1,s})$. Let $D_{sd'}$ denote the number of patients staying d days in the hospital who have undergone surgery in the previous week. We obtain:

$$\begin{aligned}
E(U_{3,1,s}) &= E(D_{s3} + D_{s4} + D_{s10} + D_{s11} + D_{s10'} + D_{s11'}) \\
&= E(D_{s3}) + E(D_{s4}) + E(D_{s10}) + E(D_{s11}) + E(D_{s10'}) + E(D_{s11'}) \\
&= E(D_{s3}) + E(D_{s4}) + 2E(D_{s10}) + 2E(D_{s11}) \\
&= 0.3 * 10 + 0.1 * 10 + 2 * 0.3 * 10 + 2 * 0.1 * 10 \\
&= 3 + 1 + 6 + 2 = 12
\end{aligned}$$

$$\begin{aligned}
var(U_{3,1,s}) =& var(D_{s3} + D_{s4} + D_{s10} + D_{s11} + D_{s10'} + D_{s11'}) \\
=& var(D_{s3}) + var(D_{s4}) + var(D_{s10}) + var(D_{s11}) + var(D_{s10'}) + var(D_{s11'}) \\
& + 2cov(D_{s4}, D_{s3}) + 2cov(D_{s10}, D_{s3}) + 2cov(D_{s10}, D_{s4}) \\
& + 2cov(D_{s11}, D_{s3}) + 2cov(D_{s11}, D_{s4}) + 2cov(D_{s11}, D_{s10}) \\
& + 2cov(D_{s11'}, D_{s10'}) \\
=& var(D_{s3}) + var(D_{s4}) + 2var(D_{s10}) + 2var(D_{s11}) \\
& + 2cov(D_{s4}, D_{s3}) + 2cov(D_{s10}, D_{s3}) + 2cov(D_{s10}, D_{s4}) \\
& + 2cov(D_{s11}, D_{s3}) + 2cov(D_{s11}, D_{s4}) + 4cov(D_{s11}, D_{s10}) \\
=& 0.3 * 0.7 * 10 + 0.1 * 0.9 * 10 + 2 * 0.3 * 0.7 * 10 + 2 * 0.1 * 0.9 * 10 \\
& - 2 * 0.1 * 0.3 * 10 - 2 * 0.3 * 0.3 * 10 - 2 * 0.3 * 0.1 * 10 \\
& - 2 * 0.1 * 0.3 * 10 - 2 * 0.1 * 0.1 * 10 - 4 * 0.1 * 0.3 * 10 \\
=& 8.3 - 3 - 2 = 3.3
\end{aligned}$$

We extend MIP1 such that the variance is taken into account. Let σ_i^2 be the variance of Z_i. Let w_μ and w_{σ^2} be the weight expressing the relative importance of respectively leveling the mean and variance of the bed occupation. Let γ be the maximal weighted sum of mean and variance. We then obtain the following MIP (MIP2):

$$\text{Minimize } \gamma \tag{3.29}$$

subject to:

$$\sum_{i \in A} x_{is} = r_s \qquad \forall s \in S \tag{3.30}$$

$$\sum_{s \in S} x_{is} \leq b_i \qquad \forall i \in A \tag{3.31}$$

$$\mu_i = \sum_{s \in S} \sum_{j \in A} \Big(\sum_{d=dist(i,j)}^{m_s} p_{sd} n_s \lceil d/l \rceil \Big) x_{js} \qquad \forall i = 1..l \tag{3.32}$$

$$\sigma_i^2 = \sum_{s \in S} \sum_{j \in A} \Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}(1-p_{sd}) n_s \lceil d/l \rceil - \sum_{d_1=dist(i,j)}^{m_s} \sum_{d_2=dist(i,j)}^{d_1-1} 2 p_{sd_1} p_{sd_2} n_s \lceil d_2/l \rceil \Big) x_{js} \qquad \forall i = 1..l \tag{3.33}$$

$$w_\mu \mu_i + w_{\sigma^2} \sigma_i^2 \leq \gamma \qquad \forall i = 1..l \tag{3.34}$$

$$x_{is} \in \{0, 1, 2, \dots, \min(r_s, b_i)\} \qquad \forall s \in S \text{ and } \forall i \in A \tag{3.35}$$

$$\mu_i \in \Re_0^+, \sigma_i^2 \in \Re_0^+ \qquad \forall i = 1..l \tag{3.36}$$

$$\gamma \in \Re_0^+ \tag{3.37}$$

3.3 Special cases

When w_{σ^2} equals 0, the variance is ignored and model MIP1 will result. When w_μ equals 0, the mean is ignored and MIP2 will minimize the maximal variance of the daily bed occupation. This means that the resulting bed occupation may exhibit peaks on certain days of the week. However, these peaks will be well predictable. This model is appropriate if the capacity of the resource is adaptable, since the peaks in the demand for resources could be anticipated by providing more capacity during these peaks. An example of such a flexible resource is manpower. Bed capacity is however generally not adaptable at short term.

The relative importance of w_{σ^2} and w_μ might be dependent on the presence (or absence) of an external stochastic process consuming the considered resource. In our example for instance beds might be occupied from emergency cases. Consider first the situation in which there is no such external process. Suppose we set w_{σ^2} equal to 0 and find an 'optimally' leveled solution. However, given uneven distributed variances, there are certain days in which there is a fair chance of bed shortage. We might obtain a better solution by slightly increasing w_{σ^2}. Assume that in the new solution, although the mean bed day occupations exhibit larger differences, the sum of the probabilities of bed shortages is much smaller. Hence, in this situation a positive value for w_{σ^2} is clearly better in the absence of external stochastic processes. However, if we do allow for external processes to consume resources, this conclusion might not hold any more. Since no single model will ever include all sources of variability in hospital environments, this is certainly an interesting point for further research.

3.4 Percentile minimization

Incorporating the variance could be done in a slightly different way. Instead of calculating the true mean and the true variance and minimizing the peak of the weighted sum, one could directly calculate the contribution of each decision variable x_{is} to some kind of weighted measure. Therefore, we take the contribution to the mean and add n_{stdev} times the square root of the contribution to the variance. For instance, the contribution for x_{js} would be:

$$\sum_{d=dist(i,j)}^{m_s} p_{sd} n_s \lceil d/l \rceil$$
$$+ n_{stdev} \left(\sum_{d=dist(i,j)}^{m_s} p_{sd}(1-p_{sd}) n_s \lceil d/l \rceil - \sum_{d_1=dist(i,j)}^{m_s} \sum_{d_2=dist(i,j)}^{d_1-1} 2 p_{sd_1} p_{sd_2} n_s \lceil d_2/l \rceil \right)^{\frac{1}{2}} \tag{3.38}$$

The model is then totally equivalent with MIP1 (3.6-3.13) except for the coefficients of constraint 3.9. Although referred to as percentile minimization, the model does not necessarily minimize the highest percentile peak. Minimizing the highest percentile peak is equivalent to minimizing the highest tail distribution and is a non-linear problem. Instead, we try to measure the contribution of each variable to each day percentile with a linear weight and solve the problem with a linear optimizer. We choose to take the root of the variance contributions, because standard deviations are more common when referring to distribution tails.

4 Is autocorrelation a problem?

Preliminary tests showed that the simulated variances of daily bed occupation (Z_i) are slightly smaller than the calculated ones given by (3.28). The reason is the occurrence of autocorrelation. Indeed, part of the variance of the number of patients in the hospital on day i of cycle t is explained by the number on day i of cycle $t-1$. This part is proportional with the number of patients expected to stay longer than the cycle time. A simulation study in Section 8.4 will try to answer the question to what amount this autocorrelation is influencing the probabilities of bed shortages.

5 Extensions

5.1 Stochastic n_s

An important drawback of our model is the assumption of deterministic numbers of patients (n_s). It would be interesting to extend our model such that it can handle stochastic n_s's. Would it still be possible to express both mean and variance as linear combinations of the decision variables?
Answer: This is no problem. Introducing stochastic n_s's following a multinomial distribution does not destroy the linearity of both average and variance. Hence, instead of assuming deterministic patient numbers, we can deal with uncertainty: for instance for a particular surgeon the number of operated patients equals 7 with probability 10%, 8 with probability 20%, 9 with probability 40% and 10 with probability 30%. We show how the expressions for both mean and variance are extended such that they incorporate this additional stochastic information. For the mean, we make use of the following theorem on conditional means:

$$E(Y) = E[E(Y|X)] \tag{5.1}$$

In words, the overall mean equals the mean of the conditional means. Applied to our problem: let N_s be a stochastic variable representing the number of patients for surgeon s. $k = 1..q_s$ are the different (discrete) states of this variable with h_{sk} being the probability and n_{sk} the corresponding number of patients in state k for patient s.

$$\begin{aligned} E(U_{ijs}) &= E[E(U_{ijs}|N_s)] && (5.2) \\ &= \sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd} n_{sk} \lceil d/l \rceil \Big) && (5.3) \end{aligned}$$

For the variance, we make use of the following theorem on conditional variances:

$$var(Y) = E[var(Y|X)] + var[E(Y|X)] \tag{5.4}$$

In words, the overall variance equals the sum of (1) the mean of the conditional variances and (2) the variance of the conditional means. Applied to our problem:

$$var(U_{ijs}) = E[var(U_{ijs}|N_s)] + var[E(U_{ijs}|N_s)] \tag{5.5}$$

Elaborating the first term gives:

$$\begin{aligned} E[var(U_{ijs}|N_s)] =& \sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}(1-p_{sd}) n_{sk} \lceil d/l \rceil \\ & - \sum_{d_1=dist(i,j)}^{m_s} \sum_{d_2=dist(i,j)}^{d_1-1} 2 p_{sd_1} p_{sd_2} n_{sk} \lceil d_2/l \rceil \Big) \end{aligned} \tag{5.6}$$

Elaborating the second term gives:

$$\begin{aligned} var[E(U_{ijs}|N_s)] &= \sum_{k=1}^{q_s} h_{sk}\Big(E(U_{ijs}|N_s) - E(U_{ijs})\Big)^2 \\ &= \sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}n_{sk}\lceil d/l\rceil - \sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}n_{sq}\lceil d/l\rceil\Big)\Big)^2 \end{aligned} \tag{5.7}$$

Combining all this gives us the variance of U_{ijs}:

$$\begin{aligned} var(U_{ijs}) = &\sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}(1-p_{sd})n_{sk}\lceil d/l\rceil \\ &- \sum_{d_1=dist(i,j)}^{m_s}\ \sum_{d_2=dist(i,j)}^{d_1-1} 2p_{sd_1}p_{sd_2}n_{sk}\lceil d_2/l\rceil\Big) \\ &+ \sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}n_{sk}\lceil d/l\rceil - \sum_{k=1}^{q_s} h_{sk}\Big(\sum_{d=dist(i,j)}^{m_s} p_{sd}n_{sq}\lceil d/l\rceil\Big)\Big)^2 \end{aligned} \tag{5.8}$$

In conclusion, incorporating numbers of patients following a discrete probability distribution preserves the linearity of both mean and variance of the daily bed occupation. Hence, the above outlined MIP's can perfectly incorporate this source of uncertainty.

6 NP-hardness proof of linearized problem

In what follows an NP-hardness proof for problem MIP1 is given. The NP-hardness is proven by means of a transformation from 3-PARTITION. This problem can be described as follows:
3-PARTITION: Given a set $T = \{1, \ldots, 3t\}$ and positive integers $a_1, \ldots, a_{3t}$ and c with $\sum_{j\in T} a_j = tc$, can T be partitioned into t disjoint 3-element subsets T_i such that $\sum_{j\in T_i} a_j = c$ $(i = 1, \ldots, t)$?
This celebrated problem was the first number problem that was proven to be NP-complete in the strong sense. A (very) small problem instance will illustrate this problem: the set T consists of 6 elements with corresponding values of 3, 3, 3, 4, 4 and 5. The values of t and c are obviously 2 (3*2=6 elements) and 11 (3+3+3+4+4+5 = 22 = 2*11), respectively. for this problem instance the answer is positive: T_1 could consist of elements 1, 2 and 6 with corresponding values of 3, 3 and 5, whereas the second set T_2 then consists of the remaining three elements 3, 4 and 5 with values 3, 4 and 4.
Given any instance of the 3-PARTITION problem, an instance of the problem MIP1 can be constructed in the following way:

- The cycle time (l) equals t; there are no inactive days ($A = \{1, \ldots, t\}$).
- The number of blocks per day (b_i) equals 3.
- The number of surgeons equals the number of different values in the set T.
- The number of patients each surgeon s can operate per block (n_s) equals the corresponding value.
- The number of requested blocks per surgeon (r_s) equals the number of times the corresponding value occurs in set T.
- The LOS of the patients is deterministic and equals 1 for each surgeon, i.e. $p_{s1} = 1, \forall s, p_{sd} = 0, \forall s, \forall d \neq 1$.

We show that 3-PARTITION has a solution if and only if there exists a feasible schedule with $\mu = c$.
Suppose that 3-PARTITION has a solution $\{T_1, \ldots, T_t\}$. A feasible schedule with value $\mu = c$ is then obtained as follows. Each set $T_1, \ldots, T_t$ represents an operating day containing 3 blocks at which the surgeons corresponding to the elements in the set are scheduled. The number of patients occupying a bed on each day amounts to c which is the sum of the operated patients during each day. In order to prove the optimality of the solution, we show that $\mu = c$ equals a lower bound. Since each patient stays exactly 1 day, the total LOS over all patients equals the total number of patients, $\sum_s n_s = \sum_{j \in T} a_j = tc$. If we manage to distribute all these patients perfectly balanced over the cycle time, we obtain a solution of $(\sum_{j \in T} a_j)/t = tc/t = c$. It follows that $\mu = c$ is a lower bound to our problem.
Conversely, suppose that there is a feasible schedule with value $\mu = c$. First of all, three blocks must have been assigned at each day, since the total number of requested blocks equals the total number of available blocks (i.e. $\sum_{s \in S} r_s = \sum_{i=1..l} b_i = 3t$). By definition, we have for each day i: $\mu_i \leq \mu$. Now, since each patient stays exactly 1 day, the total LOS over all patients equals the total number of patients, $\sum_s n_s = \sum_{j \in T} a_j = tc$. Hence, if the schedule would have a day i for which $\mu_i < \mu$, then there must be another day having a $\mu_i > \mu$. By definition, this is not possible and thus each day must have a μ_i equal to $\mu = c$. Hence, each day $i = 1..t$ represents a set of 3 elements (surgeon-block assignments) with the sum of their values (nr. of operated patients) equal to c. This is a solution to 3-PARTITION. Since MIP1 is a special case of MIP2, MIP2 is also NP-hard in the strong sense. Q.E.D.

7 Solving the original problem

MIP2 could be solved with a commercial MIP solver. Preliminary tests indicated that the LP relaxation gap of MIP2 is fairly small. This suggests that it will be difficult to develop a specific (branch-and-bound) algorithm that could solve the problem more efficiently. Nevertheless, a number of interesting research questions remain:

1. Do the proposed integer programming models provide satisfying solutions to the original problem P1 (2.1-2.4)?

2. Would it be possible to use these models in order to develop a heuristic that provides better results?

3. Which of the elaborated models/heuristics is best suited to solve the original problem P1?

4. Is the best choice dependent on certain problem dimensions?

5. How do the results compare to a metaheuristic approach in which the objective function is evaluated directly?

7.1 Objective function

If we want to solve P1, we should be able to evaluate objective function (2.1). In order to do this, we need to deduce the exact bed usage probability distributions of each day, given a particular surgery schedule. Unfortunately, computing these general discrete distribution functions involves the enumeration of an exponential number of probability states, which is computationally very hard. Therefore, we will approach the exact objective (2.1) with an easy to calculate one, making use of the central limit theorem. According to this theorem, each variable which is the sum of a number of independent variables, is approximately normally distributed with mean equal to the sum of the independent means and variance equal to the sum of the independent variances. Recall that the independent means and variances can easily be calculated exactly. Hence, for calculating the shortage probabilities we can simply make use of the standard cumulative normal distribution functions. For calculating expected shortages we have to apply numerical integration. For instance, in order to calculate the expected shortage for day i, we compute the following integral:

$$EBS_i \approx \int_{c_i+0.5}^{+\infty} (z_i - c_i) \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(z_i-\mu_i)^2}{2\sigma_i^2}} dz_i \tag{7.1}$$

This expression simply sums up all shortages $(z_i - c_i)$ multiplied by the corresponding probabilities. The reason why the integral starts at $c_i + 0.5$ (and not at c_i or at $c_i + 1$) is that we have to take into account a continuity correction for approaching a discrete function with a continue one. For calculating these integrals we made use of the numerical integration routines provided in *GNU Scientific Library* (GSL) version 1.3 (Galassi et al., 2003).

In what follows, three heuristics will be elaborated which aim at the minimization of this objective: a repetitive MIP heuristic, a quadratic MIP heuristic and a local search heuristic (simulated annealing).

7.2 Repetitive MIP heuristic

As the name suggests, the repetitive MIP heuristic involves the successive solving of a number of MIP's. After each solution, an extra constraint is added to the model, which limits the search space. For the moment we will only concentrate on the averages and thus neglect the impact of the variance. This can be motivated by the fact that the average and variance of each Z_i are positively correlated to some extent and hence low averages tend to go together with low variances and vice versa. We implemented two repetitive MIP heuristics, to which we refer as REPMIP1 and REPMIP2 respectively.

REPMIP1 works as follows:

1. $TEBS = \infty$.
2. Solve MIP1 (3.6)-(3.13). If the found schedule results in a lower total expected bed shortage, save it as being the best found. Let $\hat{\mu}$ be the optimal objective value and let i be the day with the maximal peak, i.e. $\mu_i = \hat{\mu}$.
3. Add an extra constraint to the model: $\mu_i \leq \hat{\mu} + \epsilon$.
4. Make μ_i no longer contribute to the objective function. Therefore, delete $\mu_i \leq \mu$ out of constraint set (3.10).
5. Go back to step 2. Repeat this until a certain stop criterium is met.

The idea is that after the minimization of the highest peak, the second highest peak is to be minimized, whereby the peak of the highest day is kept below a certain limit. Next, the third highest peak is minimized with constraints on the first two peaks and so on.... ϵ determines to which amount the previous peak(s) can be exceeded. If ϵ equals 0, the search space is limited most from MIP to MIP. ϵ can be made dependent on the progression of the algorithm. The solution of each MIP provides a surgery schedule which could be evaluated by calculating the total expected bed shortage (TEBS), for which we do a number of numerical integrations (7.1). The best schedule is saved.

REPMIP2 works as follows:

1. $TEBS = \infty$.
2. Solve MIP1 (3.6)-(3.13). If the found schedule results in a lower total expected bed shortage, save it as being the best found. Let $\check{\mu}_i$ be the lowest bed occupation peak and let i be the day with this minimal peak.
3. Add an extra constraint to the model: $\mu_i \geq \check{\mu}_i + \epsilon$.
4. Solve the adapted model. If the found schedule results in a lower total expected bed shortage, save it as being the best found.

5. Increase the right hand side value of the constraint, added in step 3 with ϵ over the current usage of beds on day i.

6. Go back to step 4. Repeat this until a certain stop criterium is met.

The idea is that after the minimization of the highest peak, the lowest peak is identified. Next, the model is resolved with an extra constraint which prohibits the current solution by implying an increase in the lowest peak. The aim is that the overcapacity in this lowest peak is divided over all other days but the peak day. ϵ determines to which amount the previous off-peak(s) has to be exceeded. A typical value for ϵ is 0.01. Typical end criteria include the detection of an infeasible model and/or the peak of the lowest day exceeding a certain limit (e.g. the overall average bed occupation). The solution of each MIP provides a surgery schedule which could be evaluated by calculating the total expected bed shortage (TEBS), for which we do a number of numerical integrations (7.1). The best schedule is saved.

7.3 Quadratic MIP heuristic

Also in this heuristic we neglect the variances and merely take into account the averages. We solve again a MIP, however the objective function is now quadratic (QMIP):

$$\text{Minimize} \sum_{i \in A} \mu_i^2 \tag{7.2}$$

subject to:

$$\sum_{i \in A} x_{is} = r_s \qquad \forall s \in S \tag{7.3}$$

$$\sum_{s \in S} x_{is} \le b_i \qquad \forall i \in A \tag{7.4}$$

$$\mu_i = \sum_{s \in S} \sum_{j \in A} \Big(\sum_{d=dist(i,j)}^{m_s} p_{sd} n_s(\lfloor d/l \rfloor + 1) \Big) x_{js} \qquad \forall i = 1..l \tag{7.5}$$

$$x_{is} \in \{0, 1, 2, \ldots, \min(r_s, b_i)\} \qquad \forall s \in S \text{ and } \forall i \in A \tag{7.6}$$

$$\mu_i \in \Re_0^+ \qquad \forall i = 1..l \tag{7.7}$$

$$\mu \in \Re_0^+ \tag{7.8}$$

Since $\sum_{i \in A} \mu_i$ is constant and hence independent of the surgery schedule, this model explicitly tries to level the peaks as much as possible. Note that the minimization of $x_1^2 + x_2^2$, subject to $x_1 + x_2 = a$ results in $x_1 = x_2 = \frac{a}{2}$. Remark that also $\sum_{i \in A} \sigma_i^2$ is constant and hence independent of the surgery schedule, thus we might also take into account the variances. Again, we evaluate the resulting surgery schedule by calculating the objective function by computing a number of integrals (7.1).

7.4 Simulated annealing

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. This technique stems from thermal annealing which aims to obtain perfect crystallizations by a slow enough temperature reduction to give atoms the time to attain the lowest energy state. The search tries to avoid local minima by jumping out of them early in the computation. Towards the end of the computation, when the temperature, or probability of accepting a worse solution, is nearly zero, this simply seeks the bottom of the local minimum. The chance of getting a good solution can be traded off with computation time by slowing down the cooling schedule. The slower the cooling, the higher the chance of finding the optimum solution, but the longer the run time. Thus effective use of this technique depends on finding a cooling schedule that gets good enough solutions without taking too much time. 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 an optimization technique for combinatorial (and other) problems.

Our simulated annealing (SA) implementation is very basic. Our 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. Since swaps between one surgeon and swaps between one day have no impact on the objective function, these swaps are not taken into account. 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.

The advantage of SA over the previous two methods is that the true objective can immediately be evaluated. In contrast, the repetitive MIP heuristic optimizes a series of linear objective functions which hopefully result in a schedule that minimizes the true objective. Similarly, the quadratic MIP heuristic evaluates a quadratic objective instead of the true objective. The main drawback of SA is that we have to do some experiments in order to find good values for T and the temperature decrease function. The probability of a worse solution acceptance should be large at the start of the search and small towards the end.

8 Computational experiment

8.1 Test set

In order to study the computational performance of the heuristics, a test set has been composed. Firstly, note that all test problems involve a cycle time of 7 days in which the last two days are not available to allocate OR time (weekend), which is common in practice. We identified seven factors which we thought could have an impact on the complexity of the problem. These are: (1) the number of time blocks per day, (2) the number of surgeons, (3) the division of requested blocks per surgeon, (4) the number of operated patients per surgeon, (5) the probability of a no show-up as a measure of the variability in this number, (6) the length of stay (LOS) distribution and finally (7) the bed capacity. If we consider two settings for each factor and repeat each factor combination 3 times, we obtain $2^7 * 3 = 384$ test instances. Table 2 contains the settings for these seven factors. Some of the factor settings require some further explanation.

Table 2: Design of experiment

Factor setting	Nr. blocks per day	Nr. surgeons	Division req. blocks	Nr. patients per surgeon	Prob. no show-up	LOS	Capacity
1	3-6	3-7	evenly distributed	3-5	5%	2-5	105%
2	7-12	8-15	not evenly distributed	3-12	10%	2-12	110%

The number of blocks per day is drawn from a uniform distribution with bounds 3 and 6 in the first setting and 7 and 12 in the second setting. A block is defined as the smallest time unit for which a specific operating room can be allocated to a specific surgeon (or surgical group). Note that, due to large set-up time and costs, in real-life applications the number of blocks per day in one operating room is usually 1 or 2, i.e each surgical group has the OR for at least half a day. Hence, considering more blocks can be seen as a way of considering more operating rooms as there is no difference from a computational point of view. The third factor indicates whether or not the requested blocks are evenly distributed among all surgeons; e.g. if there are 20 time blocks and 5 surgeons, each surgeon requires 4 time blocks in the evenly distributed case, whereas in the unevenly distributed case huge differences can occur. Factor 5 defines the probability of a no show-up. The higher this probability, the higher the variability in the number of operated patients distribution for each surgeon. For the LOS in factor 6 we simulated exponential distributions (made discrete by use of binomial distributions) with mean dependent on the factor setting. Finally, the capacity was set as follows. First we calculate the total bed occupation, i.e. sum up all (expected) LOS days of all (expected) patients of all surgeons. This number was divided by 7 in order to obtain the absolute minimum required capacity. Next, depending on the factor setting this capacity was increased with 5 or 10%.

8.2 Tested heuristics

Using these 384 test instances the following heuristic algorithms were tested:

Abbrev.	Description
MINMU	Minimize average peak (=MIP1)(3.6-3.13)
MINWEIGHTED	Minimize weighted peak (=MIP2 with $w_\mu = 0.8$ and $w_{\sigma^2} = 0.2$) (3.29-3.37)
REPMIP1	Repetitive MIP model 1 with ϵ=1% of previous peak
REPMIP2	Repetitive MIP model 2 with ϵ=0.01
QP	Quadratic Programming model (7.2-7.8)
MINMUPERC	Same as MINMU, but now based on percentiles (see 3.4) $n_{stdev} = 0.2$
REPMIP1PERC	Idem for REPMIP1
REPMIP2PERC	Idem for REPMIP2
QPPERC	Idem for QP
SA1	Simulated annealing (7.4) objective=min. total expected shortage initial temperature=500 temperature update interval=10 iterations temperature update function=0.95*previous temperature end criterium=max. time all previous heuristics
SA2	See SA1 except for end criterium=1000 iterations

All heuristics were implemented in Visual C++ and linked with CPLEX 8.1 (ILOG, 2002) as a callable optimization library to perform linear and quadratic optimization.

8.3 Computational Results

Table 3 contains the results of our experiment. This table contains average values (over all 384 test instances) for the expected total bed shortages (ETBS) and average values and standard deviations for the computation times (in milliseconds). The standard deviations give an indication of the variability of the computation times for each heuristic.

Table 3: Computational results

Heuristic	Avg. exp. shortage (ETBS)	Avg. comp. time (ms)	St. dev. comp. time (ms)
MINMU	9.346	76.510	228.334
MINMUPERC	9.362	78.518	405.307
MINWEIGHTED	9.219	86.174	195.423
REPMIP1	7.853	17833.776	321353.376
REPMIP1PERC	7.941	3981.865	42299.126
REPMIP2	7.278	2624.906	5145.229
REPMIP2PERC	7.278	2168.659	3888.637
QP	7.312	27.951	20.946
QPPERC	7.464	26.443	19.029
SA1	9.536	22109.503	323544.954
SA2	6.698	56808.042	20574.437

Figure 2 and 3 visualize this table. Note that the Y-axis in Figure 3 has a logarithmic scale.

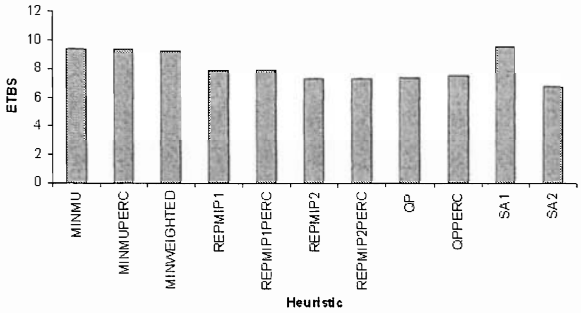


Figure 2: Comparison of heuristic results

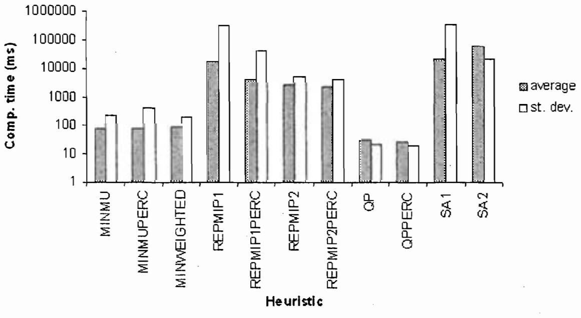


Figure 3: Comparison of heuristic computation times

From these figures we can draw a number of conclusions. First of all, if we look at the expected shortages, we see that SA2 finds the best solutions, followed by REPMIP2, REPMIP2PERC, QP and QPPERC. Additionally to these average values, a repeated measures analysis was done with the SAS software system in order to be able to draw well-founded conclusions. The solutions found by SA2 were significantly ($\alpha = 0.05$) better than those found in REPMIP2, REPMIP2PERC, QP and QPPERC, between which no significant difference could be found. The results from REPMIP1 and REPMIP1PERC are significantly worse than the previous four heuristics. Finally, MINMU, MINWEIGHTED, MINMUPERC and SA1 performed significantly worse than all other heuristics, but again no significant differences could be found between them. With respect to the computation times, four groups can be distinguished (from smallest to largest computation time): (1) the quadratic MIP heuristics (QP and QPPERC), (2) the single MIP heuristics (MINMU, MINWEIGHTED and MINMUPERC), (3) the repetitive MIP heuristics (REPMIP1, REPMIP1PERC, REPMIP2 and REPMIP2PERC) and (4) SA2. Recall that the computation time given to SA1 equals the largest of the MIP heuristics and hence, SA1 is obviously situated in the third group. From the standard deviations we may conclude that the computation time of REPMIP1 and SA2 are highly variable. Analyzing computation times in SAS yielded no significant difference between QP and QPPERC. The quadratic MIP heuristics outperform all other heuristics, although no significant differences could be found with REPMIP1, REPMIPPERC and SA1, due to the large variability in these data.

The reason why SA2 has such large computation times is that the evaluation of the true objective (via numerical integration) is very time consuming. Therefore, a second SA heuristic was implemented in which the objective is a weighted sum of the squared average daily bed occupations (like in QP) and the total shortage probability. This new objective can be evaluated instantly and hence many more iterations of SA can take place. Since the total squared sum of daily average bed occupations is much larger than the total shortage probability, this first measure is normalized such that it falls in a range from 0 (minimum) to 1 (maximum). The end criterium of this new SA heuristic (SA3) is the same as in SA2 (1000 iterations). Additionally, a new heuristic was written in which the start solution is given by the solution found by the QP heuristic followed by 250 iterations of SA (QP+SA). Here, the evaluation function is again the true objective. The results of these last two tests are given in Table 4. Figure 4 and Figure 5 provide block diagrams with the added computational results.

Table 4: Computational results

Heuristic	Avg. exp. shortage (ETBS)	Avg. comp. time (ms)	St. dev. comp. time (ms)
SA3	11.513	230.773	87.025
QP+SA	6.740	12386.804	4858.314

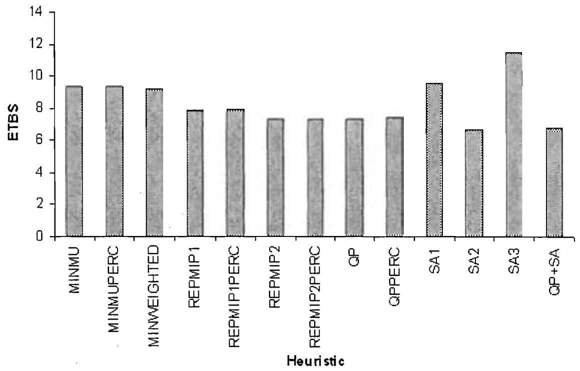


Figure 4: Comparison of new SA heuristics

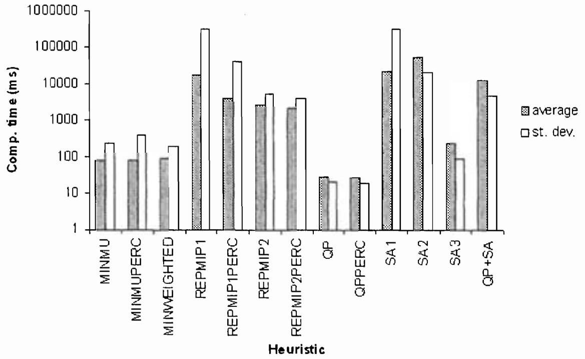


Figure 5: Comparison of computation times of new SA heuristics

From these graphs we can conclude that SA3 runs significantly faster than SA2 (and SA1), however the results are also significantly worse. As a matter of fact, the results of SA3 are amongst the worst overall. Changing the weights in the evaluation function given to the total squared sum of average bed occupations and the total shortage probability did not produce better results. QP+SA however yields almost as good results as SA2 using much smaller computation times.

The overall conclusion is that the best results are obtained by a metaheuristic approach in which the true objective is evaluated. However, MIP approaches involving linearized and/or quadratic objective functions can be useful to find good solutions within small computational effort. The quadratic MIP heuristics turn out to outperform repetitive MIP heuristics with regard to both solution quality and computational effort. When a metaheuristic approach was initiated with the solution found by a QP, good results were obtained in terms of both solution quality and computational effort.

The impact of the different factor settings on the computation time is dependent on the applied heuristic. Table 5 provides the p-values of the different factors for each heuristic. Significant factors ($\alpha = 0.05$) are indicated with a *.

Here we can distinguish between five groups. For the first group, consisting of the single MIP heuristics (MINMU, MINMUPERC and MINWEIGHTED), only the first two factors (the number of blocks per day and the number of surgeons) have a significant (positive) impact on the computation time. Due to the huge variability in computation times, no significant factors could be found for REPMIP1 and REPMIP1PERC. Since SA1 gets the largest computation time of the MIP heuristics, this heuristic obviously also belongs to this second group. REPMIP2 and REPMIP2PERC are situated in a third group for which a third factor becomes significant: the number of patients per surgeon. Also here there is a positive influence on the computation time. For the quadratic MIP heuristics (QP and QPPERC) yet another significant factor is added: the LOS (Length Of Stay of the patients). The influence of this factor is however negative. Hence, the longer the patients stay, the smaller the needed computation time to solve the quadratic program. The fifth group consists of the remaining SA heuristics (SA2, SA3 and SA+QP, in which

Table 5: Impact of factor settings: p-values

Factor	Nr. blocks	Nr.	Division	Nr. patients	Prob.	LOS	Capacity
Heur.	per day	surgeons	req. blocks	per surgeon	no show-up		
MINMU	0.0233*	$< .0001$*	0.2996	0.3956	0.0966	0.7456	0.8820
MINMUPERC	0.0903	0.0069*	0.6391	0.1143	0.2167	0.2060	0.3018
MINWEIGHTED	0.0142*	$< .0001$*	0.3489	0.5962	0.2792	0.1506	0.2724
REPMIP1	0.2945	0.2826	0.3144	0.3118	0.3322	0.2987	0.3093
REPMIP1PERC	0.1018	0.0775	0.1808	0.5891	0.5045	0.1184	0.1239
REPMIP2	$< .0001$*	$< .0001$*	0.3977	0.0286*	0.3931	0.9029	0.1704
REPMIP2PERC	$< .0001$*	$< .0001$*	0.3116	0.0030*	0.4988	0.4152	0.1612
QP	$< .0001$*	$< .0001$*	0.1083	0.0062*	0.6412	0.0186*	0.1918
QPPERC	$< .0001$*	$< .0001$*	0.0776	$< .0001$*	0.6630	0.0011*	0.9245
SA1	0.2174	0.2053	0.2680	0.3859	0.3945	0.2521	0.2500
SA2	$< .0001$*	$< .0001$*	0.0031*	$< .0001$*	0.2587	0.0006*	0.6735
SA3	$< .0001$*	$< .0001$*	0.0066*	0.0204*	0.4934	0.2370	0.7232
QP+SA	$< .0001$*	$< .0001$*	0.0008*	0.0043*	0.1545	0.0014*	0.4748

the end criterium is determined by a fixed number of SA iterations). Here also factor 3 (whether or not the blocks are equally divided over the surgeons) becomes significant. It turns out that SA can solve the problem faster when the blocks are not equally divided, which is no surprising result since the number of possible exchanges is larger when all surgeons are equally represented and hence more evaluations need to be done per iteration. This also explains why this factor is not significant in SA3, for which the computationally expensive evaluation function is replaced with an easily computable one. The probability of a no show-up and the (over)capacity do not play any role in the complexity of the problem, no matter which heuristic is applied.

8.4 Simulation study

Recall that in order to calculate expected shortages, the bed occupation distributions are approached with normal distribution functions (see Section 7.1). Alternatively, the found schedules could have been evaluated using simulation. The reason why this was not done in the computational experiments described earlier is that (reliable) simulation takes too much computation time. However, to verify the accuracy of our results, a simulation experiment was done in which the predicted values (averages, variances and shortages) are compared with simulated values. In this part we summarize the findings of this experiment.

The experiment involved all 384 test instances. Each problem was again solved with the quadratic programming heuristic (QP). For each problem the total average and total variance of the bed occupation (summed up over all 7 days) and total bed shortage resulting from the found schedule are calculated both through the theoretical results as outlined above and obtained through simulation:

1. Predicted values: average and variance are calculated using the theoretical

formulas derived above. Expected shortages are calculated by approaching the bed occupation distributions with normal distributions and applying numerical integration as described above.

2. Simulated values: average, variance and shortages are calculated by simulating 1000 periods, taking into account a warm-up period in order to reach steady-state.

The experiment provided three series (averages, variances and expected shortages) of predicted and simulated data. These series were compared using a paired Student T-test (two-tailed). In the left part of Table 6 the results are given. The extremely small p-values for both the variance and the expected shortage indicate that these predicted values are different from the simulated ones. It turns out that the predicted variances are larger and hence also the predicted shortages are larger than the simulated ones.

Table 6: Predicted versus simulated data

	All 384 instances			Only 192 instances with LOS < cycle time		
	Pred.	Sim.	p-value	Pred.	Sim.	p-value
Avg. bed occupation	967,8501	967,8160	0,4102	651,5265	651,4699	0,1996
Avg. var. bed occupation	170,0203	151,2397	1,56E-26*	130,0206	129,9518	0,7631
Avg. total bed shortage	7,2856	7,1361	7,14E-20*	11,3355	11,3301	0,7979

The reason for this discrepancy is that the theoretical results do not take into account autocorrelation in the data. Indeed, a subset of the patients occupying a bed at period t will also occupy a bed at period $t + 1$, namely those patients that stay longer than a cycle in the hospital. This means that the number of patients in the hospital at period $t+1$ can partly be explained by the number at period t. In other words, both numbers are dependent. When simulating more periods, the difference between numbers of occupied beds of subsequent periods differ less than expected from theoretical results, making the true variance smaller than the predicted one. In order to verify this explanation, the T-tests are repeated, but now only including those instances having the first setting of factor 6 (i.e. with LOS below the cycle time). The results are indicated in the right column of Table 6. As was expected, all p-values are now sufficiently high, indicating that the assumption of a (structural) difference between the predicted and the simulated data can be rejected.

9 Conclusion and future research

The purpose of this paper is to propose and compare models and algorithms for building robust surgery schedules. We concentrate on the development of cyclic master surgery schedules. A distinction is made between elective and non-elective cases producing respectively artificial and natural variability. The objective is to find a schedule for which the total expected bed shortage (from elective cases) is minimized. Since the problem is too complex to solve exactly, we develop a number of heuristics. One can distinguish between two approaches: a MIP based approach and a metaheuristic approach. In the first approach the non-linear objective function is being replaced with a linear (or quadratic) one and the resulting models are solved with a state-of-the art MIP solver. Therefore, theoretical results have been derived for both average and variance of the resulting bed occupation on each day, given a particular surgery schedule. Models have been proposed which aim at the minimization of the highest expected bed occupation peak, highest bed occupation variance or a combination of both. Additionally, a number of repetitive MIP solving algorithms have been developed. The second approach preserves the original objective function and searches a good solution by means of a metaheuristic (simulated annealing) approach. All algorithms have been extensively tested and their results compared. The best solutions are found with the simulated annealing approach. However, this approach also takes the longest computation times. Concerning the MIP based approaches, the best results are obtained with the quadratic programming (QP) models in terms of both solution quality and computation time. A hybrid approach in which a simulated annealing search is initiated with a schedule found by a quadratic program yields satisfying results with regard to both solution quality and computation time. A simulation experiment indicates that, due to autocorrelated data, there was a slight overestimation of both variance and expected shortage in the theoretically developed models.

The developed models are very basic. Only two types of constraints have been considered: surgery demand and OR capacity constraints. For real-life applications a number of additional constraints could be required like e.g. workforce capacity constraints (anaesthetists, nursing staff), surgeons preference constraints (e.g. all blocks at maximal two different days), material requirement constraints, transition constraints (change of equipment from one surgery group to another) etc... It would be interesting to implement these kinds of models for a real-life case in order to see to which extent they can improve existing practices and which model extensions are required to make them practically useful. This is the first and most important item for future research. From a theoretical point of view, it would be interesting to see which extensions could easily be handled by which solution approach and which not. Furthermore, the impact of these extensions on both solution quality and computation time could be researched.

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 Prof. Dr. W. Sermeus and Kris Vanhaecht of the Centrum voor Ziekenhuis- en Verplegingswetenschap (CZV) at the Katholieke Universiteit Leuven for drawing our attention to challenging scheduling problems in health care management.

References

Blake, J. T. & Carter, M. W. (2002). A goal programming approach to strategic resource allocation in acute care hospitals, *European journal of Operations Research* **140**: 541–561.

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.

Bowers, J. & Mould, G. (2002). Concentration and the variability of orthopaedic demand, *Journal of the Operational Research Society* **53**: 203–210.

Bowers, J. & Mould, G. (2004). Managing uncertainty in orthopaedic trauma theatres, *European Journal of Operational Research* **154**: 599–608.

Dexter, F. & Macario, A. (2002). Changing allocations of operating room time from a system based on historical utilization to one where the aim is to schedule as many surgical cases as possible, *Anesthesia and Analgesia* **94**: 1272–1279.

Dexter, F., Macario, A. & O'Neill, L. (2000). Scheduling surgical cases into overflow block time - computer simulation of the effects of scheduling strategies on operating room labor costs, *Anesthesia and Analgesia* **90**: 980–988.

Dexter, F., Macario, A. & Traub, R. D. (1999). Which algorithm for scheduling add-on elective cases maximizes operating room utilization?, *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 and Analgesia* **94**: 933–942.

Dexter, F., Traub, R. D. & Lebowitz, P. (2001). Scheduling a delay between different surgeons' cases in the same operating room on the same day using upper prediction bounds for case durations, *Anesthesia and Analgesia* **92**: 943–946.

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Booth, M. & Rossi, F. (2003). *GNU Scientific Library Reference Manual Edition 1.3*, Network Theory LTd.

Gerchak, Y., Gupta, D. & Mordechai, H. (1996). Reservation planning for elective surgery under uncertain demand for emergency surgery, *Management Science* **42**: 321–334.

Guinet, A. & Chaabane, S. (2003). Operating theatre planning, *Int. J. Production Economics* **85**: 69–81.

Hughes, W. L. & Soliman, S. Y. (1985). Short-term case mix management with linear programming, *Hospital and Health Services Administration* **30**: 52–60.

ILOG (2002). *ILOG CPLEX 8.1 User's Manual.*

Kirkpatrick, S., Gerlatt, C. D. J., & Vecchi, M. (1983). Optimization by simulated annealing, *Science* **220**: 671–680.

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., Kharraja, S. & Simonnet, G. (2003). The operating theatre planning by the follow-up of the risk of no realization, *Int. J. Production Economics* **85**: 83–90.

Metropolis, N., Rosenbluth, A., Rosenbluth, M. N., Teller, A. & Teller, E. (1958). Equations of state calculations by fast computing machines, *J. Chem. Phys.* **21**: 1087–1092.

Pincus, M. (1970). A monte carlo method for the approximate solution of certain types of constrained optimization problems, *Operations Research* **18**: 1225–1228.

Weiss, E. N. (1990). Models for determining estimated start times and case orderings in hospital operating rooms, *IIE Transactions* **22**: 143–150.