Anticipating urgent surgery in operating room departments

Authors:

M. (Marieke) van der Lans
m.vanderlans@student.utwente.nl / m.vanderlans@alumnus.utwente.nl
University of Twente, department of Operational Methods for Production and Logistics

E.W. (Erwin) Hans
e.w.hans@utwente.nl
University of Twente, department of Operational Methods for Production and Logistics

J.L. (Johann) Hurink
j.l.hurink@utwente.nl
University of Twente, department of Discrete Mathematics and Mathematical Programming

G. (Gerhard) Wullink,
Research coordinator process optimization, Erasmus MC
g.wullink@erasmusmc.nl

M. (Mark) van Houdenhoven
Division manager OR, IC and Anesthesiology, Erasmus MC
m.vanhoudenhoven@erasmusmc.nl

G. (Geert) Kazemier
Head of operating theatre department, Surgeon, Erasmus MC
g.kazemier@erasmusmc.nl

Abstract

Operating Room (OR) departments need to create robust surgical schedules that anticipate urgent surgery, while minimizing urgent surgery waiting time and overtime, and maximizing utilization. We consider two levels of planning and control to anticipate urgent surgery. At the tactical level, we study the allocation of slack for urgent surgery to one or more operating rooms, and at operational off-line level, we experiment with the sequencing of elective surgeries in the operating rooms to which slack for urgent surgery is allocated. We try to sequence the elective surgeries such that their completion times, which are break-in-moments (BIMs) for urgent surgery, are spread as equally as possible over the day. We refer to this problem as BIM optimization problem, which is NP-hard in the strong sense. In this paper, we develop and test various heuristics for this sequencing problem. By means of a simulation study, we compare five methods of anticipating urgent surgery: (1) concentrating slack for urgent surgery in a dedicated operating room, (2) allocating slack for urgent surgery to a subset of the operating rooms without BIM optimization and (3) with BIM optimization, and (4) allocating slack for urgent surgery to all operating rooms without BIM optimization, and (5) with BIM optimization. For

the test instances, the computational experiments show that urgent surgery can be anticipated best by allocating slack for urgent surgery to all available operating rooms, and thus allowing urgent surgeries to interfere with the schedule of elective surgeries. Further savings in urgent surgery waiting time can be achieved by BIM optimization, especially for the first urgent surgical cases that arrive during a day.

1. Introduction

In 2003, OECD countries spent on average 8.8% of their GDP on health care, up from 7.1% in 1990 and just over 5% in 1970. Major factors for this growth are improved methods to prevent, diagnose, and treat health conditions, as well as the ageing population (OECD, 2005). In this context, hospitals face an increasing public pressure for high quality care and cost effectiveness. Since 60-70% of all hospital admissions are caused by surgical interventions, the Operating Room (OR) department is one of the key hospital resources and should continuously strive to enhance quality and lower cost. Operations Research techniques can contribute to this process (Luss and Rosenwein, 1997). One of the fields in the OR department, where Operations Research can be applied to, is OR surgical scheduling, see, e.g. Dexter *et al.* (1999b), Guinet and Chabaane (2003), Hans *et al.* (2006) and Ozkarahan (2000).

OR surgical scheduling is one of the challenging issues for the management of the OR department. First, multiple stakeholders with conflicting interests are involved (Glouberman and Mintzberg, 2001), such as surgeons of various specialties, anesthetists, OR personnel, and, naturally, the patients that require surgery. Second, OR surgical scheduling is complex because of the uncertainty regarding the occurrence, timing and duration of surgeries. Third, the OR department faces conflicting performance criteria: schedules with a high planned utilization may lead to overtime, cancelled surgeries and long waiting times for urgent surgery.

The arrival of urgent surgical cases is one of the uncertainties that occur during the execution of surgical schedules. Urgent surgeries have to be scheduled as soon as possible to avoid medical complications or mortality. Unlike ‘elective surgeries’ that are planned in advance, we use the term ‘urgent surgeries’ for surgeries that are not scheduled in advance and turns up unexpectedly on the day of surgery (based on Fitzgerald *et al*., 2005).

OR departments can anticipate urgent surgery by reserving OR capacity: hours of staffed operating rooms in which no elective surgery is planned. This free capacity or so-called planned ‘slack’ may be concentrated in one or more operating rooms that are entirely dedicated for urgent surgery. However, these ‘dedicated rooms’ tend to result in a low utilization of the OR capacity and thus high costs (Barlow *et al*., 1993; Brasel *et al*., 1998) and the dedicated room is not immediately available if another urgent surgery is taking place in this room. Another option is to allocate the slack to a number of operating rooms that are to be used for elective surgeries, allowing urgent surgeries to be scheduled in between two elective surgeries. Although this may lead to higher OR utilization, elective surgeries may have to be cancelled due to disruptions in the surgical schedule. Besides, an urgent surgery can only be started if an ongoing elective surgery is finished, since surgeries generally cannot be interrupted. To our knowledge, this second option is not explored in the literature so far, although it is used in practice by various OR departments, for example in the Erasmus MC, the Netherlands.

If slack for urgent surgery is allocated to operating rooms with elective surgery, urgent surgery waiting time may be further decreased by optimizing the sequence of the elective surgeries in the operating rooms assigned to deal with urgent surgery. The sequence should be such, that the so-called 'break-in-moments' (BIMs), the moments that an elective surgery is expected to finish and a new urgent surgery may thus be started, are distributed as evenly as possible over the day. Sequencing methods that aim to minimize the waiting time of urgent surgery seems to be an unexplored field. Lebowitz (2003) proposes sequencing methods as well, but thereby focuses on reducing overtime.

In this paper, we examine the various options for anticipating urgent surgery that are discussed above. First, we concentrate on the sequencing aspect, to which we refer as the 'break-in-moment' (BIM) optimization problem, and we develop various off-line heuristics for this problem. We then use a simulation environment to test which option anticipates urgent surgery best with respect to urgent surgery waiting time, utilization and overtime: either a dedicated operating room for urgent surgery or slack for urgent surgery allocated to several operating rooms. For the latter, we test whether BIM optimization can reduce urgent surgery waiting time.

This paper is organized as follows: Section 2 presents an overview of relevant literature on surgical scheduling methods that anticipate urgent surgery. Section 3 gives the formal problem definition of the BIM optimization problem and proves that the problem is NP-hard in the strong sense. In Section 4 we propose various constructive and improvement heuristics to address the problem. Section 5 describes the computational experiments with these heuristics. Section 6 describes the simulation study, and Section 7 concludes this paper.

2. Literature

To describe the research on surgical scheduling that explicitly deals with the unexpected arrival of urgent surgical cases, we distinguish between four hierarchical levels of hospital planning and control: the strategic, tactical, operational off-line and operational on-line level (Van Houdenhoven *et al.*, 2006). The decisions taken by the hospital's board of directors on the long-term *strategic* level influence the amount and variety of urgent surgeries. On the medium-term *tactical* level, the amount of slack for urgent surgery is determined and allocated to the various operating rooms. For the short-term planning horizon, we distinguish between operational off-line and operational on-line level. On the *operational off-line* level, robust surgical schedules are created. The *operational on-line* level concerns the real time rescheduling due to the arrival of urgent surgeries and other disturbances. In the following subsections, we give an overview of previous work on relevant scheduling methods on the tactical, operational off-line, and operational on-line level. Altogether, these methods should contribute to robust schedules: schedules which, when implemented, minimize the effect of disruptions on the primary performance measures of the schedule (Aytug *et al*., 2005).

Tactical level

To deal with uncertainty during execution of a schedule, a buffer of extra time and/or resources (so-called *slack*) can be planned (Davenport and Beck, 2000; Davenport *et al*., 2001; Herroelen and Leus, 2005; Wullink *et al*., 2005). Quantitative research regarding the exact amount of required slack for urgent surgery in an OR surgical schedule is scarce. Discussion on this subject is rather complicated, as it is not only about cost and efficiency figures, but also about the level of medical care and personal preferences: "though utilization of an urgent room will rarely be more than 50%, the political goodwill you will buy is probably worth it" (Dexter, 2005). Three approaches for the allocation of slack for urgent surgery can be distinguished (Boer, 2006): (1) no time or operating rooms are reserved for urgent surgery, (2) dedicated operating rooms are reserved for urgent surgery, and (3) extra time is reserved for urgent surgery in each or some of the operating rooms of a specialty, so that urgent surgeries can be performed during or after the schedule of elective surgeries. According to an OR benchmarking study in the academic hospitals in the Netherlands (Van Houdenhoven *et al*., 2005), six out of eight considered hospitals choose for option 2: they use one or more dedicated rooms for urgent surgery. Lovett and Katchburian (1999) emphasize that dedicated operating rooms can help to the decrease overtime and the number of urgent surgeries served after working hours. However, Barlow *et al.* (1993) and Brasel *et al.* (1998) conclude that this is costly, because of low utilization rates. To our knowledge, studies that compare the different approaches for the allocation of slack for urgent surgery based on identical surgical case mixes are lacking. This study aims to fill this gap in literature.

Operational off-line level

On the *operational off-line* level elective surgeries are assigned to operating rooms. Methods for surgical scheduling focus on maximizing expected OR utilization and/or minimizing expected overtime, and thus cancelled elective surgeries (Dexter *et al*., 1999b; Hans *et al*., 2006; Guinet and Chaabane, 2003; Pham and Klinkert, 2005). Some approaches also address other issues, such as intensive care capacities and surgeon satisfaction (Ozkarahan, 2000). Gerchak *et al.* (1996) and Lamiri *et al.* (2005) develop methods that, besides scheduling elective surgeries, schedule individual urgent surgeries as well. Though a substantial work on surgical scheduling has appeared in literature, mostly based on Operations Research techniques commonly used in the production literature, none of the researched work addresses optimizing the sequence of the elective surgeries in order to minimize urgent surgery waiting time. Both Lebowitz (2003) and Dexter and Traub (2000) are concerned with the sequencing of surgeries, but focus on other objectives. Lebowitz claims that scheduling short procedures first can improve on-time performance and decrease staff member overtime expense without reducing surgical throughput, while Dexter and Traub use statistical decision theory to sequence surgeries to decrease the impact of limitations in equipment or personnel on surgical scheduling.

Operational on-line level

Although this paper does not address queuing and scheduling methods for urgent surgery at operational on-line level, for completeness, we mention literature findings regarding this level as well. Fitzgerald *et al.* (2005) conclude that clear objectives and procedures for organizing queues of urgent surgeries are lacking in most of the considered OR departments. Dexter *et al.* (1999a) proposes to queue urgent surgeries (1) in increasing order of expected surgery durations, (2) first come first serve, or (3) based on urgency to minimize the chance of a poor patient outcome. Regarding the scheduling of the urgent surgery, and the invoked rescheduling of elective surgeries, there are three common approaches: *right shift rescheduling*, *partial rescheduling* and *regeneration or reoptimization* (Vieira *et al*., 2003). Right shift rescheduling, that postpones all remaining elective surgeries in case an urgent surgery comes in, is the easiest methods and is therefore probably the most common method used in OR departments. Dexter *et al.* (2005) use this approach for various on-line algorithms for surgical scheduling *add-on elective cases* that need to be added to the schedule at execution time. Dexter (2000) applies partial rescheduling in deciding whether to move the last surgery of the day in an operating room to another operating room to decrease overtime labor costs, but does not address the scheduling of urgent surgery explicitly. On-line scheduling of urgent surgery and thereby rescheduling the elective surgeries remains an interesting field for further research.

3. Problem description

This section explores the break-in-moment (BIM) optimization problem in more detail. We first discuss the problem context and objective, and then we give a formal problem description. Finally, we proof that the problem is NP-hard in the strong sense.

Context description and objective

For the BIM optimization problem, we focus on sequencing the elective surgeries in the operating rooms to which a set of elective surgeries *and* slack for urgent surgery is assigned. In these operating rooms, it is allowed to schedule urgent surgeries during or after the schedule of elective surgeries. We define the interval in which all these operating rooms are in use for elective surgeries as the ‘occupied interval' (see Figure 1). During the occupied interval, an urgent surgery can only be started up upon the completion of another ongoing (elective) surgery, since it is not possible to interrupt ongoing surgery. We define all moments that can be used to start up urgent surgeries during the occupied interval as a 'break-in-moments' (BIMs). These moments include the start and end of the occupied interval, as well as all completion times of surgeries within the occupied interval. We define the interval in between two subsequent BIMs as a break-in-interval (BII). Figure 1 visualizes the occupied interval, the BIMs and BIIs for a simplified situation with two operating rooms.

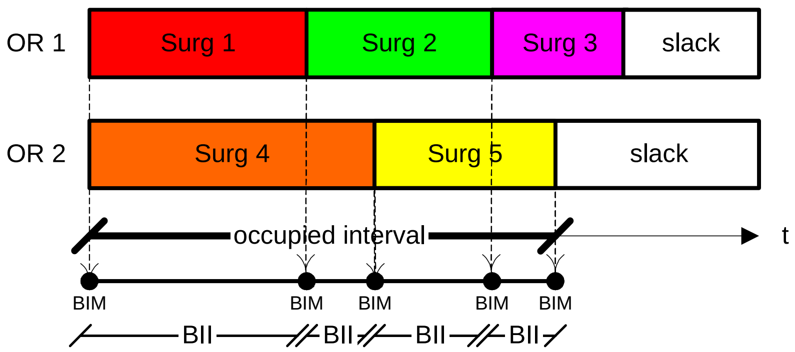

Figure 1 **BIM, BII and occupied interval**

The timing of the BIMs, and thus the length of the BIIs, depends on the sequence of the elective surgeries in the operating rooms. In practice, BIMs appear to be scheduled unevenly over the occupied interval, which results in relative long BIIs that increase the expected waiting time for urgent surgery and thus medical complications or mortality. In this paper, we aim to minimize expected waiting time for urgent surgery, by creating a schedule in which surgeries are sequenced in such way that the BIMs are spread as evenly as possible over the day. We do so by minimizing the maximum BII per day. We refer to this specific problem as the BIM optimization problem.

Formal problem description

This subsection gives the formal problem description of the BIM optimization problem for a *specific day*. Let $J = \{1,\ldots, N\}$ be the set of available operating rooms. An operating room is indexed by *j*. In these operating rooms *J* as set $I = \{1,\ldots,M\}$ of surgeries has to be performed. Surgery $i \in I$ has an expected duration p_i. Let $I_j = \{j_1,\ldots, j_{M_j}\} \subset I$ be the set of surgeries that have to be performed in operating room *j*. Since the sets I_j are a decomposition of the set I, we have $M = \sum_{j \in J} M_j$.

We assume that all surgeries in an operating room are scheduled successively with no breaks in between, as depicted in Figure 1. S_j denotes the moment that the first surgery $i \in I_j$ is allowed to start in operating room *j*, and E_j denotes the moment that all surgeries in operating room *j* are expected to be completed, i.e. $E_j = S_j + \sum_{i \in I_j} P_i$, for all *j=1,…N*. Let *O* be the occupied interval $O = [S, E]$, with $S = \max_j S_j$ and $E = \min_j E_j$

To describe the sequence of the surgeries in an operating room, we define Π^j as a permutation of I_j. Π^j_k denotes the k^{th} surgery in Π^j. It may occur that the position of a surgery is fixed at the first of the last position of the day because of diabetis, infections etcetera, which decreases the number of possible permutations Π^j. Let $C_{\Pi^j_k}$ be the completion time of surgery Π^j_k in permutation Π^j. It can be calculated as follows: $C_{\Pi^j_k} = S_j + \sum_{m=1}^{k} P_{\Pi^j_m}$ for all *k=1,…,M*$_j$. A set $\Pi = (\Pi^1,...,\Pi^N)$, including one permutation Π^j per operating room *j*, defines a schedule of the surgeries *I* in the operating rooms *J*.

To describe the maximum BII in the occupied interval *O* for a given schedule $\Pi = (\Pi^1,...,\Pi^N)$, let Ω_Π be the set of break-in-moments in $\Pi = \{\Pi^1,...,\Pi^N\}$ within the occupied interval *O*. This set includes the start of the occupied interval *S*, as well as all completion times of surgeries within the occupied interval. Thus, Ω_Π is given by $\{S\} \cup \{C_{\Pi^j_k} \mid S_j < C_{\Pi^j_k} \le E,\ \ j = 1,...N;\ \ k = 1,..., M_j\}$. Let Ω_Π^{sorted} be the sorted list of Ω_Π, ordered in non-descending order. The t^{th} element in Ω_Π^{sorted}, $\Omega_\Pi^{sorted(t)}$, is the t^{th} break-in-moment that we refer to as BIM_t. Let BII_t be the difference between two subsequent BIMs: i.e. $BII_t = BIM_t - BIM_{t-1}$, $t > 1$. The objective 'minimize the maximum break-in-interval' can now be stated as $Min\ \ Max_t BII_t$.

Problem complexity

The following theorem shows that the BIM optimization problem is NP-hard in the strong sense and that it thus is unlikely to find efficient (polynomial) solution methods which solve the problem to optimality.

Theorem: The BIM optimization problem is strongly NP-hard for two or more operating rooms.

Proof: We prove the theorem by reducing the 3-partition problem to the BIM optimization problem. The 3-partition problem is given by *3t* positive integer values $a_1,...,a_{3t}$ and a value *B* such that $\sum_{j=1}^{3t} a_j = tB$. The problem is to decide whether or not a partition of $I = \{a_1,...,a_{3t}\}$ in *t* sets

$I_1,...,I_t$ with $\sum_{j \in I_n} a_j = B$ for all $n = 1,...,t$ exists. The 3-partition problem is proven to be strongly NP-hard (Garey and Johnson, 1979).

The instance of the BIM optimization problem corresponding to a general instance of 3-partition problem has 2 operating rooms, where OR 1 has *t-1* surgeries with a length of $4B$ each, and 3t surgeries with length $a_1,...,a_{3t}$. OR 2 has 2 surgeries, each with a length of $3B$, and *t-2* surgeries with a length of $5B$ each. The question is whether this instance of the BIM optimization problem has a solution where all BIIs are smaller or equal to $2B$.

It is not possible to cover the surgeries of length $4B$ in OR 1 with BIIs smaller than $2B$, as none of the surgeries in OR 2 is smaller than $2B$ (see Figure 2). Therefore $2B$ is a lower bound on $Min\ Max_t BII_t$ in every solution of the instance.

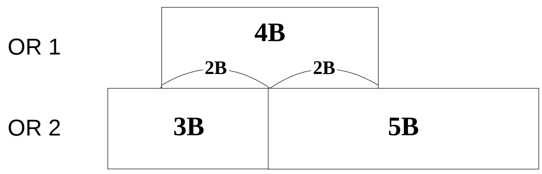

Figure 2 *2B* is a lower bound on $Min\ Max_t BII_t$

In the following, we show that a solution of the 3-partition problem exists *if and only if* the BIM optimization problem has an optimal value $Min\ Max_t BII_t$ = *2B*. First, we prove the *if* part. If an optimal solution for the 3-partition problem exists, we can create an optimal schedule for the instance, as depicted in Figure 3. This schedule consists of *t* groups of surgeries with total length B and results in a solution of the BIM optimization problem with $Min\ Max_t BII_t$ = *2B*. The *t* groups of surgeries with total length B (shaded blocks in Figure 3) correspond to the partitions in the solution of the 3-partition problem.

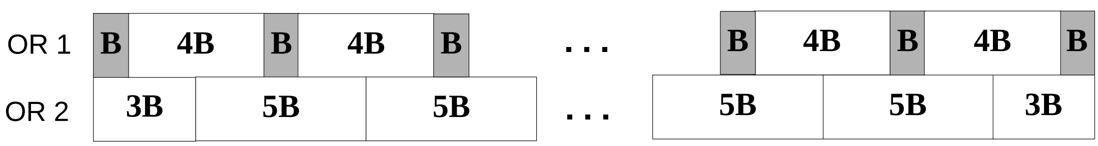

Figure 3 An optimal schedule for the instance

We now prove the *only if* part, and show that when *2B is* the optimal objective value, all optimal solutions have the structure as depicted in Figure 3. OR 1 contains *t-1* surgeries of length $4B$ each. These *t-1* surgeries all need to be covered by two equal BIIs, in order to obtain $Min\ Max_t BII_t = 2B$. OR 2 contains *t* surgeries, and thus *t-1* moments where one surgery finishes and the next surgery starts. Each of these *t-1* moments must be scheduled exactly in the middle of the surgeries of length $4B$ in OR1 to obtain $Min\ Max_t BII_t = 2B$. When applied to a surgery of length $5B$ in OR 2, this means that a 'gap' of length B occurs in between the two surgeries of length *4B* in OR 1 (see Figure 4).

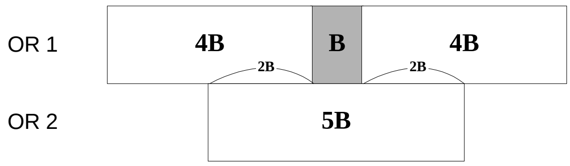

Figure 4 A surgery of length *B* must be scheduled in between two surgeries of length *2B*

When the two smaller surgeries of $3B$ in OR 2 are scheduled as first and as last surgery in OR 2, a 'gap' of length B occurs before the first surgery of $4B$ in OR 1, as well as after the last surgery in OR 1. Note that this is the only way to cover all *t-1* surgeries of length *4B* with two equal BIIs. Thus, we require exactly *t* groups of surgeries, so that a group with length *B* is placed before and after each surgery with a length of $4B$ in OR 1. This results in a solution of the 3-partition problem. Thus, the BIM optimization problem is strongly NP-hard for two or more operating rooms. ■

4. Solution approaches

In this section we propose solution approaches for the BIM optimization problem. We propose both constructive heuristics that create initial schedules, and improvement heuristics that iteratively improve initial schedules.

Constructive heuristics

We consider three constructive heuristics that sequence the set of surgeries I_j for each operating room *j* at a given day. The first heuristic is known as the Shortest Processing Time (SPT) heuristic and schedules the surgeries from I_j in increasing order of their expected durations. Furthermore we propose two new constructive heuristics, C1 and C2, which aim to sequence the surgeries such that every BII approaches lower bound λ, with $\lambda = \frac{E - S}{1 + \sum_{j \in J}(M_j - 1)}$. This lower bound reflects the distance between two subsequent BIMs if all surgeries could be completed within in the occupied interval *O*, and all BIMs would be distributed evenly in *O* so that each BII would be of equal length λ. In the following subsections, we explain C1 and C2 in more detail.

Heuristic C1

Heuristic C1 alternatively schedules surgeries forward and backward, trying to avoid large BIIs either at the beginning or at the end of the day. By forward scheduling, we mean the scheduling of surgeries in an operating room one after another from the start of the day (S_j) towards the end of the day (E_j), while the reverse holds for backward scheduling. Backward scheduling is possible, since the number and the durations of surgeries in an operating room are known, and thus the completion times per operating room. The heuristic aims to schedule surgeries such that the interval between subsequent completion (starting) times of the forward (backward) scheduled jobs approach λ.The heuristic proceeds as follows:

- Step 0: Calculate λ;
- Step 1: Forward scheduling move. Select the unscheduled surgery from one of the operating rooms *j* for which the completion time will be closest to the latest completion time of all already forward scheduled surgeries plus λ, and schedule this surgery forward in operating room *j*. If no surgeries are scheduled forward so far, select the unscheduled surgery from one of the operating rooms *j* for which the completion time will be closest to the latest starting time of all operating rooms plus λ (=*S*+ λ) and schedule this surgery forward in operating room *j;*
- Step 2: Backward scheduling move. Select the unscheduled surgery from one of the operating rooms *j* for which the starting time will be closest to the earliest starting time of all already backward scheduled surgeries minus λ, and schedule this surgery backward in operating room *j*.

  If no surgeries are scheduled backward so far, select the unscheduled surgery from one of the operating rooms *j* for which the starting time will be closest to the earliest closing time of all operating rooms minus λ (=*E*- λ)
- Step 3: Repeat step 1 and 2 until all surgeries are scheduled.

Figure 5 visualizes the decisions to be made for the second iteration of step 1 and step 2, for a simplified situation with two operating rooms.

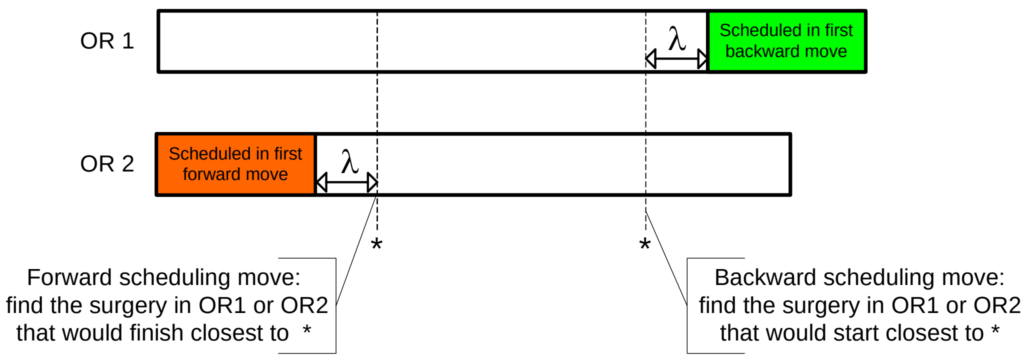

Figure 5 Scheduling according to C1

Heuristic C2

Heuristic C2 also strives every BII to approach λ. Heuristic C2 is based on the SPT heuristic, and schedules one whole set of surgeries I_j in an operating room at a time, starting with the operating room with the highest number of surgeries, and ending with the operating rooms with the lowest number of surgeries. We iteratively schedule the surgeries for the remaining operating rooms. If possible, we try to avoid scheduling a surgery of which the completion time would be too close (< ½ λ) to S and one of the completion times of the already scheduled surgeries. The heuristic proceeds as follows:

- Step 0: Calculate λ
- Step 1: Select the operating room *j* with the highest number of surgeries. Schedule the surgeries in this operating room *j* according to the SPT rule
- Step 2: Select the operating room *j* with the highest number of surgeries out of the remaining operating rooms of which the surgeries are not scheduled so far. Schedule the surgeries I_j in this room *j* according to the SPT rule, *unless* this would lead to scheduling a surgery of which the completion time is within ½ λ of S or the completion times of one of the already scheduled surgeries in one of the operating rooms. In this case, schedule the first surgery in the SPT list of this operating room for which holds that the completion time is not within ½ λ of S or the completion times of one of the already scheduled surgeries in one of the operating rooms. If none of the remaining surgeries fulfills this requirement, schedule the surgery with the largest absolute difference between the completion time of this surgery and S or the nearest completion times of one of the already scheduled surgeries.
- Step 3: Repeat step 2 until all surgeries in all operating rooms are scheduled.

Figure 6 visualizes the decisions to be made for scheduling the first surgery $i \in I_2$ in operating room 2, for a simplified situation with two operating rooms.

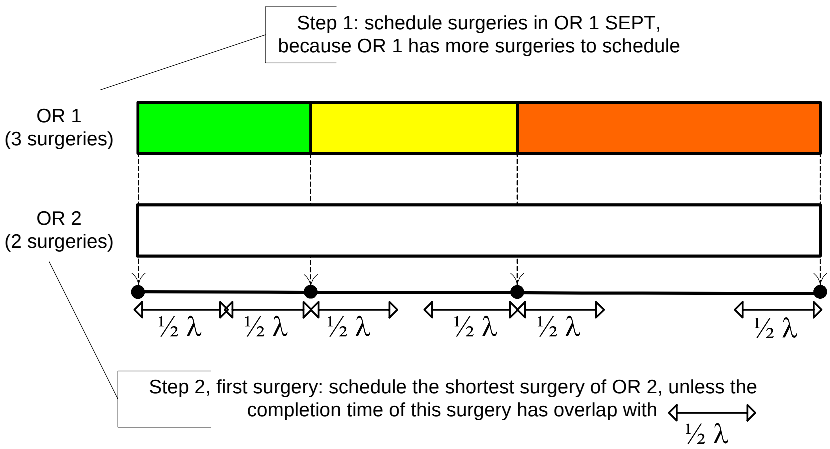

Figure 6 Scheduling according to C2

The heuristics explained above are few of many possible constructive heuristics that aim to minimize the maximum BII of a schedule. We now continue with improvement heuristics.

Improvement heuristics

Once an initial schedule has been created by a constructive heuristic, we try to improve this solution with an improvement heuristic. We propose four heuristics that make use of the 2-change neighborhood structure of the problem (Lin, 1965). This structure defines for each schedule a neighborhood consisting of the schedules that can be obtained from the given schedule by exchanging two surgeries from one single operating room. Notice that it is not allowed changing surgeries from two different operating rooms. Three of the heuristics, to which we refer as L1, L2 and L3, are based on the steepest descent method, while the fourth heuristic is based on the Simulated Annealing (SA) algorithm.

Steepest descent methods

For a given schedule, the steepest descent method (Luenberger, 1969) evaluates all 2-changes allowed. The 2-change that improves the schedule most is then carried out, after which the procedure is repeated for the new schedule. This is repeated until no better solution can be found anymore. Heuristic L1, L2 and L3 allow the following 2-changes:

- Heuristic L1: all possible 2-changes are allowed between 2 surgeries in the same operating room;
- Heuristic L2: only these 2-changes are allowed that involve a surgery with overlap with $\max BII_t$ and another surgery from the same operating room.
- Heuristic L3: only these 2-changes are allowed that involve a surgery that is scheduled *before*, and a surgery that is scheduled *after* the surgery with overlap with $\max BII_t$ in the same operating room. However, if the surgery with overlap with $\max BII_t$ is respectively the first or the last

surgery of the day, the surgery with overlap with $\max BII_t$ is allowed to be involved in the 2-change, as well as a surgery that respectively takes place after or before this surgery.

Simulated Annealing

Finally, we consider a fourth local search technique, i.e. Simulated Annealing (SA). SA (Kirkpatrick *et al*., 1983) has proven to be a successful and widely used algorithm for combinatorial problems. For an extensive description of the SA algorithm we refer to (Aarts and Korst, 1989). For the SA algorithm, we allow all possible 2-changes.

5. Off-line computational experiments

This section presents the results of the *off-line* computational experiments of the heuristics described in Section 4. With the off-line experiments, set up with the Borland Delphi 7 programming language, we create schedules of elective surgeries for which we minimize max BII_t. In the following, we first describe the instance generation and the test approach. The section continues with specifying the parameter settings of the SA heuristic. We then discuss the computational results of the various heuristics, and conclude with a detailed analysis of the results of the best heuristic.

Instance generation and test approach

Let an *instance* represent one day, comprising a given number of operating rooms with a capacity of 7.5 hours each, and a set of elective surgeries assigned to each operating room. This set of surgeries is found by the "First Fit" based algorithm proposed by Hans *et al.* (2006). The algorithm basically assigns surgeries from the top of the waiting list to an operating room plus an amount of slack to avoid for overtime caused by surgery duration variability with 69% probability, until no surgery can be found anymore that fits in the remaining capacity of the operating room.

We distinguish 12 *instance types*, characterized by a unique combination of values for three parameters: the number of operating rooms, the surgical case mix, and the so-called surgery flexibility. The number of operating rooms is 4, 8, or 12, while the surgical case mix (see Table 1) contains either many surgery types with a relative long and uncertain duration, or few surgery types with a relative short and certain duration. These data are derived from 10 years data on surgeries carried out in the OR department for inpatients in the Erasmus MC, a large academic hospital in the Netherlands.

Table 1 Surgical case mix

	Surgical case mix A	Surgical case mix B

Number of surgery types	328	172
Average duration of surgeries in minutes	142.1	103.2
Average standard deviation of all surgeries	45.1	24.1

The third parameter concerns the surgery flexibility regarding the sequencing of surgeries. In case of 100% flexibility, all possible surgery sequences per operating room are allowed. In case of restricted flexibility, with a probability of $\frac{1}{12}N$ one random surgery per day is fixed at the first position of its operating room, with N representing the number of operating rooms per day. With the same probability one random surgery per day is fixed at the last position of its operating room (Van Kempen, 2005). Table 2 summarizes the parameters and parameter values for the instance types. For each instance type, we test 2600 instances, representing a period of 10 years of 52 weeks, with 5 days per week.

Table 2 Parameters and parameter values for the instances types.

Surgical case mix	$c \in \{A, B\}$
Number of operating rooms	$N \in \{4, 8, 12\}$
Surgery flexibility	$f \in$ {100% flexibility, restricted flexibility}

To test the heuristics, for each instance we create a schedule with SPT, C1, C2, and we improve the best of these schedules with L1, L2, L3 and SA. Heuristics will be compared with respect to the average realization of $Min\ Max_t BII_t$ and the computational time. We want to prevent that the improvement heuristics stop optimizing when the maximum break-in-interval equals the length of the day's shortest surgery *and* occurs in the first break-in-interval. Therefore, for all instances, we force the shortest surgery of the day at the first position of its operating room and we do not consider the first interval when calculating $Min\ Max_t BII_t$.

Parameter settings for SA

We base the settings for SA on preliminary results of 260 instances of the instance type with the largest neighborhood structure: 12 operating rooms, surgical case mix A, and 100% surgery flexibility. The start temperature should be such that the chance that a 2-change that results in a worse solution is accepted, approaches 80% (Busetti, 2003). This yields a start temperature of 0.2. The length of the Markov chain should ideally equal the size of the largest neighborhood structure (Aarts and Korst, 1989). For the BIM optimization problem, the size of the neighborhood is given by

$\sum_j \binom{|I_j|}{2}$ possible 2-changes, which for the relevant instance type approximates 150 possible 2-exchanges. Finally, we set the final temperature at 0.001, and test various values for the decrease factor, that typically lies between 0.8 and 0.99 (Aarts and Korst, 1989). We choose a decrease factor value of 0.8. Even though $Min\ Max_t BII_t$ still slightly decreases as the decrease factor increases (see Table 3), we consider the accompanying computation times too high.

Table 3 Parameter settings for SA(C2) algorithm with a start temperature of 0.2, a final temperature of 0.0001 and a length of the Markov chain of 150

Decrease factor	Average value of $Min\ Max_t BII_t$ in minutes per day	Average of computational time in seconds per instance
0.8	17.52	8.57
0.85	17.23	11.52
0.90	16.98	17.87
0.95	16.58	36.53

Test results

Table 4 shows the results of the computational experiments, regarding the realization of the maximum interval BII_t per day and the computational time. C2 is the best constructing heuristic, while SA based on C2 yields the best results of all heuristics at the cost of a higher computational time. Note that the computational times in Table 4 are averages: the computational time depends on the number of surgeries in the instance, which is related to the number of operating rooms and the surgical case mix of the hospital.

Table 4 Computational results of off-line experiments

Heuristic	Average of $Min\ Max_t BII_t$ in minutes per day	Average of computational time in seconds per instance
SPT	67.02	0.0002
C1	64.17	0.0002
C2	48.85	0.0003

L3 (C2)	42.58	0.0386
L2 (C2)	41.62	0.0127
L2 (C2)	39.90	0.0179
SA (C2)	37.13	1.7035

Analysis

Table 5 shows the performance of the SA (C2) heuristic in more detail, by comparing it to the performance of schedules found by the randomized First Fit based algorithm. We evaluate the output in terms of the average frequency that a break-in-interval larger than 90, 75, 60, 45, 30, and 15 minutes occurs in the instance, as we expect these frequencies to be related to the expected waiting time of urgent surgery. Table 5 shows that the change in frequency is significant. For the three largest intervals measured (90,75 and 60 minutes), the frequency decreases on average 88%. Note that the change in frequency depends on the length of the measured interval and the instance: the larger the interval, and the smaller the instance in terms of the number of surgeries in an instance, the higher the relative decrease of the frequency. The restricted surgery flexibility has a minor influence on the results.

Thus, computational results show that by a heuristic approach break-in-moments can be spread more evenly over the day than is the case for a random surgery sequence, which leads to the elimination of large break-in-interval during the day that are responsible for a long waiting times for urgent surgery. SA(C2) appears to be the best performing heuristic.

Table 5 Computational results of SA (C2): average frequency of interval with random solution compared to (→) the average frequency of interval with SA(C2) solution, and the relative change in frequency

Case mix	# ORs	Restricted flexibility?	>90 min	>75 min	>60 min	>45 min	>30 min	>15 min
A	4	No	1.005→0.294	1.505→0 .670	2.012→1.499	2.717→2.835	4.086→5.524	5.508→7.113
			-70%	-56%	-25%	4%	35%	29%
A	4	Yes	1.006→ 0.300	1.505→ 0.743	2.003→1.593	2.723→2.849	4.095→5.472	5.551→7.005
			-70%	51%	-20%	5%	34%	26%
A	8	No	0.481→0.003	0.820→ 0.013	1.212→ 0.093	1.965→ 0.458	3.843→3.751	6.888→10.223
			-99%	-98%	-92%	-77%	-2%	48%
A	8	Yes	0.469→ 0.001	0.819→ 0.017	1.225→ 0.108	1.940→ 0.556	3.819→3.907	6.882-→10.017
			-100%	-98%	-91%	-71%	2%	46%
A	12	No	0.331→0	0.692→0.002	0.954→0.019	1.473→0.107	3.142→1.351	6.973→10.488
			-100%	-100%	-98%	-93%	-57%	50%
A	12	Yes	0.357→0	0.697→0.001	0.952→0.022	1.456-0.128	3.148-1.575	6.924→10.263
			-100%	-100%	-98%	-91%	-50%	48%
B	4	No	0.287→0.018	0.633→0.065	1.243→0.306	2.128→1.084	4.9→4.775	8.284→10.86
			-94%	-90%	-75%	-49%	-3%	31%
B	4	Yes	0.295→0.024	0.64→0.073	1.233→0.327	2.112→1.127	4.933→4.781	8.249→-10.806
			-92%	-89%	-74%	-47%	-3%	31%
B	8	No	0.018→0	0.119→0	0.354→0	0.732→0.008	3.115→0.43	8.735→11.349
			-100%	-100%	-100%	-99%	-86%	30%
B	8	Yes	0.02→0	0.116→0	0.35→-0.002	0.723→0.007	3.122→0.463	8.725→11.344
			-100%	-100%	-100%	-99%	-85%	30%
B	12	No	0.006→0	0.059→0	0.240→0	0.441→0.002	1.995→0.147	7.521→5.842
			-100%	-100%	-100%	-100%	-93%	-22%
B	12	Yes	0.003→0	0.055→0	0.222→0	0.409→0.002	1.951→0.155	7.528→6.366
			-100%	-100%	-100%	-100%	-92%	-15%

6. On-line simulation experiments

This section presents the results of the *on-line* simulation experiments. We examine which method is best at anticipating urgent surgery with respect to urgent surgery waiting time, utilization and overtime: either a dedicated operating room for urgent surgery or slack for urgent surgery allocated to several operating rooms. For the latter method, we test the situation in which the best BIM optimization heuristic, SA(C2), determines the surgery sequence, and the situation without BIM optimization. We first discuss the instance generation and the simulation model. The remainder of the section concerns the results of the simulation study.

Instance generation

Let an *instance* represent one day, comprising 12 operating rooms with a capacity of 7.5 hours each, and a set of elective surgeries assigned to each operating room found by the First Fit based algorithm. We assume restricted surgery flexibility, as explained in Section 5. We distinguish 10 *instance types*, characterized by a unique combination of values for the three parameters presented in Table 6. Each test instance is either based on the surgical case mix A or surgical case mix B, as explained in Section 5. The number of urgent surgeries per day differs per case mix, as shown in Table 7. The slack required for urgent surgery is based on the total expected capacity required for urgent surgery per day, and thus differs per case mix. The second parameter concerns the method for the allocation of slack for urgent surgery for an instance. Notice that besides slack for urgent surgery, we also schedule slack for the variability of surgery duration, as explained in Section 5. Slack for urgent surgery is either concentrated in dedicated operating room(s) or scheduled at the end of the day in 4 or 12 operating rooms, equally distributed. The remainder of the capacity of these operating rooms is filled with elective surgeries and its slack for the variability of surgery durations. Slack for urgent surgery is always scheduled at the end of the day. In case of a dedicated operating room, slack for urgent surgery is concentrated in $y = \left\lceil \frac{total\ \ slack\ \ per\ \ day}{capacity\ \ per\ \ operating\ \ room} \right\rceil$ operating room(s). For the y^{th} operating room, the remainder of its capacity is filled with elective surgery. Finally, the third parameter concerns the methods that determines the sequence of the surgeries in the operating rooms where slack for urgent surgery is allocated to: either the First Fit based algorithm or the best BIM heuristic SA(C2).

Table 6 Parameter values for the test instances.

Surgical case mix	$h \in$ {surgical case mix A, surgical case mix B}
Number of operating rooms with slack for urgent surgery	$N \in$ {dedicated, 4, 12}
Method that determines surgery sequence within the operating rooms with slack	$b \in$ {First Fit, SA(C2)} | $N \in$ {4, 12}

Table 7 Hospital characteristics for urgent surgery

Surgical case mix	Average no. of urgent surgeries per day	Total slack for urgent surgery
A	5.1	743 minutes
B	2.6	372 minutes

Simulation model

We simulate the execution of a surgical schedule with help of an OR department simulation model created in eM-Plant 7.0 (Technomatrix). The surgery durations have a lognormal distribution (Strum *et al.*, 2000). Urgent surgical cases arrive according to a Poisson process, i.e. interarrival times between urgent surgeries are independent and exponentially distributed. An urgent surgery is started first-come-first-serve as soon as one of the operating rooms assigned for urgent surgery is or becomes available, i.e. during the period that no (more) elective surgeries are scheduled, or at any break-in-moment between surgeries. Ongoing surgeries are not interrupted for urgent surgery. When an urgent surgery is scheduled to start at a break-in-moment all subsequent elective surgeries are postponed, which may result in overtime. We use the sequential procedure of Law and Kelton (2000) to determine the number of instances that guarantees a precision of a maximal relative error of 0.10, and a minimal confidence level of 90%. We apply this procedure to the performance indicator with the instance type in which the output of the performance indicator fluctuates most strongly. This turns out to be the waiting time for urgent surgery in case of a dedicated operating room for surgical case mix A. After running 708 instances, we obtain the specified precision. As the simulation model handles 260 days at a time, we choose to set the number of instances per instance type at 780 days.

Results

First we compare the various methods for the allocation of slack for urgent surgery in terms of urgent surgery waiting time, utilization and overtime. We then discuss the reduction of urgent surgery waiting time due to BIM optimization.

Allocation of slack for urgent surgery

Table 8 compares the three methods for allocating slack for urgent surgery regarding the planned and the realized utilization. We define utilization as the ratio of used OR capacity to the allocated OR capacity, without considering overtime. From Table 8, we conclude that best utilization figures are realized by scheduling slack for urgent surgery in all available 12 operating rooms. Differences in planned utilization of the three methods can be explained by the different bin-packing problems. Differences in realized utilization can be explained by both the different bin-packing problems and the flexibility the methods offer for scheduling urgent surgery. In case urgent surgery is allowed in 12 operating rooms, all unused capacity can be used, including non-used slack for the variability of surgery duration, and several urgent surgeries can take place at the same time. Differences in utilization figures of surgical case mix A and surgical case mix B are caused by the required slack to deal with surgery duration variability, and the duration of the surgeries itself: on average, the surgeries in surgical case mix B have a shorter duration, which makes it easier to fill the capacity of an operating room.

Table 8 Planned and realized utilization for surgical case mix A and surgical case mix B

	Surgical case mix A		Surgical case mix B	
Slack for urgent surgery allocated to:	Planned utilization	Realized utilization	Planned utilization	Realized utilization
Dedicated room(s)	67.6%	73.6%	79.7%	81.7%
4 operating rooms	67.3%	74.9%	79.6%	82.3%
12 operating rooms	68.6%	76.7%	80.9%	83.2%

Table 9 shows figures on overtime, caused by both urgent surgery and the variability of surgery durations. Slack for urgent surgery allocated to 12 operating rooms gives the best results: it offers the most flexibility for scheduling urgent surgery. Differences in overtime between surgical case mix A and B can be attributed to the difference in average value and variability of the surgery durations.

Table 9 Average overtime and number of operating rooms with overtime per day

	Surgical case mix A		Surgical case mix B	
Slack for urgent surgery allocated to:	Average overtime per day	Average no. of operating rooms with overtime per day	Average overtime per day	Average no. of operating rooms with overtime per day
Dedicated room(s)	10.6h	3.6	6.2h	3.0
4 operating rooms	9.1h	3.9	5.4h	3.2
12 operating rooms	8.4h	3.8	5.6h	3.3

Finally, Figure 7 and 8 show the percentage of the urgent surgery started within the specified interval of time after arrival, as depicted on the x-axis of the graphs. We observe that allocating slack for urgent surgery in 12 operating rooms guarantees the shortest waiting time for urgent surgery, for both surgical case mix A and surgical case mix B.

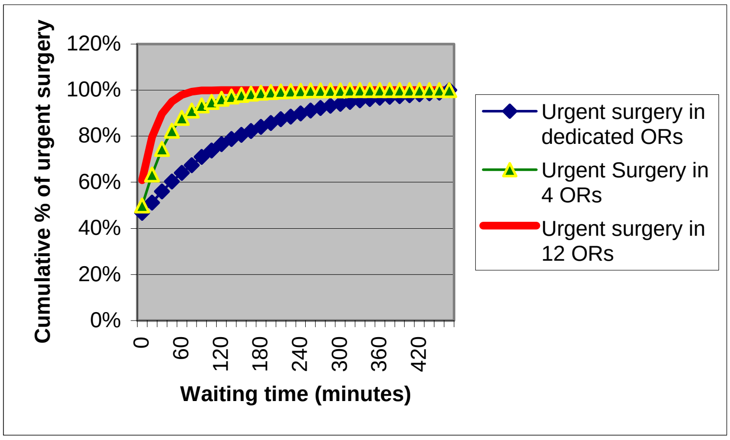

Figure 7 Cumulative percentages of urgent surgery waiting time for surgical case mix A

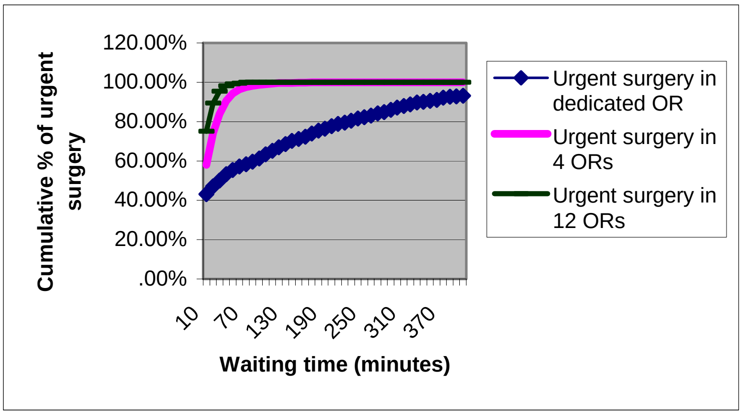

Figure 8 Cumulative percentages of urgent surgery waiting time for surgical case mix B

From Figures 7 and 8 above we can conclude that regarding the all performance measure used, allocation of slack for urgent surgery in 12 or 4 operating rooms outperforms concentrating slack in a dedicated operating room. In the following, we study whether adjusting the sequence of elective surgeries by BIM optimization can lead to a further decrease of the waiting time for urgent surgery.

Sequence of elective surgeries

To measure the effect of BIM optimization, we show figures of the waiting time of urgent surgical cases that arrive during the occupied interval. We first examine the situation in which slack for urgent surgery is allocated to 12 operating rooms, both for surgical case mix A and surgical case mix B. Table 11 and 12 show the percentage of the 1^{st}, 2^{nd} and 3^{rd} urgent surgery that is started within 10, 20 or 30 minutes after arrival. Figures of these three groups of urgent surgeries show the effect of BIM optimization most clearly: BIM optimization certainly has impact on the waiting time for urgent surgery, although the later the urgent surgical case arrives, the less the impact of BIM optimization is. As time goes by, the optimized schedule is increasingly disturbed by surgeries that take more or less time then expected, and the arrival of urgent surgeries. Note that, for both BIM optimization and no BIM optimization, the later the urgent surgical case arrives, the shorter the waiting time is. This is due to the fact that for almost all instances with slack in 12 operating rooms both with and without BIM optimization, the length of the first surgery with the shortest duration determines the maximum break-in-interval. The later the urgent surgical case arrives, the smaller the chance that the urgent surgical case arrives in this first interval. Furthermore, Table 10 and 11 show that the waiting time for urgent surgeries are shorter for surgical case mix B than for surgical case mix A: shorter surgery durations, and thus more surgeries per OR, lead to more BIMs.

Table 10 Cumulative percentages of waiting times for urgent surgeries for surgical case mix A and slack in 12 ORs

	1st urgent surgery		2nd urgent surgery		3rd urgent surgery	
Interval (minutes)	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization
10	28.8%	48.6%	34.9%	44.9%	40.4%	46.2%
20	53.0%	75.8%	56.9%	73.6%	63.0%	69.8%
30	70.5%	90.9%	71.8%	87.2%	76.3%	86.7%

Table 11 Cumulative percentages of waiting times for urgent surgeries for surgical case mix B and slack in 12 ORs

	1st urgent surgery		2nd urgent surgery		3rd urgent surgery	
Interval (minutes)	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization
10	51.7%	66.7%	56.6%	61.0%	60.4%	67.6%
20	77.2%	90.5%	83.5%	88.4%	83.5%	91.0%
30	89.4%	97.8%	93.4%	96.5%	93.5%	98.6%

For the situation in which slack for urgent surgery is allocated to 4 operating rooms, Table 12 and 13 show the cumulative percentages of the urgent surgery waiting times. BIM optimization yields better results for the 1st and 2nd urgent surgery, for the third urgent surgery, however, BIM optimization results in a longer waiting time. This is explained by the fact that BIM optimization schedules BIMs at the beginning of the day, that were first scheduled at the end of the day. Note that, in general, the later the urgent surgical case arrives, the longer the waiting time: the first break-in-interval is not necessarily the longest interval anymore because the small number of surgeries, and elective surgeries at the beginning of the day tend to be shorter than (mostly urgent) surgeries at the end of the day, as we assigned the slack to operating rooms with surgeries with a relatively short duration.

Table 12 Cumulative percentages of waiting times for urgent surgeries for surgical case mix A and slack in 4 ORs

	1st urgent surgery		2nd urgent surgery		3rd urgent surgery	
Interval (minutes)	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization
10	19.4%	29.4%	18.3%	21.4%	18.2%	17.0%
20	37.4%	50.8%	35.0%	35.4%	37.0%	31.4%
30	53.2%	67.6%	51.8%	54.8%	51.2%	45.8%

Table 13 Cumulative percentages of waiting times for urgent surgeries for surgical case mix B and slack in 4 ORs

	1st urgent surgery		2nd urgent surgery		3rd urgent surgery	
Interval (minutes)	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization	Without BIM optimization	With BIM optimization
10	32.6%	44.3%	34.4%	34.1%	37.9%	31.5%
20	58.2%	72.2%	59.3%	64.3%	64.9%	51.9%
30	75.8%	85.4%	76.3%	79.4%	78.2%	67.9%

7. Conclusion

Computational experiments show that in terms of urgent surgery waiting time, utilization, and overtime, urgent surgery can be anticipated best by allocating slack for urgent surgery to all available operating rooms, and thus allowing urgent surgery to interfere with the elective surgical schedule. Further savings in urgent surgery waiting time can be achieved by sequencing the surgeries such that the break-in-moments in the schedule of elective surgeries are spread equally over the day (BIM optimization). However, the effect of BIM optimization decreases as time goes by, since the optimized schedule is increasingly disturbed by the variability of the surgery durations and the arrival of urgent surgeries. Absolute and relative savings in urgent surgery waiting time are largest for a hospital with a surgical case mix with relative long surgery durations with a high variability.

Further research should focus on practical constraints when implementing BIM optimization: so far, we did not account for the availability of scarce resources such as microscopes or X-ray machines, surgeon's preferences for the sequence of surgeries, and (variable) set-up times in between elective surgeries. We also assumed that urgent surgery can be performed in any of the available operating rooms, and should be performed as soon as possible. In practice, part of the urgent surgeries can only be performed in a subset of the operating rooms that command specialized resources, and some surgeries are more urgent than others. Additional modeling of the BIM optimization problem and simulation experiments should clarify the effect of BIM optimization if all these issues are taken into account. Furthermore, it should be investigated whether on-line rescheduling algorithms can further reduce urgent surgery waiting time, especially for the urgent surgical cases that arrive once the BIM optimized surgical schedule has been disturbed.

Bibliography

Aarts,E.H.L. and Korst,J.H.M. (1989). Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. New York: Wiley.

Aytug,H., Lawley,M.A., McKay,K., Mohan,S., and Uzsoy,R. (2005). Executing production schedules in the face of uncertainties: a review and future directions. *European Journal of Production Research* 161:86-110.

Barlow,A.P., Wilkinson,D.A., Wordsworth,M., and Eyre-Brook,I.A. (1993). An emergency daytime theatre list: utilization and impact on clinical practice. *Annals of The Royal College of Surgeons of England* 75:441-444.

Boer, F. (2006). Praktisch OK management. Personal communication.

Brasel,K.J., Akason,J., and Weigelt,J.A. (1998). Dedicated operating room for trauma: a costly recommendation. *Journal of Trauma* 44:832-836.

Busetti,F. (2003). Simulated annealing overview.
Available from http://www geocities.com/francorbusetti/saweb pdf.

Davenport,A.J. and Beck,J.C. (2000). A survey of scheduling techniques with uncertainty.
Available from http://www eil utoronto ca/profiles/chris/chris papers html.

Davenport,A.J., Gefflot,C., and Beck,J.C. (2001). Slack-based techniques for robust schedules. *Paper presented at the Sixth European Conference on Planning* (ECP-2001).

Dexter,F. (2000). A strategy to decide whether to move the last case of the day in an operating room to another empty operating room to decrease overtime labor costs. *Anesthesia & Analgesia* 91:925-928.

Dexter,F. (2005). What's the best way to manage urgent and emergent OR cases? *OR Manager* 21.

Dexter,F., Macario,A., and Traub,R.D. (1999a). Optimal sequencing of urgent surgical cases. *Journal of Clinical Monitoring and Computing* 15:153-162.

Dexter,F., Macario,A.M.D., and Traub,R.D. (2005). Which algorithm for scheduling add-on elective cases maximizes operating room utilization? Use of bin packing algorithms and fuzzy constraints in operating room management. *Anesthesiology* 91:1491.

Dexter,F., Macario,A.M.D., Traub,R.D., Hopwood,M., and Lubarsky,D.A. (1999b). An operating room scheduling strategy to maximize the use of operating room block time: computer simulation of patient scheduling and survey of patient's preferences for surgical waiting time. *Anesthesia & Analgesia* 89:7-20.

Dexter,F. and Traub,R.D. (2000). Sequencing cases in the operating room: predicting whether one surgical case will last longer than another. *Anesthesia & Analgesia* 90:975-979.

Fitzgerald, J. A., Lum, M., and Dadich, A. (2005). How can human technologies improve decision-making practices in the operating theatre? *Paper presented at the 4th Health Care Technology Management Conference, Aalborg, Denmark.*

Garey,M.R. and Johnson,D.S. (1979). Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco.

Gerchak,Y., Gupta,D., and Henig,M. (1996). Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science* 42:321-334.

Glouberman,S. and Mintzberg,H. (2001). Managing the care of health and the cure of disease--Part I: Differentiation. *Health Care Manage Review* 26:56-69.

Guinet, G., and Chaabane,S. (2003). Operating theatre planning. *International Journal of Production Research* 85:69-81.

Hans,E.W., Wullink,G., Houdenhoven,M., and Kazemier,G. (2006). Robust surgery loading. *Accepted for publication in European Journal of Operational Research /BETA working paper WP-141, University of Twente*.

Herroelen,W.S. and Leus,R. (2005). Project scheduling under uncertainty; survey and research potentials. *European Journal of Operational Research* 165:289-306.

Kirkpatrick,S., Gerlatt,C.D., and Vecchi,M.P. (1983). Optimization by simulated annealing. *Science* 220:671-680.

Lamiri,M., Xie,X., Dolgui,A., and Grimaud,F. (2005). A stochastic model for operating room planning with elective and emergency surgery demands. *Paper presented at the EURO Working Group on OR Applied to Health Services, 31st Annual Meeting*.

Law,A.M. and Kelton,W.D. (2000). Simulation modeling and analysis. McGraw-Hill Inc., Singapore.

Lebowitz,P. (2003). Schedule the short procedure first to improve OR efficiency. *AORN Journal* 78:651-659.

Lin,S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal* 2245-2269.

Lovett,B.E. and Katchburian,M.V. (1999). Emergency surgery: half a day does make a difference. *Annals of the Royal College of Surgeons of England* 81:62-64.

Luenberger,D.G. (1969). Optimization by vector space methods. New York: John Wiley & Sons, Inc..

Luss,H. and Rosenwein,M.B. (1997). Operations research applications: opportunities and accomplishments. *European Journal of Operational Research* 97:220-244.

OECD (2005). Health at a Glance - OECD Indicators 2005. Available from *http://www.oecd.org.*

Ozkarahan,I. (2000). Allocation of surgeries to operating rooms by goal programming. *Journal of Medical Systems* 24:339-378.

Pham,D. and Klinkert,A. (2005). Surgical case scheduling as generalized job shop scheduling problems. Pham, D.; Klinkert, A.. *Paper presented at ESI2005: OR in Health Care*.

Strum,D.P., May,J.H., and Vargas,L.G. (2000). Modeling the uncertainty of surgical procedure times - comparison of log-normal and normal models. *Anesthesiology* 92:1160-1167.

Van Houdenhoven,M., Krijgsman,H., Van Barneveld,R., Boer,F., Hans,E.W., Van Hoorn,A., and Pullen,N. (2005). Benchmarking OK: van appels met peren naar Elstar met Jonagold. *NFU projectgroep benchmarking OK, internal document in Dutch*.

Van Houdenhoven,M., Wullink,G., Hans,E.W., and Kazemier,G. (2006). A framework for hospital planning and control. *Working paper Erasmus MC*.

Van Kempen,A.N.O. (2005). Personal communication.

Vieira,G.E., Herrmann,J.W., and Lin,E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies and methods. *Journal of Scheduling* 6:39-62.

Wullink,G., Gademann,A.J., Hans,E.W., and Van Harten,A. (2005). Scenario-based approach to resource loading under uncertainty. *International Journal of Production Research* 42:5079-5098.