Flexible Services and Manufacturing manuscript No.
(will be inserted by the editor)

A rolling horizon framework for the operating rooms planning under uncertain surgery duration

the date of receipt and acceptance should be inserted later

Abstract We consider the Advanced Scheduling Problem (ASP) assuming a block scheduling strategy. A set of patients and the related surgery waiting list are given, together with a set of Operating Room (OR) blocks and a planning horizon. The problem asks to determine the subset of patients to be scheduled and their assignment to the available OR blocks.

We consider a so-called rolling horizon approach in order to solve the ASP over a planning horizon of several weeks. The approach is iterative and readjusts the schedule each week: at each iteration the mid-term schedule over the next n weeks is generated by solving an optimization problem, minimizing a penalty function based on patients' delay and tardiness; the first week schedule is then implemented. Unpredictable extensions of surgeries and new arrivals may disrupt the schedule. The schedule is then repaired in the next week iteration, again optimizing over n weeks the penalty function while limiting the number of disruptions from the previously computed plan.

The total delay and tardiness minimization problem is formulated as an ILP model and solved with a commercial solver. A deterministic formulation and a robust one are proposed and compared over different stochastic realization of surgery times.

Keywords operating room planning, rolling horizon, robust optimization, block scheduling, re-optimization

1 Introduction and problem addressed

In the last decades, the growing pressure on budget cuts and performance evaluation led health care managers to improve hospital organization, by reducing cost, optimizing resource use and increasing operational efficiency. The crucial role that surgery departments play within hospitals has been raising an increasing number of research studies aimed at planning Operating Room (OR) activities. This is due both to the significant costs of surgical facilities and to the impact that surgical activities have on the demand for hospital services and on patients waiting times [Testi et al., 2007]. Exhaustive literature reviews on operating room planning and scheduling are reported in [Cardoen et al., 2010a] and [Guerriero and Guido, 2011]. Several different versions of the OR problem have been considered in literature [Cardoen et al., 2010b] which differ w.r.t. features and assumptions.

In this paper we focus on an OR planning problem assuming a closed block planning approach. Following this strategy, during each week, a given number of pre-assigned OR blocks is assigned to each surgical specialty/surgeon, in which surgical cases can be operated [van Oostrum et al., 2010]. The OR blocks cannot be shared among surgical specialties. In block scheduling approach, OR planning and scheduling problem is often divided into three phases/sub-problems each one associated with a different decision level [Testi et al., 2007]. In the first phase, number, type and working hours of the available ORs, and OR capacity assignment to surgical groups or specialties are determined at a strategic level. In the second phase, a cyclic timetable, denoted as Master Surgical Schedule (MSS), is built on a medium term stand point to define the tactical assignment of specialties to days and ORs. The last phase, referred to as Surgery Process Scheduling, is divided into two sub-problems: Advanced Scheduling and Allocation scheduling [Magerlein and Martin, 1978, Blake and Carter, 1997]. The Advanced Scheduling Problem (ASP) assigns a surgery date and OR to the each scheduled patient, while the Allocation Scheduling problem determines the sequence of surgeries in each OR block.

The Surgery Process Scheduling, i.e. scheduling and sequencing patients into OR blocks, has recently received growing attention and many solution approaches have been proposed. Langragian relaxation approaches [Augusto et al., 2008], branch and price algorithms [Cardoen et al., 2009, Fei et al., 2008], heuristics [Marques et al., 2012, Tànfani and Testi, 2010, Herring and Herrmann, 2012] and metaeuristics algorithms [Rizk and Arnaout, 2012, Herring and Herrmann, 2011] have been recently proposed to solve the ASP with deterministic surgery times.

Real life OR scheduling decisions are further complicated by inherent variability of surgical cases durations [Tyler et al., 2003].

Stochastic optimization models have been proposed to tackle ASP with uncertain surgery duration. In [Denton et al., 2007] a two-stage stochastic model with recourse, taking into account patient waiting times and OR idle time and overtime is proposed. Different heuristics are compared and the influence of patient sequencing within OR blocks is analyzed. In [Min and Yih, 2010] a stochastic programming model with recourse is presented. A sample average approximation method to obtain an optimal surgery schedule with the aim of minimizing patient costs and OR overtime costs is used. In [van Oostrum et al., 2008] a mathematical programming model is proposed in which uncertain surgery durations are dealt with through probabilistic constraints. The proposed model tries to optimize OR utilization without increasing overtime and cancellations.

Researchers are recently moving towards robust optimization approaches. In [Hans et al., 2008] different heuristics for the robust surgery loading problem are proposed, with the aim of maximizing operating theatre utilization and minimizing overtime risk by introducing planned slack times. In [Tànfani et al., 2010] a two-level framework is proposed. In the first level, a MIP model computes a deterministic solution for the OR planning problem. In the second level, variability of surgery duration is taken into account by means of individual chance constraints for each OR block and a robust solution is achieved by iteratively adding safety slacks to the first level deterministic model solutions. In [Denton et al., 2010] two models aimed at minimizing overall OR cost including a fixed cost of opening ORs and a variable cost of overtime are compared. The first is a two-stage stochastic linear model with binary decision variables in the first stage and simple recourse in the second stage. The second is its robust counterpart, in which the objective is to minimize the maximum cost associated with an uncertainty set for surgery durations. They show that the robust method is much faster than stochastic recourse model, and limits the worst-case outcome of the recourse problem. In [Addis et al., 2014] a cardinality constrained robust optimization approach based on [Bertsimas and Sim, 2004] is proposed. The method allows to exploit the potentialities of a linear programming model without the necessity of generating scenarios. Different levels of robustness are evaluated and the solutions found are compared with the deterministic one in terms of number of operated and tardy patients, OR utilization rate and number of rescheduled patients.

In general, the previous reported robust approaches solve a single run planning and scheduling phase, i.e. they anticipate future disruptions of OR schedules caused by real operating time realizations and guarantee the solutions to be feasible taking into account changes in surgery duration. The main aims of the so called proactive scheduling are optimizing OR utilization rate and reducing the risk of patient cancellations. In [Stuart and Kozan, 2012] the disruption management and rescheduling problem, i.e. reactive scheduling, of a single OR is solved focusing on the day-to-day running of a day-surgery unit. To the best of the authors' knowledge, reactive scheduling and re-optimization of patient assignment over a longer planning horizon have not been already addressed in the literature.

In this paper we focus our attention on the ASP including uncertainty in surgery duration. We consider a planning horizon of several weeks. The initial waiting list is given , and successive weekly arrivals are considered. The problem is tackled with a rolling horizon approach. The set of patients to be scheduled in each OR block and week of the planning horizon is determined by solving an optimization model, which spans several weeks, yet usually not the whole planning horizon. Only the first week is applied and its impact on real surgery times is computed, possibly canceling patients. Then the waiting list is updated with new arrivals and cancelled patients, and a new optimization phase is performed on the following weeks. We consider the previous solution as a baseline schedule, and we search for a new re-optimized solution that does not "differ too much" from the previously computed schedule. In fact, changes in the scheduling, also refereed as disruptions, represent a reduction in the quality of service for patients. We also force the scheduling of cancelled patients. Both a deterministic optimization model and its robust counterpart (in the sense of Bertsimas and Sim [2004]) are proposed and evaluated to solve the ASP.

The paper is organized as follows. In Section 2 we describe the proposed solution approach. In Section 3 we present the basic optimization model and its robust counterpart. The proposed approach is evaluated in a realization based environment and computational results are presented in Section 4.

2 Optimization–re-optimization framework

In the considered environment ORs are assumed to be assigned to specialties according to a predefined master schedule. The general idea of the approach is to schedule one week (or, more in general, one time slot) at a time, and to iterate for all the weeks of the planning horizon. The framework provides the schedule for one week at a time, yet according to a look-ahead policy, the optimization step takes into account a longer planning horizon. The first week is applied and its outcome, in terms of cancelled patients, is combined with new arrivals to produce the input for the next week re-optimization. As patients must be summoned in advance, the number of differences between the re-optimized and the previously planned schedule is limited. The architecture of the proposed solution approach is detailed in Fig 1.

An aggregate optimization model is applied every week (B1), providing a mid-term patient schedule which spans a horizon of several weeks. The output of the model is the assignment of a subset of selected patients to OR blocks in the considered time horizon. The first week schedule is then completed by deciding the fine scheduling and applied. The fine scheduling of patients assigned to one block is handled and refined on a tactical level by a dispatcher (B2) that provides the order according to which patients are operated in a single block.

Real surgery durations may differ from forecasted values used in the mid-term and fine scheduling planning; thus, it may happen that not all the scheduled patients can be operated in the block they are assigned to. As an outcome, a set of cancelled patients can be generated, who must be rescheduled in following weeks. Cancelled patients and new arrivals, together with patients who leave the system, are taken into account in updating the waiting list (B4).

The four components of Figure 1 are to be iteratively activated in a sequence ($B1, B2, B3, B4$), and the output of each one feeds the next component. For evaluation purposes the framework has been implemented in a realization based environment.

2.1 Optimization model ($B1$)

This is the most complex element in the system, and a detailed description is deferred to Section 3. Basically the optimization model deals with scheduling a set I of patients over a planning horizon composed of a set K of consecutive weeks, and it assigns patients to blocks in such weeks. At the p-th iteration, K is made up of the $p, p+1, \ldots, (p+n-1)$-th weeks, where $n = |K|$. In general, we cannot expect all the patients of I to be scheduled, hence the model aims to produce a schedule

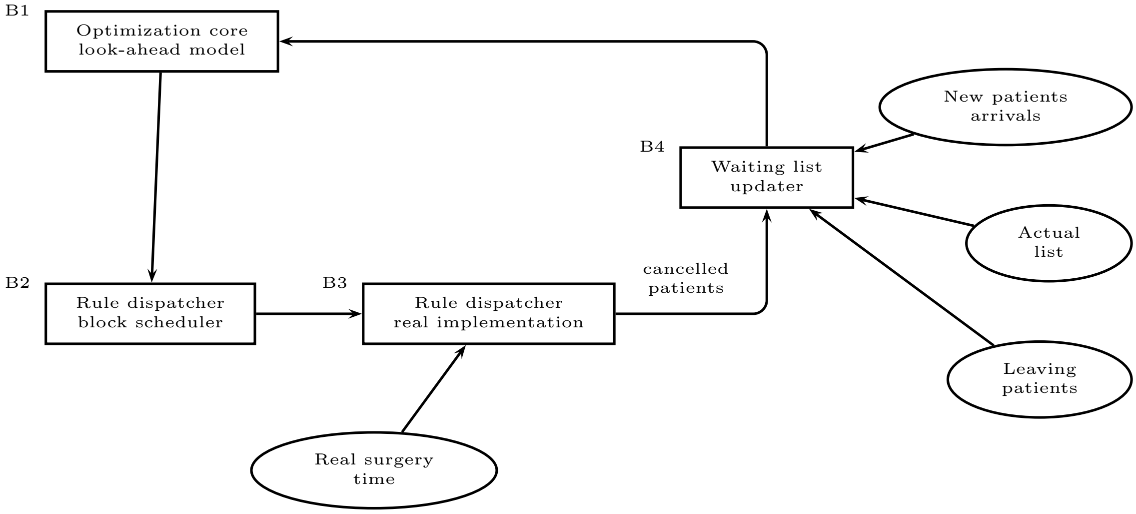

Fig. 1 Structure of the optimization framework

S for a subset $I' \subset I$ minimizing a total weighted delay measure, For patients in $I \setminus I'$ (that will be scheduled out of the planning horizon) a lower bound on their delay is computed.

The model is fed with a list of patients, where for each patient his/her expected operating time, waiting time, due date and an urgency coefficient are specified.

We stress that at each iteration a schedule over n weeks is produced, but only the first week of the schedule is applied, the remainder of the schedule is kept as a reference for the next iteration. Besides minimizing the tardiness measure, the model also aims to be conservative with respect to decisions taken in the previous iteration, limiting the number of changes in the schedule.

2.2 Rule dispatcher (B2)

This component takes care of the allocation problem in the first week of the planning horizon. It takes the first week of the produced schedule and defines the order according to which patients assigned to each block are to be operated. In a real-life environment, such decisions would be taken by the surgical staff; in order to run in a simulated environment, this block implements a simple “longest surgery time first” rule, assuming heuristically that long operations are more complex and hence are given higher priority in accessing the operating room.

2.3 Rule dispatcher real implementation ($B3$)

In applying the solutions to real data, this component generates realizations of the random parameters that describe the surgery time of each patient scheduled in the given week. Here is where unexpected events come into play — in a real-life environment, and operation can require more time than expected. According to the realized operating times, some operations may be cancelled and patients sent back without entering the operating room.

2.4 Waiting list updater ($B4$)

Cancelling operations cause patients to "reenter" the waiting list, along with newly arrived patients. The updated list is sent as an input plan for the remaining to the optimization model ($B1$) together with the plan for the remaining weeks. At the next iteration, the optimization model will again produce a schedule over the next planning horizon — say weeks $p+1, p+2, \ldots, p+n$, while also trying to ensure that reentering patients will be scheduled early in the next planning horizon and limiting disruptions with respect to the old schedule.

3 Models

We introduce the optimization model to solve the ASP with stochastic surgery durations (Stochastic Advanced Scheduling, SAS). We assume a block scheduling approach and focus on a single surgical specialty, but the approach can be easily adapted to take into account more than one specialty. The objective function aims at minimizing an overall penalty due to delay in serving patients. As proposed in Tànfani and Testi [2010] it takes into account both urgency and waiting time of scheduled and not scheduled patients. Besides, a penalty for due date violation (i.e. patient tardiness) is also considered as in (Addis et al. [2014]).

A set of elective patients I to be scheduled in a planning horizon of D days is given. A set J of OR blocks and their schedule during each week are given. Each block is described by an operating room and a week day. The planning horizon is then represented by a sequence of repetitions of the same group of blocks in a set of weeks K. The available total time of a time block j in week k, i.e. the OR block length, is denoted as γ_{jk}.

Patients in the set I belong to a waiting list, where patients are registered at the moment they arrive in the service. For each patient i, let w_i denote the number of days which the patient has already spent in the waiting list at the beginning of the planning horizon. Moreover, a maximum waiting time l_i and a corresponding urgency parameter u_i are given for each patient i. If the patient has spent w_i days in the waiting list, he/she must receive surgery before a due date $dd_i = l_i - w_i$, otherwise he/she is considered tardy. According to the block weekly based pattern, if a patient is scheduled in block $j \in J$ and in week $k \in K$, he/she waits a total number of days $d_{jk} = 7(k-1)+j$. The surgery time $\tilde{t}_i$ for each patient i is consider to follow a given probability distribution.

The set of weeks in which a cancelled patient must be rescheduled is denoted as $K_r \subset K$. For each patient i belonging to the set I of patients to be scheduled in the next optimization step, let introduce the parameter r_i which is equal to 1 if patient i must be rescheduled in the next weeks $k \in K_r$ and 0 otherwise.
To limit the impact of rescheduled patients and newly arriving ones, we accept a limited number of disruptions in the first weeks on the rolling period. Let us denote with K_d the first week set in which disruptions have to be limited. Urgent new arrivals are dealt with by leaving some empty space in each week.

The problem can be formulated using the following sets of binary decision variables:

- x_{ij}^k, such that $x_{ij}^k = 1$ if patient i is assigned to block j in week $k \in K$, and zero otherwise.
- z_i^k, such that $z_i^k = 1$ if pre-planned patient i is cancelled from the current schedule
- y_i^k, such that $y_i^k = 1$ if patient i, who was not previously pre-planned, is added to the schedule

The objective function is formulated as follows:

$$\min \sum_{i \in I} \left\{ \sum_{j \in J} \sum_{k \in K} \left[d_{jk} + (w_i + d_{jk} - l_i)^+ \right] u_i x_{ij}^k \right. \tag{1}$$

$$\left. + \left[(w_i + D + 1) + (w_i + D + 1 - l_i)^+ \right] u_i \left(1 - \sum_{j \in J} \sum_{k \in K} x_{ij}^k \right) \right\},$$

where $(w_i + d_{jk} - l_i)^+ = \max\{w_i + d_{jk} - l_i, 0\}$ is the patient tardiness, that is the number of days waited after the due date. The first term represents the penalty for the scheduled patients. For each scheduled patient i the penalty is composed by two parts: the number of days d_{jk} spent before receiving surgery in the planning horizon and the tardiness $(w_i + d - l_i)^+$ of the patient. The term is weighted by patient urgency parameter u_i, in order to give priority to most urgent patients. The second term is associated with the penalty of unscheduled patients. It is the sum of the tardiness and the overall days spent waiting for surgery before and after the beginning of the planning horizon, while for the scheduled patients, the waiting days term do not consider the days before the beginning of the planning horizon. As real tardiness and waiting days cannot be computed for unscheduled patients (we do not know when they will be scheduled), we use a lower bound to take them into account. The bound is calculated assuming that all the remaining patients are scheduled in the first day after the end of the planning horizon $(D + 1)$. Also for unscheduled patients waiting time and tardiness are weighted by urgency parameter u_i.

The set of constraints is the following:

$$\sum_{j \in J} \sum_{k \in K} x_{ij}^k \leq 1 \qquad \forall i \in I : r_i = 0 \tag{2}$$

$$\sum_{j \in J} \sum_{k \in K_r} x_{ij}^k = 1 \qquad \forall i \in I : r_i = 1 \tag{3}$$

$$\sum_{i \in I} \tilde{t}_i x_{ij}^k \leq \gamma_{jk} \qquad \forall j \in J, \quad \forall k \in K \tag{4}$$

$$\sum_{i \in I} \sum_{j \in J} \tilde{t}_i x_{ij}^k \leq \alpha_k \sum_{j \in J} \gamma_{jk} \qquad \forall k \in K \tag{5}$$

$$z_i^k \geq 1 - \sum_{j \in J} x_{ij}^k \qquad \forall i \in I, k \in K_d : \sum_j \tilde{x}_{ij}^k = 1 \tag{6}$$

$$y_i^k \geq \sum_{j \in J} x_{ij}^k - 1 \qquad \forall i \in I, k \in K_d : \sum_j \tilde{x}_{ij}^k = 0 \tag{7}$$

$$\sum_{i \in I} z_i^k + \sum_{i \in I : r_i = 0} y_i^k \leq \delta_k \qquad \forall k \in K_d \tag{8}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i \in I, j \in J, k \in K \tag{9}$$

$$y_i^k \in \{0, 1\}, z_i^k \in \{0, 1\} \qquad \forall i \in I, k \in K_d \tag{10}$$

Constraints (2) ensure that each patient is operated at most once, while constraints (3) ensure that each patient cancelled is scheduled in one block belonging to week $k \in K_r$. Constraints (4) are stochastic capacity constraints for each block forcing the total time in block j of week k to be lesser than or equal to the maximum available time γ_{jk}. Constraints (5) are week utilization constraints which bounds the total occupation of blocks for week k to be less than given occupation parameters α_k; in our implementation the value of α_k is set equal to 1 for the first week of the planning horizon and decreases for the following weeks — this leaves increasing slack capacity to manage new patient arrivals and emergency cases in the future. Constraints (6) and constraints (7) compute the number of variations between the actual solution and the previous one, namely disruptions. We denote by $\tilde{x}_{ij}^k$ the solution found in the previous iteration in the scheme of Figure 1. Constraints (8) bound the total number of disruptions between the actual and the previous schedule to the value δ_k, for the set of weeks K_d. Finally (9) and (10) are variable domain constraints.

The Deterministic Advanced Scheduling (DAS) model is obtained from the SAS model using for each patient i a deterministic surgery time $\bar{t}_i$. Constraints (4) and (5) are replaced by

$$\sum_{i \in I} \bar{t}_i x_{ij}^k \leq \gamma_{jk} \qquad \forall j \in J, \quad \forall k \in K \tag{11}$$

$$\sum_{i \in I} \sum_{j \in J} \bar{t}_i x_{ij}^k \leq \alpha_k \sum_{j \in J} \gamma_{jk} \qquad \forall k \in K \tag{12}$$

Beside, we propose a robust model, the Robust Advanced Scheduling (RAS) model, to deal with uncertainty.

The RAS model is based on the robust optimization approach proposed in [Bertsimas and Sim, 2004]. Random parameters are assumed to vary in a given interval. Uncertainty is dealt with so as to guarantee than any solution is feasible even if, for each constraint involving uncertain parameters, at most a fixed number Γ of them assume the maximum value in the interval and all the others assume the central one. The parameter Γ controls the "level of robustness" for the solution. In our case uncertain parameters are the surgery times $\tilde{t}_i$. We consider the interval $\bar{t} - \hat{t}, \bar{t} + \hat{t}$, where the central value of the interval is denoted as $\bar{t}$ and the maximum value we want to protect from is equal to $\bar{t}_i + \hat{t}_i$.

Uncertainty of surgery times have an impact on capacity constraints (4) and (5). We enforce robustness on the single block capacity constraints (4) as follows. For each block j, and week k, a subset S_{jk} of patients, of cardinality (at most) Γ, assigned to the block is assumed to get the worst possible realization of surgery times. The solution is guaranteed to be feasible even with respect to the selection of elements of subset S_{jk} having the worst impact on the capacity constraint:

$$\sum_{i \in I} \bar{t}_i x_{ij}^k + \max_{S_{jk} \subset I : |S_{jk}| = \Gamma} \left\{ \sum_{i \in S_{jk}} \hat{t}_i x_{ij}^k \right\} \leq \gamma_{jk} \tag{13}$$

Following [Bertsimas and Sim, 2004] the robust capacity constraints (13) are linearized and transformed into constraints (17) in the RAS model that follows, via dualization: this also requires new variables ζ^{jk}, π_i^{jk} and new constraints (21) (see appendix A for details). In order to limit the computational effort required by the solution process, we decided to impose robustness only on the set E_K of the first $|E_K|$ weeks of the planning horizon. Week capacity constraints (5) are transformed into constraints (18) by forming their left-hand sides as the sum of the left-hand sides of (17), without introducing new worst-impact subproblems. In this way we guarantee that the worst case capacity evaluated with the robust constraints does not exceed the fraction α_k for each week.

$$\begin{aligned} \min \sum_{i \in I} \Bigg\{ & \sum_{j \in J} \sum_{k \in K} [d_{jk} + (w_i + d_{jk} - l_i)^+] u_i x_{ij}^k \\ & + [(w_i + D + 1) + (w_i + D + 1 - l_i)^+] u_i \left(1 - \sum_{j \in J} \sum_{k \in K} x_{ij}^k \right) \Bigg\} \end{aligned} \tag{14}$$

s.t.

$$\sum_{j\in J}\sum_{k\in K} x_{ij}^k \le 1 \qquad \forall i \in I : r_i = 0 \qquad (15)$$

$$\sum_{j\in J}\sum_{k\in K_r} x_{ij}^k = 1 \qquad \forall i \in I : r_i = 1 \qquad (16)$$

$$\sum_{i\in I} \bar{t}_i x_{ij}^k + \Gamma\zeta^{jk} + \sum_{i\in I}\pi_i^{jk} \le \gamma_{jk} \qquad \forall j \in J, \forall k \in E_K \qquad (17)$$

$$\sum_{j\in J}\left(\sum_{i\in I} \bar{t}_i x_{ij}^k + \Gamma\zeta^{jk} + \sum_{i\in I}\pi_i^{jk}\right) \le \alpha_k \sum_{j\in J}\gamma_{jk} \qquad \forall k \in E_K \qquad (18)$$

$$\sum_{i\in I} \bar{t}_i x_{ij}^k \le \gamma_{jk} \qquad \forall j \in J, \quad \forall k \in K \setminus E_K \qquad (19)$$

$$\sum_{i\in I}\sum_{j\in J} \bar{t}_i x_{ij}^k \le \alpha_k \sum_{j\in J}\gamma_{jk} \qquad \forall k \in K \setminus E_K \qquad (20)$$

$$\zeta^{jk} + \pi_i^{jk} \ge \hat{t}_i x_{ij}^k \qquad \forall j \in J, \forall i \in I, \forall k \in E_K \qquad (21)$$

$$z_i^k \ge 1 - \sum_{j\in J} x_{ij}^k \qquad \forall i \in I, k \in K_d : \sum_j \tilde{x}_{ij}^k = 1 \qquad (22)$$

$$y_i^k \ge \sum_{j\in J} x_{ij}^k - 1 \qquad \forall i \in I, k \in K_d : \sum_j \tilde{x}_{ij}^k = 0 \qquad (23)$$

$$\sum_{i\in I} z_i^k + \sum_{i\in I: r_i=0} y_i^k \le \delta_k \qquad \forall k \in K_d \qquad (24)$$

$$x_{ij}^k \in \{0,1\} \qquad \forall j \in J, \forall k \in K, \forall i \in I \qquad (25)$$

$$y_i^k \in \{0,1\}, z_i^k \in \{0,1\} \qquad \forall i \in I, k \in K_d \qquad (26)$$

$$\zeta^{jk}, \pi_i^{jk} \ge 0 \qquad \forall j \in J, \forall k \in E_K, \forall i \in I \qquad (27)$$

4 Computational tests

The optimization–re-optimization framework has been tested on real life based data. The framework has been coded in C++, models have been solved with CPLEX 12.2.0.0 with single thread option. Tests have been run on a Intel Xeon CPU E5335 (2 quad core cpus at 2GH). We set a two hours time limit and a keep the default 10^{-4} as acceptable relative gap. We limit the size of the search tree to at most 2000MB.

At each optimization phase a planning horizon of 4 weeks is considered. The framework is applied for 8 consecutive times, then covering 8 weeks of effective scheduling. We consider two patients lists, one with 80 initial patients rising up to a total of 132 due to new arrivals and one with 120 initial patients rising up to 176 during the planning period. The arrivals span the considered planning horizon. Patients are divided into five urgency classes as proposed in [Valente et al., 2009]. Each urgency class is associated with a maximum waiting time expressed in days, that is the maximum number of days that a patient can wait without deteriorating his/her clinical conditions. The maximum waiting times are set to 8, 30, 60, 180 and 360 days, respectively. The corresponding urgency coefficients of the five considered classes are 45, 12, 6, 2, 1, respectively. For each list we generated different instances, by assigning surgery times derived from [Hans et al., 2008] to patients. We considered three surgery time lists, each described by an average surgery time, a standard deviation and the percentage of this type over the total number of surgeries in the list. According to these percentages, an average surgery time ($\bar{t}_i$) and a standard deviation (σ_i) is randomly assigned to each patient i. The maximum deviation is assumed to be equal to the standard deviation, i.e. $\hat{t} = \sigma_i$. Based on such data, 10 random realization are generated for each patient set and surgery time list, and the framework is applied to each of them.

Determ.	# op	# op < dd	# canc	# wait	# wait >dd
surgery list 1	65.70	31.50	11.80	66.30	18.40
surgery list 2	78.20	45.80	11.40	53.80	10.30
surgery list 3	68.30	38.30	9.00	63.70	15.00
$\Gamma = 1$					
surgery list 1	59.60	26.20	3.50	72.40	25.10
surgery list 2	75.50	42.90	3.70	56.50	10.60
surgery list 3	65.40	35.70	2.70	66.60	17.40
$\Gamma = 3$					
surgery list 1	34.50	25.50	0.10	97.50	44.50
surgery list 2	65.90	38.40	0.40	66.10	17.60
surgery list 3	57.29	30.30	0.50	74.70	24.40

Table 1 Results on cancelled, operated and waiting patients for 80 initial patients instances

Three blocks are considered for the 80 patients list, and 4 for the 120 case. The length of blocks is assumed to be equal to 6 hours for all the cases.

A previously cancelled patient is forced to be rescheduled in the first 2 weeks of the planning horizon – i. e. with horizon $K = \{p, \ldots, p+3\}$ we set $K_r = \{p, p+1\}$ –, while disruptions are must be at most 2 in the first week of the planning horizon – $K_d = \{p\}$ and $\delta_k = 2$. The value of α_k depends on the week: $\alpha_p = 1, \alpha_{p+1} = 0.73, \alpha_{p+2} = 0.73, \alpha_{p+3} = 0.73$. Finally, when applying the robust model, robustness is required in the first 2 weeks – namely $E_K = \{p, p+1\}$, $|E_K| = 2$.

4.1 Quality of solutions: patient point of view

To asses quality of obtained solutions from the patients point of view, Table 1 and 2 report the total number of operated patients (# op), the total number of patients operated before their due date (# op <dd), the total number of cancelled patients (# canc), the number of patients still in the waiting list at the end of the planning period (# wait), and, among them, the number of those whose due date has been exceeded (# wait >dd), for the 80 and 120 initial patients case, respectively. Values are averaged out over the 10 realizations.

Results on 80 initial patients case show that by applying the deterministic model about one half of the patients entering the systems are operated in the considered planning horizon for the surgery list 1 and 3. The number slightly increases for surgery list 2. More than 50% of operated patients receive surgery before their due date is exceeded. The number of patients in the waiting list at the end of the planning horizon is smaller than 80. The size of the waiting list oscillates along weeks. Although the length differs for the different realizations, in general the final waiting list is shorter then the initial one, and the reduction may rise up to 11. The number of tardy waiting patients usually increases in the first weeks, and then reduces, but at the end is higher than the initial value.

Increasing the level of robustness reduces the number of operated patients and increases the final waiting list size. Besides, the waiting list size may increase along the planning horizon, for surgery list 1 and 3. The increase may be significant, up to more than 20 patient if high robustness is required $\Gamma = 3$. The number of tardy patient at the end of the planning horizon increases, as well.

On the other hand, the deterministic solutions produce high number of cancelled patients, while the robust ones reduce them dramatically. In fact, by imposing $\Gamma = 3$, the number of cancelled patients is negligible (0.5 in the worst case), while it may rise up to 12 if robustness is not required.

A similar behavior is obtained for the 120 initial patients case.

4.2 Quality of solutions: hospital point of view

From the hospital point of view, operating rooms utilization rate and overtime are important metrics, as they describe the impact on usage of limited resources. We provide per week utilization

Determ.	# op	# op < dd	# canc	# wait	# wait > dd
surgery list 1	86.10	58.00	12.70	89.90	30.10
surgery list 2	102.30	69.90	13.80	73.70	18.50
surgery list 3	92.40	61.50	12.30	83.60	23.00
$\Gamma = 1$					
surgery list 1	79.40	50.50	5.50	96.60	35.50
surgery list 2	100.30	64.60	4.10	75.70	22.10
surgery list 3	87.00	64.90	3.30	89.00	26.80
$\Gamma = 3$					
surgery list 1	44.40	40.20	0.80	131.60	57.50
surgery list 2	85.90	58.30	0.60	90.10	26.90
surgery list 3	76.10	54.20	0.40	99.90	33.00

Table 2 Results on cancelled, operated and waiting patients for 120 initial patients instances

rate, fraction of overtime and undertime blocks, and average overtime and undertime (in minutes)for different initial set of patients and values of Γ in Tables 3, 4 and 5, for the 80 initial patients, and 6, 7 and 8 for the 120 patients case. Values are averaged out on the 10 random realizations. In the first row, for each surgery list, values averaged out over 8 weeks are reported.

The average utilization rate is between 0.86 and 0.89 for the deterministic solutions and decreases if robustness is required. For the 80 patients case it is between 0.74 and 0.8 if $\Gamma = 1$ and between 0.57 and 0.68 if $\Gamma = 3$. It is even lower for the 120 patients case, being between on average 0.68 and 0.71 for $\Gamma = 1$ and between 0.57 and 0.62 for $\Gamma = 3$.

The fraction of undertime and overtime blocks are similar for different realizations, while average overtime and undertime amount vary.

The fraction of overtime blocks is very low: it is always less than 0.2, and it drops down to less than 0.05 if $\Gamma = 3$. On the contrary, of course, the fraction of undertime blocks is very high, especially when a high level of robustness is required: for $\Gamma = 3$ it is almost always above 0.9. In some weeks, for the 80 patients case, all blocks are undertime.

The small number of overtime blocks is mainly due to the conservative robust capacity constraints. When robustness is not required, although the fraction of overtime blocks is limited, the amount of overtime almost balanced the amount of undertime. Instead, when the level of required robustness increases, the amount of undertime may become significantly greater than the overtime, and may rise up to half the length of the block.

The overall approach is able to manage successfully a waiting list. If the level of robustness is low the waiting list size is reduced, while increasing the desired level of robustness reduces the number of operated patients. According to results, robustness has a positive impact on the number of cancelled patients, which is reduced dramatically. On the other hand, it increases the average undertime and reduces the utilization rate. The level of robustness must be therefore carefully set to take into account these conflicting goals.

4.3 Model performance

Having assessed the quality of obtained solutions, the computational performance of the applied models must be evaluated, to verify whether the approach can be used for a real life planning, which is usually performed on a weekly basis. The computational behavior is described in Tables 9 and 10, for the 80 and 120 initial patients instances, respectively.

The number of cases in which optimality (# opt) is proved, the number of cases in which time limit (#TL) or memory limit (# mem) are reached are reported.

Concerning the 80 initial patients instances, the deterministic model is solved to optimality within the time limit for all the weeks and all realizations for surgery lists 1 and 3. For surgery list 2, 38 of the considered cases out of 80 (namely number of weeks times number of realizations) is solved to optimality, while 26 exceed the time limit and in 16 cases CPLEX stops because of the search tree memory limitation. However, the gap never rises above 2.5%. When $\Gamma = 1$ the number of instances solved to optimality is similar for surgery list 1 and 3. For both it is about 63%. The

Instance	Util rate	over blocks	under blocks	Avg overtime	Avg undertime
surgery list 1	*0.86*	*0.18*	*0.83*	*76.92*	*79.13*
1	0.85	0.13	0.87	28.95	68.08
2	0.89	0.23	0.77	59.16	70.97
3	0.86	0.17	0.83	40.62	68.45
4	0.82	0.17	0.83	33.30	83.16
5	0.91	0.23	0.77	109.98	78.00
6	0.94	0.30	0.70	135.32	87.67
7	0.79	0.07	0.93	41.15	83.53
8	0.79	0.10	0.90	86.99	92.63
surgery list 2	*0.89*	*0.16*	*0.84*	*29.64*	*53.90*
1	0.86	0.10	0.90	17.76	59.96
2	0.92	0.17	0.83	44.03	41.96
3	0.90	0.13	0.87	26.00	46.32
4	0.89	0.13	0.87	18.86	46.57
5	0.89	0.27	0.73	27.53	66.36
6	0.88	0.07	0.93	33.04	49.69
7	0.89	0.27	0.73	29.28	63.86
8	0.88	0.17	0.83	36.51	59.82
surgery list 3	*0.88*	*0.13*	*0.87*	*36.41*	*57.12*
1	0.89	0.17	0.83	46.66	58.32
2	0.86	0.03	0.97	18.62	51.73
3	0.88	0.13	0.87	25.12	53.98
4	0.87	0.13	0.87	63.86	62.90
5	0.92	0.20	0.80	24.96	43.76
6	0.87	0.10	0.90	29.03	53.46
7	0.86	0.13	0.87	21.61	62.04
8	0.86	0.17	0.83	46.81	71.09

Table 3 Utilization rate, overtime and undertime, 80 initial patients, deterministic model

Instance	Util rate	over blocks	under blocks	Avg overtime	Avg undertime
surgery list 1	*0.74*	*0.10*	*0.90*	*91.10*	*116.06*
1	0.70	0.03	0.97	7.16	111.70
2	0.77	0.13	0.87	48.00	103.97
3	0.66	0.03	0.97	6.12	128.19
4	0.69	0.10	0.90	24.89	125.36
5	0.76	0.10	0.90	191.15	115.60
6	0.83	0.27	0.73	130.23	128.47
7	0.72	0.07	0.93	78.26	113.76
8	0.76	0.10	0.90	75.24	102.80
surgery list 2	*0.80*	*0.07*	*0.93*	*14.51*	*79.80*
1	0.82	0.10	0.90	9.97	73.10
2	0.82	0.00	1.00	-	66.56
3	0.80	0.03	0.97	3.44	73.37
4	0.78	0.03	0.97	0.07	80.81
5	0.80	0.07	0.93	11.93	78.55
6	0.80	0.10	0.90	17.06	83.76
7	0.75	0.03	0.97	1.32	92.02
8	0.80	0.17	0.83	24.48	92.18
surgery list 3	*0.80*	*0.07*	*0.93*	*27.11*	*80.55*
1	0.81	0.07	0.93	21.14	75.63
2	0.78	0.03	0.97	14.24	82.95
3	0.83	0.03	0.97	0.14	62.64
4	0.82	0.10	0.90	40.18	76.59
5	0.77	0.13	0.87	14.68	95.85
6	0.84	0.13	0.87	8.00	68.49
7	0.80	0.03	0.97	186.18	82.72
8	0.73	0.03	0.97	6.76	99.47

Table 4 Utilization rate, overtime and undertime, 80 initial patients, robust model $\Gamma = 1$

Instance	Util rate	over blocks	under blocks	Avg overtime	Avg undertime
surgery list 1	*0.57*	*0.05*	*0.95*	*78.86*	*166.06*
1	0.57	0.03	0.97	39.27	159.96
2	0.72	0.03	0.97	204.87	112.73
3	0.64	0.07	0.93	121.71	145.79
4	0.45	0.00	1.00	-	197.06
5	0.52	0.03	0.97	17.51	178.09
6	0.46	0.00	1.00	-	194.05
7	0.59	0.17	0.83	72.30	192.08
8	0.62	0.07	0.93	39.87	148.95
surgery list 2	*0.68*	*0.03*	*0.97*	*10.78*	*120.05*
1	0.66	0.00	1.00	-	123.73
2	0.76	0.07	0.93	12.01	95.13
3	0.67	0.00	1.00	-	120.29
4	0.68	0.07	0.93	2.40	124.31
5	0.67	0.03	0.97	10.78	123.20
6	0.65	0.00	1.00	-	125.38
7	0.68	0.07	0.93	21.62	125.76
8	0.67	0.03	0.97	3.42	121.80
surgery list 3	*0.69*	*0.03*	*0.97*	*16.61*	*115.48*
1	0.68	0.07	0.93	16.09	124.44
2	0.72	0.03	0.97	4.48	103.64
3	0.68	0.00	1.00	-	116.90
4	0.73	0.10	0.90	26.56	110.16
5	0.73	0.03	0.97	16.37	100.75
6	0.65	0.00	1.00	-	127.34
7	0.66	0.00	1.00	-	121.36
8	0.68	0.03	0.97	0.22	118.50

Table 5 Utilization rate, overtime and undertime, 80 initial patients, robust model $\Gamma = 3$

Instance	Util rate	over blocks	under blocks	Avg overtime	Avg undertime
surgery list 1	*0.83*	*0.17*	*0.83*	*90.88*	*92.97*
1	0.93	0.23	0.78	124.75	68.13
2	0.87	0.25	0.75	106.20	98.87
3	0.82	0.10	0.90	128.63	86.68
4	0.81	0.08	0.93	38.60	75.13
5	0.74	0.15	0.85	49.70	118.65
6	0.84	0.20	0.80	86.97	92.04
7	0.80	0.15	0.85	117.04	106.05
8	0.82	0.23	0.78	54.13	99.14
surgery list 2	*0.82*	*0.11*	*0.89*	*49.43*	*80.54*
1	0.88	0.10	0.90	34.38	50.45
2	0.82	0.08	0.93	43.00	72.17
3	0.83	0.10	0.90	38.98	72.89
4	0.89	0.20	0.80	51.95	61.67
5	0.78	0.15	0.85	76.78	106.76
6	0.80	0.13	0.88	59.95	88.94
7	0.80	0.08	0.93	29.27	78.49
8	0.72	0.08	0.93	31.08	111.96
surgery list 3	*0.78*	*0.09*	*0.91*	*32.55*	*89.09*
1	0.85	0.05	0.95	68.37	61.11
2	0.88	0.10	0.90	29.66	51.36
3	0.80	0.10	0.90	46.30	86.37
4	0.73	0.05	0.95	54.87	106.79
5	0.72	0.10	0.90	19.99	115.21
6	0.71	0.08	0.93	17.56	114.63
7	0.82	0.13	0.88	15.42	75.76
8	0.76	0.10	0.90	37.83	100.97

Table 6 Utilization rate, overtime and undertime, 120 initial patients, deterministic model

Instance	Util rate	over blocks	under blocks	Avg overtime	Avg undertime
surgery list 1	*0.68*	*0.10*	*0.90*	*81.79*	*139.39*
1	0.72	0.13	0.88	86.92	128.86
2	0.76	0.08	0.93	175.29	107.81
3	0.62	0.08	0.93	80.31	154.35
4	0.72	0.15	0.85	34.79	124.34
5	0.60	0.10	0.90	41.02	164.29
6	0.68	0.10	0.90	78.01	138.29
7	0.62	0.08	0.93	157.47	160.67
8	0.70	0.13	0.88	68.11	135.09
surgery list 2	*0.71*	*0.05*	*0.95*	*24.55*	*111.01*
1	0.80	0.05	0.95	17.84	78.32
2	0.76	0.05	0.95	9.37	92.72
3	0.72	0.05	0.95	45.86	108.37
4	0.74	0.10	0.90	20.59	107.17
5	0.67	0.05	0.95	9.07	125.37
6	0.72	0.05	0.95	52.89	108.58
7	0.66	0.05	0.95	19.50	128.24
8	0.63	0.03	0.98	25.85	138.43
surgery list 3	*0.70*	*0.06*	*0.94*	*26.73*	*115.26*
1	0.76	0.08	0.93	48.32	97.27
2	0.81	0.13	0.88	24.27	83.31
3	0.78	0.10	0.90	15.73	88.86
4	0.69	0.08	0.93	25.94	122.12
5	0.60	0.03	0.98	14.01	146.69
6	0.69	0.05	0.95	45.80	119.70
7	0.62	0.03	0.98	18.92	139.14
8	0.68	0.03	0.98	3.06	119.26

Table 7 Utilization rate, overtime and undertime, 120 initial patients, robust model $\Gamma = 1$

Instance	Util rate	over blocks	under blocks	Avg overtime	Avg undertime
surgery list 1	*0.57*	*0.06*	*0.94*	*68.92*	*166.84*
1	0.67	0.10	0.90	39.01	135.64
2	0.60	0.05	0.95	53.46	153.04
3	0.66	0.10	0.90	51.22	142.71
4	0.59	0.00	1.00	-	149.05
5	0.49	0.05	0.95	68.61	197.27
6	0.54	0.08	0.93	158.24	190.01
7	0.49	0.08	0.93	53.59	204.21
8	0.55	0.00	1.00	-	162.64
surgery list 2	*0.62*	*0.03*	*0.97*	*30.16*	*142.29*
1	0.68	0.00	1.00	-	115.22
2	0.64	0.03	0.98	30.39	133.88
3	0.57	0.00	1.00	-	153.21
4	0.69	0.05	0.95	13.26	117.45
5	0.53	0.03	0.98	33.11	175.38
6	0.61	0.05	0.95	36.66	149.52
7	0.61	0.03	0.98	66.66	146.20
8	0.61	0.05	0.95	20.71	147.57
surgery list 3	*0.61*	*0.03*	*0.97*	*43.20*	*147.26*
1	0.68	0.08	0.93	53.43	127.05
2	0.73	0.00	1.00	-	98.03
3	0.70	0.03	0.98	25.91	112.65
4	0.66	0.08	0.93	16.13	133.30
5	0.43	0.00	1.00	-	205.15
6	0.44	0.00	1.00	-	201.67
7	0.60	0.05	0.95	78.02	154.57
8	0.62	0.03	0.98	41.37	142.44

Table 8 Utilization rate, overtime and undertime, 120 initial patients, robust model $\Gamma = 3$

	Week	surgery list 1 # opt	# TL	# mem	surgery list 2 # opt	# TL	# mem	surgery list 3 # opt	# TL	# mem
deterministic	1	10	0	0	0	10	0	10	0	0
	2	10	0	0	1	8	1	10	0	0
	3	10	0	0	0	6	4	10	0	0
	4	10	0	0	2	2	6	10	0	0
	5	10	0	0	5	0	5	10	0	0
	6	10	0	0	10	0	0	10	0	0
	7	10	0	0	10	0	0	10	0	0
	8	10	0	0	10	0	0	10	0	0
	total (out of 80)	80	0	0	38	26	16	80	0	0
	percentage	100	0	0	48	33	20	100	0	0
$\Gamma = 1$	1	0	10	0	0	0	10	10	0	0
	2	0	1	9	0	1	9	3	7	0
	3	0	1	9	0	2	8	1	9	0
	4	10	0	0	0	1	9	1	9	0
	5	10	0	0	0	1	9	6	4	0
	6	10	0	0	2	3	5	10	0	0
	7	10	0	0	4	3	3	10	0	0
	8	10	0	0	10	0	0	10	0	0
	total (out of 80)	50	12	18	16	11	53	51	29	0
	percentage	63	15	23	20	14	66	64	36	0
$\Gamma = 3$	1	10	0	0	0	10	0	0	10	0
	2	10	0	0	0	3	7	0	10	0
	3	8	2	0	0	3	7	0	10	0
	4	0	10	0	0	5	5	0	10	0
	5	0	10	0	0	5	5	0	10	0
	6	0	10	0	0	8	2	0	10	0
	7	0	10	0	0	10	0	0	10	0
	8	0	10	0	0	10	0	0	10	0
	total (out of 80)	28	52	0	0	54	26	0	80	0
	percentage	35	65	0	0	68	33	0	100	0

Table 9 Computational performance on 80 initial patients instances

remaining cases of surgery list 3 reach the time limit, while 23% of surgery list 1 cases run out of memory. For the remaining 15% time limit is reached. Surgery list 2 has a large number of cases, 53 out of 80, in which CPLEX runs out of memory, and only 20% is solved to optimality. The gap never rises above 6%, but it is in general significantly lower (about 1.2%). When $\Gamma = 3$ for none of the cases in surgery list 1 and 3 CPLEX runs out of memory. As for surgery list 1, 35% of the cases are solved to optimality, and for the remaining the gap never rises above 6%. For surgery list 3 all the cases reached the time limit with a worst gap of about 11.5%.

Concerning the 120 initial patients case, the deterministic model always proves optimality for surgery list 1, it solves to optimality 19% of cases for surgery list 2 and 90% for surgery list 3. CPLEX runs out of memory for 80% of cases in surgery list 2. However the gap is limited, being always below 2%. With $\Gamma = 1$ 33% of cases in surgery list 1 is solved to optimality, and none for surgery list 2 and 3. For surgery list 2 CPLEX runs out of memory for 93% of cases, while 90% of cases reach the time limit for surgery list 3. On the overall, setting $\Gamma = 1$ produces an average gap of about 4% and a maximum of about 9%. Increasing the level of robustness reduces the number of optima found: in fact optimality is never proved. The time limit is reached in 60% of cases for surgery list 1, and in 15% for surgery lists 2 and 3. The overall average and maximum gaps rise up to about 11% and 21%, respectively.

5 Conclusions

In this work we proposed and tested an approach aimed to solve the Advanced Scheduling problem assuming a block scheduling strategy. To guarantee a certain level of performance with respect of quality of service, we consider a penalty function associated to waiting time, urgency and tardiness of patients. We considered two sources of uncertainty that complicate the problem:

	Week	surgery list 1 # opt	# TL	# mem	surgery list 2 # opt	# TL	# mem	surgery list 3 # opt	# TL	# mem
deterministic	1	10	0	0	0	0	10	10	0	0
	2	10	0	0	0	0	10	10	0	0
	3	10	0	0	0	0	10	9	0	1
	4	10	0	0	0	0	10	7	1	2
	5	10	0	0	0	0	10	10	0	0
	6	10	0	0	1	0	9	10	0	0
	7	10	0	0	6	0	4	10	0	0
	8	10	0	0	8	1	1	10	0	0
	total (out of 80)	80	0	0	15	1	64	76	1	3
	percentage	100	0	0	19	1	80	95	1	4
$\Gamma = 1$	1	0	0	10	0	0	10	0	10	0
	2	0	0	10	0	0	10	0	8	2
	3	0	0	10	0	1	9	0	9	1
	4	0	1	9	0	0	10	0	9	1
	5	0	0	10	0	0	10	0	7	3
	6	7	1	2	0	1	9	0	10	0
	7	9	1	0	0	3	7	0	9	1
	8	10	0	0	0	1	9	0	10	0
	total (out of 80)	26	3	51	0	6	74	0	72	8
	percentage	33	4	64	0	8	93	0	90	10
$\Gamma = 3$	1	0	0	10	0	0	10	0	0	10
	2	0	0	10	0	0	10	0	0	10
	3	0	1	9	0	0	10	0	0	10
	4	0	7	3	0	0	10	0	0	10
	5	0	10	0	0	0	10	0	0	10
	6	0	10	0	0	1	9	0	8	2
	7	0	10	0	0	2	8	0	1	9
	8	0	10	0	0	9	1	0	3	7
	total (out of 80)	0	48	32	0	12	68	0	12	68
	percentage	0	60	40	0	15	85	0	15	85

Table 10 Computational performance on 120 initial patients instances

(1) new patients arrivals that occur within the planning horizon and (2) surgery times, that are only roughly predictable.

We tackled issue (1) by adopting a rolling horizon approach with reoptimization. At the beginning of each week a time window of several weeks is planned, and the first planned week is implemented; the plan for the other weeks is kept as a reference for the next iterations. Each plan is produced by solving an optimization model.

We tackled issue (2) by adopting robust optimization models that allow to specify a robustness level Γ. We stress that, although the realization of operating times were drawn from probability distributions in testing, the considered model are not stochastic — instead, they guarantee the feasibility of the generated solution when at most Γ operating times get the worst possible realization in a given interval. When $\Gamma = 0$ the robust model reduces to the deterministic version where all operating times are assumed to be known without uncertainty.

The computational results, obtained from testing with data and probability distributions from real-life instances, are promising. First of all the models remain reasonably well solvable in all cases: by this we mean that small optimality gaps are reached within the specified time limit for computations — since the planning is an offline activity, even allowing two full hours of CPU time for is not a severe restriction. Furthermore, starting from the deterministic model and moving towards more stringent robustness requirements (increasing Γ) we saw a shift from solutions that show a better resources utilization, hence possibly appealing to the management of an hospital, to solutions that strongly limit the number of cancelled operations, hence somehow more patient-oriented. Because of this we think that the proposed models could be effective decision-support tools in the mid-term Operating Rooms planning.

References

B. Addis, G. Carello, and E. Tànfani. A robust optimization approach for the operating room planning problem with uncertain surgery duration. In A. Matta, J. Li, E. Sahin, E. Lanzarone, and J. Fowler, editors, *Proceedings of the International Conference on Health Care Systems Engineering*, volume 61 of *Springer Proceedings in Mathematics & Statistics*, pages 175–189. Springer International Publishing, 2014.

V. Augusto, X. Xie, and V. Perdomo. Operating theatre scheduling using lagrangian relaxation. *European Journal of Industrial Engineering*, 2(2):172–189, 2008.

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

J.T. Blake and M.W. Carter. Surgical process scheduling: a structured review. *Journal of the Society for Health Systems*, 5(3):17–30, 1997.

B. Cardoen, E. Demeulemeester, and J. Beliën. Sequencing surgical cases in a day-care environment: An exact branch-and-price approach. *Computers & Operations Research*, 36(9):2660–2669, 2009.

B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201:921–932, 2010a.

B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling problems: A classification scheme. *International Journal of Health Management and Information*, 1(1): 71–83, 2010b.

B. Denton, J. Viapiano, and A. Vogl. Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 10:13–24, 2007.

B. Denton, J. Miller, H.J. Balasubramanian, and T.R. Huschka. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research*, 58:802–816, 2010.

H. Fei, C. Chu, N. Meskens, and A. Artiba. Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics*, 112:96–108, 2008.

F. Guerriero and R. Guido. Operational research in the management of the operating theatre: a survey. *Health Care Management Science*, 14:89–114, 2011.

E. Hans, G. Wullink, M. van Houdenhoven, and G. Kamezier. Robust surgery loading. *European Journal of Operational Research*, 185:1038–1050, 2008.

W.L. Herring and J.W. Herrmann. Local search for the surgery admission planning problem. *Journal of Heuristics*, 17:389–414, 2011.

W.L. Herring and J.W. Herrmann. The single-day surgery scheduling problem: sequential decision-making and threshold-based heuristics. *OR Spectrum*, 34:429–459, 2012.

J.M. Magerlein and J.B. Martin. Surgical demand scheduling: A review. *Health Services Research*, 13:418–433, 1978.

I. Marques, M.E. Captivo, and M.V. Pato. An integer programming approach to elective surgery scheduling. *OR Spectrum*, 34:407–427, 2012.

D. Min and Y. Yih. Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206:642–652, 2010.

C. Rizk and J.P. Arnaout. Aco for the surgical cases assignment problem. *Journal of Medical Systems*, 36:1191–1199, 2012.

K. Stuart and E. Kozan. Reactive scheduling model for the operating theatres. *Flexible Services and Manufacturing Journal*, 24:400–421, 2012.

E. Tànfani and A. Testi. A pre-assignment heuristic algorithm for the master surgical schedule problem (mssp). *Annals of Operations Research*, 178(1):105–119, 2010.

E. Tànfani, A. Testi, and R. Alvarez. Operating room planning considering stochastic surgery durations. *International Journal of Health Management and Information*, 1(2):167–183, 2010.

A. Testi, E. Tànfani, and G.C. Torre. A three-phase approach for operating theatre schedules. *Health Care Management Science*, 10:163–172, 2007.

D.C. Tyler, C.A. Pasquariello, and C.H. Chen. Determining optimum operating room utilization. *Anesthesia and Analgesia*, 96(4):1114–1121, 2003.

R. Valente, A. Testi, E. Tànfani, M. Fato, I. Porro, M. Santo, G. Santori, G.C. Torre, and G.L. Ansaldo. A model to prioritize access to elective surgery on the base of clinical urgency and

waiting time. *BMC, Health Services ResearchAnnals of Operations Research*, 9(1), 2009.
J.M. van Oostrum, M. van Houdenhoven, J.L. Hurink, E.W. Hans, G. Wullink, and G. Kazemier. A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum*, 30:355–374, 2008.
J.M. van Oostrum, E. Bredenhoff, and E.W. Hans. Suitability and managerial implications of a master surgical scheduling approach. *Annals of Operations Research*, 178(1):91–104, 2010.

A Robust model linearization

We report here the linearization of the robust conterpart of the capacity contraints. We recall that surgery times $\tilde{t}_i$ are the uncertainty parameters, and that we want to guarantee feasibility to variations in a given interval $\bar{t}-\hat{t}, \bar{t}+\hat{t}$. In particular, at most Γ parameters are allowd to assume the maximum value $\bar{t}_i + \hat{t}_i$, while all the others assume the central value $\bar{t}$.

Then, for each block j, and week k, a subset S_{jk} of patients, who require their maximum surgery time, such that $|S_{jk}| = \Gamma$, is chosen among the patients assigned to the block in the given week. Among all the possible subsets, the one having the worst impact on the capacity constraint is selected, and the solution is guaranteed to be feasible even with respect to this subset:

$$\sum_{i\in I} \bar{t}_i x_{ij}^k + \max_{S_{jk}\subset I:|S_{jk}|=\Gamma} \left\{ \sum_{i\in S_{jk}} \hat{t}_i x_{ij}^k \right\} \leq \gamma_{jk} \qquad \forall j \in J, \quad \forall k \in K \tag{28}$$

The value $\max_{S_{jk}\subset I:|S_{jk}|=\Gamma} \left\{ \sum_{i\in S_{jk}} \hat{t}_i x_{ij}^k \right\}$ can be computed for each block j and each week k solving the following Linear Programming problem:

$$\beta^{jk} = \max \left(\sum_{i\in I} \hat{t}_i x_{ij}^k \right) z_i \tag{29}$$

$$\sum_{i\in I} z_i \leq \Gamma \tag{30}$$

$$z_i \leq 1 \qquad \forall i \in I \tag{31}$$

$$z_i \geq 0 \qquad \forall i \in I \tag{32}$$

Let denote with ζ^{jk} the dual variables associated to constraints (30) and with π_i^{jk} the dual variables associated to constraints (31). The dual of (β^{jk}) can be formulated as follows:

$$\min \quad \Gamma\zeta^{jk} + \sum_{i\in I} \pi_i^{jk} \tag{33}$$

$$\zeta^{jk} + \pi_i^{jk} \geq \hat{t}_i x_{ij}^k \qquad \forall i \in I \tag{34}$$

$$\zeta^{jk}, \pi_i^{jk} \geq 0 \tag{35}$$

The optimal values of the objective functions (29) and (33) coincide. Thus, constraints (28) can be linearized, by replacing them with:

$$\sum_{i\in I} \bar{t}_i x_{ij}^k + \Gamma\zeta^{jk} + \sum_{i\in I} \pi_i^{jk} \leq \gamma_{jk} \qquad \forall j \in J \quad \forall k \in K \tag{36}$$

$$\zeta^{jk} + \pi_i^{jk} \geq \hat{t}_i x_{ij}^k \qquad \forall j \in J \quad \forall i \in I \tag{37}$$

$$\zeta^{jk}, \pi_i^{jk} \geq 0 \tag{38}$$