A hybrid optimization algorithm for surgeries scheduling

A hybrid optimization algorithm for surgeries scheduling

Paolo Landa, Roberto Aringhieri, Patrick Soriano, Elena Tànfani, Angela Testi

PII: S2211-6923(15)30008-4
DOI: http://dx.doi.org/10.1016/j.orhc.2016.01.001
Reference: ORHC 96

To appear in: *Operations Research for Health Care*

Received date: 7 May 2015
Accepted date: 4 January 2016

A hybrid optimization algorithm for surgeries scheduling

Paolo Landa[a,*], Roberto Aringhieri[b], Patrick Soriano[c], Elena Tànfani[a], Angela Testi[a]

[a]*Department of Economics and Business Studies, University of Genova, Italy*
[b]*Department of Computer Science, University of Torino, Italy*
[c]*Department of Management Sciences and CIRRELT, HEC Montréal, Canada*

Abstract

This paper deals with the Operating Room (OR) planning problem at an operational planning level. The problem addressed consists in two interrelated sub-problems usually referred to as "advance scheduling" and "allocation scheduling". In the first sub-problem, the decisions considered are the assignment of a surgery date and an OR block to a set of patients to be operated on over a given planning horizon. The second aims at determining the sequence of selected patients in each OR and day. We assume that the duration of surgeries are random variables with known probability distributions. For each sub-problem an integer linear stochastic formulation is given. A hybrid two-phase optimization algorithm which exploits the potentiality of neighborhood search techniques combined with Monte Carlo simulation is developed to solve the overall problem. The approach developed searches for a feasible and robust solution designed to balance the trade-off arising between the hospital and patient perspectives, i.e. maximizing the OR utilization and minimizing the number of patient cancellations. The contribution of this paper is twofold. The former, more methodological, is to provide an efficient algorithmic framework to solve the joint advance and allocation scheduling problem taking into account the inherent uncertainty of surgery durations. The latter, more practical, is to provide a tool to develop robust offline OR schedules which consider the trade-off between reducing surgery cancella-

[*]Corresponding author

Email addresses: paolo.landa@unige.it (Paolo Landa), roberto.aringhieri@unito.it (Roberto Aringhieri), patrick.soriano@hec.ca (Patrick Soriano), etanfani@economia.unige.it (Elena Tànfani), testi@economia.unige.it (Angela Testi)

tions and postponements while maximizing the operating theater utilization. To evaluate the efficiency of the proposed algorithmic approach, in terms of quality of solutions and solution time, we provide a computational analysis on a set of instances based on real data.

Keywords: Operating room planning and scheduling, uncertain surgery duration, Metaheuristic, Monte Carlo simulation
2000 MSC: 90B50, 90C11, 90C27, 90C59

1. Introduction

Operating Rooms (ORs) are one of the most expensive resources in hospitals. Their management typically needs to take into account numerous factors (e.g., personnel availability, surgical instruments, ICU and ward bed capacities, etc.) and involves the actions of different players, such as surgeons, nurses and patients [1].

The management of ORs has been a challenging research topic over the last decades. Recently, exhaustive literature reviews [2, 3] on operating room planning and scheduling problems have been published classifying the different problem versions by using multiple fields and perspectives. The literature on OR planning problems is usually classified starting by the level of decisions taken in the analysis. In particular strategic (long term), tactical (medium term) and operational (short term) problems deal, respectively, with the case mix planning, master surgery scheduling and patient scheduling problems, even if there is not a univocal definition of the problems addressed [2]. The operational level is usually decomposed into two phases: advance scheduling and allocation scheduling. The first, also referred to as surgical case assignment problem (SCAP), assigns a surgery day and an OR to a set of elective patients waiting for surgery; while the allocation scheduling determines the sequencing of the assigned patients in each OR and day.

Another major classification item refers to whether or not uncertainty is incorporated into the analysis (stochastic versus deterministic contexts). The main sources of uncertainty considered are usually associated with the arrival of emergency patients, the patients' length of stay and surgery durations.

In this paper we deal with the joint advance and allocation scheduling problem (operational level) while assuming that patient surgery durations are stochastic variables which follow a priori given distribution functions. Note that the uncertainty pertaining to surgery duration has a major and direct impact on the quality of the schedules and it has been largely addressed in the literature. In [4] the authors develop a stochastic programming model with recourse and a sample average approximation method for the SCAP with the aim of minimizing patients costs and OR overtime costs. Dealing with the same problem, in [1] a column generation approach to maximize the OR utilization and level the requirements for hospital beds in two subsequent phases is proposed, while in [5] the authors develop a local search heuristic aimed at maximizing the utilization of operating theater and minimizing the overtime risks by introducing planned slack times. In [6] a two-stage stochastic model with binary decision variables and simple recourse is introduced. The model determines the assignment of surgeries to ORs by minimizing the maximum cost associated with uncertain surgery durations. In [7] a cardinality-constrained robust optimization approach is proposed. The approach allows to exploit the flexibility of a linear programming model but does not require the generation of a huge number of scenarios. The model minimizes a patient centered objective function, which takes into account waiting time, urgency and tardiness of patients. In [8], the authors propose a chance constrained model which maximizes the expected utilization of ORs, the solution algorithm solves a series of MIP models based on a normal approximation of cumulative surgery durations. In [9, 10] the authors propose an online approach to adjust a patient schedule affected by a delay during its execution. Possible adjustments are the allocation of some overtime or rescheduling the patient surgery. A hybrid simulation and offline optimization model is developed to evaluate the impact of the online algorithm.

Few papers have addressed the simultaneous assignment and sequencing of patients, i.e. the advance scheduling combined with the allocation scheduling. In [11] the authors develop an optimization model to manage surgery time uncertainty using a two-stage stochastic model with recourse, including in the objective function the patient waiting times as well as the OR idle time and overtime. Other authors [12] develop a two-stage stochastic MIP model that determines the allocation of patients to ORs, the sequence of surgeries and the start time for each surgeon. The aim in their work is to minimize the total expected operating cost. More recently in [13], a rolling horizon approach combined with robustness is used to schedule and reschedule patients into OR blocks. The sequencing of patients is introduced as a component of the overall procedure which aims at keeping limited the number of disruptions and patient cancellations.

Finally, some authors have used simulation to compare different scheduling and sequencing strategies and tested the solution robustness against the randomness of surgery duration (see, e.g., [9, 10, 14–17]).

The contribution of this paper is twofold. The former, more methodological, is to provide an efficient algorithmic framework to solve the joint advance and allocation scheduling problem taking into account the inherent uncertainty of surgery durations. The latter, more practical, is to provide a tool to develop robust offline OR schedules which consider the trade-off between reducing surgery cancellations and postponements while maximizing the operating theater utilization.

For each problem a stochastic mathematical programming formulation is given. Afterwards, a hybrid two-phase optimization algorithm is developed. The method exploits neighborhood search techniques combined with Monte Carlo simulation to deal with the uncertainty of surgery durations. The hybrid approach searches for a feasible and robust solution designed to balance the trade-off arising between the hospital and patient perspectives, i.e., maximizing the OR utilization and minimizing the number of patient cancellations. This trade-off is strongly influenced by the sequencing of the patients within the OR blocks [17] which is explicitly taken into account here.

The proposed hybrid approach is tested on a set of instances based on real data. Experiments are reported to analyze the impact of varying search criteria, operating time distributions parameters, as well as the critical overtime probability level. The paper is organized as follows. In Section 2 the problem is formally introduced and the stochastic optimization formulations for the two problems reported. The components of the proposed hybrid solution algorithm are then given and explained in Section 3 while Section 4 reports and analyzes the computational experiments. Section 5 closes the paper discussing also the implementation of our approach in the hospital practice.

2. Problem statement and model formulations

We consider a surgery department composed of several specialties or disciplines which share a set of available ORs. We assume a block scheduling management strategy where the surgical specialties are assigned a given number of OR blocks over a given planning horizon in which they can schedule their elective patients. The duration of each OR block is determined a priori but we assume that for the whole operating theater a certain amount of overtime can be used to avoid surgery cancellations if some blocks are running late. The problem addressed herein aims at determining the surgery dates and ORs for a set of patients to be operated on, while determining the sequence of patients within each OR and day. A planning horizon of one week is considered.

Let I, J, and K be respectively the sets of patients, surgical specialties, and operating rooms, each indexed by the corresponding lower cased letter, i, j, and k. The surgery days within the planning horizon are from Monday to Friday. Therefore $T = \{1, ..., 5\}$, indexed by t, denotes the set of dates corresponding to the surgery days to be scheduled.

For each patient i, the expected deterministic surgery duration p_i, the stochastic surgery duration ε_i and the expected Length of Stay (LOS) μ_i are given. In addition, let I_j be the subset of patients that belong to specialty j, $j \in J$, and I_h the subset of patients having LOS $\mu_i = h$, $h = 1, ..., \mu_{max}$, where $\mu_{\max}$ represents the longest LOS. Clearly, subsets I_j define a partition of I as do subsets I_h.

In the practical setting of the hospital considered here, each specialty j has its own post-surgery beds from Monday to Friday and there is no availability restriction, whereas the weekend stay beds are a shared resource among all surgical specialties for which availability is limited (given limited hospital staffing over the weekends). Let χ be the maximum number of common stay beds available for the department in the weekend. Let us also define T_h as the subset of T for which patients with a LOS h require a bed for the weekend if scheduled on any of the days in T_h. As an example, those patients having a LOS equal to 3 days can be scheduled on Monday, Tuesday or Wednesday without requiring a weekend bed (since they can be discharged before Friday evening). To the contrary, if one such patient is scheduled either on Thursday or Friday it will need to stop hospitalized during the weekend and therefore $T_3 = \{4, 5\}$.

Each OR time block within the planning horizon is uniquely defined by a pair of indices (k, t) which represents the OR k and the day t of the week when the block is scheduled. Let s_{kt} be the length of OR time block (k, t), i.e., the time available for surgery in that block. Following a block scheduling operating policy, each specialty j is assigned to a different set of OR blocks where it can schedule its surgical cases. The number and distribution of the OR blocks (k, t) available for each specialty during the week is given by the cyclic timetable of the department, referred to as the Master Surgical Schedule (MSS). Let Π be a matrix of binary values used to input the MSS, such that element π^j_{kt} is set to 1 if specialty j is assigned to OR k on day t

and 0 otherwise.

Finally, we suppose that a maximum number of overtime units, noted L, are available for the overall planning horizon. Each overtime unit has a duration equal to ℓ and we suppose that $\ell \leq \min_{i \in I} \varepsilon_i$ so that the addition of an overtime unit will not enable the realization of a surgery that otherwise could not have been started before the normal end of the OR block. The total amount of overtime available for the department is therefore equal to $L \times \ell$.

Our problem can be seen as a special case of the “surgery process scheduling” problem which is usually composed of two subsequent steps [18, 19]. The first step (advance scheduling or SCAP) consists in assigning a specific OR time block to each patient over the planning horizon. The second step (allocation scheduling) determines the detailed sequencing of surgical procedures within each block and the allocation of the resources needed to perform them as efficiently as possible.

According to this hierarchical view, in Section 2.1 and 2.2 we propose two mathematical formulations that address these two steps and allow to outline the different decisions and objectives considered in our approach. To face the surgery duration variability, we adopt a modeling approach based on chance constraint programming [20]. In this approach, the focus is on the reliability of the system, i.e., the ability of the system to satisfy feasibility in an uncertain environment. This reliability is expressed as a minimum requirement on the probability of satisfying constraints [21].

2.1. The patient assignment model

The first step seeks a robust assignment of the set I of patients to the available OR time blocks (k, t) in such a way as to maximize the overall OR utilization while leveling the OR blocks utilization over the planning horizon. The solution must satisfy the operational constraints regarding the OR blocks length and the maximum number of beds available during the weekend, following the “week surgery” ward organization discussed earlier.

In health care settings characterized by long waiting list, the set of patients I who must be scheduled in the planning horizon is usually determined on the basis of patient expected surgery durations, while taking into account the patients urgency and priority.

We consider x_{ikt} as the binary decision variables of the problem, with the following definitions:

$$x_{ikt} = \begin{cases} 1 & \text{if patient } i \in I \text{ is assigned to OR } k \in K \text{ on day } t \in T; \\ 0 & \text{otherwise.} \end{cases}$$

Variables x_{ikt} allow the following formulation of the problem:

$$\begin{aligned}
\max \quad & z = y && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{k\in K, t\in T} x_{ikt} \le 1 && \forall i \in I && \text{(1b)}\\
& \sum_{i \in I_j} x_{ikt} \le \pi^j_{kt} |I_j| && \forall j \in J, k \in K, t \in T && \text{(1c)}\\
& \sum_{h=1}^{\mu_{\max}} \sum_{i\in I_h} \sum_{t\in T_h} \sum_{k\in K} x_{ikt} \le \chi && && \text{(1d)}\\
& \sum_{i\in I} \varepsilon_i x_{ikt} = s_{kt} + y_{kt} && \forall k \in K, t \in T && \text{(1e)}\\
& \mathbb{P}\left[y_{kt} \ge 0\right] \le \alpha && \forall k \in K, t \in T && \text{(1f)}\\
& y_{kt} \ge y && \forall k \in K, t \in T && \text{(1g)}\\
& x_{ikt} \in \{0,1\}\,,\ y_{kt} \in \mathbb{R},\ y \in \mathbb{R} && && \text{(1h)}
\end{aligned}$$

Constraints (1b) are the assignment constraints implying that each patient can be scheduled at most once during the planning horizon. Constraints (1c) ensure that each patient $i \in I_j$ can only be assigned to an OR time block that is assigned to the specialty $j \in J$, to which the patient belongs, i.e., that is one for which $\pi^j_{kt} = 1$. The term $|I_j|$ enables the assignment of more than one patient of the specialty to a given OR block (provided that the available time is sufficient). The weekend stay bed availability constraint (1d) ensures that the number of patients requiring a bed for the weekend is less than or equal to χ, the maximum number of beds available for the weekend. Note that constraint (1d) can be easily applied to deal with bed constraints for every day of the planning horizon if required.

Let us introduce a set of auxiliary stochastic slack variables $y_{kt} \in \mathbb{R}$. Constraints (1e) define their values in such a way that when $y_{kt} > 0$, y_{kt} measures the overtime needed to complete all the surgeries assigned to OR block (k,t). On the other hand, when $y_{kt} < 0$, the variable measures the unused operating time (undertime) in OR block (k,t). The solution reliability is expressed by the chance constraints (1f), which limits the probability $\alpha \in [0, 1)$ that a given OR block (k, t) needs some amount of overtime to complete the surgeries assigned.

To maximize the overall utilization, we adopted a bottleneck objective function aimed at maximizing the minimal utilization of OR time blocks (k, t): the basic idea is to minimize the undertime of the OR time block less utilized. This is modeled as usual by introducing an auxiliary variable $y \in \mathbb{R}$ forced by constraints (1g) to be lesser or equal to the minimum value of y_{kt}, and then maximizing the variable y in the objective function (1a). The chance constraint (1f) guarantees the correctness of this formulation since it imposes that a given number of OR time blocks (k, t) are in undertime, that is the corresponding y_{kt} are less than or equal to zero.

Let us recall that the chance constrained model (1a)–(1f) solves the stochastic patient advance scheduling step by determining the robust assignment of patients to OR blocks over the planning horizon. The patient sequence within each OR block (allocation scheduling step) is not yet determined. In fact, the overtime and undertime computed in (1e) are not affected by the patient sequencing because we assume that surgery duration distributions are mutually independent. In order to deal with this aspect, the overtime allocation model is developed and reported in the next section.

2.2. The overtime allocation model

The schedule determined by the previous model gives the robust assignment of patients to OR blocks and indicates the OR blocks which need some amount of overtime in order to operate on all patients in I. Indeed the obtained schedule maximizes the OR utilization and levels the occupation among the set of available OR blocks.

What remains to be determined is how should any available overtime be allocated to the different OR blocks over the planning horizon in order to avoid surgery cancellations or at least reduce them as much as possible.

Recall that $L \times \ell$ is the total amount of overtime available for the whole department during the planning horizon considered, where L and ℓ are the number of overtime units available and the duration of each overtime unit, respectively. This assumption is quite close to what is observed in practice in many hospitals where a given predetermined *budget* of overtime will generally be available. This overtime is then allocated, as the need arises, among specific ORs that are running late in multiples of 15 or 30 minutes – in order to complete the procedures in progress or perform some that have not yet started but will clearly go beyond the duration of the block so as to reduce cancellations. It cannot be divided exactly as the y_{kt} values would indicate since the assignment of overtime must be decided *online* before the real duration of the procedures becomes known with certainty.

Starting with the solution of model (1a)–(1h), let B be the set of OR blocks (k,t) such that $y_{kt} \geq 0$. The value of $y_{kt} \geq 0$ measures the overtime required to accomplish all the surgeries assigned to OR block (k,t). The sum $\sum_{(k,t)\in B} y_{kt}$, gives the total overtime required to perform all the surgeries in the schedule. If it is greater than $L \times \ell$, the decision regarding the overtime allocation becomes crucial in order to minimize the number of patients canceled from the schedule.

Let I_{kt} be the set of patients assigned to OR block $(k,t) \in B$, i.e. $i \in I_{kt}$ if and only if $x_{ikt} = 1$. Let us introduce the set O_{kt} as the set of all possible (ordered) sequences of patients in I_{kt}, $\forall (k,t) \in B$, and define $\mathcal{O} = \cup_{(k,t)\in B} O_{kt}$. Let us also define I^o_{kt} as an ordered subset of patients for $(k,t) \in B$ and $o \in O_{kt}$, where the order of patients in I^o_{kt} is such that if $i < i'$ then the surgery of patient i is scheduled before that of patient i'. Finally, let p^o_{kt} denote the first patient in I^o_{kt} whose surgery needs overtime to be finished and let $\varphi_{p^o_{kt}}$ be the amount of surgery time for that patient that exceeds s_{kt}, the duration of block (k,t), with $\varphi_{p^o_{kt}} < \varepsilon_{p^o_{kt}}$ of course.

The decision variables $u_{kt} \in \mathbf{Z}^+$ give the number of overtime units assigned to each time block $(k,t) \in B$. Indeed, for a given $o \in O_{kt}$, let z^i_{kt} be an auxiliary binary variable equal to 1 if patient $i \in I^o_{kt}$ is not covered by the overtime assigned to OR time block $(k,t) \in B$, 0 otherwise.

For any given $o \in \mathcal{O}$, the overtime allocation problem can be formulated as follows, where to simplify notation we use p in place of p^o_{kt}, i.e., $p \equiv p^o_{kt}$.

$$\min \quad z(o) = \sum_{(k,t)\in B} \sum_{i \in I^o_{kt}} z^i_{kt} \tag{2a}$$

$$\text{s.t.} \sum_{(k,t)\in B} u_{kt} \leq L \tag{2b}$$

$$\ell u_{kt} \geq \varphi_p \left(1 - z^p_{kt}\right) \qquad \forall (k,t) \in B \tag{2c}$$

$$\ell u_{kt} \geq \varphi_p + \sum_{h=p+1}^{i-1} \varepsilon_h + \varepsilon_i \left(1 - z^i_{kt}\right) \qquad \forall (k,t) \in B, i \in I^o_{kt}, i > p \tag{2d}$$

$$z^i_{kt} \in \{0,1\}\,,\ u_{kt} \in \mathbb{Z}^+ \tag{2e}$$

The objective function (2a) minimizes the number of patients not covered by the overtime allocation, that is the number of patients canceled from the schedule. Constraints (2b) limits the total number of overtime units that can be assigned to blocks (k,t) to the maximum number available L. Constraints (2c) and (2d) impose that variables z^i_{kt} be equal to 1 if and only if the total amount of overtime assigned to the time block $(k,t) \in B$ is not enough to cover the time required for operating patient $i \in I^o_{kt}$. It is worth pointing out that the solution of the overtime allocation problem finds the sequence $o \in \mathcal{O}$ which minimizes $z(o)$, i.e., argmin $\{z(o) : o \in \mathcal{O}\}$.

3. Hybrid Solution Approach

In this section, the hybrid two-phase algorithm developed to solve the overall surgery scheduling problem is described. As reported in [22], health care optimization problems are challenging, often requiring the adoption of unconventional solution methodologies. The solution approach proposed in this paper belongs to this family. In the proposed hybrid optimization approach, Monte Carlo simulation is exploited to take into account the uncertainty of surgery durations while an optimization engine computes a solution. We should point out that the patient assignment model corresponding to the first sub-problem of the overall problem studied here is in fact a stochastic version of the SCAP which is NP-hard in its deterministic version [23]. Efficient exact solutions for solving realistic sized instances of the problem studied here are therefore unlikely.

The proposed hybrid algorithm starts from any deterministic solution which gives the assignment of patients to the available OR blocks (pre-schedule), which can be obtained by means of deterministic models or algorithms as those proposed in [23, 24].

The first phase of the algorithm adapts the pre-schedule to find a feasible and robust solution with respect to the stochastic surgery durations that maximizes the occupation of the operating rooms while evenly distributing the workload among the OR blocks. This solution is computed by means of a neighborhood search algorithm which first seeks to regain the feasibility of the starting solution with respect to the chance constraints (1f), and then tries to improve its reliability while preserving feasibility.

The second phase of the algorithm seeks to minimize the number of possible cancellations by assigning the available overtime while maintaining the reliability of the first phase schedule. This solution is computed by using a pre-defined sequencing to order the selected patients within each OR block and then by applying a sequence of greedy procedures to assign the available overtime. Note that the impact of different sequencing rules is evaluated and compared.

This Section is organized as follows. Section 3.1 describes the basic elements of the proposed algorithm, sections 3.2–3.4 detail the three optimization modules which compose the overall solution approach, while the whole hybrid algorithm is summarized in Section 3.5.

3.1. Basic elements of the algorithm

The algorithm is based on two main elements. The first is the set of neighborhoods exploited by the optimization modules introduced in Section 3.2 and 3.3, while the second is the scenario generation and the sampling needed by the Monte Carlo simulation.

Neighborhoods. The optimization procedure exploits the following neighborhoods already discussed in [23].

The first, named *p-swap*(in, in), evaluates and performs exchanges of two patients that belong to two different OR blocks assigned to the same surgical specialty. The second neighborhood, named *p-swap*(in, out), evaluates and performs exchanges of two patients where one patient is included in the current OR schedule while the other is not. Note that the two neighborhoods operate on the decision variable x_{ikt} in two different ways depending on how the first patient is selected. In the first case, the patient can belong to any feasible OR blocks (k, t) while, in the second, the patient is selected from the OR block (k, t) whose utilization is the minimal one. In the first case, the complexity is quadratic in the number of patients while it is linear in the second case.

In order to disrupt a current solution, we introduce the *p-drop*(in) neighborhood which removes from the solution the patient with the minimum surgery duration in the OR block having the highest probability to go into overtime. The rationale here is to free operating time in order to promote swaps between patients. The complexity of this neighborhood is linear in the number of patients.

The last neighborhood, named $p\text{-}add(out, in)$, adds patients that are not currently scheduled in order to fill as much as possible the OR blocks, without exceeding the capacity chance constraints (1e)-(1f). Patients can be added only to an OR block assigned to their surgical specialty. The complexity of this neighborhood is linear in the number of patients.

Scenarios for Monte Carlo simulation. Monte Carlo simulation is exploited to deal with the variability of patient surgery durations. We developed a stochastic scenario generation procedure in order to generate the value ε_i for each patient $i \in I$ following a predetermined distribution probability $F(\varepsilon_i)$.

In the literature, several authors recommend the use of the lognormal distribution for simulating surgery times [25–29]). In [30] the authors propose a new approach based on a three parameter lognormal, however we did not adopt this approach because it requires more data than what was available in our dataset such as age and experience of surgeon, hospital organization, type of surgical intervention, and so on. Other works propose the Gaussian distribution to generate surgery times [5].

In our study, we have used lognormal distributions but the method can easily be adapted to other continuous distributions. The stochastic surgery times are generated by lognormal random variables using mean μ equal to the patient expected surgery time and standard deviation σ a function of the patient expected surgery duration p_i, as follows:

$$\sigma(F_{\epsilon_i}) = \vartheta p_i \quad \text{with} \quad (0 \leq \vartheta \leq 1).$$

Each scenario is composed of a realization of the surgery duration following the F_{ϵ_i} distribution for each patient $i \in I$. The main idea is therefore to check the feasibility and to measure the reliability of each incumbent solution during the neighborhood exploration exploiting the set of scenarios generated by a Monte Carlo simulation procedure.

Clearly, the quality of the final solution may depend significantly on the number of scenarios considered during the computation. At the same time, the number of scenarios affects the running time of the method. To reduce the computational overhead while maintaining the quality, we introduce the idea of *scenario sample*, that is the N scenarios generated by the Monte Carlo procedure are partitioned into S samples, each one composed of $\frac{N}{S}$ scenarios. We will denote the s^{th} scenario sample with $\mathcal{S}_s$.

3.2. Local Search for Feasibility

The first optimization module is the *Local Search for Feasibility* (LS-F). LS-F consists of an iterated sequence of swaps and deletions of patients in such a way as to increase the reliability of the solution by reducing the probability that an OR block (k, t) goes into overtime.

Given a pre-schedule coming from an initial deterministic feasible solution of the patient assignment problem and a sample $\mathcal{S}_s$, LS-F aims at making feasible the chance constraints (1f) with respect to the scenarios belonging to $\mathcal{S}_s$.

At each iteration, LS-F evaluates – for each OR block (k, t) and for each scenario in $\mathcal{S}_s$ – if considering the stochastic surgery duration the current solution is not feasible, (i.e., if the percentage of OR blocks in overtime among the considered scenarios sample is greater than the critical overtime probability level α).

If the current solution is not feasible, LS-F tries to reach feasibility through an iterated local search which exploits the three neighborhoods *p-swap*(in, in), *p-swap*(in, out) and *p-drop*(in). The local search performs swaps until the overall overtime can be reduced; when the local search stops the exploration, if the solution is feasible with respect to the chance constraints (1f) then LS-F finishes its execution, otherwise a *p-drop*(in) is applied and the local search starts again.

Note that the local search applies the two swap neighborhoods selecting patients belonging to any pair of OR blocks. Furthermore, the neighborhood *p-swap*(in, out) can only be applied after the first drop.

3.3. Tabu Search for Improvement

The second optimization module referred to as *Tabu Search for Improvement* (TS-I) is applied to the feasible solution determined by LS-F (i.e., it is the input of TS-I). It aims at improving the quality of the solution computed by LS-F, that is to optimize the objective function (1a), without decreasing reliability.

At each iteration, TS-I first performs the exploration of the two neighborhoods (*p-swap*(in, in) and *p-swap*(in, out)) following a first improvement criterion. For a swap, the exploration selects the former patient from the OR block (k, t) less utilized and the latter from one of the remaining OR blocks in such a way as to optimize the objective function (1a). Only feasible swaps are allowed, that is swaps preserving the feasibility of the LS-F solution. If no improvement is found, TS-I tries to add a patient (dropped during LS-F) exploring the neighborhood *p-add*(out, in).

In order to avoid looping over already visited solutions, TS-I employs two tabu lists having fixed length l_1 and l_2, respectively, and implemented using tabu tags [31]. The first tabu list forbids a patient to be part of a swap for the next l_1 iterations while the second tabu list forbids a patient to be assigned back to the previously assigned OR block for the next l_2 iterations. Obviously, it is required that $l_1 < l_2$. The rationale here is to block the patient $i \in I$ in order to allow TS-I to adjust the solution after moving it from OR block (k, t); then, we prevent patient i from being assigned back to (k, t) to allow the algorithm to compose an exchange in two stage when the exchange is not allowed to be done directly. The effectiveness of this setting is already discussed in [32, 33]. TS-I performs N_I iterations.

3.4. Overtime Allocation and Patient Sequencing

The last optimization module is the *Overtime Allocation and patient Sequencing* (OA-S). OA-S first selects a suitable patient sequencing and then determines the allocation of the available overtime in order to minimize the number of patient canceled from the schedule according to model (2a)–(2e). Given a solution x, the list $\mathcal{L}$ of patients not included in the schedule after performing LS-F and TS-I, and the amount of overtime units available L, OA-S is heuristically solved as follows.

First, the worst case scenario $g \in \mathcal{N}$ (or one of the worst case scenarios) is identified, i.e., the scenario which results in the largest number of OR blocks going into overtime with solution x. OA-S then orders the patients according to the selected sequencing criterion and assigns the overtime units in order to maximize the number of patients covered by the assigned overtime, i.e., the patients whose surgeries can be completed within the available OR block operating time (i.e., OR block length s_{kt} plus the allocated overtime). Finally, if some overtime units are still available, the algorithm allocates them in order to insert as much patients as possible from $\mathcal{L}$ in the schedule. Using g as reference to perform these adjustments to the schedule will enable the procedure to preserve the reliability level of x as well as reduce the computational overhead.

Patient Sequencing. We will consider the four patient scheduling strategies that were studied in [17, 34], which are:

- Increasing duration (ID): patients are ranked by their surgery duration and the OR block sequence begins with the shortest surgery time and finishes with the longest one;

- Decreasing duration (DD): patients are ranked by their surgery duration, the OR block sequence begins with the longest surgery time and finishes with the shortest one;

- Half increasing duration (HID): patients are sorted by increasing surgery duration, the OR block sequence begins with the shortest surgery time, then the second shortest is placed at the end of the block, then the third starts after the first one, and so on;

- Half decreasing duration (HDD): patients are sorted by decreasing surgery duration, the OR block sequence begin with the longest surgery time, then the second longest is placed at the end of the block, then the third starts after the first one and so on.

Overtime allocation. Let us introduce y^g_{kt} with

$$y^g_{kt} = \sum_{i \in I} \varepsilon^g_i x_{ikt} - s_{kt},$$

which represents the overtime ($y^g_{kt} > 0$) or undertime ($y^g_{kt} < 0$) of OR block (k, t) under the worst case scenario $g \in \mathcal{N}$.

The overtime allocation is then solved by a sequence of two greedy algorithms: the first greedy allocates units of overtime in such a way as to maximize the number of patients operated and then the second tries to insert patients not scheduled exploiting any the residual overtime units still available.

In particular, the first greedy algorithm assigns the required overtime to the OR blocks in order to cover the maximum number of patients operated while minimizing the unused operating time for all blocks. More specifically, let u_{kt} be the units of overtime required by block (k, t) such that $y^g_{kt} > 0$

$$u_{kt} = \left\lceil \frac{y^g_{kt}}{\ell} \right\rceil$$

and let L' be the number of overtime units still available. At each iteration, the greedy algorithm assigns $u_{kt} \leq L'$ overtime units to the OR block (k, t)

requiring the minimal value of u_{kt}. If there is a tie, the algorithm selects the OR block (k,t) for which the unused portion of the allocated overtime $\ell u_{kt} - y^g_{kt}$ is minimized.

After the end of the first greedy procedure, some overtime units could still be available, i.e., $L' > 0$. In that case, a second procedure will seek to increase the utilization of OR blocks in undertime, i.e., those having $y^g_{kt} < 0$. To exploit the residual overtime L', the second greedy heuristic tries to insert patients from $\mathcal{L}$ in the current schedule adopting the Best Fit Decreasing (BFD) greedy criterion for the Bin Packing Problem (see, e.g., [35]). In this way, a patient p is inserted in the OR block (k,t) in order to minimize the unused overtime allocated to block (k,t) computed as follows

$$(s_{kt} + \ell u_{kt}) - \left(\varepsilon^g_p + \sum_{i \in (k,t)} \varepsilon^g_i x_{ikt} \right)$$

where

$$u_{kt} = \left\lceil \frac{y_{kt} + \varepsilon^g_p}{\ell} \right\rceil .$$

3.5. Hybrid algorithm

The whole hybrid solution approach, which exploits the optimization modules described in the previous sections, is depicted by the pseudo code of Algorithm 1 below.

After generating the N scenarios and dividing them into S samples $\mathcal{S}_s$, the first phase of the algorithm consists in determining a list of robust solutions $x^\star_s$ for each sample s. Starting from an optimal deterministic solution (pre-schedule), each robust solution is computed by applying first LS-F, to guarantee the reliability of the solution, and then TS-I, to improve its quality by maximizing the OR utilization without deteriorating the reliability. Note that in our computational analysis we will consider different ways to determine the initial pre-schedule.

From the list $x^\star_1, \ldots, x^\star_S$, the second phase starts by discarding those solutions which are not feasible with respect to the whole set of scenarios $\mathcal{N}$. Among the remaining (feasible) solutions, the algorithm selects the best one $\bar{x}$, that is the one which maximizes the objective function (1a). If no feasible solutions are available, the algorithm selects the least infeasible solution and makes it feasible by applying LS-F. Then, OA-S computes the final solution $x^\star$ by minimizing the number of patients canceled from the pre-schedule $\bar{x}$.

Algorithm 1: two-phase algorithm

```
Input   : problem data.
begin
    generate $N$ scenarios; split them into $S$ samples $\mathcal{S}_s$;
    // start first phase
    for $s = 1, \dots, \frac{N}{S}$ do
        $x_s^0$ := getInitialSolution ( );
        $x_s^1$ := LS-F ( $x_s^0$, $\mathcal{S}_s$ );
        $x_s^\star$ := TS-I ( $x_s^1$, $\mathcal{S}_s$ );
    // end first phase, start second phase
    $\bar{x}$ := selectBestFeasibleSolution ( $x_1^\star, \dots, x_S^\star, \mathcal{N}$ );
    $x^\star$ := OA-S( $\bar{x}$, $L$, $\mathcal{L}$ );
    // end second phase
Output: return $x^\star$;
```

4. Computational Analysis

The proposed algorithm has been tested on a set of instances based on real data. The computational environment and benchmark instances are described in Section 4.1.

A series of computational experiments was carried out to evaluate the impact of the main parameters and components of the algorithm. First, in Section 4.2, the impact on the algorithm behavior of using alternative initial solutions x_s^0 is evaluated. The default value has been set to the deterministic solution (DET), i.e., the pre-schedule, while the effects of choosing the feasible solution computed after LS-I or the best solution computed during TS-I, have been analyzed and compared. In Section 4.3 we perform a sensitivity analysis to assess the impact of alternative values of parameters α and σ, while in Section 4.4 changes of the patient sequencing rule and the overtime availability are evaluated. Finally, in Section 4.5 the solutions obtained by the algorithm are analyzed with respect to the resulting OR occupation rate.

4.1. Computational environment and parameter tuning

In order to generate realistic test instances, real waiting list data provided by the Department of General Surgery of the San Martino University Hospital, Genova, Italy, have been used. In particular, the data provided contain all the relevant information regarding a real waiting list composed of

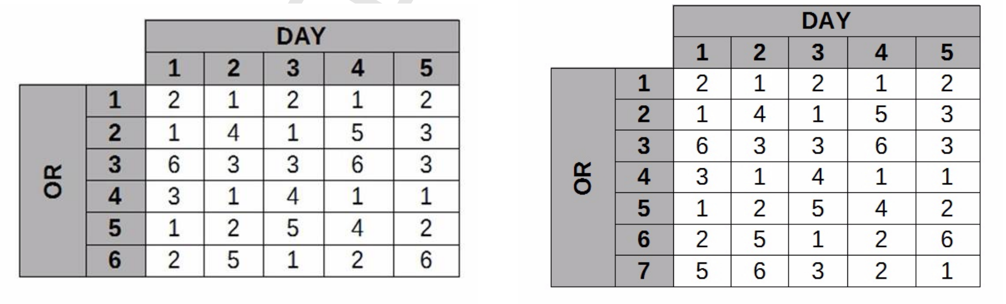
400 patients. For each patient i in the waiting list the date of referral d_i, the expected surgery duration p_i and the expected LOS μ_i are given.

B_1						B_2					
Id	$\|I\|$	$\|J\|$	$\|K\|$	s_{kt}	χ	Id	$\|I\|$	$\|J\|$	$\|K\|$	s_{kt}	χ
1	85	6	6	6	14	5	128	6	6	9	14
2	98	6	6	7	14	6	163	6	6	12	14
3	100	6	7	6	14	7	199	6	6	15	14
4	115	6	7	7	14	8	230	6	6	18	14

Table 1: Characteristics of the benchmark sets B_1–B_2.

From these information, we generated 2 different benchmark sets whose characteristics are summarized in Table 1: each benchmark is composed of 4 instances by varying the number of patients $|I|$, the number of operating rooms $|K|$ and the OR time block duration s_{kt}. All instances are characterized by the same number of surgical specialties $|J| = 6$ and number of weekend stay beds $\chi = 14$. The duration of one overtime unit is set to $\ell = 30$ minutes.

The set of patients I, the pre-schedule and the MSS are determined by solving the deterministic model reported in [23]. Since both MSS and the set I can have a significant effect on performance measures in our computational experiments, we report the MSSs used in Figure 1 while the average distribution of surgery durations of the patients belonging to set I is reported in Figure 2.



		DAY				
		1	2	3	4	5
OR	1	2	1	2	1	2
	2	1	4	1	5	3
	3	6	3	3	6	3
	4	3	1	4	1	1
	5	1	2	5	4	2
	6	2	5	1	2	6

		DAY				
		1	2	3	4	5
OR	1	2	1	2	1	2
	2	1	4	1	5	3
	3	6	3	3	6	3
	4	3	1	4	1	1
	5	1	2	5	4	2
	6	2	5	1	2	6
	7	5	6	3	2	1

Figure 1: MSS adopted for instances with 6 and 7 ORs.

Preliminary computational tests (not detailed here) were performed to tune some of the algorithm parameters and set their default values. For

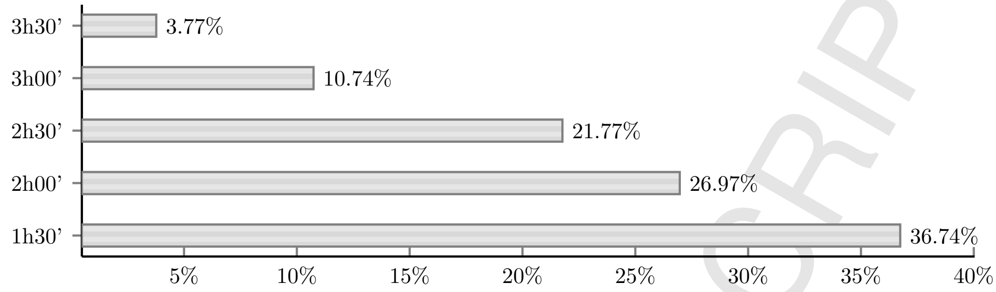


Figure 2: Average distribution of patient surgery durations in test data.

instance, the tabu lists lengths l_1 and l_2 were set to 8 and 12, respectively, while the number of scenarios N and of samples S were set to 1000 and 10. These values proved to produce better results on average and shorter running time. Table 2 reports the default values used for all the algorithm parameters as well as the alternative values explicitly tested in the following computational experiments. In the following sections, the results reported were obtained using the default parameter values (i.e., those in the third column of Table 2) except when explicitly stated otherwise.

Parameter	Description	Default value	Other values
l_1, l_2	length of tabu list 1 and 2	8 and 12	
N_I	number of iterations for TS-I	300	
N	number of scenarios	1000	
S	number of samples	10	
α	reliability (1f)	0.05	0.10, 0.15
σ	variability of p_i	0.1	0.15, 0.20, 0.30, 0.40
L	number of overtime units	5	10, 15, 20
x_s^0	initial solution	DET	LS-F, TS-I
–	sequencing	DD	ID, HID, HDD

Table 2: Algorithm parameters.

Finally, the algorithm was coded in standard C++ and compiled with `gcc 4.4.3`. All tests were performed on a 1.73 GHz Intel core i7 processor, with 4 GB of RAM running under Linux Ubuntu 14.10.

4.2. Impact of the initial solution

Recall that, for each sample exploration, the procedure is initiated from an *initial* solution (see line 4 of Algorithm 1). Obviously, for the first iteration the deterministic solution is the only one available. However, after the first loop of the procedure has been completed (on the first sample), different choices regarding the initial solution can be taken at the start of each of the following sample explorations.

In Table 3 the impact of the following three decision rules on the quality of the final solution and on running time is analyzed: always starting from the deterministic solution (DET); starting from the solution computed after LS-F; and starting from the best solution computed during TS-I. For each instance in B_1, three different reliability levels $\alpha \in \{5\%, 10\%, 15\%\}$ are also evaluated. The number of patients deleted p_d for each instance and for each value of α are reported, while in the last column the average running time is given.

	Instances												
init	1			2			3			4			avg.
sol.	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%	secs
DET	15	15	8	17	16	10	28	27	23	26	26	23	498.7
LS-F	15	15	8	17	16	10	28	29	23	23	26	23	485.6
TS-I	23	23	23	24	24	0	28	28	28	22	22	0	193.3

Table 3: Number of patients canceled and running time using different initial solutions.

As can be observed, the algorithm's behavior is very similar, both in terms of number of patients deleted and average running time, when using the deterministic solution or the feasible solution computed after LS-F. However, when using the best solution computed during TS-I, the behavior is quite different. First, the running times are much shorter, roughly about 25% of what they are when using either of the other two strategies. In addition, the number of cancellations varies significantly from one instance to another when compared with the other strategies: it is much higher for instances 1 and 2, somewhat lower for instance 4, and when the reliability level α is set to 15%, no cancellations occur for instances 2 and 4.

In order to get a clearer understanding of the difference in behavior that the different strategies produce, Figure 3 depicts the evolution of the total number of OR blocks that are in overtime at the beginning of each successive sample exploration loop (cf. line 4 of Algorithm 1) for instance 3 (the one where the quality of the final solutions is similar). Note that the total number of OR blocks in each sample is equal to 3500, i.e., the number of OR blocks in the planning horizon (35 for instance 3) times the number of scenarios in

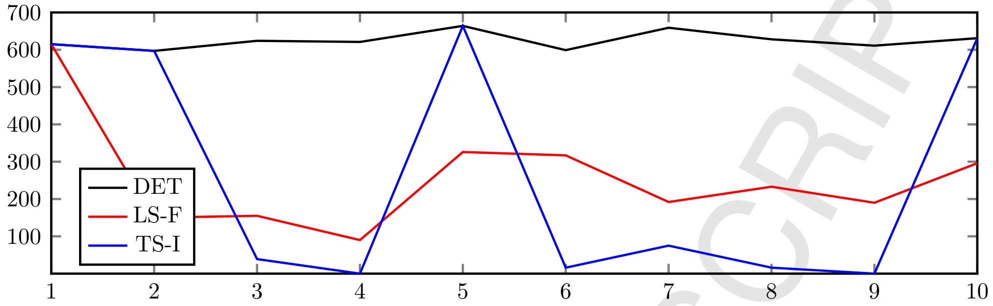


Figure 3: Instance 3: number of OR blocks in overtime at the beginning of each iteration.

each sample (100).

As can be seen, the total number of OR blocks that are in overtime when the procedure passes to a new sample or scenarios is quite different depending on the chosen strategy. When restarting from the deterministic solution every time, this number is rather high and quite stable, which is to be expected since it is the same schedule that is tested against different yet similar samples of scenario realizations. Likewise, the behavior when using the LS-F solution at the start of the next sample exploration is also somewhat predictable exhibiting a lower number of overtime blocks than with the first strategy (since the initial solution was already feasible with respect to the previous sample) but this number varying somewhat when changing samples (since the scenarios composing successive samples are different).

To the contrary, the behavior when using the TS-I solution is counter-intuitive at first glance since one might expect it to be even lower than with the previous strategy and quite stable, which is not the case. It varies significantly from one sample to the next and it is close to 0 several times. On second thought, this is not totally surprising in fact since the best solution found by TS-I on the previous sample was obtained after applying the LS-F to make it feasible and then improving its quality by inserting additional cases without reducing the reliability. It should therefore share similar qualities with respect to the feasibility with the one used in the second strategy. If two successive samples are quite similar, then it is quite normal that the number of OR blocks in overtime be low or even non existing. If the consecutive samples are quite different, then one should expect that the number of OR blocks in overtime be high since the schedule is fuller than the one obtained after the LS-F only and it will need much modifications by LS-F to recover feasibility. In fact, the first situation explains the much shorter running times of this strategy since in that case LS-F does not need to runs as long to restore feasibility and LS-F is the more computationally expensive part of the overall procedure.

Figure 3 also provides some insights to understand the ability of the algorithm with the TS-I solution strategy to drastically reduce the number of patients canceled. When starting from the solution computed by TS-I, the information regarding the previous sample is in some sense transferred to the next one via this initial solution. This process may act as a kind of of continuous refining of the solution and allow a better use of the available overtime units.

4.3. Solution reliability vs. variability of the surgery duration

The choice of the α and σ parameter values can have a significant impact on the final solutions. A sensitivity analysis of their impact was performed to provide additional insights on the algorithm behavior. Recall that the value of α measures the level of reliability of the obtained solution (when α increases, the level of reliability decreases). Whereas, the value of σ represents the level of variability when generating the stochastic surgery duration as described in Section 3.1.

For each instance in B_1 and for each value of α and σ (see Table 2) the number of patients canceled (p_d) and the corresponding hours of unused OR capacity (h_d) are reported in Tables 4 and 5, respectively.

		Instances											
		1			2			3			4		
	α →	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%
	10%	15	15	8	17	16	10	28	27	23	26	26	23
σ	15%	16	16	8	24	15	10	28	27	22	22	25	22
	20%	15	15	19	16	15	11	28	28	23	24	22	22

Table 4: Number of patients canceled varying α and σ parameters.

The results reported in Table 4 give several interesting insights. As expected, increasing the value of α decreases the number of patients canceled. On the other hand, increasing the value of σ does not affect the solution significantly. This tends to illustrate that the algorithm seems to be capable of dealing quite well with the larger variability of the surgery durations. It can also be seen from these results that the particular characteristics of an instance may have a significant impact on the quality of the solutions obtained. Indeed, analyzing in more detail instances 2 and 3, one can note that they are very similar with respect to the number of patients to be scheduled and OR capacity (see Table 1). Nevertheless, there is a considerable difference in the number of patients canceled in the schedules obtained for the two instances, this value in instance 3 in the case of $\alpha = 15\%$ being almost double than for instance 2 and same α (and this for all values of σ).

The results reported in Table 5 confirm the previous remarks: the hours of unused OR capacity decrease as the value of α increases. On the other hand, the metric increases slightly when the value of σ increases. This is most probably related to the behavior of the algorithm reported above which seems to be capable of keeping at a relatively similar level the number of patient cancellations as the variability of surgery duration increases.

			Instances											
			1			2			3			4		
	α	→	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%
	10%		29.9	29.9	13.9	31.3	31.8	18.4	69.0	70.3	59.2	49.7	51.8	49.8
σ	15%		28.8	28.8	18.7	49.7	31.0	19.9	68.4	69.4	59.2	43.6	47.1	46.1
	20%		30.3	30.3	28.4	34.5	31.1	21.7	67.8	67.7	59.8	57.6	45.9	53.5

Table 5: Unused hours of OR capacity with varying α and σ parameters.

4.4. Impact of sequencing rules and overtime availability

The algorithm has been tested comparing four alternative sequencing rules to schedule patients within the OR blocks (see Section 3.4).

Preliminary tests on the B_1 instances show a smaller reduction of the OR capacity (2.5 hours on average) when using DD and HDD with respect to ID and HID. This limited impact is essentially due to the fact that the solutions of the instances derived from real data usually have a small number of patients scheduled in each OR block (k, t), i.e., between 2 and 4 patients.

Thus, the analysis is given for benchmark set B_2 which is characterized by a greater OR capacity time s_{kt} as reported in Table 1. The main purpose here is to evaluate the impact of the four sequencing rules when more patients can be scheduled in each OR block. From an operational point of view, this means that we have two or more surgery teams that will follow each other in the same OR block, in sequence. The first team will perform their set of surgeries, then the second team will perform theirs, and so on.

	Instances							
	5		6		7		8	
Seq. Rule	pre	post	pre	post	pre	post	pre	post
ID	2.7	8.7	4.1	13.6	3.8	10.2	4.3	18.9
DD	1.6	7.6	2.0	11.4	1.5	7.9	1.4	16.1
HID	1.9	7.9	2.1	11.5	3.1	9.5	2.3	16.9
HDD	2.5	8.5	3.6	13.0	3.6	10.0	3.6	18.2
L	5		10		15		20	
	pre	post	pre	post	pre	post	pre	post
s_o	1	2	1	3	1	5	1	7
p_d	26	24	23	19	25	20	19	14

Table 6: Impact of sequencing rules on benchmark set B_2.

The results are reported in Table 6. The upper part of the table presents the total number of operating time hours that would be lost if no overtime was available. For each instance and for each sequencing rule, this value is reported before (pre) and after (post) applying the OA-S procedure. The lower part of the table reports for each instance the number of overtime units L available, the number of OR blocks in overtime s_o, and the number of canceled patients p_d. It is important to note that these information are not affected by the sequencing rule selected but only by the patients scheduled in each OR block and the realization of surgery durations.

The results reported in Table 6 illustrate the superiority of DD rule as expected. Surprisingly, the HID rule results as the second best option and gives results close to those of DD except for Instance 7. Finally, the worst results are obtained using the ID sequencing rule. Note that similar results have been discussed in [9, 10, 17].

A particular remark is in order regarding the value s_o which is always 1 before applying OA-S. This means that the reliability after applying TS-I is greater than that imposed by the chance constraints (1f). This can be justified by the fact that more than one scenario can become feasible just after dropping a patient during the LS-F. Finally note that this behavior is maintained by the algorithm when solving instances in B_1 and when α increases, that is s_o is usually less than the expected number of scenarios (e.g., if $\alpha = 0.05$ and $\frac{N}{S} = 100$ then s_o should be around 5).

The impact of changing the overtime availability to be allocated by the

OA-S has also been investigated. Recall that after selecting the best and feasible solution $\bar{x}$ over the set of solutions computed by LS-F and TS-I for each sample, OA-S starts allocating the L available overtime units with the aim of reducing the number of patient cancellations. Recall also that OA-S works on the solution $\bar{x}$ obtained after performing LS-F and TS-I on the worst scenario $g \in \mathcal{N}$, that is the one with the largest number of OR blocks (k, t) in overtime.

		Instances								
		1		2		3		4		avg.
L	$L \times \ell$	p_d	h_d	p_d	h_d	p_d	h_d	p_d	h_d	imp.
5	2.5	15	29.9	17	31.3	28	69.0	26	49.7	–
10	5.0	13	24.3	15	25.9	26	63.1	24	44.7	5.5
15	7.5	12	21.9	14	23.9	25	60.6	23	42.3	7.8
20	10.0	11	18.9	13	22.4	25	60.6	23	42.3	8.9
40	20.0	8	13.4	9	15.4	21	51.2	19	34.8	16.3
45	22.5	7	10.9	9	15.4	20	47.7	18	32.3	18.4
50	25.0	6	9.5	8	13.0	19	44.7	17	30.8	20.5

Table 7: Impact of varying the number of available overtime units L.

Table 7 reports the results obtained considering different amounts of overtime units available for a planning horizon of one week. In the first and second columns the number of overtime units L and the corresponding total hours of overtime available are reported, respectively. Seven cases are considered which correspond to values of L ranging from 5 to 50. As stated previously, the length of an overtime unit ℓ is set to 30 minutes. Let us also recall that in our test instances the minimum value of expected surgery durations is equal to 1.5 hours, which corresponds to 3 overtime units.

For each instance in B_1 the number of patients canceled (p_d) and the corresponding unused hours of OR capacity (h_d) are reported. Finally, the last column gives the average decrease in the unused hours of OR resulting from the increase in overtime availability computed with respect to the first row, $L = 5$, which corresponds to an overtime availability of 2.5 hours. This metric can be viewed as a proxy of the benefit of additional overtime capacity in terms of decreasing the OR time wasted when some patients need to be canceled from the pre-schedule.

As should be expected, in almost all instances and cases, increasing the overtime availability results in a decrease in both the number of patients canceled and the unused OR time. The greatest decrease of the number of patients canceled is observed when increasing the number of overtime units from 5 to 10, while shifting from 15 to 20 does not allow to reduce the number of canceled patients for instances 3 and 4. Further increases in overtime availability result in decreasing benefits.

The decrease in the unused OR time capacity follows the same pattern as that of canceled patients, i.e., that of diminishing returns. For $L = 5$, 10 and 15, the average decrease of the unused OR capacity (last column) is always greater than the additional hours of overtime provided. For instance, increasing the number of overtime units from 5 to 10, i.e., increasing the overtime availability by 2.5 hours, results in a reduction of 5.5 hours of the unused OR hours. This means that the benefit almost doubles the additional overtime hours. However, this positive impact decreases as the number of additional overtime units increases, and starting with $L = 20$ the benefits become less than the additional overtime provided.

4.5. Evaluating OR utilization

As reported in Section 2 in order to achieve a high level of OR utilization, which is the second main objective of the solution approach, a max min objective function is introduced in the patient assignment model (1a). The aim is therefore to maximize the utilization of the OR block which has the minimal utilization. In this section we analyze further the quality of the algorithm solutions in terms of OR occupation. We use as OR occupation metric the minimal utilization of the OR blocks denoted z as in (1a).

		Instances											
		1			2			3			4		
	α →	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%
σ	10%	61.0%	61.0%	21.7%	29.4%	29.4%	57.1%	0.0%	0.0%	0.0%	14.7%	28.8%	15.2%
	15%	42.4%	42.4%	43.3%	1.2%	29.8%	44.9%	0.0%	0.0%	0.0%	16.6%	28.6%	15.5%
	20%	27.8%	27.8%	0.0%	32.1%	32.1%	46.4%	0.0%	0.0%	0.0%	2.6%	30.1%	2.0%
avg. z		54.7%	54.7%	46.3%	55.8%	60.2%	69.1%	38.2%	38.1%	38.2%	51.5%	59.6%	51.3%

Table 8: Impact on OR utilization: improvement of z between the solutions computed after LS-F and TS-I.

Table 8 reports the percentage improvement of the solution computed after LS-F and TS-I, the latter referred as solution $\bar{x}$ (see line 7 in Algorithm 1), which measures the utilization of the OR block which has the minimal utilization. For each instance, we also report the results when varying the values of

α and σ. It is worth recalling that the α and σ parameters measure, respectively, the level of reliability of the final solutions (i.e., the critical probability level used in the chance constraints (1f)) and the variability of the stochastic surgery times (i.e., the standard deviation of the distributions used to generate the surgery duration). The last row reports the average value of $\bar{x}$ with respect to the OR capacity s_{kt}.

As can be observed, the percentage improvement decreases when the variability of the surgery durations increases. Nevertheless, the behavior is greatly affected by the different instances and also varies significantly with respect to the value of α. The surgery time variability impacts in a slightly dissimilar way when combined with different values of α, as shown by instances 2 and 4. A somehow particular behavior is experienced solving instance 3. In this case, the TS-I is not capable to improve the objective function z. This seems due to the characteristics of the instance which make it more difficult to maintain the solution feasibility (with respect to the chance constraints) during the improvement phase. Note that for this instance the average occupation rate never reaches the 40%. Recall however that this corresponds to the OR block having the least utilization in the worst case scenario so it is bound to be low.

			Instances										
			1			2			3			4	
	α	→	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%
	10%		64.6%	64.6%	–	–	32.2%	61.1%	2.3%	2.1%	2.0%	–	–	–
σ	15%		–	–	50.1%	5.4%	33.2%	49.3%	5.3%	4.8%	5.9%	20.2%	34.4%	21.7%
	20%		38.4%	38.4%	6.4%	38.2%	37.0%	58.8%	9.5%	11.9%	–	12.5%	–	13.3%

Table 9: Impact on OR utilization: percentage improvement of z between the solutions computed after LS-F and OA-S.

In Table 9 the percentage improvements between the solutions computed after LS-F and OA-S are reported for the same instances and for each value of α and σ parameters. In order to have a consistent comparison among the results in Table 8 and Table 9, the percentage improvement is reported just for those instances in which the solution $\bar{x}$ is the one computed using the sample $\mathcal{S}_s$ which contains the worst scenario $g \in \mathcal{N}$. This solution is in fact the solution used by the OA-S to start the overtime allocation phase.

The solutions after OA-S always increase the percentage improvement of z with respect to the solutions obtained after the TS-I (see Table 8). On average, the increase is higher for the $\sigma = 20\%$ case, where it is below 8 for instances 1 and 2 and rises up to 11 for instances 3 and 4. The impact of different values of α is confirmed. In fact looking at the $\sigma = 20\%$ case, the greatest and lowest percentage increases of OR utilization are both found in instance 2 (about 12%, from 46.4% to 58.8%, for $\alpha = 15\%$ and around 5% for $\alpha = 5$ and 10%).

5. Conclusions

In this paper we presented an efficient algorithmic framework for solving the problem of assigning and sequencing a set of elective surgical patients to a set of available OR time blocks, over a one week planning horizon, while taking explicitly into account the uncertainty of surgery durations. As in many practical contexts, a given amount of overtime is available to mitigate the unavoidable consequences of surgery time variability. The method therefore seeks to assign this overtime resource in order to balance the trade-off between hospital and patient perspectives, i.e., maximize OR utilization while minimizing surgery cancellations (due to OR blocks running late).

The problem is decomposed into two sub-problems which are solved in two sequential phases. Integer stochastic formulations are proposed for both sub-problems. The hybrid algorithm developed combines neighborhood search techniques with Monte Carlo simulation, which is used to deal with the variability of surgery times. In addition, greedy algorithms are developed for the overtime allocation.

A set of instances based on real data have been constructed to test the algorithm behavior and the robustness of the solutions produced. An extensive computational results analysis has been performed to analyze the impact of the main algorithmic components and parameters. The computational analysis demonstrated the capability of the proposed method to deal efficiently with the trade-off between hospital and patient perspectives in reasonable computational times.

The proposed approach could represent a useful decision tool to be used by OR managers to determine reliable/robust OR schedules (i.e., planning and sequencing of patients). The approach has the advantage of exploiting the trade off between achieving an acceptable level of OR utilization rate while limiting the negative effects of surgery cancellations and postponements. Despite the efficiency demonstrated by the proposed approach, it has not yet been integrated in the hospital practice. The main reasons are linked to difficulties in introducing and interfacing stand alone resolution methods into the hospital information systems.

Finally, the proposed approach can be considered as the starting point for defining a new methodological approach to deal with stochastic health care problems arising in real life applications. As a matter of fact, this approach allows the combination of the potential of local search metaheuristics to deal with combinatorial optimization problems, with the well known capability of Monte Carlo simulation to represent stochastic phenomena. Furthermore, the idea of gathering scenarios in samples makes the algorithm framework more flexible and computationally efficient.

Acknowledgment's

All the authors acknowledge support from the Italian Ministry of Education, University and Research (MIUR), under the grant n. RBFR08IKSB, “Firb – Futuro in Ricerca 2008”. Patrick Soriano's research on this project was also supported by the Natural Sciences and Engineering Research Council of Canada and the Fonds québécois de la recherche sur la nature et les technologies under grants OGP0218028, OGP0184121, and 01-ER-3254. This support is hereby gratefully acknowledged.

References

[1] J. van Oostrum, M. van Houdenhoven, J. Hurink, E. Hans, G. Wullink, G. Kazemier, A master surgical scheduling approach for cyclic scheduling in operating room departments, OR Spectrum 30 (2008) 355–374.

[2] B. Cardoen, E. Demeulemeester, J. Beliën, Operating room planning and scheduling: A literature review, European Journal of Operational Research 201 (2010) 921–932.

[3] F. Guerriero, R. Guido, Operational research in the management of the operating theatre: a survey, Health Care Management Science 14 (2011) 89–114.

[4] D. Min, Y. Yih, Scheduling elective surgery under uncertainty and downstream capacity constraints, European Journal of Operational Research 206 (2010) 642–652.

[5] E. Hans, G. Wullink, M. van Houdenhoven, G. Kamezier, Robust surgery loading, European Journal of Operational Research 185 (2008) 1038–1050.

[6] B. Denton, J. Miller, H. Balasubramanian, T. Huschka, Optimal allocation of surgery blocks to operating rooms under uncertainty, Operations Research 58 (2010) 802–816.

[7] B. Addis, G. Carello, 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, J. Fowler (Eds.), Proceedings of the International Conference on Health Care Systems Engineering, Vol. 61 of Springer Proceedings in Mathematics & Statistics, Springer International Publishing, 2014, pp. 175–189.

[8] O. Shylo, O. Prokopyev, A. Schaefer, Stochastic operating room scheduling for high-volume specialties under block booking, INFORMS Journal of Computing 25 (4) (2013) 682–692.

[9] R. Aringhieri, D. Duma, A hybrid model for the analysis of a surgical pathway, in: Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (HA-2014), 2014, pp. 889–900. doi:10.5220/0005148408890900.

[10] D. Duma, R. Aringhieri, An online optimization approach for the Real Time Management of operating rooms, Operations Research for Health Care 7 (2015) 45–51. doi:10.1016/j.orhc.2015.08.006.

[11] B. Denton, J. Viapiano, A. Vogl, Optimization of surgery sequencing and scheduling decisions under uncertainty, Health Care Management Science 10 (2006) 13–124.

[12] S. Batun, B. Denton, T. Huschka, A. Schaefer, Operating room pooling and parallel surgery processing under uncertainty, INFORMS Journal of Computing 23 (2) (2011) 220–237.

[13] B. Addis, G.Carello, A. Grosso, E. Tànfani, Operating room scheduling and rescheduling: a rolling horizon approach, Flexible Services and Manufacturing Journal. Advanced publication on line: 31 Jan 2015. doi:10.1007/s10696-015-9213-7.

[14] M. Persson, J. Persson, Analysing management policies for operating room planning using simulation, Health Care Management Science 13 (2010) 182–191.

[15] A. Testi, E. Tànfani, G. Torre, A three-phase approach for operating theatre schedules, Health Care Management Science 10 (2007) 163–172.

[16] B. Sobolev, V. Sanchez, C. Vasilakis, Systematic review of the use of computer simulation modeling of patient flow in surgical care, Journal of Medical Systems 35 (2011) 1–16.

[17] I. Beaulieu, M. Gendreau, P. Soriano, Operating room scheduling under uncertainty, in: E. Tànfani, A. Testi (Eds.), Advanced Decision Making Methods Applied to Health Care, International Series in Operations Research and Management Science, Springer, Milan, 2012, pp. 13–32.

[18] J. Magerlein, J. Martin, Surgical demand scheduling: A review, Health Services Research 13 (1978) 418–433.

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

[20] A. Ruszczynski, A. Shapiro, Stochastic Programming, Elsevier, Amsterdam, The Netherlands, 2003.

[21] N. Sahinidis, Optimization under uncertainty: State-of-the-art and opportunities, Computers & Chemical Engineering 28 (7) (2004) 971–983.

[22] R. Aringhieri, E. Tànfani, A. Testi, Operations Research for Health Care Delivery, Computers & Operations Research 40 (9) (2013) 2165–2166.

[23] R. Aringhieri, P. Landa, P. Soriano, E. Tànfani, A. Testi, A two level Metaheuristic for the Operating Room Scheduling and Assignment Problem, Computers & Operations Research 54 (2015) 21–34.

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

[25] W. Hancock, P. Walter, R. More, N. Glick, Operating room scheduling data base analysis for scheduling, Journal of Medical Systems 12 (6) (1988) 397–409.

[26] W. Spangler, D. Strum, L. Vargas, H. Jerrold, Estimating procedure times for surgeries by determining location parameters for the lognormal model, Health Care Management Science 7 (2004) 97–104.

[27] D. Strum, J. May, L. Vargas, Modeling the uncertainty of surgical procedure times: Comparison of lognormal and normal models, Anesthesiology 92 (4) (2000) 1160–1167.

[28] J. May, D. Strum, L. Vargas, Fitting the lognormal distribution to surgical procedure times, Decision Sciences 31 (1) (2000) 129–148.

[29] F. Dexter, J. Ledolter, Bayesian prediction bounds and comparisons of operating room times even for procedures with few or no historic data, Anesthesiology 103 (2005) 1259–1267.

[30] P. Stepaniak, C. Heij, G. D. Vries, Modeling and prediction of surgical procedure times, Statistica Neerlandica 64 (1) (2010) 1–18. doi:10.1111/j.1467-9574.2009.00440.x.

[31] M. Gendreau, A. Hertz, G. Laporte, A tabu search heuristic for the vehicle routing problem, Management Science 40 (10) (1994) 1276–1290.

[32] R. Aringhieri, M. Dell'Amico, Comparing metaheuristic algorithms for sonet network design problems, Journal of Heuristics 11 (1) (2005) 35–57.

[33] R. Aringhieri, Composing medical crews with equity and efficiency, Central European Journal of Operations Research 17 (3) (2009) 343–357.

[34] E. Marcon, F. Dexter, Impact of surgical sequencing on post anesthesia care unit staffing, Health Care Management Science 9 (1) (2006) 87–98.

[35] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley-Intersci. Ser. Discrete Math. Optim., John Wiley and Sons, 1990.