Journal of Applied Operational Research (2014) **6**(3), 145–157

ISSN 1735-8523 (Print), ISSN 1927-0089 (Online)

Optimization methods for the operating room management under uncertainty: stochastic programming vs. decomposition approach

Raul Pulido [1,]*, Adrián M. Aguirre [2], Natalia Ibáñez-Herrero [1], Miguel Ortega-Mier [1], Álvaro García-Sánchez [1] and Carlos A. Méndez [2]

[1] *Escuela Técnica Superior de Ingenieros Industriales (ETSII-UPM), Madrid, Spain*
[2] *INTEC (UNL-CONICET), Güemes, Santa Fe, Argentina*

Abstract. The operating theatres are the engine of the hospitals; proper management of the operating rooms and its staff represents a great challenge for managers and its results impact directly in the budget of the hospital. This work presents a MILP model for the efficient schedule of multiple surgeries in Operating Rooms (ORs) during a working day. This model considers multiple surgeons and ORs and different types of surgeries. Stochastic strategies are also implemented for taking into account the uncertain in surgery durations (pre-incision, incision, post-incision times). In addition, a heuristic-based methods and a MILP decomposition approach is proposed for solving large-scale ORs scheduling problems in computational efficient way. All these computer-aided strategies has been implemented in AIMMS, as an advanced modeling and optimization software, developing a user friendly solution tool for the operating room management under uncertainty.

Keywords: MILP; stochastic optimization; decomposition approach; scheduling; operating room

Received November 2013. Accepted April 2014

Introduction

Nowadays, hospitals managers are focusing on providing higher utilization of their resources with the reduction in operative costs. In this context, operating theatres represent a critical area due to any improvement impact directly in the budget of the hospital (Souki, 2011). Operating theatres usually content a set of surgical and recovery rooms with limited number of beds and personnel, such as nurses, surgeons, anesthetists, etc. According to this, the best way to improve the performance of operating theatres is trying to synchronize the surgery activities in a better way. Then, planning and scheduling of surgery activities seems to be the most useful and efficient strategy for this purpose. Many contributions about planning and scheduling of operating theatres have been developed in literature. Few contributions decompose the problem in planning and scheduling decisions in two levels. In the first one, surgical cases are priori assigned to a particular block time in a week (date) whereas in the second level daily surgical cases are scheduled (see Augusto *et al*., 2010; Cardeon *et al*., 2010; Fei *et al*., 2010).

* *Correspondence*: Raul Pulido, Escuela Técnica Superior de Ingenieros Industriales (ETSII-UPM), José Gutierrez Abascal 2, 28006 Madrid, Spain. E-mail: raul.pulido@upm.es

Similarly, we could compare the block-scheduling strategy in where surgeries, pre-assigned to surgeons according to the surgical service, have to be scheduled in blocks prior to the working day, to the open-scheduling strategy where surgeries submit a request for OR time and a detailed schedule is generated during the day of surgery. The latter strategy is common, for example, in Neurosurgery operations where a patients list is only known 24h before a surgical day. This flexible scheme avoids unfilled blocks in a working day. In addition, urgent/emergent surgeries should not be delayed until an available surgeon is free and also this strategy eliminates the ORs idle times when surgeons have already finished (Denton *et al*., 2007).

A real application problem appears in Batun *et al*. (2011) where they study the impact of different operative costs in the ORs. They suggested accounting a negative operating cost of an operation room when is not in use by the staff and the overtime as a penalty cost. An also considers the operative cost of the surgeons when they are idle or waiting for another surgery. This problem has attracted the attention of numerous researchers and practitioners in recent years. In 2013 the optimization modeling competition MOPTA selected this problem as a relevant one for the study in the operational research community (MOPTA, 2013). According to the participants and organizers, this kind of problem commonly appeared in many hospitals in different countries where the scheduling process is done without any support system.

In this paper we consider the scheduling problem of daily surgical cases in the operating theatre presented in MOPTA 2013. Thus, we will study the situation where a hospital is already working and the number of ORs and surgeons are given. Due to the hospital administration has decided to use more efficiently their ORs, the manager requires to allocate and sequence a set of already planned surgeries in a given number of available ORs and surgeons in each particular surgical day. For that, we have to find the best sequence of surgeries that minimize the total surgical cost composed by ORs idle time and ORs overtime and surgeons waiting times. In order to do this, let us assume that the set of surgeries to be scheduled is known 24h in advance at the surgical day and the number of available ORs and surgeons are fixed. Then, all planned surgeries have been done during the surgical day. Also, we must consider that all the surgeries can be performed at any of the ORs and surgeries could be performed by any of the surgeons. In our case also surgical operation durations (pre-incision, incision and post-incision times) are imprecise and have to be modeled as a random distribution. Finally, surgeons move between ORs performing surgeries until all are finished. Then, based on the principal ideas of global precedence (Méndez and Cerdá, 2003), we formulated a MILP model for the scheduling of multiple surgeries in homogeneous ORs with several available surgeons. This problem can be tackled as a generalized scheduling problem with multiple resources, as was presented in Capón *et al*. (2007). Other formulations have been developed previously for a similar problem by Batun *et al*. (2011) but they do not exploit the real strengths of precedence concepts (Méndez *et al*., 2006) and also take priori decisions as pre-assign surgeries to surgeons.

Thus, the main contribution of this work is the development of a tightened model in terms of integer and continuous variables that allow us to take into account all the features of this problem without considering predefined decisions. This model is formulated taking into account a single surgeon or multiple surgeons working in several ORs and also is able to consider different types of surgeries during a normal working day. In addition, based on this model, a stochastic strategy and a decomposition approach were proposed to solve the problem considering the uncertainty in surgical operation durations. Finally, all these approaches were implemented in the advanced modeling and optimization software AIMMS® widely used for industrial and educational applications. For this, we created a user-friendly interface for hospital managers, where they can easily configure the basic parameters and obtain a reliable solution in short computational time.

The work is organized as follows. Section 2 presents a daily surgical scheduling problem in operating theaters. In section 3, a general MILP formulation is presented considering all the features of this problem. The solutions using a deterministic approach and heuristic-based methods are compared in section 4. Then, section 5 presents the results provided by a stochastic optimization and a MILP-based decomposition approach for solving large-scale scheduling problems. Final conclusions and references are presented at the end.

Daily Surgical Scheduling problem in Operating Theatres

This work studies the scheduling problem of surgical cases raised in operating theatres. This problem assumes that multiple homogeneous ORs and surgeons are available to perform surgical activities, like pre-incision, incision, post-incision operations. According to this, surgeries must be scheduled in order to minimize the total surgical cost formed by OR vacant cost, surgeon waiting cost and OR overtime (see Table 1).

Table 1. Hourly Cost

OR Vacant Cost	*Surgeon Waiting Cost*	*OR Overtime Cost*
CV	*CW*	*CO*
$1,209.60	$1,048.80	$806.40

The set of planned surgeries to be scheduled is known in advance at the surgical day and the number of available ORs and surgeons are given. Thus, different type of surgeries (A-J) must be performed during the surgical time horizon defined by T between 4-12 hours. Each surgery type is characterized by their preparation time (TP), surgery time (TS) and cleaning time (TC). The complete set of data related to the preparation, surgery and cleaning times of different type of surgeries can be found in MOPTA 2013. Then, according to the number of surgeries to be done, the type of surgeries and the time horizon, a set of problem instances are defined (see Table 2).

Table 2. Sequencing Instances

Instance	*T (in hours)*	*No. Surgeries*	*Surgeries to be Sequenced (by Type)*										
			I1	*I2*	*I3*	*I4*	*I5*	*I6*	*I7*	*I8*	*I9*	*I10*	*I11*
1	4	4	A	A	C	J							
2	4	5	A	A	G	H	J						
3	4	5	A	D	G	G	J						
4	8	6	A	B	F	G	G	H					
5	8	7	C	D	F	H	J	J	J				
6	8	10	A	A	A	C	D	G	I	J	J	J	
7	8	11	A	A	F	F	G	H	H	I	I	J	J
8	12	7	A	B	D	E	G	G	J				
9	12	10	A	A	B	D	G	G	I	I	J	J	
10	12	11	A	A	C	E	E	F	G	H	I	I	J

General MILP Formulation

In this section, we present a general MILP continuous time formulation for the daily surgical scheduling problem in the operating theatres with the uncertain in the surgery durations. This model takes into account the surgeries *s, s'*, of each types *i,* to be scheduled during a surgical day and also, considers the set of available surgeons and operation rooms denoted by *k, k'* and *r*. The set of scenarios to be solved are presented by *w* index. The following table provides the full notation about sets, parameters and variables used by the model.

Table 3. Notation of sets, parameters and variables

Index Set	
S	surgeries to be scheduled in a surgical day (s, s')
I	type of surgery (i)
I_s	subset of surgeries s of type i (i_s)
K	surgeons (k, k')
R	operation rooms (r)
W	scenarios (w)

Parameters

TP_{iw}	preparation time of type of surgery $i \in I$ in scenario $w \in W$
TS_{iw}	surgery time of type of surgery $i \in I$ in scenario $w \in W$
TC_{iw}	clean-up time of type of surgery $i \in I$ in scenario $w \in W$
CV	cost per minute of having an OR vacant
CW	cost per minute of having the surgeon waiting
CO	cost per minute of using an OR beyond the normal shift length
T	normal shift length
M	large scalar value much more longer than the normal shift length

Variables

x_{sr}	binary variable, 1 if surgery $s \in S$ is done in room $r \in R$; 0 otherwise
$y_{ss'k}$	binary variable, 1 if $s \in S$ precedes $s' \in S$ in surgeon $k \in K$, 0 otherwise
$z_{ss'kk'}$	binary variable, 1 if s precedes $s' \in S$ and it is done by different surgeon k and $k' \in K$, 0 otherwise
q_{sk}	binary variable, 1 if surgery $s \in S$ is done by surgeon $k \in K$, 0 otherwise
ts_{sw}	start time of the surgery $s \in S$ in scenario $w \in W$
tsS_{kw}	start time of the surgeon $k \in K$ in scenario $w \in W$
msR_{rw}	makespan of room $r \in R$ in scenario $w \in W$
msS_{kw}	makespan of surgeon $k \in K$ in scenario $w \in W$
vt_{rw}	vacant time of room $r \in R$ in scenario $w \in W$
ot_{rw}	overtime of room $r \in R$ in scenario $w \in W$
wt_{kw}	waiting time of surgeon $k \in K$ in scenario $w \in W$
tc	total surgical cost

The principal aim of this MILP model is to minimize the expected total surgical cost represented by tc for a set of selected scenarios w. According to this, two sets of decision variables need to be evaluated. First, the assignment binary variable x_{sr} that determine the allocation of surgery s in operation unit r while q_{sk} provides information about if surgery s in done by surgeon k, adopting both value 1. And then, sequencing binary variables, using the ideas of precedence-based, are proposed to determine if surgery s is done after or before s' in the same surgeon k or in different surgeons k, k' by $y_{ss'k}$ or $z_{ss'kk'}$ respectively.

Note that, all continuous variables associated to the start times of surgeries ts_{sw} and surgeons tsS_{kw}, completion time of rooms msR_{rw} and surgeons msS_{kw} and operating times, as operation room vacant time vt_{rw} and overtime ot_{rw}, and surgeon waiting time wt_{kw}, depends on w and so take a specific value for each scenario.

The main equations of this model are explained as follow. Equation (1) represents the mean total surgical cost *(tc)*, formed by the overtime cost, vacant time cost and waiting time cost, to be minimized by the model for the considering scenarios W. Equation (2) shows that each surgery s must be performed in only one OR r by $x_{sr}=1$. Equation (3) ensures that each surgery is supported by a single surgeon k by adopting $q_{sk}=1$. Sequencing and timing constraints in the same OR and also in the same surgeon are presented by equations (4-5) and equations (8-9) by using binary variables $y_{ss'k}$. In addition binary variable $z_{ss'kk'}$ is introduced in order to consider the sequencing and timing decisions of surgeries performed by different surgeons but in the same OR, as is shown in equations (6-7). Equation (10) defines the completion time of the operation rooms msR_{rw} while equation (11) estimates the completion time in the surgeons msS_{kw}. After that, equation (12-13) is proposed to determine the initial time of each surgery ts_{sw} and surgeon tsS_{kw} in the system, respectively. In addition, the overtime ot_{rw}, vacant time vt_{rw}, and waiting time wt_{kw} variables are calculated in equations (14-16) by using the information of the initial and the completion time of surgeries and surgeons in the system.

$$min.\, tc = \frac{CO}{||W||}\sum_{rw} ot_{rw} + \frac{CV}{||W||}\sum_{rw} vt_{rw} + \frac{CW}{||W||}\sum_{kw} wt_{kw} \quad (1)$$

$$\sum_{r} x_{sr} = 1 \qquad \forall s \in S \quad (2)$$

$$\sum_k q_{sk} = 1 \qquad \forall s \in S \tag{3}$$

$$ts_{sw} + TS_{i_s w} + TC_{i_s w} \le ts_{s'w} - TP_{i_{s'} w} + M(1 - y_{ss'k}) + M(2 - x_{sr} - x_{s'r}) + M(2 - q_{sk} - q_{s'k}) \quad \forall s, s', r, k, w \mid (s' < s) \tag{4}$$

$$ts_{s'w} + TS_{i_{s'} w} + TC_{i_{s'} w} \le ts_{sw} - TP_{i_s w} + M(y_{ss'k}) + M(2 - x_{sr} - x_{s'r}) + M(2 - q_{sk} - q_{s'k}) \quad \forall s, s', r, k, w \mid (s' < s) \tag{5}$$

$$ts_{sw} + TS_{i_s w} + TC_{i_s w} \le ts_{s'w} - TP_{i_{s'} w} + M(1 - z_{ss'kk'}) + M(2 - x_{sr} - x_{s'r}) + M(2 - q_{sk} - q_{s'k}) \quad \forall r, s, s', k, k' \mid (s' < s), (s'k') \wedge (sk) \tag{6}$$

$$ts_{s'w} + TS_{i_{s'} w} + TC_{i_{s'} w} \le ts_{sw} - TP_{i_s w} + M(z_{ss'kk'}) + M(2 - x_{sr} - x_{s'r}) + M(2 - q_{sk} - q_{s'k}) \quad \forall r, s, s', k, k' \mid (s' < s), (s'k') \wedge (sk) \tag{7}$$

$$ts_{sw} + TS_{i_s w} \le ts_{s'w} + M(1 - y_{ss'}) + +M(2 - q_{sk} - q_{s'k}) : \forall s, s', k, w \mid (s' < s) \tag{8}$$

$$ts_{s'w} + TS_{i_{s'} w} \le ts_{s'w} + M(1 - y_{ss'}) + M(2 - q_{sk} - q_{s'k}) : \forall s, s', k, w \mid (s' < s) \tag{9}$$

$$msR_{rw} \ge ts_{sw} + TS_{i_s w} + TC_{i_s w} - M(1 - x_{sr}) : \forall s, r, w \tag{10}$$

$$msS_{kw} \ge ts_{sw} + TS_{i_s w} - M(1 - q_{sk}) : \forall s, k, w \tag{11}$$

$$ts_{sw} \le TP_{i_s w} \ : \forall s, w \tag{12}$$

$$tsS_{kw} \le ts_{sw} + M(1 - q_{sk}) : \forall s, k, w \tag{13}$$

$$ot_{rw} \ge msR_{rw} - T \ : \forall r, w \tag{14}$$

$$vt_{rw} \ge msR_{rw} - \sum_s \left(\left(TP_{i_s w} + TS_{i_s w} + TC_{i_s w} \right) \times q_{sk} \right) : \ \forall r, w \tag{15}$$

$$wt_{kw} \ge msS_{kw} - tss_{kw} - \sum_s \left(TS_{i_s w} \times q_{sk} \right) : \ \forall k, w \tag{16}$$

Deterministic Problem

In this section different approaches are tested using deterministic data for surgical activities. First, we present some heuristic approaches to obtain an initial solution of this problem by considering two operating rooms ORs ($|r|=2$) and a single surgeon ($|k|=1$) with the information of the average scenario ($|w|=w$). Then, we are going to compare the solutions of these heuristic approaches with the ones provided by the full-space MILP model presented above.

Dispatching rules-based heuristic algorithms

In this section, five dispatching rules are evaluated in order to provide, to the hospital manager, a fast and a reliable solution to be implemented. These heuristics are inspired in Iser *et al*. (2008) and Souki (2011). The principal aim of these quick heuristics is to evaluate the solution of the system without using an optimization tool. The first four heuristics, in Algorithms 1-4, are based in a simple sorting criterion ordering the surgeries according to their preparation times (TP) and/or surgery times (TS). The heuristics have been named as "parameter to sort" / type of sorting (A for ascending or D for descending). Finally, we developed a more accurate heuristic specially proposed for this problem structure. This heuristic named as "Ad-Hoc Heuristic" is described as follow in Algorithm 5.

Algorithm 1: Heuristic TS/A

Step 1: Surgeries I of the instance are ordered in ascending order of incision time TS_{iw}.

Step 2: The surgeon in which every surgery is realized is decided taking into account the sequence previously obtained in the Step 1.

Algorithm 2: Heuristic TS/D

Step 1: Surgeries I of the instance are ordered in descending order of incision time TS_{iw}.

Step 2: The surgeon in which every surgery is realized is decided taking into account the sequence previously obtained in the Step 1.

Algorithm 3: Heuristic (TS+TP)/A

Step 1: Surgeries *i* of the instance are ordered in ascending order of the addition of the incision time TS_{iw} and the preparation time TP_{iw}.

Step 2: The surgeon in which every surgery is realized is decided taking into account the sequence previously obtained in the Step 1.

Algorithm 4: Heuristic (TS-TP)/A

Step 1: Surgeries *i* of the instance are ordered in ascending order of the subtraction of the incision time TS_{iw} minus the preparation time TP_{iw}.

```
Step 2: The surgeon in which every surgery is realized is decided taking into account the
sequence previously obtained in the Step 1.
```

Algorithm 5: Ad Hoc Heuristic

```
Create two ascending ordered list using the TS_iw and TP_iw
repeat
       in the first OR, the surgery with the longest TS_iw is selected
       if the surgery has been sequenced before. then
       it is eliminated from the TS_iw list.
       else
       the surgery is assigned and eliminated from the TS_iw list.
       end if
       in the second OR, the surgery with the longest TP_iw that has not been sequenced is assigned
       and the surgeries are eliminated from the list
       if the surgery has been sequenced before. then
       it is eliminated from the TP_iw list
       else
       the surgery is assigned and eliminated from the TP_iw list
       end if
until No more than one surgery is left in the lists
if Both list are empty then
       finish
else this surgery is assigned in the OR which be available first and finish
end if
```

Results

The heuristics and the MILP model presented above were modeled using AIMMS 3.13 (Bisschop and Roelofs, 2011). The solver used was Gurobi 5.0 Optimization (2012) in a PC Intel Core i3-2350M 2.30 GHz with 6 GB RAM under Windows 7. Termination criterion was imposed in 3600 sec. in order to provide good-quality results in reasonable CPU time for the hospital manager.

Solutions obtained in Table 4, demonstrate that in all instances our model solves up to optimality in only few seconds or minutes. For eight of ten cases analyzed the CPU time was less than 1 minute and only for most complex instances (7 and 10) our model takes more time (3 min. and 6 min.). Model size is reported in this table by the number of variables and constraints while the complexity of the solution is demonstrated by the number of nodes and iterations explored. The performance of the model is measured by the relative gap between the initial and final solution and also by the CPU time consumed. The initial solution was reported in all cases in less than 5 seconds. And the relative gap between de initial and final solution was less than 7.0 percent for all cases analyzed.

Table 4. Results of the deterministic problem using (2R, 1k)

Instance	*Total Cost*	*CPU Time*	*Binary Variables*	*Continuous Variables*	*Equations*	*Nodes*	*Iterations*	*Initial solution*
1	480	0.02	14	34	68	193	893	480
2	449	0.03	20	41	98	524	2123	449
3	261	0.03	20	41	98	493	2238	261
4	630	0.06	27	49	134	2378	8845	630
5	943	3.61	35	58	176	98451	376165	943
6	2,165	17.65	65	91	338	547572	2162232	2,299
7	6,186	82.3	77	104	404	2688876	10712959	6,583
8	983	0.87	35	58	176	21767	90654	983
9	1,915	18.34	65	91	338	466586	2098862	2,007
10	4,363	359.17	77	104	404	11043003	44030805	4,701

Our formulation provides a reduced number of binary sequencing variables in comparison with other MILP formulations reported in literature up to now, e.g., Batun *et al*., (2011). Then, our model is much more tightened due to associates a unique general precedence variable to the surgeon when in other formulations the sequencing variables are proposed for each ORs using the concepts of unit-specific precedence-based representation. So, the number of sequencing variables grows up with the number of ORs and always the number of ORs is greater or equal than the number of surgeons. In addition, the unit-specific precedence formulation have to considers all the combinations between two different surgeries *s,s'* where *s≠s'* and the number alternative sequencing decisions for each OR should be $|S|^*|S|-1$. In our model, the number of sequencing decisions for each surgeon is reduced at half.

Figure 1 shows the model behavior for the most complex instance 10 (2R, 1k) of the deterministic problem, drawing the lower bound and the upper bound solutions over time. As we can see in Figure 1, the lower bound was initialized in zero. This is a critical point in the solution performance due to our model could find good-quality initial feasible results in only few seconds but requires plenty of time to assure the optimality of the solution found. Then based on the behavior of our model we could offer optimal solutions within few minutes, or if the instance is small or there is not enough time, you can select an upper time limit. A detailed schedule and costs of this particular instance 10 using (2R, 1K) are reported by Figure 2.

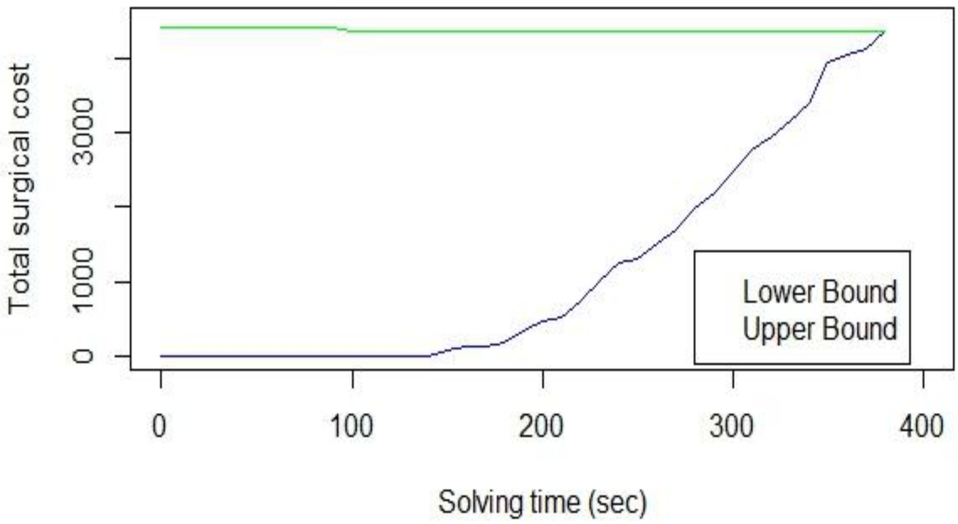

Fig. 1. Solution behavior of the MILP for instance 10 using (2R, 1k)

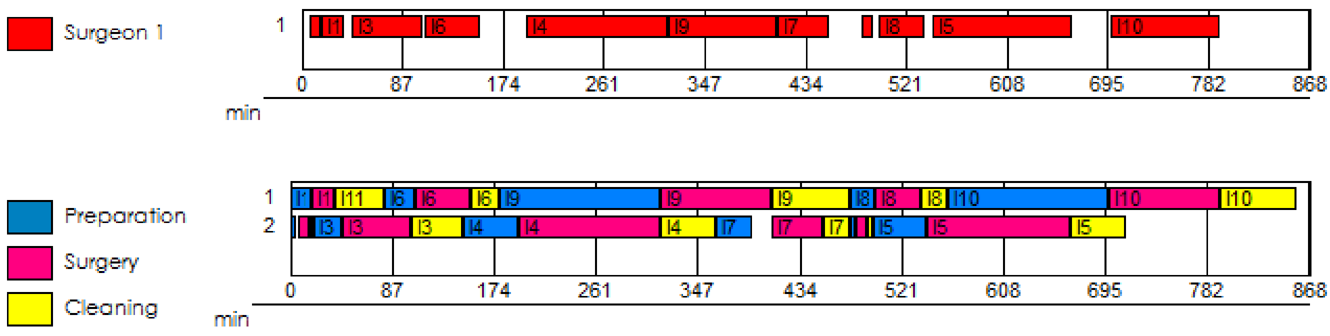

Fig. 2. Solution Schedule of instance 10 using (2R, 1k).

Table 5 shows the results of the heuristics by using the mean scenario. Both, "Heuristic TS/A" and "Heuristic (TS+TP)/A", presents better solutions than the other ("Heuristic TS/D" and "Heuristic (TS-TP)/A") while our "Ad-Hoc Heuristic" provides the best result for each instance. Despite of this the solutions reported by heuristics are still far from the optimal ones obtained for each particular problem instance (see Table 5). In conclusion, heuristic methods based on a simple sorting criterion have poor performance but are fast and can be implemented even by hand. Our MILP model could be used to address multiple surgeons and ORs in simultaneous but in this work we only present the case of a single surgeon and multiple ORs problem.

Table 5. Result comparison for problem (2R, 1k)

Instance	*TS/A*	*TS/D*	*(TS+TP)/A*	*(TS-TP)/A*	*Ad Hoc*	*MILP*
1	2,222	3,710	2,222	3,028	1,569	480
2	3,236	3,538	3,236	4,095	1,098	449
3	4,116	3,836	4,116	4,348	1,059	261
4	5,408	7,531	5,440	5,432	2,359	630
5	6,456	6,853	6,456	5,772	1,470	943
6	9,759	14,440	9,759	15,391	9,362	2,165
7	16,832	18,303	16,760	19,654	9,018	6,186
8	8,615	10,168	8,615	9,302	3,362	983
9	10,918	11,870	10,918	13,866	2,495	1,915
10	18,730	20,809	17,877	20,484	7,930	4,363

Stochastic Problem

In this section we will study the problem in which the duration of surgical activities, closely linked with the surgery's type, is uncertain. All the input data provided by MOPTA 2013 Competition represents historical information which is assumed to be independent and can be modeled by a standard probabilistic distribution with their own parameters. So, no correlations exist among the duration of the pre-incision, the incision, and the post-incision times for each surgery type. According to this, a good solution for a stochastic model will be the one that minimize the expected total cost for all scenarios together. Other approaches that consider only a certain type of cost or use the most likely scenario for the evaluation can be easily implemented.

An extra index for the scenarios w is considered by MILP model in this problem. In here, binary variables do not depend on w assuring the same sequencing and assignment decisions for all the scenarios evaluated. Only timing decisions of surgical activities differ in each scenario. The model is solved considering two operating rooms ORs ($|r|=2$) and a single surgeon ($|k|=1$) minimizing the expected value of the total cost assuming that all proposed scenarios ($|w|=100$) have the same probability of occurrence.

Scenario reduction

In stochastic programming the number of scenarios plays a key role to obtain a reliable solution. For this problem we emulate 100 scenarios using Monte Carlo simulation. We assume that use of the entire set of 100 scenarios gives us the "Optimal value". Then, a suitable reduction of scenarios decrease the solution time but increase the result error. This error will be calculated with the following expression according to the best solution found.

$$Result\ Error = \frac{ObtainedValue - OptimalValue}{OptimalValue} \times 100$$

On the other hand, explore all the scenarios increase the solution time in some cases beyond the threshold. Only the first five instances will be evaluated with the MILP model since it can be solve up to optimality (see Table 6).

Figure 3 shows the % error between the values obtained using a specific number of scenarios from 0-100. When the number of scenarios is below to 20 the error in some cases is above 30%. Then, the error decreases gradually with the number of scenarios. Analyzing that, we conclude that over 50 scenarios the error remains under 10% resulting unnecessary to consider much more number of scenarios for the resolution of the stochastic problem.

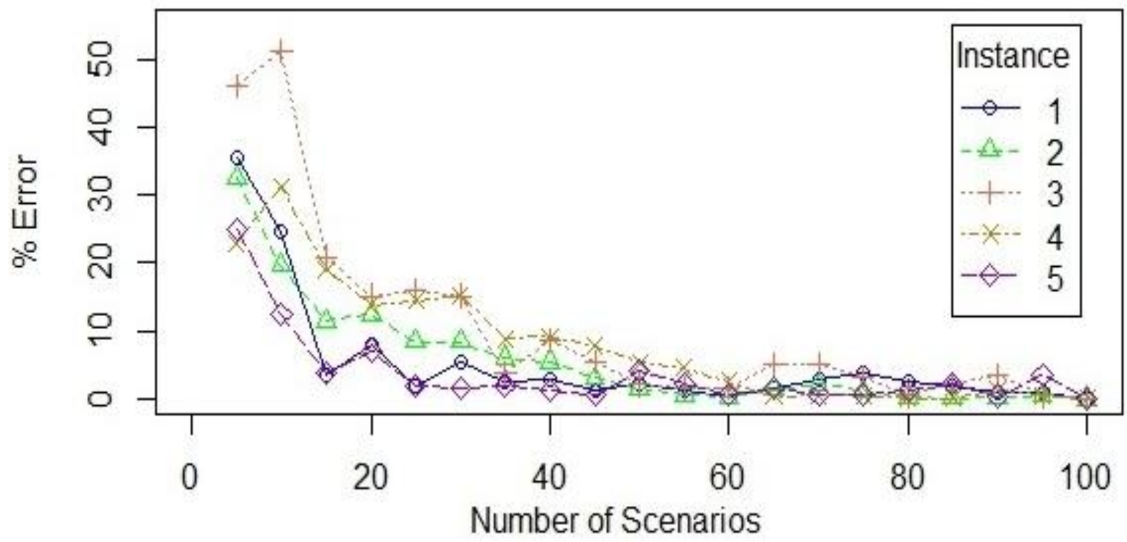

Fig. 3. Analysis of Error vs. Number of scenarios.

Decomposition approach: Constructive-Improvement methods

The MILP-based decomposition approach was developed "ad-hoc" for the specific structure and features of this problem. The constructive-improvement methods were proposed using MILP models, as the one presented above, taking the advantages of General-Precedence (GP) concepts and also the strengths of exchanging information between them. This iterative solution allows decomposing the problem in small sub-problems that can be solved separately, in a sequential way, consuming moderate computational effort. Each algorithm consists in five sequential steps: initialization, selection procedure, setting binary variables, model resolution and updating parameters.

In the constructive algorithm, a reduced MILP model is solved in each iteration obtaining an aggregated schedule with minimum Mean Total Cost (z). When all surgeries are inserted in the system, this phase finishes reporting an Initial Solution (see Algorithm 6).

Then, starting from this solution, the improvement algorithm determinates the surgeries to be released per iteration by chosen the first N consecutive surgeries in the Surgery List. Released surgeries are re-scheduled in the system by optimizing $q_{sk}, x_{sr}, y_{ss'k}, z_{ss'kk'}$ while binary variables of non-released surgeries remain fixed. After solving, the result of the MILP model is compared with the Best solution obtained until this iteration. The Best solution is reported and its schedule is updated. This improvement phase finish when no released surgery can enhance the Best solution found (see Algorithm 7).

Algorithm 6: Constructive Method

Step 1: Initialize parameters *iter, N* and variables $ts_{sw}, q_{sk}, x_{sr}, y_{ss'k}, z_{ss'kk'}$

Step 2: Select N consecutive surgeries to be scheduled in each iteration *iter* by following their lexicographic order from the Surgery List $(s_1, s_2, \ldots, s_S)$

Step 3: Set fixed all binary variables $q_{sk}, x_{sr}, y_{ss'k}, z_{ss'kk'}$ of inserted surgeries.

Step 4: Solve the MILP model for selected surgeries and optimize $ts_{s,w}$ variables of all inserted surgeries.

Step 5: Update parameters and report aggregate schedule. (Back to Step 2)

Algorithm 7: Improvement Method

Step 1: Initialize parameters *iter, N* and start from the initial solution found (Schedule list).

Step 2: Select N consecutive surgeries to be re-scheduled in each iteration *iter* by following their lexicographic order from the Surgery List $(s_1, s_2, \ldots, s_S)$

Step 3: Set fixed all binary variables $q_{sk}, x_{sr}, y_{ss'k}, z_{ss'kk'}$ of non-released surgeries.

Step 4: Solve the MILP model for released surgeries and optimize $ts_{s,w}$ variables of all inserted surgeries.

Step 5: Update parameters and report improvement schedule. (Back to Step 2)

The solution obtained by the "ad-hoc" decomposition approach can be also enhanced by exploiting the strength of the proposed General Precedence MILP formulation, releasing and optimizing much more number of binary variables of non-released surgeries per iteration. According to this, we can play with the assignment variables q_{sk}, x_{sr} improving the model behavior without increasing the number of released variables significantly.

Finally, in both constructive and improvement methods we have decided to use a lexicographic order for the selection procedure since incorporate randomness make impossible the reproducibility of the results. The analysis of different selection rules to improve the solution performance of the algorithm needs to be study in details in future works. The algorithm ends when no another released surgery could improve the Best solution found or after 3600 sec. of CPU time. We adopt this termination criterion in order to make a fair comparison with full-space MILP model presented above.

Results

Increasing the number of scenarios will improve the quality of the solution at the expense of more duration of the experiment. The solution for the 100 scenarios is presented in Table 6 considering all scenarios together and the solution for the mean case using the average scenario. Here, it can be seen that the relative difference between the full case and the mean case is very high. According to this, the use of the full case is much better than using the mean case.

Table 7 shows the principal results and analysis among stochastic, constructive and improvement methods. For the first 4 instances analyzed, both stochastic and decomposition methods provide optimal solutions in short CPU time (< 5min). But, for biggest instances 5 to 10, stochastic model could not ensure optimal results in 1 hour of CPU time. According to this, decomposition approach (constructive method + improvement method) emerges as an efficient solution tool for solving large scale problems with reasonable computational effort.

Table 6. Result of the stochastic problem using 100 scenarios.

Instance	*Total Cost*	*CPU Time*	*Bin Var.*	*Cont Var.*	*Eqs.*	*Nodes*	*Iterations*	*Mean Cost*	*Diff*	*CPU Time*
1	675.84	27	18	1319	6109	171	40730	739	9.3	1.5
2	1172.74	36	25	1426	9011	3002	461512	1220	4.1	4.2
3	1612.88	37	25	1426	9011	1912	322763	1612	0	2.9
4	1670.33	354	33	1534	12513	5201	881252	2046	22.5	6.7
5	2045.37	3600	42	1643	16615	160652	29823967	2299	12.5	12.7
6	4531.56	3600	75	1976	32521	53526	17994882	5364	22.5	47.6
7	10465.19	3600	88	2089	39023	46982	14530619	10084	7.8	132.3
8	2511.68	3600	42	1643	16615	33409	4708673	2921	16.3	9.3
9	5510.55	3600	75	1976	32521	83891	20361739	5238	2.5	43.3
10	10405.46	3600	88	2089	39023	37142	15621953	9549	10.1	410.5

Thus, our constructive algorithm could provide initial good-quality solutions for all these cases in less than 5 min. For the biggest instances in some cases the constructive algorithm obtains better solution in 5 minutes than the stochastic result in one hour. The improvement method increases the quality of the solution in less than half an hour. In almost all cases the decomposition approach obtains the same or a best solution in the half of the time of the stochastic method and average of 3.22% of improvement.

The parameter *N* plays a key role in those algorithms. In the constructive algorithm, it is the number of surgeries inserted at each iteration while in the improvement algorithm it represents the number of release surgeries to be re-scheduled. A small number of *N* narrows the search space with the possibility of eliminate the global optimal but decreasing the CPU time.

Table 7. Comparative using Constructive Method *N*=1, Improvement Method *N*=1

	Stochastic Model		*Constructive Method*			*Improvement Method*			*Total*
Instance	*Total Cost*	*CPU Time*	*Total Cost*	*CPU Time*	*% Imp*	*Total Cost*	*CPU Time*	*% Imp*	*% Imp*
1	675.84	27	675.84	5	0	675.84	24	0	0
2	1172.74	36	1172.74	17	0	1172.74	52	0	0
3	1612.88	37	1612.88	15	0	1612.88	77	0	0
4	1670.33	354	1670.88	29	0	1670.32	85	0	0
5	2045.37	3600	2134.57	51	-4.36	2134.57	126	0	-4.36
6	4531.56	3600	4978.67	192	-9.87	4418.14	1856	12.32	2.50
7	10465.19	3600	9658.81	277	7.71	9089.16	2049	5.44	13.15
8	2511.68	3600	2511.68	48	0	2511.68	121	0	0
9	5510.55	3600	4938.77	156	3.36	4938.77	1324	0	3.36
10	10405.46	3600	8704.83	281	16.34	8579.72	1500	1.20	17.55
Mean				*107.1*	*1.32*		*721.4*	*1.90*	*3.22*

Table 8 shows the results of giving more degree of freedom to the algorithm by inserting and releasing two surgeries instead of one in the both constructive and improvement steps. The constructive method improves the results for only two of ten instances analyzed, since it has more flexibility to construct a better solution. Regrettably, it takes much more time to solve the problem. For the improvement part, when *N=2*, the algorithm takes much more time and no significant improvement can be seen after 3600 sec.

Table 8. Comparative using Constructive Method *N*=2, Improvement Method *N*=2

	Stochastic Model		*Constructive Method*			*Improvement Method*			*Total*
Instance	*Total Cost*	*CPU Time*	*Total Cost*	*CPU Time*	*% Imp*	*Total Cost*	*CPU Time*	*% Imp*	*% Imp*
1	675.84	27	675.84	9	0	675.84	24	0	0
2	1172.74	36	1172.74	34	0	1172.74	112	0	0
3	1612.88	37	1612.88	31	0	1612.88	130	0	0
4	1670.33	354	1670.88	63	0	1670.32	153	0	0
5	2045.37	3600	2134.57	121	-4.36	2134.57	375	0	-4.36
6	4531.56	3600	4480.447	658	1.13	4380.92	3600	2.22	3.32
7	10465.19	3600	9573.68	1174	8.52	9573.68	3600	0	8.52
8	2511.68	3600	2511.68	116	0	2511.68	383	0	0
9	5110.55	3600	4938.77	598	3.36	4938.77	2780	0	3.36
10	10405.46	3600	8704.83	1223	16.34	8579.72	3600	1.44	17.55
Mean	*4020.16*	*2205.4*	*3747.6*	*402.7*	*2.50*	*3725.11*	*1475.7*	*0.36*	*2.83*

Table 9 presents the experimentation of the constructive phase, *N=2*, and the improvement phase, *N=1*. The construction phase obtained better results, but took a longer time since more possibilities are being evaluated at each iteration. The improvement phase makes some improvements of the results of the other options using some extra time.

More experimentation was done using $N > 2$, but the performance was poor. The time stop criterion was applied for the majority of the instances with almost no improvement. As *N* became the Total number of surgeries, the problem transformed into the stochastic model, which had to be solved several times, offering poor performance. As was discussed, small values of *N* should be used.

Table 9. Comparative using Constructive Method *N*=2, Improvement Method *N*=1

	Stochastic Model		*Constructive Method*			*Improvement Method*			*Total*
Instance	*Total Cost*	*CPU Time*	*Total Cost*	*CPU Time*	*% Imp*	*Mean Total Cost*	*CPU Time*	*% Imp*	*% Imp*
1	675.84	27	675.84	9	0	675.84	36	0	0
2	1172.74	36	1172.74	34	0	1172.7	61	0	0
3	1612.88	37	1612.88	31	0	1612.88	162	0	0
4	1670.33	354	1670.88	63	0	1670.33	88	0	0
5	2045.37	3600	2134.57	121	-4.36	2134.6	1200	0	-4.36
6	4531.56	3600	4480.447	658	1.13	4380.9	2891	2.2	3.32
7	10465.19	3600	9573.68	1174	8.52	9089.16	3600	5.06	13.15
8	2511.68	3600	2511.68	116	0	2511.68	176	0	0
9	5110.55	3600	4938.77	598	3.36	4938.77	743	0	3.36
10	10405.46	3600	8704.83	1223	16.34	8579.7	1476	1.4	17.55
Mean	*4020.16*	*2205.4*	*3747.6*	*402.7*	*2.50*	*3711.48*	*1043.3*	*0.88*	*3.30*

As a conclusion, our decomposition method, by using only the constructive phase, could provide even better results than the stochastic model for large problem instances with a significant reduction in CPU time. Also, our algorithm could solve these problems using much more scenarios without significant decrement in the efficiency of the solution found. Finally, possible enhancements can be tested in the algorithm by using different *NxN* parameters and selection rules according to the case study analyzed.

Conclusion

This work presents the main contributions and results obtained for the daily scheduling problem of surgical cases in operating theatres under uncertainty. An efficient and also tightened MILP model was developed taking into account all the features of this challenge problem. In addition, stochastic strategies were implemented in order to deal with the uncertain in surgery durations. Results show that our MILP-based model represents an efficient solution approach for solving deterministic cases, in which timing information is known, providing optimal results in short computational time (< 5min). Also, in stochastic cases, when the prior information is unknown, our stochastic model provides good-quality results but does not assure optimality in a time limit imposed of 1 hour for the largest cases.

In order to improve the solution found and also reduce the CPU time consumed by the stochastic model, a decomposition approach based on constructive and improvement methods was developed. This approach allows decomposing the problem finding an initial good-quality result in less than 5 minutes even for the more complex case in comparison with the full-space stochastic model. Then, an improvement method was applied to enhance the solution in 3.22% (in avg.) in less than 3600 seconds. For example, for the most complex case, our approach was able to improve the solution reported by the full-space model in more than 17% using only 1500 sec. which is quite acceptable for this offline solution purpose. All these solution strategies were implemented in AIMMS® using the principal strength of this modeling and optimization based software. Thus, an end user application was developed with a friendly interface for the hospital manager to introduce and remove data and solve deterministic and stochastic cases without needing any previous information about the result of the problem.

The feedback received from surgeons about the tool was useful to simplify our tool, since the majority of them do not understand operation research terminology, and they want a user-friendly tool with their own terminology that offer reliable and quick results. Unfortunately, many of the surgical scheduling operations in public hospitals are done by hand which represents a lack between data and IT systems. This becomes a challenging opportunity for our application to be implemented at any hospital reducing total surgical costs and at the same time improving resource utilization. As a conclusion, we can realize now how much money is the hospital loosing for do not use the proper scheduling system. If we compare with traditional heuristic rules, the ones probably used in real life, our MILP model could provide a total saving between 25-75%.

In the stochastic problem, the difference from use a decomposition method against the traditional full-space method gives savings of 5% average for largest instances analyzed reducing the CPU time in more than 50%. Use this kind of tools represents a high reduction in total surgical cost and its utilization is really important to everyday scheduling. The specific requirements of the hospital manager will be added in a future step representing the real life conditions more accurately.

Acknowledgments. The financial support receives from the bilateral agreements between EMJD Programme European Doctorate in Industrial Management (EDIM) by the European Commission Erasmus Mundus Action 1 and Universidad Politécnica de Madrid (UPM) and from Erasmus Mundus ARCOIRIS Action 2 between Universidad Nacional del Litoral (UNL) and UPM is fully appreciated.

References

Augusto, V., Xie, X., & Perdomo, V. (2010). Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. Computers & Industrial Engineering, 58(2), 231-238.

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

Bisschop J, Roelofs M (2011). AIMMS 3.13: "The User Guide". Haarlem: Paragon Decision Technology BV, Netherlands.

Capón, E, Bonfill, A, Espuña, A, & Puigjaner, L (2007). Scheduling of flexible multipurpose back-up chemical/ assembly process systems. Proc. of European Congress of Chemical Engineering (ECCE-6) Copenhagen, Denmark, pp 1-11.

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. European Journal of Operational Research, 201(3), 921-932.

Denton, B., Viapiano, J., Vogl, A., (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. Health care management science 10: 13-24.

Fei, H., Meskens, N., & Chu, C. (2010). A planning and scheduling problem for an operating theatre using an open scheduling strategy. Computers & Industrial Engineering, 58(2), 221-230.

Iser, J. H., Denton, B. T., King, RE (2008). Heuristics for balancing operating room and post-anesthesia resources under uncertainty. In: Simulation Conference, 2008. WSC 2008. Winter. IEEE, pp. 1601–1608.

Méndez CA, Cerdá J (2003). Short-term scheduling of multi-stage batch processes subject to limited finite resources. Computer Aided Chemical Engineering 15, 984-989.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers and Chemical Engineering, 30, 913-946.

MOPTA. Competition (2013): Lehigh University Bethlehem, PA, USA. In: http://coral.ie.lehigh.edu/~mopta2013/

Gurobi optimizer reference manual (2012).

Souki, M. (2011). Operating theatre scheduling with fuzzy durations. Journal of Applied Operational Research, 3(3), 177-191.