Employing Fast Heuristics for Operating Room Planning

J.M. Molina-Pariente[1], J. M. Framinan[1], P. Perez-Gonzalez[1]

[1]Industrial Management. School of Engineering. University of Seville, Spain
jmolinap@esi.us.es, framinan@us.es, pazperez@us.es

Keywords: Operating Room Planning Problem, Open Scheduling Policy, Constructive Heuristics.

1. Introduction

Operating theatres, comprising both in facilities and personnel, are one of the most critical and expensive resources in hospitals (Guerriero and Guido 2011). Therefore, adequate decision-making for operating theatre planning and scheduling is of vital importance, as it may greatly impact the quality and the costs of the surgical process.

Operating theatre planning and scheduling is usually decomposed into three hierarchical decision levels (Blake et al. 2002, Testi et al. 2007): Strategic, tactical and operational. This paper focuses on the operational level, where the decision is to specify the date, time and place of each surgery in the waiting list. In most works, this level is solved in two steps (Ozkarahan 2000). In the first step (called *advance scheduling*), the so-called Operating Room Planning Problem (ORPP) is addressed, and a surgery plan is determined to establish which patients in the waiting list will be planned in a given planning horizon, also specifying where (OR) and when (date) each surgery will be performed. The second step (called *allocation scheduling*) determines the patient sequence (time indication) for each OR in each day (Fei et al. 2009).

In this work, we focus on the ORPP under an Open Scheduling policy (in the following called $ORPP_{OS}$), where the surgeons can choose any OR-shift for planning their surgeries (Dexter et al. 2003), in contrast to e.g. Block Scheduling, where fixed OR time blocks are booked by the surgeons. The maximum operating time available for a surgeon on any day is known by the OR scheduler. Objectives usually considered in this problem are related to the minimization of OR costs (e.g. Roland et al. 2010), and to the maximization of the OR utilization (Dexter et al. 1999a).

In our paper, the objective function is the maximization of the Quality of Service, defined as the sum of the quotients between the patient's clinical weight and his/her planned surgery date. This objective is related both to patient waiting time and to the maximization of the OR usage and it is widely employed in the Regional Healthcare System in Andalusia, Spain. The clinical weight (w) of a patient is usually calculated in the literature either as a function of the arrival date and the medical priority of the patient (see e.g. Ogulata and Erol 2003), or simply as a function of the medical priority, which is set by clinicians depending on the diagnostic (see e.g. Ozkarahan 2000, Sier et al. 1997). We will consider here the first option, where w is a linear combination of his/her medical priority (mp) and the number of days the patient has been in the waiting list (dwl).

The $ORPP_{OS}$ is a heavily constrained problem, as there are constraints related to the capacity of the resources (such as OR time and surgeons), time-window constraints (due dates), and forbidden OR-shifts (to e.g. prohibit the assignment of a surgery to an OR where the surgical equipment necessary to perform the surgery is not available). In addition, the number of different ORs assigned to surgeons in a day may be limited.

The $ORPP_{OS}$ can be considered a type of a general assignment problem which is NP-hard, and there have been a number of contributions to solve it. Due to the nature of the problem and its size in practice, it is unlikely to quickly found optimal plans for real life data via exact methods (Roland et al. 2010). Therefore, it seems sensible to employ approximate methods for finding good surgery plans in a short time in order to reduce or avoid disruptions caused by unforeseen events.

Regarding contributions on approximate methods find in the literature, Dexter et al. (1999a) compare on-line (in the following ON) and off-line (in the following OFF) algorithms based on Bin-Packing (BP) to solve the problem of assigning add-on surgeries into the remaining open OR time after elective surgeries are planned. These algorithms have been employed for the ORPP by Fei et al. 2009, Souki and Rebai 2010, Dexter et al. 1999b, and Hans et al. 2008. The main difference between ON and OFF algorithms lies in the knowledge of the surgeries that have to be planned. In ON algorithms, the OR scheduler does not possess information on which surgeries can be planned, planning each surgery as soon as it is submitted in the waiting list. Therefore, an ON algorithm applies a BP algorithm to determine in which OR-shift the surgery is planned. In an OFF algorithm, the OR scheduler has full knowledge of the surgeries in the waiting list before the planning is done. The OFF algorithm is divided into two steps (Dexter et al. 1999a): First, the waiting list is sorted according to a sorting indicator (based on data of surgeries) and a sorting criterion and, secondly, a BP algorithm is applied to assign add-on surgeries to OR-shifts.

Fei et al. (2009) propose a Column-Generation-Based Heuristic (CGBH). The CGBH is tested on instances for which the maximum number of surgeries is one hundred and sixty which are planned into an operating theatre composed by six ORs and eight surgeons. Liu et al. (2011) propose a heuristic procedure based on dynamic programming (in the following DPH) to solve the problem. Computational results show that DPH outperforms CGBH, therefore CGBH will not be considered in the subsequent computational experience. Guinet and Chaabane (2003) propose a primal-dual heuristic (in the following APD), which is an extended version of the Hungarian method. The maximum number of surgeries tested is eighty five which are planned into an operating theatre composed by three ORs. Roland et al. (2010) propose a Genetic Algorithm (GA) based approach. Depending on the parameters of the GA, the time to solve the instances ranges from 305 seconds to 15.1 hours. Therefore, this GA algorithm cannot be consider among the fast heuristics for the problem and will not be included in the subsequent computational experience.

In this paper, we propose several novel fast approximate methods based on BP algorithms to solve the $ORPP_{OS}$. We evaluate their performance and the adaptations of OFF, APD and DPH algorithms for the $ORPP_{OS}$ by generating and comparing the results in a large set of instances. The computational experience shows that the proposed heuristics outperform those existing up to now.

2. Two-Stage Sorting Bin-Packing Algorithms

In this section, we propose a set of constructive methods based on BP algorithms, called Two-stage Sorting Bin-Packing (TSBP). In the literature, there are several contributions (i.e. OFF and APD algorithms) based on sorting the surgeries according to an one-stage sorting (1-SS) procedure, which is characterized by a tuple (*I*, *C*) with *I* the indicator (based on data of surgeries) and *C* the criterion. In this paper, we propose a two-stage sorting (2-SS) procedure based on the fulfillment of due date constraints and on the objective function simultaneously. In order to satisfy due date constraints, surgeries with due dates within the planning horizon are sorted in ascending order of due date at the first stage, while the remaining surgeries (i.e. those with due dates outside the planning horizon) are sorted according to a sorting tuple (*I*, *C*). More specifically, the procedure adopted by our TSBP (*I*, *C*, *BP*) algorithms is composed by two steps: In step I, surgeries are sorted according to a 2-SS procedure as described before, characterized by a tuple (*I*, *C*). In step II, a surgery plan is constructed using a bin packing algorithm of type *BP*. Note that capacity, forbidden OR-shifts and the maximum number of ORs constraints are taken into account at step II.

Several sorting indicators (*I*), sorting criteria (*C*) and types of BP algorithms (*BP*) can be employed to construct the algorithms. The sorting indicators (*I*) considered are: *d* (surgery duration), *dl* (due date), *w* (clinical weight), and *ran* (random sorting), which is equivalent to not to sort the surgeries outside the planning horizon.

The sorting criteria (*C*) that can be considered are (Framinan et al. 2003): *INC* (increasing values), *DEC* (decreasing values), *HIHD* (high values of *I* in the middle of the waiting list and low figures in the beginning and in the end), *HDHI* (low values of *I* in the middle of the waiting list and high figures in the beginning and in the end), *LOHI* (choosing one surgery with low value of *I* and one with high one alternately, starting with the extreme minimum and maximum values), and *HILO* (choosing one surgery with high value of *I* and one with low one alternately, starting with the extreme maximum and minimum values).

Finally, the BP types of algorithms (*BP*) to be considered (see e.g. Dexter et al. 1999a, Dexter et al. 1999b) are:

- Next Fit (NF): The surgery is planned in the last OR-shift occupied, if possible. Otherwise, the surgery is planned into the next available OR-shift.
- First Fit (FF): The surgery is planned in the first OR-shift where it fits.
- Best Fit (BF): the surgery is planned in the OR-shift that has the least amount of available time and it fits.
- Worst Fit (WF): the surgery is planned in the OR-shift that has the most amount available time and it fits.

3. Computational Results and Evaluation

Since our aim is to select the best among the TSBP family of algorithms, and to compare them against existing approaches that may depend on different parameters (such as OFF and APD), we have used a Design of Experiments (DOE) approach for the calibration of OFF, TSBP and APD algorithms. This calibration is done in order to make a fair comparison among all algorithms.

In order to generate appropriate and comprehensive test beds both for calibration and for the comparison of the different algorithms, we review existing test beds in the literature. The parameters employed in the generation of the test bed are listed in Table 1. For the calibration, we generate a 320-instances test bed using the parameters presented above. According to this calibration, the best types of algorithms identified have been: APD (2-SS, *w*, DEC), OFF (*dl*, DEC, FF), DPH, and TSBP (*w*, DEC, FF). All these algorithms are selected for comparison. In addition, we have considered the so-called MIX-TSBP_ALL algorithm, which selects the best solution out of all combinations of *BP* and (*I*, *C*) for an instance.

Table 1. Parameters employed for the test bed design based on the literature

Parameter	Symbol	Units	Generation/Value
Number of days in the planning horizon	n^D	Days	5
Number of operating rooms	n^R	ORs	3,9
Number of surgeons	n^S	-	$\alpha \cdot \sum_{r\in n^R} \sum_{t\in n^D} l_{rt} / c \cdot mds$
Control factor to generate n^S	α	-	1.5, 2.0
Percentage of the regular OR capacity of the planning horizon used to generate surgeries (n^P)	β	-	100%, 125%
Surgery duration	d	Min.	LN[μ, σ]
Expected surgery duration	μ	Min.	Random {60, 120, 180, 240}
Standard deviation of surgery duration	σ	Min.	Random {0.1μ, 0.5μ}
Clinical guarantee	cg	Days	Random {45, 180, 365}
Number of days on waiting list	dwl	Days	U[1, cg-1]
Due date	dl	Days	$dl = cg - dwl$
Medical priority	mp	-	U[1,5]
Clinical weight	w	-	$w = 0.5\cdot mp + 0.5 \cdot dwl$
Forbidden ORs	δ	-	10% surgeries in 30% of ORs
Responsible surgeon	τ	-	U[1,n^S]
Regular OR capacity	l	Min.	480
Max. time available for a surgeon per day	c	Min.	480
Max. no. of ORs that can be assigned to a surgeon in a day	ubr	ORs	1, n^R
Max. no. of days per week in which a surgeon is available	mds	Days	3,4

Table 2. Results of the computational experience in terms of RPD, feasibility and computation times

Problem												
n^R	n^P	ubr	**APD**		**OFF**		**DPH**		**TSBP**		**MIX-TSBP**	
			RPD (feas.)	**secs.**	**RPD (feas.)**	**secs.**	**RPD (feas.)**	**secs.**	**RPD (feas.)**	**secs.**	**RPD (feas.)**	**secs.**
3	50	1	42.03 (94.2)	0.76	26.88 (100)	0.0001	24.10 (78.3)	0.80	21.45 (100)	0.0000	18.08 (100)	0.021
3	49	n^R	30.18 (98.3)	0.64	17.48 (100)	0.0001	5.43 (88.3)	0.77	10.92 (100)	0.0003	8.74 (100)	0.021
3	60	1	44.58 (93.3)	1.00	30.21 (100)	0.0000	27.29 (69.2)	1.19	22.50 (100)	0.0001	19.11 (100)	0.025
3	61	n^R	34.23 (96.7)	0.94	21.47 (100)	0.0001	5.73 (85.0)	1.53	13.62 (100)	0.0000	10.81 (100)	0.023
9	146	1	53.89 (78.3)	47.63	36.59 (100)	0.0011	38.70 (43.3)	146.68	31.32 (100)	0.0006	26.76 (100)	0.306
9	145	n^R	33.91 (96.7)	35.57	17.74 (100)	0.0009	3.90 (72.5)	187.40	10.80 (100)	0.0013	9.49 (100)	0.273
9	182	1	56.58 (61.7)	75.00	41.24 (98.3)	0.0009	41.27 (26.7)	249.81	35.05 (98.3)	0.0013	28.26 (100)	0.381
9	181	n^R	37.72 (92.5)	62.49	21.99 (100)	0.0008	4.15 (69.2)	344.33	13.75 (100)	0.0011	11.20 (100)	0.334
Average			**40.57 (89.0)**	**25.14**	**26.67 (99.8)**	**0.0005**	**15.19 (66.6)**	**95.33**	**19.90 (99.8)**	**0.0006**	**16.56 (100)**	**0.173**

For comparing the algorithms, we have generated a larger test bed consisting in a total of 960 instances. The experiments were carried out on a PC with 2.80 GHz Intel Core i7-930 processor and 16 GBytes of RAM memory. The results of experiments are shown in Table 2. Note that the mean Relative Percentage Deviation (RPD) and CPU-time values are obtained by averaging these results only for feasible solutions (indicated within brackets).

OFF, TSBP and MIX-TSBP_ALL algorithms show an excellent percentage of feasible solutions. However, DPH yields the smallest mean percentage of feasible solutions (66.6%), which we think it is due to the fact that the availability of surgeons is not taken into account. The analysis also shows that there are statistically significant differences between the MIX-TSBP_ALL and DPH algorithms, and the remaining algorithms. There are no statistically significant differences in terms of RPD for DPH and MIX-TSBP_ALL, but the former obtains the best mean RPD value (15.19%). Finally, an important difference in the CPU time used by the different algorithms can be observed. APD and DPH require, on average, more than 25 and 90 seconds, respectively, as compared to less than one second of OFF, TSBP and MIX-TSBP_ALL. The CPU differences between TSBP and MIX-TSBP_ALL are justified in view of the excellent RPD of the latter.

References

Blake, J.T., Donald, J. and Ball, S. 2002, "Mount Sinai hospital uses integer programming to allocate operating room time", *Interfaces*, vol. 32, no. 2, pp. 63-73.

Dexter, F., Macario, A. and Traub, R.D. 1999a, "Which algorithm for scheduling add-on elective cases maximizes operating room utilization? Use of bin packing algorithms and fuzzy constraints in operating room management", *Anesthesiology*, vol. 91, no. 5, pp. 1491-1500.

Dexter, F., Macario, A., Traub, R.D., Hopwood, M. and Lubarsky, D.A. 1999b, "An operating room scheduling strategy to maximize the use of operating room block time: Computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time", *Anesthesia and Analgesia*, vol. 89, no. 1, pp. 7-20

Dexter, F., Traub, R.D. and Macario, A. 2003, "How to release allocated operating room time to increase efficiency: Predicting which surgical service will have the most underutilized operating room time", *Anesthesia and Analgesia*, vol. 96, no. 2, pp. 507-512.

Fei, H., Chu, C. and Meskens, N. 2009, "Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria", *Annals of Operations Research*, vol. 166, no. 1, pp. 91-108.

Framinan, J.M., Leisten, R. and Rajendran, C. 2003, "Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem", *International Journal of Production Research,* vol. 41, no. 1, pp. 121-148.

Guerriero, F. and Guido, R. 2011, "Operational research in the management of the operating theatre: A survey", *Health Care Management Science*, vol. 14, no. 1, pp. 89-114.

Guinet, A. and Chaabane, S. 2003, "Operating theatre planning", *International Journal of Production Economics*, vol. 85, no. 1, pp. 69-81.

Hans, E., Wullink, G., van Houdenhoven, M. and Kazemier, G. 2008, "Robust surgery loading", *European Journal of Operational Research*, vol. 185, no. 3, pp. 1038-1050.

Liu, Y., Chu, C. and Wang, K. 2011, "A new heuristic algorithm for the operating room scheduling problem", *Computers & Industrial Engineering,* vol. 61, no. 3, pp. 865-871.

Ogulata, S.N. and Erol, R. 2003, "A Hierarchical Multiple Criteria Mathematical Programming Approach for Scheduling General Surgery Operations in Large Hospitals", *Journal of Medical Systems*, vol. 27, no. 3, pp. 259-270.

Ozkarahan, I. 2000, "Allocation of surgeries to operating rooms by goal programing", *Journal of Medical Systems*, vol. 24, no. 6, pp. 339-378.

Roland, B., Di Martinelly, C., Riane, F. and Pochet, Y. 2010, "Scheduling an operating theatre under human resource constraints", *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 212-220.

Sier, D., Tobin, P. and McGurk, C. 1997, "Scheduling surgical procedures", *Journal of the Operational Research Society*, vol. 48, no. 9, pp. 884-891.

Souki, M. and Rebai, A. 2010, "Heuristics for the operating theatre planning and scheduling", *Journal of Decision Systems*, vol. 19, no. 2, pp. 225-252.

Testi, A., Tanfani, E. and Torre, G. 2007, "A three-phase approach for operating theatre schedules", *Health Care Management Science*, vol. 10, no. 2, pp. 163-172.