

LIMITED-TIME DECISION MAKING WITH TABU SEARCH FOR AN OPERATING THEATRE DAILY SCHEDULING PROBLEM

Arnauld Hanset¹, Hongying Fei¹, Olivier Roux¹, David Duvivier², Nadine Meskens¹

¹{hanset, fei, roux, meskens}@fucam.ac.be
Louvain School of Management and FUCaM,
151 chaussée de Binche, 7000 Mons, Belgium.

²david.duvivier@lil.univ-littoral.fr
Université du Littoral, Côte d'Opale,
50 rue Ferdinand Buisson BP 719, 62228 Calais Cedex, France.

Abstract: *The operating theatre consists of two parts: a set of operating rooms and a recovery room. It is known as not only one of the most expensive sectors, but one of the most complex sectors to manage and schedule particularly since we consider all the hazards, the large number of actors, the surgical act standardization and coordination difficulties. In this study, we aim to solve the daily scheduling problem for an operating theatre composed of several identical operating rooms and recovery beds by taking into account their availability. The objective is to obtain an operating program minimizing the daily cost of the operating theatre in a limited time. The open-scheduling strategy is implemented in this paper to schedule surgical cases. The problem is solved by a Tabu search meta-heuristic.*

Keywords: scheduling, operating theatre, open-scheduling, Tabu search

1. Introduction

The optimization of the performance of an operating theatre is an extremely complex problem because it comprises multiple constraints, such as the schedule and the competences of specific surgeons as well as the availability of recovery beds [Dexter, 2002]. In this study, we focus on one of the short-term operating theatre management problems: a daily scheduling problem. The target problem is aimed to determine the schedule of surgical cases so as to fulfill the constraints of the recovery room and surgeons with an objective of minimizing the daily operating cost.

Generally, two strategies are used to schedule surgical cases: one is the programming with time blocks reserved in advance; i.e., block-scheduling strategy, where a surgeon has a certain number of pre-reserved blocks of time, in which he/she can assign his/her surgical cases; the other one is the “open-scheduling” strategy, where no time-slot is reserved in advance and surgeons use all available operating rooms (ORs) indifferently. When the latter strategy is applied, some negotiation may be needed to avoid collisions. Considering that the “block-scheduling” strategy can be regarded as a special case of the “open-scheduling” strategy, in this paper, the latter strategy is used to solve the daily scheduling

problem. What's more, since many hospital managers aim to improve efficiency of the operating theatre while reducing the operating cost, the goal of this study is to develop an efficient method to schedule surgical cases into this surgical suit with minimum daily operating cost. As the salary of surgical staff (surgeons, nurses, anesthetists, etc.) and the cost of equipments used in operating rooms are much higher than those in the recovery room [Fei et al., 2007], the objective function of our scheduling model is formulated as the minimization of the weighted sum of the latest completion time among all the surgeries from the operating rooms (*makespan* of the operating rooms) and the latest completion time of the recovery room (*makespan* of the recovery room, which is also the *makespan* of the operating theatre). Moreover, an auxiliary criterion is taken into account to select a final solution among equally rated/ranked solutions on the basis of the main criterion. This objective function is defined as the minimization of the occupation time of the recovery room plus the average occupation time of patients in operating rooms weighted by the ratio (ω) between the cost per hour spent in the operating room and that spent in the recovery room.

In order to solve the daily scheduling model, Fei et al. [Fei et al., 2007] proposed a Tabu search method based on that proposed by Sriskandarajah and Wagneur [Sriskandarajah, 1991]. It results that this method can find a solution of good quality within reasonable execution time. However, since the Tabu search is a stochastic method, it obtains a feasible solution without guarantee of quality. Therefore, it is better to execute the Tabu search procedure several times to improve its robustness. In this paper, our objective is to compare the performance of this metaheuristic with different parameters in order to design an operational tool for the hospital managers. If the proposed method is implemented in hospitals, the hospital managers have a choice between two solutions: one is to execute several times the Tabu search with long runtime in each loop; the other is to execute many times the Tabu search with short runtime. In addition, both of them must be finished in a predetermined time. In this paper, our contribution is to give some clues to make efficient use of available processing time in the context of this time-constrained decision-making problem. The scheduling model and the solution method are taken from Fei et al. [Fei et al., 2007], but the computational results and clues are not given in their work.

This paper is divided into four parts. The first part describes the scheduling model and related notations. Afterwards, the Tabu search procedure is proposed to solve the problem. In the third part, shown are some experimental results based on randomly computed data. This paper ends up with some conclusions and perspectives.

2. Model description and notations

Below are the hypotheses to be considered for constructing the daily scheduling model:

- All the cases are assigned to an operating theatre in advance.
- We suppose an “open-scheduling” strategy.

- The instrumental and human resources, except surgeons, are always available. A surgeon cannot undertake two surgical cases at the same time.
- All the operating rooms are opened simultaneously, and the time horizon is one day.
- All patients are ready to have surgery; i.e., since each patient is normally hospitalized the day before his surgery, he/she is supposed to be ready to have surgery whenever possible.
- A major characteristic of the problem is that, if there is no bed available in the recovery room (RR) at the end of his/her surgical operation, then the patient stays in his/her operating room until one recovery bed (RB) is released or is transferred directly to his/her room if he/she regains his/her consciousness. In the latter case, the patient has spent all his/her recovery time in the operating room. The recovery beds are identical; i.e., a patient can be transferred to any available recovery bed.
- Emergency cases are not taken into account (we assume that emergencies are assigned to a dedicated room). What's more, once a surgical case is started, it cannot be interrupted (preemption is not allowed).
- The cost of one opening hour of the operating room is much higher than that of recovery room (the ratio is $\omega=5.8$).
- The setup time, consisting of the time for cleaning OR and that for making recovery beds, is not considered. It is noteworthy that the setup time doesn't depend on the operation sequence in specificities of the Belgian healthcare system because the cases of nosocomial infections are reported at the end of the day. Consequently the setup time might be indirectly considered by artificially increasing the operating durations so as to include that setup time.

Some researchers have devoted themselves to the study of operating rooms daily scheduling problems [Dexter, 2002], [Fei et al., 2004], [Jebali et al., 2006], [Kharraja et al., 2002], [Sier et al., 1997]. However, as far as we know, few of them take into account the impact of the limited number of recovery beds on the efficiency of an operating program. There are some analogies between the operating theatre and the production workshop. Since many references related to the production scheduling problems have been explored, we based our study on them to solve the operating theatre scheduling problem. In fact, the daily scheduling problem considered in this paper is similar to a hybrid flow-shop model with two stages when regarding surgical cases as "jobs", operating rooms and recovery beds as "identical parallel machines" at two stages respectively. Unlike the classical production hybrid flow-shop scheduling problem, the operating durations in these two stages are not independent. Since the hybrid flow-shop problem is NP-hard, the involved problem is NP-hard as well. In fact, no efficient method can be found to systematically find the optimal solution for all (or even real life) instances of the NP hard problems in a reasonable amount of processing time. Therefore, many researchers focus on developing heuristic methods to generate a feasible solution with good quality in a reasonable execution time [Sriskandarajah, 1991]. We are also interested in developing an efficient heuristic method for the daily scheduling problem with an "open scheduling" strategy.

Traditionally, the objective of the classical hybrid “flow-shop” scheduling problem is to minimize the *makespan*, i.e. the completion time of the last job at the last stage. However this criterion is not sufficient for the daily scheduling problem described in this paper because the fact that one opening hour of the operating room costs much more than one opening hour of the recovery room must be taken into account while solving. What’s more, one overtime hour is more expensive than a regular one. Thus, our objective is to minimize the weighted sum of the *makespans* of operating rooms and of the operating theatre. The auxiliary criterion is formulated as the minimization of the occupation time of the recovery room plus the average occupation time of patients in operating rooms weighted by the ratio (ω) between the cost of one hour spent in the operating room and one hour spent in the recovery room.

For a scheduling day, N patients are ready to enter the operating theatre composed of M_1 identical operating rooms and M_2 identical recovery beds.

Notations are as follows:

- e : The stage of the flow-shop, operating room ($e=1$) or the recovery room ($e=2$)
- Ω : The set of surgical cases awaiting scheduling
- π : A feasible schedule, composed of a sequence of patients passing through operating rooms and the recovery room
- $\pi_k[i]$: The i^{th} patient of the sequence k
- $C_i^{(e)}$: The completion time for the case $i (i \in \Omega)$ at stage e . The time at which the patient leaves the operating room ($e = 1$) or the recovery room ($e = 2$)
- $s_i^{(1)}$: The arrival time for the case i at the first stage. It’s the beginning of the operation if all resources are available
- $s_i^{(2)}$: The arrival time for the case i at the second stage. It’s not always the beginning of the recovery time because when the patient is blocked in the operating room after his operation, he begins to regain his consciousness there.
- $\Delta_{\pi_k[i]}$: A delay resulting from the bottleneck in recovery room.
- $t_{\pi_k[i]}^{(e)}$: The theoretical time spent by a patient at stage (e)
- $t'_{\pi_k[i]}^{(e)}$: The practical time spent by a patient at stage (e) (due to a delay $\Delta_{\pi_k[i]}$)

- $C_{\max}^{(1)}$: The completion time for all cases at stage 1 of the schedule π ,
 $C_{\max}^{(1)} = \max\{C_i^{(1)} | i \in \Omega\}$.
- $C_{\max}^{(2)}$: The completion time for all cases at stage 2 of the schedule π ,
 $C_{\max}^{(2)} = \max\{C_i^{(2)} | i \in \Omega\}$.
- f : The objective function of the schedule π , $f = \omega * C_{\max}^{(1)} + C_{\max}^{(2)}$
 where ω is the ratio between the costs of one hour spent in the operating room and one hour spent in the recovery room.
- f' : The auxiliary criterion, used to select the final solution among solutions with the same value of f
 $f' = \omega * \text{average}(C_i^{(1)} | i \in \Omega) + C_{\max}^{(2)}$

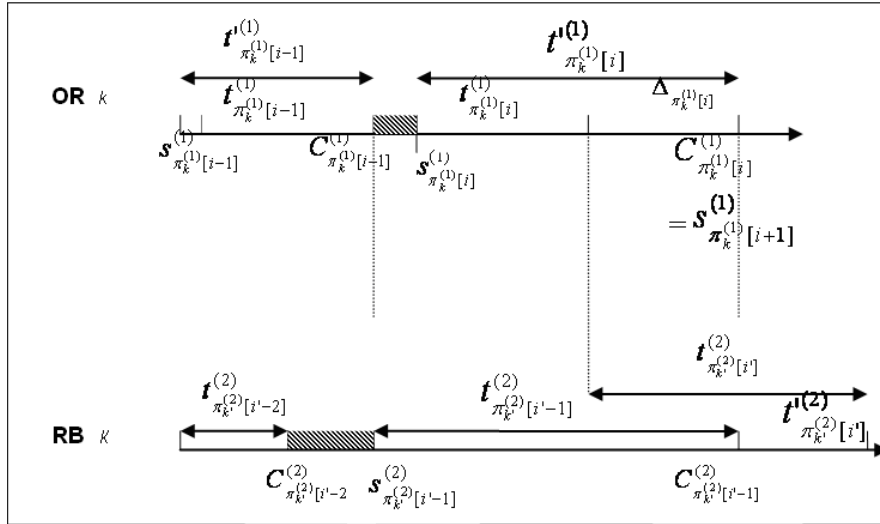


Figure 1: Illustration of our notations

For better comprehension, our notations are illustrated in Figure 1. Using these notations, the daily scheduling problem is described as follows:

Each patient i is treated by the specific surgeon ch_i with a pre-estimated duration $t_i^{(1)}$ in an operating room and then is transferred to the recovery room and stays there until he regains consciousness, unless there is no recovery bed available in the recovery room when the operation is finished in the operating room. In this case, the operating room is still occupied by the patient i after his/her operation until he leaves

or one recovery bed becomes available; i.e., the recovery time $t_i^{(2)}$ can be shared between the O.R. and the recovery room.

3. Scheduling surgical cases with a Tabu search procedure

This section describes the Tabu search method and its application to a daily scheduling problem for the operating theatre.

Developed by Glover [Glover, 1986], the main idea of the Tabu search method is to use a tabu list to keep the latest performed movements in order to avoid being trapped by a local optimum. The name “Tabu list” given by Glover came from the fact that visiting a solution recently explored is forbidden. Once a movement of Tabu search is performed, its inverse movement will be kept in the tabu list. The tabu list aims to prevent (short) cycles in the search space. The size of the Tabu list, either fixed or variable [Taillard, 1990], is an important parameter of the Tabu search method: too small, the algorithm cannot escape from local optima attractive basin; too large, all movements in the neighborhood might be labeled as Tabu (i.e. stored in the Tabu list) and the method is frozen [Hoos and Stützle, 2005].

Moreover, a basic aspiration strategy [Glover, 1989], [Glover, 1990] is implemented in the classical Tabu search procedure: a movement, forbidden by the Tabu list, may be chosen if it leads to a solution with a better cost than the best ever found.

Algorithm 1: Tabu Algorithm

- (1) **Initialization**
 - (2) Generate the initial solution x_0 ; $x^* = x_0$; $c^* \leftarrow f(x_0)$; x^* is the best ever found schedule, whose objective value is c^*
 - (3) $k \leftarrow 0$; TabuList = \emptyset
 - (4) **while** stopping-condition is not satisfied
 - (5) Choose the best (x_k) with minimum $f(x_k)$ which is not a member of TabuList, unless $c_k < c^*$ is satisfied, among the neighborhood of x_k , $V(x_k)$, generated by taking one possible movement from x_k .
 - (6) $x_{k+1} \leftarrow \text{best}(x_k)$
 - (7) **if** $c(x_{k+1}) < c^*$ **then**
 $\{x^* \leftarrow x_{k+1}, c^* \leftarrow c(x_{k+1})\}$
 - (8) $k \leftarrow k + 1$
 - (9) Update (TabuList)
-

In this algorithm, a movement corresponds to the application of an operator that allows the Tabu search procedure to move from a feasible solution to another neighboring solution (so-called neighbor). The neighborhood of one solution is composed of all the solutions reachable by one application of the chosen operator to this point.

According to the literature, the basic Tabu search method described above can be improved in some ways as follows:

- Intensification strategy: Several solutions, selected among the best solutions found, are used to generate new solutions close to the optimum. [Crainic et al., 1993].
- Diversification strategy: unlike the previous strategy, it consists in memorizing the appearance frequency of each element of the visited solutions. These frequencies are used to calculate some penalties to keep the Tabu procedure away from zones already explored [4], [Rochat and Taillard, 1995].
- Aspiration strategy: various more sophisticated strategies may be implemented instead of the basic/classical aspiration strategy.

In this paper, the Tabu search method is customized to deal with the daily open-scheduling problem under consideration. In this custom-made Tabu search (CTS) procedure, a decomposition heuristic (HD) procedure is developed to construct an initial feasible solution (details of the HD procedure can be found at point 2 of the CTS procedure hereafter).

Adaptations introduced into the CTS procedure are as follows:

1. Encoding strategy: a feasible solution is represented by an array composed of five vectors.

V_1 : a vector of size N , representing the sequence of patients at the first stage;

V_2 : a vector of separation between the different OR. The size is $(M_1 - 1)$ and the numbers of cases among V_1 affected to the first OR, next the second OR ... until the $(M_1 - 1)$ OR are indicated;

V_3 : a vector of size N , representing the distribution of patients in the recovery beds at the second stage. The patient order represented in this vector corresponds to that in the vector V_1 . One corresponding to the first RB, two the second ...

V_4 : a vector of size N , representing the sequence of patients in the recovery room;

V_5 : a vector of separation between the different recovery beds. The size is $(M_2 - 1)$ and the numbers of cases among V_2 affected to the first RB, next the second RB ... until the $(M_2 - 1)$ RB are indicated. V_5 is redundant, it can be inferred from V_3 . However it is convenient for simplifying detailed explanation about our encoding scheme.

Array	1 2 3 4 5 6 7	3	1 2 1 2 1 2 1	1 5 3 7 4 2 6	4
Vectors	V_1	V_2	V_3	V_4	V_5

Figure 2: An encoded (feasible) solution

Figure 2 shows a (feasible) solution encoded by the coding strategy described above. This array represents a feasible daily schedule for an operating theatre. The size of V_2 and V_5 is 1, thus the operating theatre is composed of two ORs and two RBs in the recovery room. V_2 shows the location of the separation between the two ORs (i.e. 3 operations in the first OR when considering the above mentioned example). Thus,

V_I represents an operating order in operating rooms, where the first three cases (1, 2 and 3) are scheduled in the first OR in this order, and the remaining four cases are scheduled in the order (4, 5,6 then 7) in the second OR. V_3 indicates that four patients go to RB number one and three to RB number two. After being operated, patients 1, 3, 5 and 7 are transferred to the first RB in the order (1-5-3-7) and the others are transferred to the second RB in the order (4-2-6). V_5 confirms that four patients go to RB number one. Once values of all elements within every array are determined, a feasible daily operating schedule is then obtained.

2. Initial solution: the initial solution is generated by the HD procedure as follows: first, the patients are randomly assigned to the operating rooms. When all patients are assigned, the patient sequence in each operating room is determined, and therefore, the vector V_I is also determined. Second, each patient leaving his operating room is assigned to a randomly selected RB, available at that moment, following the operating order at the first stage; i.e., operating rooms.

Our Tabu search procedure uses the following strategies of a neighbor: the “First fit” strategy, selecting the first solution among the neighborhood, is used; and the “Best fit” strategy; choosing the best solution found to be better than the current one is implemented.

Neighborhood definition: a two dimensional neighborhood is constructed as follows:

- a) On the first dimension, just operating rooms are taken into account: the neighborhood exploration consists in moving a patient, randomly chosen, to one of the other rooms and test all the feasible positions to determine the best position for this patient in this room (“Best fit” strategy). If a generated solution is found to be better than the best ever found solution, it becomes the current solution (“First fit” strategy), and then we start the next iteration in the Tabu search, otherwise go to step b.
- b) On the second dimension, the recovery room is considered: the same exploration as above is made for the recovery beds.
- c) If the best solution found till the last iteration has been modified in the first two steps then it becomes the current solution, else the current solution is set as the better one between those found in the first two steps.

The stochastic aspects of the Tabu search appear, firstly in the construction of the initial solution, and secondly, in the neighborhood structure.

3. Evaluation function: the objective function consists in the minimization of the weighted sum of *makespans*: $\omega C_{\max}^{(1)} + C_{\max}^{(2)}$.

If several solutions have the same value, we distinguish it by the secondary criterion: $f' = \omega * average(C_i^{(1)} | i \in \Omega) + C_{\max}^{(2)}$

4. Tabu list management: the reverse movements are stored in the Tabu list. This list is managed with a FIFO rule (*First In, First Out*) and its size is arbitrarily fixed at 7.

5. Aspiration criterion: if a movement is banned by the Tabu list but leads to a better solution than the best ever found solution, then this movement will still be accepted.

4. Experimental numerical results

All experiments, are executed on a PC compatible (CPU: PENTIUM IV 3.0 MHz, Memory: 512 MB, Operating System: Windows XP), the compiler is Microsoft Visual C++ 2003.

4.1 Data

The experimental data are as follows (durations are in minutes):

- Operating duration $ti(1)$ of the case i is generated from a Pearson III [Combes et al., 2004] distribution rule between [30, 150] with the mean 60 and the standard deviation 15.
- The recovery duration $ti(2)$ of the case i is generated from a Pearson III distribution rule between [30, 120] with the mean equal to the corresponding case's operating duration $ti(1)$ minus 10. This way of generating the recovery duration is used by some other authors as well [5], [6], [Dexter, 1995], except that a Pearson III distribution rule is used instead of a lognormal one in this paper. The standard deviation is equal to 15.
- The ratio between the cost of one opening hour in the operating room and the cost of one opening hour in the recovery room is $\omega = 5.8$;
- There are 6 ORs and 10 RBs in the operating theatre under consideration;
- All surgical cases are randomly assigned to 8 surgeons in advance;
- The number of cases is 50.

To be complete further numerical experimentations have been made, but we don't present it here.

4.2 Numerical results

Numerical results are based on only one instance of the scheduling problem.

We have compared the cost of the solutions obtained by the Tabu search with the lower bound proposed by Fei et al. [4] for the same daily scheduling problem. This lower bound is calculated as follows:

$$OLB = \omega * \max \left\{ \frac{\sum_{i \in \Omega} t_i^{(1)}}{M_1} \right\} + \max \left\{ t_{i^0}^{(1)} + t_{i^0}^{(2)}, \frac{SPT(M_2) + \sum_{i \in \Omega} (t_i^{(1)} + t_i^{(2)})}{M_1 + M_2} \right\}$$

Notations are as follows:

- i_0 is a patient that begin his/her operation at the beginning of the day and that recovery at last.
- $SPT(M_2)$ is the minimum sum of the achieving time of the M_2 patients at first stage with the shortest operating time.

The lower bound, in our example, is equal to 20654. However it is noteworthy that this cost corresponds to a non-feasible solution. Practically, the cost of the best ever found solution is 26181.

Three scenarios, running with the same execution time as that taken by one replication of a Tabu search with 300000 steps, have been tested in experimentation. These scenarios, respecting the limitation of number of steps, are characterized by different number of replications and number of steps. We observe the variations of performance of the Tabu search when the number of runs (replications) decreases (6000, 30, 20) and the number of steps in each run increases (50, 10000, 15000). Therefore, these three tested scenarios are as follows: 6000 replications of a Tabu search with 50 steps (short runs); 30 replications with 10000 steps (medium runs), and 20 replications with 15000 steps (long runs). When several runs are performed, we start from different initial solutions (using different seed for the random number generator) for the selected instance of the problem.

Run	# Steps	# Runs	Average	Std. Dev.	Time (sec.)
Short	50	6000	48986	8342	0.2
Medium	10000	30	34629	8118	40
Long	15000	20	32777	4941	60

Run	Minimum	Maximum	Q1	Median	Q3
Short	31235	43368	43036	47272	52935
Medium	26181	55405	29123	32691	36237
Long	26619	90882	29623	31030	35264

Table 1: Numerical Results

The results show that the best values, for medium and long runs, are taken close to each other. The limited-time decision to execute the short runs or the medium runs or the long runs is fixed, by construction ($\#Steps * \#Runs = Const$), at approximately 20 minutes.

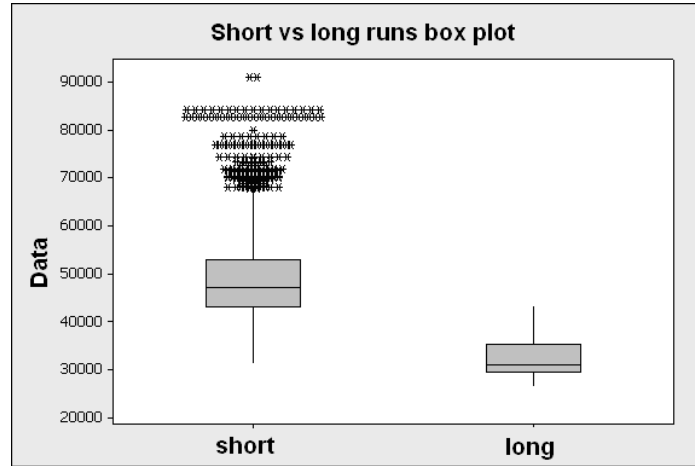


Figure 3: Short vs. long runs box plot

The results shown in Table 1 are as follows: the number of steps (# Steps) per execution (replication), the number of executions of the Tabu method (# Runs), some statistics of the objective values obtained over the runs (average, standard deviation, minimum, maximum, first quartile, median, third quartile). The box plots of the objective value are shown in Figures 3, 4 and 5.

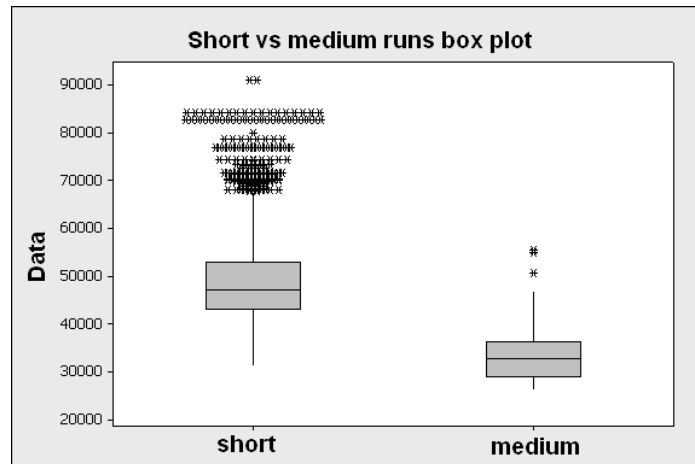


Figure 4: Short vs. medium runs box plot

The results given (in Table 1) show that the scenario with long runs performs better in average than that with short runs. Furthermore, this conclusion is confirmed by the box plot represented in figure 3, and the results of Mann-Whitney non-parametric statistical test, which rejects the assumption of equality at 99% with a significant test at 0. In addition, the results in Figure 4 show that the scenario with medium runs performs better than that with short. This result is confirmed by the

results of the Mann-Whitney test rejecting the assumption of equality at 99% with a significant test at 0.

By contrast (cf. Figure 5), no significant difference is found between the medium-runs and long-runs group (the Mann-Whitney test doesn't reject the assumption of equality at 95%, with a significant test at 0.7589).

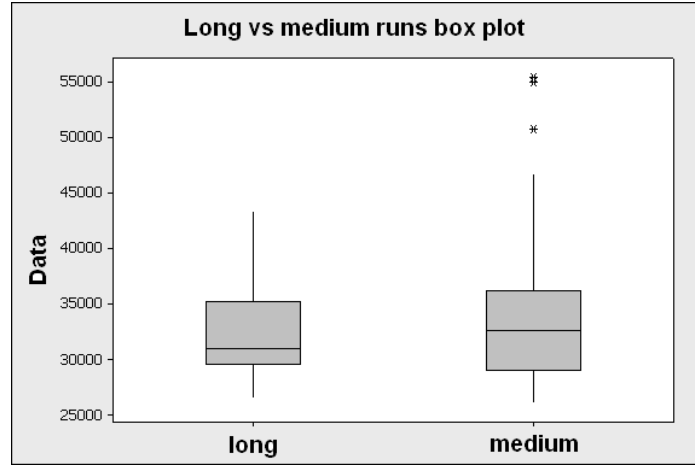


Figure 5: Long vs. medium runs box plot

5. Conclusions and some perspectives

In this paper, a daily scheduling problem of the operating theatre including the management of recovery beds has been presented. This scheduling problem is modeled as a hybrid “flow-shop” scheduling problem with two stages and an efficient dedicated Tabu search method has been described.

Comparing the results obtained by the Tabu search with the lower bound, we have shown that our method can provide the hospital managers with a good quality solution in a reasonable time. Furthermore, we conclude that the medium-runs group is the best strategy because it not only obtains better results than the short-runs group, but also has a performance similar to the long-runs group. In addition, as the medium-runs group leads to more exploration of the search space than the long-runs group, there is a higher probability to find a better solution. In the light of those results, we can conclude that we have improved the metaheuristic, which was used by [2] as local operator improvement.

In our future research, to go out of the Belgian healthcare context, we will take into account the setup time between two surgical cases, especially the necessary time for changing materials and preparing the equipments. We will also take into account the availability of other necessary human and material resources in our model. We will tune the Tabu search using several sizes of the Tabu list (either static or dynamic

sizes) and several instances of the problem so as to judiciously balance robustness and efficiency in our time-limited decision-making context.

References

- [Combes et al., 2004] C. Combes, A. Dussauchoy, S. Chaabane, N. Smolski, J.P. Viale and J.C. Souquet (2004). Démarche méthodologique d'analyse des données pour la planification des blocs opératoires : une application à un service d'endoscopie, *Actes GISEH'04, Mons, Belgium*.
- [Crainic et al., 1993] T. Crainic, M. Gendreau, P. Soriano and M. Toulouse (1993). A tabu search procedure for multicommodity location/allocation with balancing requirements. *Annals of Operations Research*, 42((1-4)), 359-383.
- [Dexter, 1995] F. Dexter, J. Tinker (1995). Analysis of strategies to decrease postanesthesia care unit costs, *Anesthesiology*, 82(1), 94-101.
- [Dexter, 2002] F. Dexter, R.D. Traub (2002). How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time, *Anesthesia & Analgesia*, 94, 933-942.
- [Fei et al., 2004] H. Fei, A. Artiba, C. Chu (2004). A memetic algorithm for operating room scheduling, *International Conference on Computational & Experimental Engineering and Sciences, Portugal*.
- [Fei et al., 2007] H. Fei, N. Meskens, C. Chu (2007). An operating theatre planning and scheduling problem in the case of an 'open scheduling' strategy, *IESM2007, China*.
- [Glover, 1986] F. Glover (1986). Future paths for integer programming and link to A.I.. *Computers and Operations Research*, 533-549.
- [Glover, 1989] Glover F. (1989). Tabu search - part 1. *ORSA Journal on Computing*, 1(3), 190-206.
- [Glover, 1990] Glover F. (1990). Tabu search - part 2. *ORSA Journal on Computing*, 2(1), 4-32.
- [Hoos and Stützle, 2005] H.H. Hoos and T. Stützle, (2005), Stochastic local search: foundations and applications, Elsevier/Morgan Kaufmann, San Francisco
- [Jebali et al., 2006] A. Jebali, A.B. Hadj Alouane, P. Ladet (2006). Operating room scheduling, *International Journal of Productions Economics*, 99, 52-62.
- [Kharraja et al., 2002] S. Kharraja, S. Chaabane, E. Marcon (2002). Evaluation de performances pour deux stratégies de programmation opératoire de bloc, *In Proceedings of second French international conference of automatic, France*.
- [Rochat and Taillard, 1995] Y. Rochat and E.D. Taillard (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 147-167.
- [Sier et al., 1997] D. Sier, P. Tobin, C. McGurk (1997). Scheduling surgical procedures, *Journal of Operating Research Society*, 48, 884-891.
- [Skorin-Kapov, 1990] J. Skorin-Kapov (1990). Tabu search applied to the quadratic assignment problem. *ORSA Journal of Computing*, 2 (1), 33-45.
- [Sriskandarajah, 1991] C. Sriskandarajah, E. Wagneur (1991). Hierarchical control of the two processors flow-shop with state dependent processing times: complexity

analysis and approximate algorithms, INFOR, *Canadian Journal of Information Systems and Operational Research*, 29(3), 193-205.

[Taillard, 1990] E.D. Taillard (1991). Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17, 443-455.