HAL
archives-ouvertes.fr

Operating theatre scheduling with patient recovery in both operating rooms and recovery beds

Vincent Augusto, Xiaolan Xie, Viviana Perdomo

▶ **To cite this version:**

HAL Id: hal-00510507

https://hal.archives-ouvertes.fr/hal-00510507

Submitted on 19 Aug 2010

Operating theatre scheduling with patient recovery in both operating rooms and recovery beds

Abstract : This paper investigates the impact of allowing patient recovery in the operating room when no recovery bed is available. Three types of identical resources are considered : transporters, operating rooms and recovery beds. A fixed number of patients must be planned over a term horizon, usually one or two weeks. The surgery process is modelled as follows : each patient is transported from the ward to the operating theatre. Then the patient visits an operating room for surgery operation and is transferred to the recovery room. If no recovery bed is available, the patient wakes up in the operating room until a bed becomes available. The operating room needs to be cleaned after the patient's departure, before starting another operation. Finally, the patient is transported back to the ward after his recovery. We consider several criteria based on patients' completion times. We propose a Lagrangian relaxation-based method to solve this operating theatre scheduling problem. The efficiency of this method is then validated by numerical experiments. A comprehensive numerical experiment is then performed to quantify the benefit of allowing patient recovery in operating rooms. We show that the benefit is high when the workload of the recovery beds is high.

1 Introduction

1.1 Context

Application of operational research methods in health care systems has proven effective for design and operation of the operating theatres. New organisations of health care systems and new health care delivery technologies emerge to better cope with cost pressure while keeping the quality of service and care high. Guinet and Chaabane (2003) [5] described an hospital as a multi-service production system which is constrained by limited material and human resources. Optimal usage of expensive resources is highly important in cost containment. Surgeries are critical processes in hospital operation, not only because of their high costs but also because of their direct impact on patient's health and on quality perception. Therefore the operating theatre scheduling has been widely studied in order to maximize the utilization of operating rooms and enhance the patient's satisfaction. Surgery procedures involve about 9% of total costs of the hospital (Gordon et al., 1988) [4]. Dexter (2002) [1] showed that one hour overtime cost of any operating room is about 1.75 time of its usual cost.

In an operating theatre, when a surgery operation ends, the patient recovers slowly under appropriate monitoring of competent personal and with necessary equipment. Patient recovery usually takes places in a bed in the recovery room. As a result, an operated patient must be transferred in a recovery bed after the end of the surgery. However recovery process can be longer than expected, and situations where no recovery beds are available occur sometimes. Instead of delaying surgery operations, letting the recovery occurs in the operating room could be a cleaver solution.

1.2 Literature review

The purpose of this paper is to investigate the impact of allowing patient recovery in operating rooms. This strategy has been scarcely addressed in the literature.

The operating theatre scheduling problem consists in scheduling surgery operations over a certain time period (usually a week), taking into account available resources. Operating theatres can be classified into several categories depending on medical equipments, operating rooms features and localisation in the hospital (centralised theatre or dedicated theatre). Depending of the operating theatre type, there are two surgery scheduling strategies in the literature : « block scheduling » and « open scheduling » [6].

In the block scheduling policy, a preliminary time table called master schedule is established in order to allocate time slots to surgeons, groups of surgeons or medical specialities. Determination of an optimal schedule is a difficult problem [15]. Guinet and Chaabane (2003) [5] proposed a two-step approach where patients are first assigned to operating rooms, and then each operating room is scheduled using an extension of the Hungarian method. Jebali *et al.* (2006) [7] studied the same problem by focusing on operating rooms and provided a mixed integer linear programming formulation that they solved with commercial software package. Fei *et al.* (2006b) [3] also proposed a two-step approach : a weekly schedule is created using column generation, then the daily scheduling problem is solved using an hybrid genetic algorithm. Finally, Kharraja *et al.* (2005) [8] studied using simulation a problem similar to ours where patient recovery in operating room was allowed but patient transportation was not taken into account.

In the open scheduling policy, no pre allocation is used, patients are scheduled without any specialty related restriction. In this context, operating theatre scheduling with constraints of patient recovery and transportation is scarcely addressed. Most existing models do not consider patient recovery in operating rooms. The work most related to this paper is Fei *et al.* (2006a) [2], where a daily surgery schedule is built for an operating theatre. Operating rooms and recovery beds are identical. Patient recovery in operating rooms is allowed. No transporters are considered. A hybrid genetic algorithm was proposed to solve the problem. Marcon *et al.* (2003) [14] developed a simulation model to find an optimal scheduling strategy, taking into account the whole surgery process. A mathematical model is proposed to minimize the overtime for operating rooms. The simulation model is used to model the patient pathway throughout the operating theatre. Bottleneck resources of the system are the transporters. Emergencies have been considered in the operating theatre by Lamiri *et al.* (2006) in [10] : a stochastic scheduling model has been presented in this article. In [11] the same problem is solved using a column generation procedure.

The modified block scheduling policy is an intermediate strategy which combines the open and block schedulin strategies. First the weekly planning is divided into time slots. Two strategies are commonly used : (i) some time slots are allocated to surgeons, surgeons teams or medical specialities, and the remaining time slots are common (unassigned block) ; (ii) after a given time (block release time), unaffected time slots are re-assigned in order to maximize the operating theatre utilisation [13].

1.3 Scientific contribution

In this paper we focus on the open scheduling policy. The operating theatre scheduling problem considered here is similar to the one considered in (anonymous ref.) and both take into account not only availability of operating rooms but also availability of porters and recovery beds. In contrast to the previous work which does not allow recovery in the operating room, patient recovery in the operating room is allowed in this paper when no recovery bed is available at the end of the surgery. This situation occurs when the when workload of the recovery beds is high with respect to the workload of operating rooms, i.e. when the operating theatre is crowded : a large number of emergency patients have to be operated in regular operating rooms could be a reason to allow recovery in operating rooms if the recovery room is full. Furthermore allowing patient recovery in operating rooms could be a cleaver strategy when recovery process times grow high, i.e. when workload of recovery beds becomes greater than that of operating rooms. We consider here critical situations in the operating theatres that often occur according to surgeons' expertise. A Lagrangian relaxation method is proposed. A comprehensive numerical experiment is performed to investigate the benefit of allowing patient recovery in operating rooms. In particular, we show that the benefit is significant when the workload of the recovery beds is high with respect to the workload of operating rooms.

Notice that cost of operating room utilization during one hour is two to five times greater than cost of recovery bed utilization during the same amount of time, depending on surgery type. From this point of view, allowing recovery in operating rooms during patient scheduling may increase surgery costs. However we focus in this problem on makespan criterion (minimize operating theatre closing time), in order to process as much patients as possible during a short amount of time. Taking into account costs comparison may be examined in perspectives of this work.

The rest of the paper is organized as follows. The problem setting is introduced in Section 2. Section 3 presents a mathematical model and Section 4 describes the Lagrangian relaxation approach. Numerical results are given and analyzed in Section 5.

2 Problem setting

The surgery process followed by a patient can be divided in several steps ; we consider that a patient must go through the surgery process on a fixed day, at a fixed hour.

1. The patient is transported from the ward to the operating theatre by a porter team.

2. If no operating room is available, the patient must wait in a waiting room. We consider the surgeon, the nurses, the medical equipment and the operating room as an unique resource.

3. When an operating room becomes available, the surgery operation starts.

4. At the end of this operation, the patient is transferred to a recovery bed in the recovery room. If no recovery bed is available, the patient recovery begins in the operating room which becomes blocked. The patient leaves the operating room when the recovery is completed or when a recovery bed becomes available. In the latter case, the patient is immediately transferred to the recovery room.

5. The operating room is cleaned as soon as the patient leaves.

6. When the patient has recovered from his surgery operation, he must wait for a porter team to be transferred to the ward.

Fig. 1 – Surgery Process

The entire process is illustrated in figure 1.

Taking into consideration the processing time of each activity, the surgery and recovery operations are the most critical activities of the operation process. In this paper, we assume deterministic processing times for all operations although surgery time for example is hard to predict and directly affects the utilization of the most critical resources, the operating rooms. Operating time estimation depends on surgeon expertise and on patient pathology (Wright, 1996) [16].

The problem studied in this paper can be considered as a four-stage hybrid flowshop, where each resources (transporters, operating rooms and recovery beds) are identical parallel machines. The four stages are : (i) the transport from the ward to the operating theatre, (ii) the surgery and recovery, (iii) the cleaning task and (iv) the transport from the operating theatre to the ward. We also have a blocking constraint between the surgery operation and the recovery operation for each patient. A finite time horizon is given for scheduling all patients. The goal is to minimize a regular criterion of patients' completion times in order to minimize the total cost of the operating theatre activity.

The hybrid flowshop scheduling problem consists in optimizing a criterion (usually minimizing the makespan) in a serial multiple-stage manufacturing system, where each stage consists of parallel identical machines. During the last decade, both hybrid flowshop scheduling problems and its numerous variations alike have been intensely investigated in the scheduling literature. The reader is referred to [12] and [9] for comprehensive surveys.

3   Mathematical model

In this paper, N surgery operations must be scheduled in the operation theatre during H time units, for example one day. Let p_{ij} be the processing time of task j for patient i. The surgery of each patient i involves five different tasks : transport from the ward to the operating theatre (p_{i1}), surgery operation (p_{i2}), recovery in a recovery bed (p_{i3}), transport from the operating theatre to the ward (p_{i4}) and operating room cleaning (p_{i5}).

The following assumptions are made. (H_1) Surgeries from the emergency department are not taken into account : in reality, emergency surgeries are often treated in a dedicated operating room which is not used for elective interventions. In addition, there is a surgeon in charge of emergency cases. This is why we do not consider emergency surgeries in our scheduling model. (H_2) Surgery team resource constraints are not taken into account : we consider all needed resources (operating room, surgeon, nurses, etc.) for surgery as only one resource called "operating room". (H_3) The recovery operation starts immediately after the surgery operation, and can occur in a recovery bed, in an operating room, or in both places.

Considering surgery team resources individually could be possible : we would have to take into account a certain number of additional resources, with various availability time constraints. Complexity of the problem would be much greater. Moreover, considering the operation room as several individual resources is not really relevant in this case because surgeon, nurses, anaesthesiologist, etc. are considered as a whole team on the field. Some studies focused on taking into account surgeon's availability, which is still a problem in operating theatre planning. Considering this problem is part of our perspectives.

The following resources are involved in the problem.
– P_t : Number of identical porter teams at time t.
– R_t : Number of identical operating rooms at time t.
– B_t : Number of identical recovery beds at time t.

Decision variables are s_{ij}, the starting time for task j of patient i and α_i, the recovery time patient i spent in an operating room. We can now state the mathematical model of the problem :

Minimize :

$$J = \sum_i f(c_i) \tag{1}$$

Subject to the following constraints :

$$s_{i1} + p_{i1} \leq s_{i2} \quad \forall i \in \{1 \dots n\} \tag{2a}$$

$$s_{i2} + p_{i2} + \alpha_i = s_{i3} \quad \forall i \in \{1 \dots n\} \tag{2b}$$

$$s_{i2} + p_{i2} + \alpha_i = s_{i5} \quad \forall i \in \{1 \dots n\} \tag{2c}$$

$$s_{i3} + p_{i3} - \alpha_i \leq s_{i4} \quad \forall i \in \{1 \dots n\} \tag{2d}$$

$$c_{ij} = s_{ij} + p_{ij} \quad \forall i \in \{1 \dots n\} \tag{2e}$$

$$\alpha_i \leq p_{i3} \quad \forall i \in \{1 \dots n\} \tag{2f}$$

$$\sum_i (\delta_{i1t} + \delta_{i4t}) \leq P_t \quad \forall t \in \{1 \dots H\} \tag{2g}$$

$$\sum_i (\delta_{i2t} + \delta_{i5t}) \leq R_t \quad \forall t \in \{1 \dots H\} \tag{2h}$$

$$\sum_i \delta_{i3t} \leq B_t \quad \forall t \in \{1 \dots H\} \tag{2i}$$

where

$$\delta_{ijt} = \mathbf{1}\big\{ s_{ij} \leq t \leq s_{ij} + p_{ij} - 1 \big\} \; \forall j \in \{1, 4, 5\} \tag{3a}$$

$$\delta_{i2t} = \mathbf{1}\big\{ s_{i2} \leq t \leq s_{i2} + p_{i2} + \alpha_i - 1 \big\} \tag{3b}$$

$$\delta_{i3t} = \mathbf{1}\big\{ s_{i3} \leq t \leq s_{i3} + p_{i3} - \alpha_i - 1 \big\} \tag{3c}$$

The criterion J to optimize is the sum of non-decreasing functions of c_i, where c_i is the completion time for patient i, i.e. $c_i = c_{i4}$. Therefore, we can have $f(c_i) = c_i$, $f(c_i) = c_i^2$ or $f(c_i) = c_i^3$, for example. A criterion like $f(c_i) = c_i^3$ or $f(c_i) = c_i^4$ allows us achieve similar results as if we use the makespan as the assessment measure.

Constraints (2a-2d) are the precedence constraints. Constraint (2e) defines the completion time c_{ij} of the task j for the patient i. Constraints (2g-2i) are the capacity constraints for the porter teams, the operating rooms and the recovery beds, respectively. Constraints (3a-3c) indicate whether task j of patient i is active in discrete time period t.

We can deduce from (2b) and (2c) that $s_{i3} = s_{i5}$. Therefore variable s_{i5} and constraint (2c) can be removed from the model.

4 Lagrangian relaxation heuristic

4.1 Lagrangian relaxation of the problem

To apply the method, the capacity constraints (2g-2i) are relaxed using the Lagrangian multipliers, which are interpreted as utilization costs of resources. In this case the multipliers λ_t, β_t and γ_t act as prices for using transporters, operating rooms and recovery beds at time t, respectively. This way we obtain the following relaxed problem :

$$RP = L(\lambda_t, \beta_t, \gamma_t) \tag{4}$$

where :

$$
\begin{aligned}
L(\lambda_t, \beta_t, \gamma_t) = \min \Big\{ & \sum_i f(c_i) \\
& + \sum_t \lambda_t \Big(\sum_i (\delta_{i1t} + \delta_{i4t}) - P_t \Big) + \sum_t \beta_t \Big(\sum_i (\delta_{i2t} + \delta_{i5t}) - R_t \Big) + \sum_t \gamma_t \Big(\sum_i \delta_{i3t} - B_t \Big) \Big\}
\end{aligned}
\tag{5}
$$

subject to :

$$(2a - 3c)$$

Problem (5) can be further expressed as follows :

$$L(\lambda_t, \beta_t, \gamma_t) = -\sum_t \lambda_t P_t - \sum_t \beta_t R_t - \sum_t \gamma_t B_t + \sum_i SP_i(\lambda_t, \beta_t, \gamma_t) \tag{6}$$

where :

$$SP_i(\lambda_t, \beta_t, \gamma_t) = \min \left\{ f(c_i) + \sum_t \lambda_t \delta_{i1t} + \sum_t \lambda_t \delta_{i4t} + \sum_t \beta_t (\delta_{i2t} + \delta_{i5t}) + \sum_t \gamma_t \delta_{i3t} \right\} \tag{7}$$

which can be simplified as :

$$SP_i(\lambda_t, \beta_t, \gamma_t) = \min \left\{ f(c_i) + \sum_{t=s_{i1}}^{s_{i1}+p_{i1}-1} \lambda_t + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t + \sum_{t=s_{i2}}^{s_{i2}+p_{i2}+\alpha_i+p_{i5}-1} \beta_t + \sum_{t=s_{i2}+p_{i2}+\alpha_i}^{s_{i2}+p_{i2}+p_{i3}-1} \gamma_t \right\} \tag{8}$$

subject to :

$$(2a - 3c)$$

Hence we have to solve N sub-problems SP_i in order to solve the relaxed problem RP. This can be achieved using dynamic programming.

4.2   Solving the sub-problems

The N sub-problems (8) can be solved separately using dynamic programming. We will therefore obtain the optimal starting times s_{ij}^{RP} for each patient i. Let SPP_{4i}, SPP_{2i} and SPP_{1i} be the following sub-problems :

$$SPP_{4i}(T) = \min \left\{ f(c_i) + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t \right\} \tag{9}$$

subject to :

$$(2a - 3c) \tag{10a}$$
$$s_{i4} + p_{i4} - 1 \leq H \tag{10b}$$
$$s_{i4} = T \tag{10c}$$

where T is the decision variable, the starting time of the transport from the operating theatre to the ward, hence s_{i4}. Recall that H is the fixed time horizon. Constraint (10b) ensures that the transport operation does not exceed the time horizon H.

$$SPP_{2i}(T, A) = \min \left\{ f(c_i) + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t + \sum_{t=s_{i2}}^{s_{i2}+p_{i2}+A+p_{i5}-1} \beta_t + \sum_{t=s_{i2}+p_{i2}+A}^{s_{i2}+p_{i2}+p_{i3}-1} \gamma_t \right\} \tag{11}$$

subject to :

$$(2a - 3c) \tag{12a}$$
$$s_{i2} + p_{i2} + \alpha_i + p_{i3} + p_{i4} - 1 \leq H \tag{12b}$$
$$s_{i2} = T \tag{12c}$$
$$\alpha_i = A \tag{12d}$$

where T and A are the decision variables, respectively the starting time of the surgery operation (s_{i2}) and the recovery time spent in the operating room (α_i). Constraint (12b) ensures that s_{i2} does not cause a time horizon violation.

$$SPP_{1i}(T) = \min \left\{ f(c_i) + \sum_{t=s_{i4}}^{s_{i4}+p_{i4}-1} \lambda_t + \sum_{t=s_{i2}}^{s_{i2}+p_{i2}+\alpha_i+p_{i5}-1} \beta_t + \sum_{t=s_{i2}+p_{i2}+\alpha_i}^{s_{i2}+p_{i2}+p_{i3}-1} \gamma_t + \sum_{t=s_{i1}}^{s_{i1}+p_{i1}-1} \lambda_t \right\} \quad (13)$$

subject to :

$$(2a-3c) \tag{14a}$$
$$s_{i1} + p_{i1} + p_{i2} + \alpha_i + p_{i3} + p_{i4} - 1 \le H \tag{14b}$$
$$s_{i1} = T \tag{14c}$$
$$\tag{14d}$$

where T is the decision variable, the starting time of the transport from the ward to the operating theatre, hence s_{i1}. Constraint (14b) ensures that s_{i1} does not cause a time horizon violation. These sub-problems can be rewritten as follow :

$$SPP_{2i}(T, A) = \sum_{t=T}^{T+p_{i2}+A+p_{i5}-1} \beta_t + \sum_{t=T+p_{i2}+A}^{T+p_{i2}+p_{i3}-1} \gamma_t + \min_{s_{i2}+p_{i2}+p_{i3} \le t \le H} SPP_{4i}(t) \quad (15)$$
$$s.t. \ (2a-3c), \ s_{i2} + p_{i2} + \alpha_i + p_{i3} + p_{i4} - 1 \le H, \ s_{i2} = T, \ \alpha_i = A$$

and :

$$SPP_{1i}(T) = \min \sum_{t=T}^{T+p_{i1}-1} \lambda_t + \min_{s_{i1}+p_{i1} \le t \le H} SPP_{2i}(t) \quad (16)$$
$$s.t. \ (2a-3c), \ s_{i1} + p_{i1} + p_{i2} + \alpha_i + p_{i3} + p_{i4} - 1 \le H, \ s_{i1} = T$$

where solution of $SPP_{4i}(T)$ is obvious. The sub-problem can thus be solved using dynamic programming. Once the sub-problem is solved, the solution for the relaxed problem can be easily found. The solution of the relaxed-dual problem gives a lower bound of the initial problem J, namely L.

4.3   Building a feasible surgery schedule

The solution of the relaxed problem is in general not feasible. The schedule built with the starting times given by the solution of the dual problem might violate the capacity constraints (2g-2i) in some time periods. This subsection presents two algorithms to transform the infeasible solution into a feasible one. These algorithms take as input the infeasible solution given by the resolution of the dual problem and return a feasible solution.

4.3.1   Algorithm 1

Patients are sorted according to the starting time of their surgery operation and scheduled myopically one after another : the schedule is built without tanking into account the future, i.e. the remaining patients to be scheduled. For each patient, all tasks are scheduled one after another at the earliest date : the first transport is scheduled, then the surgery. At the end of the surgery process, the recovery begins in a recovery bed if available, otherwise in the operating room. If a recovery bed becomes available during this task, the patient is transferred and the operating room is released. The last two tasks (operating room cleaning and transport to the ward) are then scheduled. Patients are assigned as soon as possible when resources are available.

The main advantage of this algorithm is its simplicity and therefore its execution speed. Its complexity is $O(2N)$. It allows the generation of a surgery planning which is not too far from the original one. Critical tasks are the surgery and recovery operation (with a part of recovery eventually in the operating room) : keeping the order of these tasks allow us to generate a planning which have the same structure as the original infeasible solution. Transport tasks are short and can be scheduled fairly easily once the central « surgery-recovery » block is scheduled.

4.3.2 Algorithm 2

This algorithm takes as input the whole schedule given by the infeasible solution. The process is the following : each patient is picked from the planning and rescheduled as soon as possible, without violating the capacity constraints. The objective function value is evaluated after rescheduling each patient and its value is recorded. The patient is then replaced where he/she was, and we test the next patient. At the end of the process (i.e. when all patient has been rescheduled one time), the patient who gives the best objective function value is definitely rescheduled and the schedule is saved. The process restarts for the remaining patients who violate the capacity constraints.

The insertion of the surgery and recovery tasks may be an issue. First, the surgery task is inserted as soon as possible in the planning. Several cases appear when inserting the recovery task :

– A recovery bed is available right after the surgery until the end of the task : no problem here, the recovery task is inserted in the schedule.

– A recovery bed is not available right after the surgery, but becomes available some time after until the end of the task : we have to test here if the operating room is still available after the end of the surgery until the recovery bed becomes available. Otherwise, we have to remove the surgery and find another place.

– All other cases are infeasible : we have to remove the surgery and find another place.

If the recovery task could be inserted respecting the capacity constraints, the cleaning task must be also inserted : if there is no sufficient time when the operating room is released, we have to remove the surgery and recovery tasks and find another place. Finally, algorithm 2 can be summarized as follow :

1. **do**
2.   **for all** patient i **do**
3.   remove patient i from the planning
4.   insert patient i as soon as possible, respecting capacity constraints
5.   evaluate objective function value J_i
6.   replace patient i where he was
7.   **end for**
8.   reschedule patient i who maximize J_i definitely
9. **until** planning is feasible **and** solution cannot be improved anymore

This algorithm is more complex than the first one : its complexity is $O(N^2)$, and we will see in the results that computation times are higher. Nevertheless, this algorithm takes into account the entire infeasible schedule here. The way we build the feasible solution assure that it will be very close of the original one.

4.4   Solving the dual problem

The goal of the Lagrangian relaxation method is to obtain the best lower bound for the optimal solution. The subgradiant method is iteratively applied to solve the dual problem. For iteration $n + 1$, the multipliers are then updated as follows :

$$u^{(n+1)} = u^{(n)} + S^{(n)} \nabla L(u^{(n)}) \tag{17}$$

where $u = (\lambda; \beta; \gamma)$. The following converging step size $S^{(n)}$ is used :

$$S^{(n)} = \epsilon \frac{L^* - L(u^{(n)})}{||\nabla L(u^{(n)})||} \tag{18}$$

where L^* is the optimal dual, L^n is the value of the transformed relaxed problem at the nth iteration and $\nabla L(u^{(n)})$ is the subgradient of the relaxed problem L with respect to the multiplier. ϵ chosen between 0 and 2 and initialized at 0.5, is a variable factor used to adjust the convergence.

Finally the Lagrangian relaxation heuristic can be expressed as follow :

1. Initialize λ_t, β_t and γ_t with $\lambda_t^{(0)}$, $\beta_t^{(0)}$ and $\gamma_t^{(0)}$. $n \leftarrow 0$.
2. Solve $RP = L(\lambda_t, \beta_t, \gamma_t)$. We obtain the s_{ij}^{NF} and L.
3. Deduce a feasible solution. We obtain the s_{ij}^{F} and J.
4. Evaluate the gap :
$$gap = \frac{J - L^*}{L^*}$$
5. Update the multipliers. $n \leftarrow n + 1$.
6. Go to step (2) until $gap < GAP$ or $n > MAX\_IT$.

Values of GAP and MAX_IT are fixed. The duality gap gives a measure of optimality of the feasible schedule calculated in step (3). Notice that the optimal value L^* is needed to evaluate the duality gap. As we do not know its value, we replace L^* by the best L found until now.

Note that, during the Lagrangian relaxation-based heuristic, we also record the schedule that minimizes the makespan, i.e. the schedule that minimizes $\max(c_i)$ among all feasible schedules encountered in the Lagrangian relaxation-based heuristic.

5 Numerical results

Two types of numerical results are presented in this section : (i) problem instances were generated and tested to evaluate the efficiency of the method ; then (ii) a comparison study between two scheduling strategies (allowing recovery in the operating room or not) has been done. We were not able to get data set from a real operating theatre because processing times for surgery operations or recovery are rarely recorded. However, we generate data set as near as possible to the reality, according to our observations.

5.1 Efficiency of the Lagrangian relaxation heuristic

In order to test the efficiency the method, eight classes of problem instances were generated and tested (fifteen instances each). Each problem instance class has a different size and is described in table 1. The first column of the table indicates the problem instance number, the second one the number of surgeries cases to perform during the time horizon, the next three columns respectively show the number of porter teams, operating rooms and recovery beds available. The last four columns present the ranges of the processing times of the activities involved in the operation process, which were described at the beginning of Section 3. Data are generated using a uniform distribution. The data sets are very specific for our problem in order to make the recovery in operating room occurs. In reality the number of recovery beds is always greater than the number of operating rooms.

Moreover we observed that the number of porter teams is often very low compared to operating rooms number in small operating theatres of local hospitals. In particular, porter teams are sometimes replaced by nurses in order to save some time. In the state of the art we saw that porter teams are often considered as a bottle-neck resource. This is why we found relevant to test our method with a wide range for the number of porter teams : indeed when we do not have enough porter teams, allowing recovery in the operating rooms could be a cleaver solution to avoid operating theatre overtime or to minimize waiting time of scheduled patients.

TAB. 1 – Problem instances description

In order to test our method the time was discretized into time units of ten minutes. The time horizon H is fixed between 100 and 200 time units depending of the size of the problems. The maximum number of iterations (MAX_IT) is 3000, the minimum gap value (GAP) is 0.01% and the step size for updating multipliers is 1.9. The following four different criteria of the c_i function are tested and compared :

- $f_1(c_i) = \sum c_i$;
- $f_2(c_i) = \sum c_i^2$;
- $f_3(c_i) = \sum c_i^3$;
- $f_4(c_i) = \sum (c_i - LB)^+$.

LB is a lower bound for the makespan, which is calculated by making the sum of surgery times and cleaning times over all patients, and the result is divided by the total number of available operating rooms. As a result, we have :

$$LB = \frac{\sum_i (p_{i2} + p_{i5})}{\max_t(R_t)} \tag{19}$$

When used with the Lagrangian relaxation, f_4 is the criterion which gives best results for makespan : LB is a rough lower bound which only takes into account usage of the operating rooms where processing times are high. As described in the next sub-section, feasible solutions are farther from the optimal solution with f_3 or f_4 criteria, however makespans are quite good compared to solutions found with f_1 and f_2 criteria.

Results presented in tables 2 and 3 are average values over the 15 cases. All processing times for each data input file are generated randomly. Tables 2 and 3 present the duality gap expressed in percentage for the different assessment criteria tested as well as the computation time between parenthesis in seconds to achieve the solution for algorithm 1 and 2 respectively. The heuristic was coded using C++ and tested with a Pentium 4 at 3.2 GHz.

A first overview of gap evaluations for the two algorithms shows that results are quite good in general. For algorithm 1, the smallest and greatest gaps are 0.04% (class 8) and 16.5% (class 4), and for algorithm 2 the smallest is 0.01% (class 8) and the greatest 31.25%. For each algorithm, the best gaps are reached for criterion f_1 : even for a large size problem as class 5 with 30 patients, the gap is only 1.62%, which suggests that larger problems could be successfully solved with this method. Duality gaps are larger for other criteria. Indeed, the two last criteria (f_3 and f_4) show poor gap performances for large problems (classes 4 and 5). It is worth noticing that feasible solutions of the last two criteria have smaller makespan (not shown here). Finally, computation times remain very high for almost all instances.

TAB. 2 – Gap and time performance (algorithm 1)

TAB. 3 – Gap and time performance (algorithm 2)

With respect to the performances of the two algorithms, it can be observed that the second one gives better results for small size problems (classes 1, 2, 7 and 8), whereas the first one achieves smaller gaps for large size problems (classes 4 and 5). Although the second algorithm should be chosen for solving small size problems, the benefit in this case is not really significant. For large size problems the first algorithm works far better with gaps less than one third of the ones obtained with the second algorithm. Computation times are also higher for algorithm 2, which can be explained by its complexity, as said in Section 4.4.2. Hence, we recommend the use of first algorithm for real size problems.

5.2 Benefit of patient recovery in operating rooms

In a previous paper (anonymous ref.) we studied a similar problem where recovery is not allowed in the operating room : we had to build a feasible schedule with a no-wait constraint between the surgery

operation and the recovery/cleaning operations. In order to decide which method is better depending of the case study, we designed several experiments presented tables 4 and 5. The number of patients is fixed and equals to ten. Comparisons are made using algorithm 1. We define the improvement of the solution in percentage as follows :

$$\frac{f_i^{NW} - f_i}{f_i} \tag{20}$$

The first experiment (table 4) tests the effect of resources capacity variation. Processing times are fixed to the same values as class 1 in the last section. Classes 1 to 4 display results for an operating theatre with 4 operating rooms and 2 to 8 recovery beds. Classes 5 to 8 display results for 6 operating rooms and 4 to 10 recovery beds. At last, classes 9 to 11 display results for 8 operating rooms and 4 to 8 recovery beds. The numbers of transporters is fixed and will not effect the performances of the two strategies.

The second experiment (table 5) describes the influence of resources workload. There are 4 operating rooms and 6 recovery beds, which is the common ratio 3/2 for operating theatres. Two porter teams are also dedicated to the system. Instances of classes 12 to 16 display results for the case with average operating room workload of 1.38 and average recovery bed workload from 0.5 to 2.5. Classes 17 to 20 display results for operating rooms workload of 2.13 and recovery beds workloads from 1 to 2.5. Finally, classes 21 to 24 display results for operating rooms workload of 3.63 and recovery beds workloads from 2 to 3.5.

Tab. 4 – Experiments 1

Tab. 5 – Experiments 2

As expected, the benefit of patient recovery in operating rooms depends on the workload ratio of operating rooms/recovery beds. We observe in tables 4 and 5 that all criteria are improved when the recovery beds load becomes greater than the operating rooms load. In experiment 1 (see table 4), classes 1, 5 and 9 where the recovery beds are heavily loaded with respect to the operating rooms, the makespan is improved by at least 12%. In classes 2, 6 and 11 where the recovery beds and the operating rooms are equally loaded we manage to get makespan improvements from 0.1% to 2.16% for smaller operating theatres. In experiment 2 (see table 5), the makespan is improved by 17.94% for class 16 and 4.73% for class 20. We did not compare the methods with greater loads for recovery beds in order to keep consistency with reality.

If we compare results among the four criteria f_1, f_2, f_3 and f_4, best improvements are reached for criterion f_3 with 65.34% in class 1, 48.59% in class 9 and 42.59% in class 16. These results are obtained for heavily loaded recovery beds, even for a capacity-compliant operating theatre. To the contrary, criterion f_4 is only improved by 20% in classes 9 and 16. Allowing recovery in the operating rooms can even lead to slightly worse performance as we can see in class 12, 17 and 21 where recovery beds load is equal to operating rooms load. This is mainly due to the heuristic nature of the Lagrangian relaxation heuristics. Note however that any feasible schedule not allowing patient recovery in operating rooms is a feasible solution in our case.

The experiments presented in this section allow us to know what is the best strategy depending of the operating theatre configuration and/or the type of surgeries. Results presented in table 4 show that allowing the recovery in the operating room improve performances as soon as the ratio recovery

beds/operating rooms is lower than 3/2. This kind of situation occurs when the operating theatre is crowded : a large number of emergency patients have to be operated in regular operating rooms could be a reason to allow recovery in operating rooms if the recovery room is full. On the other hand, allowing recovery in the operating room could be a good strategy when recovery processing times grow high, i.e. when workload of recovery beds becomes greater than that of operating rooms. This kind of situation occurs when operated patient begin to bleed after a surgery and must be re-operated immediately, increasing the duration of recovery. Notice that transporters represent also a bottleneck resource : patients waiting for transporters in recovery room consume resources uselessly and prevent patients in operating rooms to be transferred into recovery room.

6   Conclusions and perspectives

In this paper, we presented a new operating theatre scheduling problem where patient recovery is allowed in operating rooms. Recovery in operating rooms is allowed when the recovery room is crowded, which occurs in specific cases (large number of emergency surgeries, recovery time longer than expected, etc.). The problem has been modelled as a 4-stage hybrid flowshop problem with blocking constraint, and successfully solved thanks to the Lagrangian relaxation method. By evaluating the duality gap, we were able to measure the quality of the results obtained. Several data sets were generated in order to test the efficiency of the method on real size problems.

We also compared two scheduling strategies : allowing the recovery in the operating room when to recovery bed is available or not. Using a design of experiments, we were able to determine in which case one strategy should be applied to the operating theatre : a load ratio limit has been found when resources capacity/processing times vary.

Surgeons were enthousiastic about the creation of an automatic surgery planning generation which consider recovery in operating room, as medical teams have to take into account this possibility. Unfortunately, we were not able to apply our method in a real opearting theatre yet.

Further experiments are needed to test the robustness of the model against risks. The randomness of surgery or recovery times is likely to lengthen during the process itself. Allowing recovery in the operating room will certainly lead to better results and further optimize resources utilisation, offering a more flexible schedule. Another future research direction is the introduction of resource costs into our model in order to compute accurate criterion regarding health care staff demands : opening an operating room during an hour is more expensive than using a recovery bed during the same time. Introducing over-time costs is also part of our perspectives. Finally, considering emergent surgeries in the operating theatre would constitute an interesting study.

Références

[1] F. Dexter. Cost implications of various operating room scheduling strategies. *American Society of Anesthesiology - Refresher Courses in Anesthesiology*, 30(1) :87–95, 2002.

[2] H. Fei, D. Duvivier, N. Meskens, and C. Chu. Ordonnancement journalier dans un bloc opératoire dans le cadre d'une stratégie open scheduling. In *Actes de la conférence GISEH*, pages 615–622, 2006.

[3] H. Fei, N. Meskens, and C. Chu. An operating theatre planning and scheduling problem in the case of a "block scheduling" strategy. In *Proceedings of the IEEE Conference on Service Systems and Service Management*, pages 422–428, 2006.

[4] T. Gordon, S. Paul, A. Lyses, and J. Fountain. Surgical unit time utilization review : resource utilization and management implications. *J. of Med. Sys.*, 12(3) :169–179, 1988.

[5] A. Guinet and S. Chaabane. Operating theatre planning. *International Journal of Production Economics*, 85 :69–81, 2003.

[6] A. Jebali. *Vers un outil d'aide à la planification et à l'ordonnancement des ressources dans les services de soins*. PhD thesis, Laboratoire d'Automatique de Grenoble (LAG) - Ecole Doctorale Organisation Industrielle et Systèmes de Production, 2004.

[7] A. Jebali, A. B. Hadj Alouane, and P. Ladet. Operating rooms scheduling. *Int. J. Production Economics*, 99 :52–62, 2006.

[8] S. Kharraja, R. Abbou, F. Albert, and E. Marcon. Open scheduling : A block scheduling strategy-based operating theatre planning. In *Proceedings of the IEEE Conference on Industrial Engineering and Systems Management*, 2005.

[9] T. Kis and E. Pesch. A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. *European Journal of Operational Research*, 164 :592–608, 2005.

[10] M. Lamiri, X. Xie, A. Dolgui, and F. Grimaud. A stochastic model for operating room planning with elective and emergency demand for surgery. *European J of Operational Research*, accepted, in press, 2006.

[11] M. Lamiri, X. Xie, and S. Zhang. Column generation for operating theatre planning with elective and emergency patients. *IIE Transactions*, accepted, in press, 2007.

[12] R. Linn and W. Zhang. Hybrid flow shop scheduling : A survey. *Computers and Industrial Engineering*, 37 :57–61, 1999.

[13] E. Marcon and S. Kharraja. Survol des différentes stratégies de programmation opératoire : Analyse de leurs performances et des outils informatiques de leur mise en œuvre. *Journal Europén des Systèmes Automatisés*, 2002.

[14] E. Marcon, S. Kharraja, N. Smolski, B. Luquet, and J.-P. Viale. Determining the number of beds in the postanesthesia care unit : A computer simulation flow approach. *Anesth Analg*, 96 :1415–1423, 2003.

[15] M. Pinedo. *Scheduling theory, algorithms and systems*. Prentice Hall, 1995.

[16] I. H. Wright, C. Kooperberg, B. A. Bonar, and G. Bashein. Statistical modeling to predict elective surgery time. *Anesthesiology*, 85 :1235–1245, 1996.

Figure 1
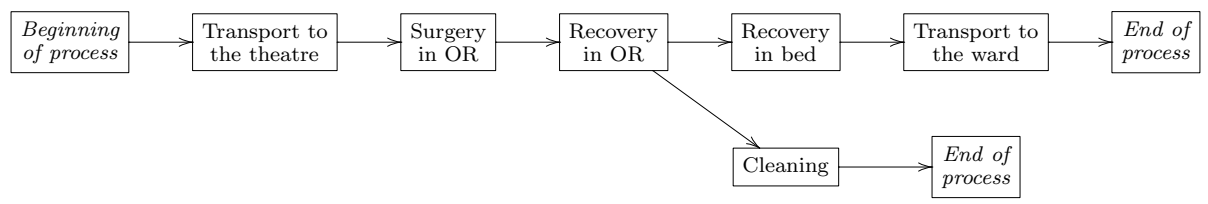
Fig. 1 – Surgery Process

Table 1

TAB. 1 – Problem instances description

Class	n	P	R	B	$p_{i,1}$ & $p_{i,4}$	$p_{i,2}$	$p_{i,3}$	$p_{i,5}$
1	10	2	4	4	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
2	10	2	4	2	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
3	15	2	4	2	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
4	20	2	4	3	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
5	30	2	6	4	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
6	10	1	2	1	$\{1,3\}$	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$
7	10	2	4	2	$\{1,3\}$	$\{18,24\}$	$\{18,24\}$	$\{2,4\}$
8	10	2	4	2	2	$\{4,22\}$	$\{6,24\}$	$\{2,3\}$

Table 2

Tab. 1 – Gap and time performance (algorithm 1)

Class	Criterion			
	f_1	f_2	f_3	f_4
1	0.06 (37.2)	0.26 (40.8)	0.73 (40.8)	1.37 (40.5)
2	0.37 (50.4)	1.43 (51.3)	1.79 (52)	0.84 (49.8)
3	0.31 (169.1)	1.16 (174.2)	2.81 (174.01)	2.31 (160.6)
4	0.96 (387.2)	3.40 (387.8)	6.56 (388.2)	16.5 (398.4)
5	1.62 (499.5)	5.25 (576.6)	9.60 (577.8)	11.25 (533.3)
6	0.66 (125.5)	1.32 (153)	2.10 (136)	10.64 (154.1)
7	0.08 (159.1)	0.18 (149.7)	0.32 (156.1)	0.20 (164.8)
8	0.04 (53.3)	0.36 (55.2)	0.47 (53)	0.57 (54)

Table 3

TAB. 1 – Gap and time performance (algorithm 2)

Class	Criterion			
	f_1	f_2	f_3	f_4
1	0.05 (36.4)	0.12 (40.4)	0.75 (41.2)	1.27 (39.3)
2	0.06 (45.1)	0.22 (54.8)	0.31 (42)	1.01 (49.2)
3	0.77 (169.1)	3.30 (142.2)	12.44 (172.4)	5.40 (169.2)
4	2.87 (376.2)	9.09 (393.5)	17.03 (393.1)	22.5 (396.4)
5	7.16 (545.2)	16.96 (545.4)	24.49 (549.2)	31.25 (561)
6	0.85 (136)	3.69 (142.2)	7.21 (129.5)	10.64 (154)
7	0.02 (93.5)	0.09 (109.8)	0.22 (147)	0.22 (159.6)
8	0.01 (29.5)	0.18 (37.1)	1.76 (49.9)	0.38 (57.4)

Table 4

<div align="center">

Tab. 1 – Experiments 1

Class	N	M	Resource load		Improvement of solution (%)				
			OR	Beds	f_1	f_2	f_3	f_4	$Cmax$
1	4	2	3.88	7.5	10.28	30.45	65.34	44.61	34
2	4	4	3.88	3.75	0.03	0.19	0.64	1.74	2.16
3	4	6	3.88	2.5	0	0	0.1	0	0.4
4	4	8	3.88	1.88	0	0	0	-0.1	0
5	6	4	2.58	3.75	2.01	6.63	15.06	5.66	12.46
6	6	6	2.58	2.5	0.25	1.08	2.36	0.58	1.8
7	6	8	2.58	1.88	0	0	0.04	-0.06	0.43
8	6	10	2.58	1.5	0	0	0	0	0
9	8	4	1.94	3.75	9.08	24.25	48.59	19.29	16.4
10	8	6	1.94	2.5	0.91	2.97	6.67	1.82	2.6
11	8	8	1.94	1.88	0	0.19	0.87	0	0.1

</div>

Table 5

Tab. 1 – Experiments 2

Class	$p_{i,2}$	$p_{i,3}$	Resource load		Improvement of solution (%)				
			OR	Beds	f_1	f_2	f_3	f_4	$Cmax$
12	$\{2,4\}$	$\{2,4\}$	1.38	0.5	0.04	0.04	-0.32	-0.07	0.33
13	$\{2,4\}$	$\{4,8\}$	1.38	1	0	0.06	0.02	-0.14	0.53
14	$\{2,4\}$	$\{6,12\}$	1.38	1.5	0.53	2.11	4.84	2.61	2.16
15	$\{2,4\}$	$\{10,14\}$	1.38	2	4.47	11.34	19.82	11.14	6.85
16	$\{2,4\}$	$\{12,18\}$	1.38	2.5	9.18	23.09	42.59	21.13	17.94
17	$\{4,8\}$	$\{4,8\}$	2.13	1	0	0	0.02	-0.37	0
18	$\{4,8\}$	$\{6,12\}$	2.13	1.5	0.29	0.65	0.62	-0.86	0.42
19	$\{4,8\}$	$\{10,14\}$	2.13	2	0.05	0.09	0.43	0.24	1.17
20	$\{4,8\}$	$\{12,18\}$	2.13	2.5	1.88	4.91	9.26	5.61	4.73
21	$\{10,14\}$	$\{10,14\}$	3.63	2	0	0.01	0	0	-0.12
22	$\{10,14\}$	$\{12,18\}$	3.63	2.5	0	0.05	0.05	0.08	0
23	$\{10,14\}$	$\{16,20\}$	3.63	3	0.07	0.21	0.44	0.7	0.11
24	$\{10,14\}$	$\{20,22\}$	3.63	3.5	0.74	1.48	2.12	3.1	0

December 1st, 2007

Dear Editor of the Computers & Industrial Engineering journal,

Enclosed is a paper, entitled "**Operating theatre scheduling with patient recovery in both operating rooms and recovery beds**". Please accept it as a candidate for publication in the Computers & Industrial Engineering journal.

This paper addresses the surgery operation scheduling problem under the open scheduling policy. This NP hard problem is solved using a Lagrangian relaxation approach. A comprehensive numerical experiment is then performed to quantify the benefit of allowing patient recovery in operating rooms.

Health care system environments are evolving. New policies of health care budget control have been introduced in order to improve the whole system. The operating theatre is the most critical ressource of the hospital, therefore surgery operations scheduling and human ressources planning is operating theatre are crucial issues both for patient's health and resources optimization. This study deals with a concrete problem of surgery scheduling in the operating theatre in order to reduce costs. The Lagragian relaxation method is used . We strongly believe the contribution of this study warrants its publication in the Computers & Industrial Engineering journal.

Finally, this paper is our original unpublished work and it has not been submitted to any other journal for reviews. Some results have been improved since the first paper for the IESM'07 international conference, especially about the benefit of patient recovery in operating room.

Sincerely,

Vincent AUGUSTO, Xiaolan XIE and Viviana PERDOMO

Health Care Systems Operation Department

Engineering and Health Division

Ecole Nationale Supérieure des Mines de Saint-Etienne

158 cours Fauriel

42023 Saint-Etienne cedex 2, France

E-mails: augusto@emse.fr ; xie@emse.fr