Operating Theatre Scheduling Under Constraints

L. Dekhici and K. Belkadi
LAMOSI Laboratory, Department of Computer Science, Faculty of Sciences, University of Sciences and Technology, Oran USTO, BP. 1505 Oran M’Naouer 31000, Algeria

Abstract: We present in this study some manufacturing systems scheduling constraints that we adapt to the operating theatre scheduling. The scheduling that should comply with all operating constraints, related to quality, to patients and numerous resources information can be considered as a two-stage hybrid flow shop problem without buffer. We confirmed the existence of the constraints after a large study with operating theatre managers at several hospitals. For resolution, we use two methods: Local search for constraint satisfaction in the initial feasible solution and Tabu search with restricted neighborhood system for Makespan Optimization. In one hand, we use yearly data from an existing multidisciplinary which has 3 operating rooms and 9 beds and releases 10 surgeries per day. In the other hand, we schedule others examples with different variables. The amelioration rates of both makespan and constraints conflict were interesting according to program execution time.

Key words: Hybrid flowshop, constraints satisfaction, precedence constraint, local search, Tabu search, surgeries

INTRODUCTION

The operating theatre involves two sectors: Operating Rooms (ORs) and Post-Anesthesia Care Unit (PACU) or revival beds. The operating process procedure is divided into three phases:

• The pre-operative step during which the patient undergoes surgical and anesthetics consultations. It extends from the admission of the patient until the eve of the intervention
• The per-operative step which ranges from the patient mental preparation before surgery intervention, until he wakes up and leaves the PACU. This phase takes place the day of the intervention. In cases where the patient’s condition is considered critical, it will rather lead to Intensive Care Unit (ICU) (Hammami et al., 2007)
• The post-operative phase after his revival

The operating theatre program refers to two sequential sub-problems: Advanced scheduling which plans the surgeries a week before and the allocation scheduling which determines the order of surgical interventions passages in one day and their assignments in the operating theatre.

In this study, we limited our study to the per-operative (during surgery) steps of the operating procedure.

OPERATING THEATRE SCHEDULING PROBLEM

Problematic and contribution: Many solutions for the surgeries scheduling problem can be found in the literature. Some approaches are based on the simulation (Dexter et al., 1999). Jebali et al. (2006) introduced a two-step approach for operating room scheduling, where operations are assigned to ORs in the first step and then these operations are sequenced in the second step. The reader may also refer by Guinet and Chaubane (2003), Hsu et al. (2003), Van Costrum et al. (2008) and Cardoen et al. (2009b). A summary of recent approaches can be found by Cardoen et al. (2010).

The criteria of scheduling surgical interventions depend on the strategy adopted at the Advanced Scheduling. We can find: Over-use and under-use of the operating rooms (Fei et al., 2010), Makespan (Khannaja and Marcon, 2003) operating rooms efficiency (Dexter et al., 2002) patients waiting and total interventions cost (Cardoen et al., 2009b).

Scheduling seeks to balance needs expressed by patients, surgeons with clearly defined policies and supplies. The presence of a large number of constraints is a difficulty in constructing this schedule. In the field of manufacturing systems, many works were interested by developing of assignment and scheduling algorithms under constraints (Espinoose et al., 1999; Baptiste et al., 2001; Alem-Tabriz et al., 2009). In the area

Corresponding Author: L. Dekhici, LAMOSI Laboratory, Department of Computer Science, Faculty of Sciences, University of Sciences and Technology, Oran USTO, BP. 1505 Oran M’Naouer 31000, Algeria
of operating programming, constraints studied had economic nature so they were limited to time's constraints and such as: Opening hours of the operating rooms (Kharraga and Marcon, 2003), time available for each time slot, average waiting time (Dexter et al., 2002) deterioration of the no-idle constraint. To our knowledge, only the study of Sier et al. (1997) and Jebali et al. (2006) considered patients satisfiability or surgeries quality by taking into account number of constraints such as priority by age, assignment laws and respect of rooms’ specialization and recently Cardoen et al. (2009a) solved a case sequencing problem in which they propose to minimize the date of surgeries of children and prioritized patients. In this last study, authors ignore assignments to beds. They took into account a good sized PACU with identical beds and without intensive care beds. Present study is to adapt some manufacturing systems constraints of sequencing and assignment to operating theatre scheduling problem. Thus, we undertake to propose constraints of quality and of service feasibility not present in the literature of surgeries scheduling as they are present in the literature of manufacturing systems scheduling. We got these constraints after a view of the set of constraints known in manufacturing systems literature and a deep study with experts at different operating theaters especially in the academic and hospitalable establishment of Oran (EHUO: Etablissement Hospitalier et Universitaire d’Oran).

Analogy with the manufacturing system scheduling problem: On the one hand, the difference between manufacturing system and the operating theatre lies in their products. Compared to the manufacturing system where the processes are deterministic, in the Operating theatre system services are offered to patients whom health vary and preferences vary. In addition, there are more frequent emergencies in the hospital than in the manufacturing systems. On the other hand, unlike the Manufacturing System where a machine is normally occupied by a technician, surgical teams are composed of members from different specialties (surgeons, nurses, anesthetists etc.) So, it needs to coordinate activities. In Table 1 we consider the surgical as tasks, the operating units as service centers and we can deduce that the operating theatre is composed of two stages where the first are the parallel operating rooms and the second are beds in PACU. This system is literarily classified as a hybrid flow shop.

HYBRID FLOW SHOP PROBLEM

Presentation of hybrid flow shop: A Hybrid Flow Shop (HFS) also called flexible flow shop, is a system composed of a set of stages, where each stage is composed of one or more parallel machines. The different jobs visit the stages in the same order. On each stage, a job is treated by one machine only. Between each stage, the jobs can wait or not in limited or unlimited buffers (Vignier et al., 1999).

Scheduling in the HFS consists to find a adequate sequence of the jobs in entry and an assignment of the jobs on the various machines at the various stages. The objective is Optimization of a criterion of performance among the criteria one can quote the makespan or $C_{\text{max}}$, $F_{\text{max}}$, etc.

Number of tasks scheduled is $N$. The assignment of a task $i$ to the $j$th machine in the stage $k$ is noted $v_{i,k,j}$. $I_k$ is the number of machines in stage $k$.

Figure 1 is an example of a hybrid flow shop with 2 stages and 3 machines on the first stage and 2 machines on the second one. A buffer of infinite capacity is incorporated between stages of the system. Moreover, all jobs are assuming to be available at the system entrance with release date with value 0 (Bellkadi et al., 2006; Sahraoui et al., 2003).

If we take as criterion the $C_{\text{max}}$ (the completion time of the last job on the last stage) and using the notation of (Vignier et al., 1999), the system can be defined by $\text{FH}2(M(3),M(2))|C_{\text{max}}$.

Survey of the two-stage HFS scheduling: The flow shop scheduling problem has been shown to be NP-Hard.

![Fig. 1: HFS,$M(3)$,$M(2)$|$C_{\text{max}}$](image-url)
General HFS studies are found by Baker (1974) and Carlier (1993). Shen and Chen (1972) were the first to tackle a two-stage HFS with \( l_1 \geq 2, l_2 \geq 2 \). They proposed a permutation heuristic to solve the Makespan Minimization Problem (MMP) when preemption is not allowed. Buten and Shen (1973) have studied the same problem but with jobs having precedence constraints, which they solved with a modified version of Johnson (1954) algorithm. Gupta (1988) has studied the MMP in a two-stage HFS with \( l_1 \geq 2 \) and \( l_2 \geq 1 \). He states that the MMP in a two-stage HFS is NP-hard when \( \max(l_1, l_2) \geq 1 \). This result is very important because it shows that any MMP in a K-stage HFS is NP-hard, since the K-stage HFS can always be reduced to a two-stage HFS. He proposed a heuristic and lower bounds for the MMP case.

Sriskantharajah and Sethi (1989) focused on the two stages HFS with single machine on the first stage and parallel identical machines on the second stage to minimize the makespan.

**CONSTRAINTS ADAPTED TO OPERATING THEATRE**

A notation adapted to the operating theatre scheduling problem are presented. We submit from the 3rd sub-title, this problem to many constraints.

**Problem notation:** We consider the problem as a two-stage hybrid flow shop with without buffer or with Blocking, where the first stage represents no-identical ORs and the second, beds in the PACU (Fig. 2).

The Input Parameters are:

\[
\begin{align*}
K & = \text{2 number of stage} \\
N & = \text{No. of patients} \\
l_1 & = \text{No. of ORs in stage 1} \\
l_2 & = \text{No. of beds in stage 2}
\end{align*}
\]

The Indices used are:

\[
\begin{align*}
k & = \text{Stage where, } k = 1 \ldots 2 \\
j & = \text{Processor in stage, where, } j = 1 \ldots l_k \\
i, i' & = \text{Patient where, } i, i' = 1 \ldots N \\
t_{ik} & = \text{Processing times of a patient } i \text{ in the stage } k
\end{align*}
\]

Decision variables are:

\[
\begin{align*}
d_{ik} & = \text{Beginning time of the patient } i \text{ at stage } k \\
f_{ik} & = \text{Departure time of patient } i \text{ from stage } k \\
C_{\text{max}} & = \text{Makespan. It is calculated by the Eq. 1:}
\end{align*}
\]

\[
C_{\text{max}} = \max_{i=1 \ldots N} f_{i1} - \min_{i=1 \ldots N} d_{i1}
\]

We obtain HFS2 (M(11),M(12)) block | C\text{max} according to Vignier notation(Vignier et al., 1999).

For the solution notation, we consider:

\[
\begin{align*}
p_i & = \text{1..N is the patient of the order } i \\
o_{ij} & = \text{1..l_1 is the OR assigned to the patient } p_i \\
b_{ij} & = \text{1..l_2 is the bed assigned to the patient } p_i
\end{align*}
\]

A solution is a combination of the three vectors (p,o,b).

**Example:** A solution for 4 patients in a HFS2(M3,M5) |(1,2,3,4)| block | C\text{max} can be notated as below:

\[
(4,3,2,1)(1,1,3,2)(1,2,3,4)
\]

Where:

\[
\begin{align*}
p_1 & = 4: \text{The patient number 4 passes in 1st order} \\
o_4 & = 1: \text{The OR assigned to patient 4 is the 1st one} \\
b_4 & = 1: \text{The bed assigned to patient 4 is the 1st one}
\end{align*}
\]

**Time and block constraints:** Hwan and Finoedo (1986), Espinouse et al. (1999), Abadi et al. (2000) and Kalozy and Kamburovski, (2005) blocking flow shop problem as follows: The flow shop has no intermediate buffer therefore a job cannot leave a machine until the next machine downstream is free. If that is not the case, the job (and the machine as well) is said to be blocked. Detailed survey of the application and research on the problem is given by Hall and Sriskantharajah (1996).

We can deduce Eq. 2 about the date of leaving the OR and Eq. 3 about the relation between two patients assigned to the same bed in PACU.

\[
\forall i = 1 \ldots N (d_{ik} = f_{ik})
\]

\[
\forall i,i' = 1 \ldots N (b_{ip} = b_{ip'} \forall i < i' \rightarrow f_{ip} > f_{ip'})
\]
In addition to the particularity of blocking constraint, the time of execution on the second stage decreases if the patient began to wake up in the OR. We calculate the real time which must be spent in the PACU, noted $t_{r,i}$ as Eq. 4.

$$\forall i \in 1..N(t_{r,i} = t_{p,i} + t_{r,i-1})$$

(4)

**Order constraints:** Here, we present the constraints that affect on the sequence of the patients. We quote:

- Connective precedence
- Unitary precedence
- Sequencing relation
- Temporal localization

**Connective precedence:** This constraint was discussed by Botta-Genoulaz (2000). We can say that we have a connective precedence where, a patient $i$ must be scheduled above all others by priority (5). Let note $p^{-1}$ the order or the inverse function of $p$.

We note: $\text{pri}_i$, as an integer and a default value is 0.

$$\forall i,j' = 1..N(\text{pri}_i < \text{pri}_{j'} \rightarrow p_i^{-1} < p_{j'}^{-1})$$

(5)

Some data as the age of patients, operating and revival times or having already a canceled surgery, allow deciding priorities for the sequence of surgeries.

**Unitary precedence:** Many definitions were given to this constraint (Buten and Shen, 1973; Rimmoy, 1976; Kuriš, 1976; Espinouse et al., 1999) as a relation between only two tasks when one must be schedule above another in the same stage or when they use the same machine. We define the unitary precedence where a patient $p$, must be scheduled above another $p_i$. Eq. 6 and 7.

We note: $\text{pre}_{p_i}^p$, as a Boolean and a default value is 0:

$$\forall i,j' = 1..N(\text{pre}_{p_i}^p = 1 \rightarrow i < j')$$

(6)

$$\forall i,j' = 1..N(\text{pre}_{p_i}^p \lor \text{pre}_{j'}^p \leq 1)$$

(7)

If $p_i$ must be scheduled directly above $p$, we talk about strict precedence Eq. 8 and 9.

We note: $\text{pre}_{p_i}^p$, as a Boolean and a default value is 0.

$$\forall i,j' = 1..N(\text{pre}_{p_i}^p = 1 \rightarrow i = j')$$

(8)

$$\forall i,j' = 1..N(\text{pre}_{p_i}^p \lor \text{pre}_{j'}^p \leq 1)$$

(9)

An example of the use of this last restriction is the case of transplantation.

Those two constraints can be transformed in priorities.

**Sequencing relation:** The Sequencing Relation was defined by Esquirol et al. (2001) for projects scheduling. In our case it prohibits for a given pair of patients ($p_i$, $p_j$) any scheduling where, $p_i$ is directly above $p_j$, or $p_j$ is directly above $p_i$. Eq. 10 and 11.

We note: $\text{noseq}_{i,j}$, as a Boolean and a default value is 0.

$$\forall i,j' = 1..N(\text{noseq}_{i,j'} = 1 \rightarrow (i < j' + 1) \lor (j < i' - 1))$$

(10)

$$\forall i,j' = 1..N(\text{noseq}_{i,j'} = \text{noseq}_{j,i'})$$

(11)

We consider this constraint when the surgeries of $p_i$ and $p_j$ require a special effort from the surgical team or a special supply from the pharmacy.

**Temporal localization:** A temporal localization gives earliest Eq. 13 and latest Eq. 12 dates (Giard, 1991) for the start of a surgery.

We note: $\text{dmax}$, or $\text{dmin}$, as float and default values are -1;

$$\forall i = 1..N(\text{dmax} \rightarrow -1 \rightarrow d_{i,j} \leq \text{dmax})$$

(12)

$$\forall i = 1..N(\text{dmin} \rightarrow -1 \rightarrow d_{i,j} \leq \text{dmin})$$

(13)

This constraint can be defined where the fasting duration is paramount.

**Assignment constraints:** Here, we present the constraints that affect on the assignment of the patients to ORs and beds. We quote:

- Dedication constraint
- Technology dependence

**Dedication constraint:** The constraint of dedicated machine for a type of task was used for the HSF by Riane et al. (2001), Hung and Ching, (2003), Longmin et al. (2007) and Dekhioc and Belkadi (2008) for the clothing workshop where it dedicates a special sewing machine to an appointed task.

In present cases it assigns a patient to a special operating room (Eq. 14) or a special bed (Eq. 15) when the state is considered critical.

We note: $\text{ded}_{i,k}$ as integer and a default value -1.

$$\forall i = 1..N, j = 1..11(d_{i,j} = i + j)$$

(14)
\[ v_i = 1 \land N, j = 1.12(\text{ded}_d, j \rightarrow b_j = j) \quad (15) \]

For example, if a patient \( p_i \) which is a biologic danger must be assigned to bed \( y \) in stage 2 we note: \( \text{ded}_p, 2 \rightarrow y \) and if a patient \( p_i \) must be assigned to an efficient operating room \( z \) in stage 1, we note: \( \text{ded}_p, 1 = z \).

**Technology dependence:** The technology dependence constraint was defined by Vignier et al. (1999) for the HFS with \( k \) stages and a same number of machine 1 in each stage \( HFS, M(1), M(1), \ldots, M(1) | C_{\text{max}} \) as follow.

For all job \( i \), if \( j \) is the machine assigned to the job \( i \) in the stage \( k \) (\( v_{ik} \) according to Vignier Notation), then the machine assigned to \( i \) in the stage \( k+1 \) must be the \( j \)th machine in this stage.

\[ v_i = 1 \land N, \forall k = 1.1.K(v_{ij}) = 1.1, v_{ik} = j \rightarrow v_{ij+1} = j \quad (16) \]

This definition was modified in (Dekhici and Belkadi, 2008). We define the technology dependence for some OR only. If a patient goes to the operating room \( j \), he must go to the bed \( j' \) (Eq. 17).

We note: \( \text{tec}_b, j' \) as integer and a default value is -1.

\[ J = 1.1, j' = 1.1, j' \rightarrow \text{tec}_j = j' \rightarrow v_i = 1.1.N(v_i = j, b_j = j) \quad (17) \]

As an example let propose that every patient who goes to the cardiac OR must go to an intensive care bed.

**New notation:** After submitting the habitual problem to constraints, we get a new problem: \( HFS2, M(1), M(1), | t, d, l, d, \text{max}, l, \text{min}, b, \text{prio}, t, \text{tec}, \text{ded}, \text{poseq} \) \( C_{\text{max}} \).

**Possible conflict:** In manufacturing systems, assignment constraints may contradict (Dekhici and Belkadi, 2008). An expert confirms that in Hospital Systems, they are complementary and that only order constraints can disrupt planning.

**Example:** \( \text{ded}_p, 1 = x \) and \( \text{ded}_p, 2 = y \) and \( \text{tec}_j = z \). So, both of the assignment of the room \( x \) and the bed \( y \) and the assignment of the room \( x \) and the bed \( z \) are refused.

**RESOLUTION**

We have presented earlier the notation of operating theatre scheduling under constraint. And we can identify that there are two problems: How to give a which satisfies constraint and how to minimize the \( C_{\text{max}} \) of operating procedure in one day. We propose two essential algorithms (Fig. 3).

![Fig. 3: Notation and Methods used in the problem resolution](image)

The first one solves a Maximal Constraint Satisfaction problem (MCS) in order to give a feasible schedule and the second one solves a scheduling problem to give an optimal schedule.

**Constraint satisfaction in the initial schedule:** For the dependence technology and the dedication constraints, a simple check is sufficient given priority to dedication one, if ever a bad manipulation has led a conflict.

To find an initial feasible order, we implement an algorithm for Maximal Constraints Satisfaction (MCS) with a local search or min-conflicts. One of the advantage of the min-conflicts is that its runtime is roughly independent of the problem size.

The MCS problems consist either of the optimization of the constraints violated number or the optimization of the inconsistency cost. After a discussion with experts at EHUO, we admit that constraints have not the same importance in the planning. So, we choose the second criterion.

To evaluate the inconsistency cost, we first sort order constraints per importance and we give a coefficient for each one according to our study:

- Temporal localization, coeff. = 4
- Unitary precedence, coeff. = 3
- No sequencing constraint, coeff. = 2
- Priorities, coeff. = 1

Our MCS can be formulated as a triple \( (X, D, C) \), where \( X \) is a set of variables which are the patients \( p_i \), \( D \) is a domain of values which is the schedule and \( C \) is a set of constraints. Every constraint is in turn a pair \( (T, R) \) where, \( T \) is a tuple of variables and \( R \) is a set of relations.
\( X = \{p_i| i=1..N\} \) scheduling of patients.

\( D_i = [1..N], i=1..N \)

\( C = (T,R)\) the set of the order constraints. It contains the relations of priorities \((X, Prior(p_1,p_2,p_3,...))\), of unitary precedence \((i(i+1),prec(i,i+1))\), of no sequencing \((i(i+2),naoseq(i,i+2))\), of temporal localization \((i, t_{\text{max}}(i))\) and no equality of the variables (Eq 18):

\[
\forall i \neq j(p_i = p_j)
\]

With this last constraint, we can deduce that we cannot talk about a reselection of a new value for a variable from \( X \) without talking about a permutation between two values.

We admit that priorities can be erased by other order constraints as the no sequencing constraint. All the constraints are soft. Present algorithm (algorithm 1) uses local search to check for conflicts and to select assignments that minimize conflict with other variables.

**Algorithm 1:**

```plaintext
function MIN-COST(csp, max_steps): D //returns a solution
inputs: csp=(X,D,C);
max_steps// the number of steps allowed before giving up
Pcurrent=next(RPrior); // an initial complete assignment for csp which is the sort per priority.
for step = 1 to max_steps do
begin
if Cost(Pcurrent,csp) >= 0 then return Pcurrent; //if Pcurrent satisfies all constraints.
Pnext=Pcurrent;
enddo;
I= random(N); J= random(N);
Pf=X[I]; Pj=X[J]; // choose random variables from X
while (I=E or Pj and Pf are conflicted variables // One at least must appear in csp C
T=Pnext[Pj].value;
Pnext[Pj]=Pnext[Pj].value;
Pnext[Pf]=value; //reselection of values by permutation
if Cost(Pnext,csp)/Cost(Pcurrent,csp) then
Pcurrent=Pnext;
step++;
enddo;
return Pcurrent;
enddo;
```

Where **Cost** \( (P,csp) \) is a function that calculates the number of constraints violated multiplied by their coefficients in an assignment \( P \).

The generation of a feasible schedule is in algorithm 2 and follows the steps below:

- Generate a feasible sequence of the patients with min conflict
- Generate a feasible assignment to either ORs or Beds with totally respect of dedication and technology dependence constraints

**Algorithm 2:**

```plaintext
Initial_schedule(i,j,N,p,ob)// order of patients
p=MIN-COST(csp, max_steps);//give a feasible sequence with min conflict.
Assignment of patients
for i=1..N do
begin
/assignment to ORs
if ded(i) <= 1 then ob; //respect dedication constraint
else
j=random(i); // randomly chosen
ob=j;
enddo;
/assignment to beds in PACU
if ded(i) <= 1 then ob; //respect dedication constraint
else if tech(i) <= 1 then ob; //respect technology dependency
else
j=random(i); // randomly chosen
ob=j;
enddo;
endfor;
enddo;
```

**Optimization schedule:** In order to provide an efficient schedule within a very reasonable time, we use a Tabu Search (TS) method (Glover and Laguna, 1997). The performance of the Tabu Search algorithm for the HFS2\( C_{\text{max}} \) was analyzed from the computational point of view. It gave good results with the size of the Tabu list fixed to 7, the candidate list to 500 and iterations number to 1000 (Belkadi et al., 2006, Sahraoui et al., 2003).

The algorithm 3 shows the adaptation of this search to our case. Details can be found by Belkadi et al. (2006).

**Algorithm 3:**

```plaintext
Tabu_search(max_iter)
Initialize Tabu memory T to \( \emptyset \);
Initial_schedule(i,D,N,p,ob);
X_init=p[ob];
X_current = X_best = X_init;
iter = 0;
begin
while((iter=max_iter) or (SOX_current(T)<v)) do
// SOX_current(T) is set of possible moves of X_current
begin iter = iter + 1;
Best_Move = 0;
for (all candidate moves \( E \) restricted neighborhood) do
choose sk \( E \) SOX_current(T) as:
sk(X_current) = optimum(s(X_current)
S(OX_current(T)) // sk is a possible candidate moves which is in not tabu
memory
X_next = sk(X_current); //X_next is given by the possible move on the
//current solution
move_value = H(X_next) - H(X_current);
if (move_value < best_value)
then Best_move = move_value;
X = X_successor;
endif;
done;
update tabu memory
if (H(X) < H(X_best))
then X_best = X;
X_current = X;
Endfor;
endwhile;
return X_best;
enddo;
```

1385
We study on 6 types of neighbor as the permutation of order, of assignment, the insertion etc. A restricted neighborhood system (Baptiste et al., 2001) must satisfy all constraints. An example of a neighbor is given in algorithm 4.

Algorithm 4:
neighb_change_assignment_i_1stage_i_patient
(o_p,b_p)
trial=0;
Repeat trial++;
    k = random (2)+1; // Select a random stage j: ORs or beds in PACU
    l = random (N)+1; // Select a random order i of a patient
    if dec_{ij} =1 then possible = true;
    if k=2 and dec_{pi} > 1 then possible=false;
Until (possible or trial> limit )
If possible then // no constraint on his current assignment
    if k=1 then
do j = random (l) while o_pj // select another OR
    o_pj // change allocation
else do j = random (l) while b_{ij} = j // select another bed
    //change allocation
end:
else choice_neighborhood ( ).
end:

Where choice_neighborhood( ) is a function for reselection a of another type of neighborhood.

RESULTS AND DISCUSSION

Data used in Experimentations are extracted from an implemented database. We supply the database from the history of the multidisciplinary Operating theatre of the EHUO. This center opens at 8 am and the operating theatre comprises 3 ORs and 9 beds in PACU and 2 beds in ICU dedicated to critical cases. The daily number of surgeries performed is less than 10.

First, we decrease number of beds to only 4 to give a simple example of a bad sized operating theater under constraint.

Input data: N = 5 Patients noted A,B,... E. L_i=3 ORs (the first ordinary, the second dedicated to cardiac state and the third has high strelity). L_i=4 Beds in PACU(th the latest is for Intensive Care).The Execution times are shown in Table 2.

Constraints:

\[ \text{Ded}_{01} = 2 \] (patient D must go to the cardiac OR)
\[ \text{Ded}_{01} = 4 \] (patient D must go to the Intensive Care bed)
\[ \text{Noseq}_{E,E} = 1 \]

Feasible schedule: A run without optimization give the solution below: abdce, 12212 and 12441. Assignment to

<table>
<thead>
<tr>
<th>Patient</th>
<th>Operating time (h)</th>
<th>Reweaking time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2.0</td>
<td>5.5</td>
</tr>
<tr>
<td>B</td>
<td>0.5</td>
<td>2.0</td>
</tr>
<tr>
<td>C</td>
<td>3.0</td>
<td>8.5</td>
</tr>
<tr>
<td>D</td>
<td>3.0</td>
<td>5.0</td>
</tr>
<tr>
<td>E</td>
<td>2.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table 2: Operating and reweaking times

<table>
<thead>
<tr>
<th>Order</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>A</td>
<td>B</td>
<td>D</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>( \delta_{01} )</td>
<td>08</td>
<td>08</td>
<td>08.5</td>
<td>10</td>
<td>11.5</td>
</tr>
<tr>
<td>( \delta_{22} )</td>
<td>10</td>
<td>08.5</td>
<td>11.5</td>
<td>16.5</td>
<td>15.5</td>
</tr>
<tr>
<td>( \delta_{44} )</td>
<td>15.5</td>
<td>10.5</td>
<td>16.3</td>
<td>21.5</td>
<td>17.5</td>
</tr>
</tbody>
</table>

Table 3: Allocation schedule of the example

Fig. 4: Gantt diagram of the feasible solution of the example

OR, to PACU and dates of beginning and ending of process in both OR and PACU are shown in Table 3. As an example, Patient D has the third order. He is assigned to the second OR and the fourth bed in PACU. His surgery begins at 8.5 a clock, finishes at 11.5 a clock. He lives the PACU at 16.5 a clock. The makespan is calculated with the max of \( f_{22} \), date of departure from PACU and the min of \( \delta_{22} \), date of beginning to OR, \( C_{max}=21.5-08=13.5 \) (h).

We validate result with a Gantt diagram (Fig. 4) which highlights the resources occupation and demonstrates the feasibility of scheduling.

An optimization give a better schedule (bdace,12311,24312) with a \( C_{max} = 11.5 \) h and the Gantt diagram of Fig. 5.

Some results of experimentation done on the existing database are in Table 4.

The aim was to define constraints of surgeries scheduling. We have submitted an operating theatre scheduler to many constraints that we had token from reality. For resolution we proposed for \( C_{max} \) scheduling a Tabu search which is already known well in this area and local search for constraints satisfaction problem. Results of many Fictive examples of bad sized operating theaters and real examples from the multidisciplinary
surgery or constraints increase, shown the capacity of algorithms to give a good rates of criteria amelioration. In its practical aspect, the advantage of our algorithms is that they could give computerized feasible optimal schedule of 10 surgeries in less than 14.3 seconds where a simple feasible hand-made could take 5 days.

**CONCLUSION**

We presented manufacturing constraints accustomed to the Operating theatre scheduling. The aim was to present the problem of surgeries scheduling differently in its real image, by adding these constraints. In order to give a satisfactory schedule, we implemented a local search for Maximal Constraints Satisfaction. Tabu Search was used to optimize the Cmax. We envision the use of several other meta heuristics and parallel versions of these metaheuristics in order to choose the best method.

**REFERENCES**


