European Journal of Operational Research 247 (2015) 401–413

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Scheduling operating theatres: Mixed integer programming vs. constraint programming

Tao Wang[a,*], Nadine Meskens[b], David Duvivier[c]

[a] University of Lyon, INSA-LYON, University of Jean Monnet Saint-Etienne, DISP - EA 4570, F-69621 Villeurbanne, France
[b] Catholic University of Louvain, Louvain School of Management, B-7000 Mons, Belgium
[c] University of Valenciennes and Hainaut-Cambresis, LAMIH - UMR CNRS 8201, Campus du Mont Houy, F-59313 Valenciennes, France

ARTICLE INFO

Article history:
Received 30 January 2014
Accepted 5 June 2015
Available online 22 June 2015

Keywords:
Combinatorial optimization
Constraints satisfaction
Linear programming
OR in health services
Models comparison

ABSTRACT

The daily scheduling of an operating theatre is a highly constrained problem. In addition to standard scheduling constraints, many additional constraints on human and material resources encountered in real life should be taken into account. These constraints concern the priority of operations, the affinities between surgical team members, renewable and non-renewable resources, various sizes in the block scheduling strategy, and the surgical team's preferences/availabilities. We developed two models in our research work, using mixed-integer and constraint programming respectively. These were compared using a real-life case in order to determine which one coped better with a highly constrained problem. A cross-comparison of the experimental results shows that the mixed-integer programming model provides a better performance using the weighted sum objective function than using the makespan minimization objective function. Conversely, the constraint programming model is better suited to the makespan minimization objective function than to the weighted sum objective function. The originality of this research lies on three levels: (1) two models are presented in detail and compared using real data; (2) constraint programming is used to schedule the operating theatre; (3) some new constraints are taken into account, such as the affinities between team members in the composition of surgical teams, and the priorities of patients such as diabetics.

1. Introduction

Health systems in many countries are in crisis. This situation is not new; what is new, however, is the awareness that if the current trend continues most health systems will no longer be viable by 2015 (IBM, 2006). Belgium is no exception to this trend, even though it is sometimes said to have one of the best healthcare system in the world. Its strengths are its almost complete coverage of the population (Durant, 2006), the dynamics produced by the global supply system and the actual delivery of care, the quality of health education and healthcare providers, and the increasing professionalization of the healthcare management structures (Itinera Institute, 2008). Nevertheless, Belgium is losing ground to other countries as regards the overall quality of its healthcare system. The country was ranked sixth in the *Euro Health Consumer Index* for 2013. First place went to the Netherlands, followed by Switzerland, Iceland, Denmark and Norway (EHCI, 2013).

Healthcare systems are fundamental to European societies, making a huge contribution to human activity and hence economic development. Health and its preservation are human needs that everyone wants to obtain and maintain.

During the 1960s and 1970s, the prosperous economic situation in Belgium allowed almost unlimited expenditure on health, which led to enough people being hired and enough material being bought to ensure the supply of good quality services. However, from the 1980s to the present day, major rationalizations have been undertaken focusing notably on hospitals.

The research reported in this paper focuses on one of the central and most expensive of hospital activities: the management of operating theatres. The operating theatre in a hospital is composed of operating rooms and one recovery room. This research aims to help the operating theatre manager improve the organization of these facilities, in particular the assignment of surgical cases. The development of an efficient algorithm is required for assigning a set of surgical cases to operating rooms during a period (often a week). This weekly operating theatre planning and scheduling problem is solved in two phases. In other words, decisions are made at the tactical and then the operational level. First, a planning problem is solved to obtain the date of surgery for each patient, allowing for the

* Corresponding author. Tel.: +33 477448954; fax: +33 477448921.
E-mail addresses: tao.wang@univ-st-etienne.fr (T. Wang), Nadine.meskens@uclouvain-mons.be (N. Meskens), David.Duvivier@univ-valenciennes.fr (D. Duvivier).

http://dx.doi.org/10.1016/j.ejor.2015.06.008
0377-2217/

availability of operating rooms and surgeons. Second, a daily scheduling problem is solved to determine the sequence of operations in each operating room (Fei, Chu, & Meskens, 2009).

Managing the operating theatre is a complex task because surgical cases must be planned and scheduled so as to minimize the costs of operating rooms and satisfy the needs and requests of surgeons, anesthetists and nurses. Satisfying the patient's needs and managing the material resources must also be taken into account. Moreover, human and material resources are in limited supply, and legal regulations have to be fulfilled.

Our study focuses on the daily scheduling of surgical cases at the operational level, taking into account human and material constraints. The role of human beings in the decision-making process cannot be ignored. The operating theatre employs various teams of workers (surgical, nursing, anesthetic, maintenance, etc.). Given the limitations on resources, scheduling their activities is crucial to the efficient performance of their work. Some constraints are linked to restrictions on the availability of resources: these include the opening hours of the operating rooms, the availability of surgeons, anesthetists and nurses, and the availability and numbers of surgical instruments, and recovery beds. Other constraints are linked to the specificities of the resources (e.g. the versatility of the operating rooms, staff qualifications). Similarly, just as the operating rooms may be dedicated to certain types of procedures, the medical staff may also be specialized. The operating theatre manager therefore has to try to harmonize resources and increase the versatility of the staff so as to optimize the use of the operating theatre.

The daily scheduling of the operating rooms is a highly constrained problem. A recent review by Meskens, Duvivier, and Hanset (2013) suggests that very few studies of this problem have taken into account the constraints of both material and human resources. The only constraints on human resources which are normally considered are the number and availability of surgeons, stretcher-bearers and anesthetists and sometimes of nurses and surgeon assistants (Beliën & Demeulemeester, 2007; Fei et al., 2009, 2010; Guinet & Chaabane, 2003; Ghazalbash, Sepehri, Shadpour, & Atighehchian, 2012; Hashemi, Rousseau, & Pesant, 2014; Marcon, Kharraja, & Simonnet, 2003; Roland, Di Martinelly, Riane, & Pochet, 2010; Vijayakumar, Parikh, Scott, Barnes, & Gallimore, 2013; Wang et al., 2014). Recent works (Marques, Captivo, & Vaz Pato, 2012; Van Huele & Vanhoucke, 2014) also consider the surgeon's workload. A very small number of research papers have taken into account priorities between patients (Cardoen, Demeulemeester, & Belien, 2009a, 2009b; Min & Yih, 2010a, 2010b). In general, the material constraints taken into account are limited to the number of operating rooms, the availability of specific materials and the number of beds in the recovery room (Augusto, Xie, & Perdomo, 2010; Bulgarini, Di Lorenzo, Lori, Matarrese, Schoen, 2014; Fei, Meskens, & Chu, 2010; Ghazalbash et al., 2012; Kharraja, Hammami, & Abbou, 2004; Pham, & Kinkert, 2008; Santibanez, Begen, & Atkins, 2007; Testi & Tànfani, 2009; Vijayakumar et al., 2013; Van Huele & Vanhoucke, 2014). Stochastic aspects have been integrated by some authors in the planning and scheduling problem of operating theatre such as Denton, Viapiano, and Vogl (2007), Hans, Wullink, Van Houdenhoven, and Kazemier (2008), Min and Yih, 2010a, 2010b), Lamiri (2008), Lamiri, Xie, Dolgui, and Grimaud (2009).

However, in real life other constraints are also very important and we propose in this research to take them into account. These include human constraints such as the priority of some surgical cases (for example, children and people suffering from diabetes should be scheduled at the beginning of the day), the preferences of surgeons, nurses and anesthetists, and material constraints such as the versatility of operating rooms and the availability of medical supplies.

In particular, the affinity relationship between two persons who work on the same surgical cases has been taken into account in this research. This is an important constraint for team building that has not carried weight in previous research works (except in Meskens et al., 2013). It has frequently been noted that the same people work together every day in the operating room for long periods of time. A strong affinity in teamwork can generate synergy through a coordinated effort, and allows each member to maximize his or her strengths and minimize his or her weaknesses. Obviously, the surgical team's effectiveness contributes to better communication and so to quality and safety in the health care received by patients. A growing number of studies have demonstrated this point (Carney, West, Neily, Mills, & Bagian, 2010; Kurmann et al., 2010; Sekhar and Mantovani, 2015; Tibbs & Moss, 2014; Weaver et al., 2010).

Each resource limitation included in an operating theatre management model leads to an increase in its complexity, in terms of the number of variables and/or the number of constraints and the computational time required to solve the optimization problem. In the case of highly constrained problems, however, where the solutions space might be narrow and fragmentary, exact methods could provide optimal solution within reasonable time. Constraint programming (CP) may also be another adequate tool for our highly constrained target problem. Indeed, the goal of CP is to solve problems when combinatorial optimization has to handle many variables and constraints.

The major difficulty encountered in solving a combinatorial optimization problem is the explosion in the number of combinations as the number of variables increases. The main problem when considering such a huge set of combinations is the amount of computing hours or days required to find an optimal solution. This is not acceptable in the context of time-limited decision-making. However, unlike the classical mathematical approach, the constraint programming paradigm is based on "reasoning" about the constraints and is still efficient when a lot of constraints are involved. This is one of the reasons why we chose to use constraint programming. This method also has the advantage of providing the user with different modes of functioning when searching for a solution in the path tree of solutions (Krzysztof, 2003).

Based on logic programming and graph theory, constraint programming is an alternative to mathematical programming for complex problems that have a slow convergence. It is also an efficient approach to solving and optimizing problems in the event of nonlinear constraints, logical statements, or non-convex solution space. This includes time-tabling problems, sequencing problems, and allocation or rostering problems (IBM ILOG, 2010). Constraint programming has been widely applied in industrial scheduling problems (Baptiste, LePape, & Nuijten, 2001; El Khayat, Langevin, & Riopel, 2006; Harjunkoski & Grossmann, 2002; Novas & Henning, 2014), but has rarely been used in the field of healthcare. Some perspectives in the literature have recommended the use of constraint programming for solving nurse rostering problems in operating theatres (Trilling, 2006), staff scheduling in healthcare (Bourdais, Galinier, & Pesant, 2003), and medical resident scheduling problems (Topaloglu & Ozkarahan, 2011). Recently, Hashemi et al. (2014) as well as Van Huele and Vanhoucke (2014) used constraint programming based on a column generation approach to resolve a surgery-planning and scheduling problem. Zhao and Li (2014) resolved a scheduling of elective surgeries problem in an ambulatory surgical center. They proposed and compared constraint programming and Mixed Integer Nonlinear Programming models to solve the scheduling problem. But the treated problem was not highly constrained. The authors did not take into account any human or material constraints.

In order to demonstrate the advantages of applying CP to a highly constrained problem as against a traditional method, we compare a constraint programming model (CP-MOD) to a traditional mathematical programming one (MP-MOD).

In our two models, new constraints have been taken into account, such as the priority of operations, the affinities between surgical team members, renewable and non-renewable resources, various sizes in the block scheduling strategy, and the surgical team's preferences/availabilities. The comparison between the two models is the main contribution.

Following this introduction, the problem is defined mathematically in the second part of the paper, which sets out its notations and hypothesis. The third and fourth parts describe respectively the MP and CP scheduling models, and the fifth part compares the two models. This comparison is made on the one hand by the minimization of the makespan, and on the other hand by the minimization of the weighted sum of the completion time. Finally we present our conclusions and perspectives.

2. The problem

In this research, we first need to describe mathematically the problem of daily scheduling of surgical cases. The time horizon is set to one day and divided into T time slots ($t \in \{1..T\}$). In our implementation each time slot is fixed to 30 minutes, the most common divisor of the surgery durations that we collected from a field study. This temporal granularity is quite close to reality, but does not substantially increase the size of the search space. There is a set of R operating rooms. Each room will then be available for T time slots, which will correspond to free periods for the surgeons.

In practice, there are two types of operations: elective operations, which are planned by the surgeon in consultation with the patient; and emergency operations, which — as their name suggests — are unplanned and arrive unexpectedly. In general, operations for emergency patients are performed in dedicated operating rooms; some of them are allowed to be scheduled together with elective operations, if patients' clinical conditions are relatively stable. In our model, each operation included in the set of operations to be assigned is denoted by $o \in \Omega$.

A set of material and human resources is also required for each operation. Concerning human resources, the legislation requires the presence of at least one surgeon, one anesthetist, and two nurses for each surgical case. The affinity relationships between members of surgical team are taken into account. Some specialized works (Leach, Myrtle, & Weaver, 2011; Mazzocco et al., 2009; Weaver et al., 2010) showed that affinities in a surgical team had an impact on the performance of the team in terms of communication, cooperation, and coordination, which itself has a significant impact on the quality of care delivered to the patient. Their estimations are obtained by means of a field survey and interviews with the surgical staff. Since affinity relationships are not all evaluated equal between members, the lower value is thought to be acceptable to qualify a bilateral affinity relationship. Hence an overall evaluation of these bilateral affinity relationships will contribute to building efficient surgical teams.

We make a distinction between renewable and non-renewable resources. Material resources can be renewable (K^{ρ}) or non-renewable (K^{υ}). For example, sterile medical trays, which are a basic requirement for operations, need to be completely sterilized between uses, and so are non-renewable resources in the course of a day. Human resources are typically renewable resources (K^{ρ}).

We consider the existence of three kinds of priority for operations: High, Medium and Low. Operations with a high priority (set Ω_b) start earlier than the other operations; they cover children, diabetics, ambulatory operations, etc. Operations with a low priority (set Ω_e) are carried out later than the others, and are mainly infectious cases that contaminate the room and require more cleaning after the operation. We have also added preferential constraints related to the preferences of the surgeons for particular operating rooms and to the availability of human resources.

The limited capacity of recovery beds is taken into account. In practice, the ratio of operating rooms to recovery beds is 1–1.5 (legal constraint), or 1–2 if precautionary measures are taken. The operation-recovery process is a two-stage no-wait flow-shop, so the scheduling of operating theatres should ensure that there will be at least one available recovery bed at the end of each operation. When no bed is (or will be) free at the end of the surgery, the operation is not (or will not be) performed. The patient can be transferred to the normal bed only after his/her recovery.

2.1. Problem statement

The notations of input data used for the two models are listed below:

Ω	The set of all the surgical cases
O	The number of operations: $O = \mid \Omega \mid$
o	An operation: $o \in \Omega$
T	The set of time slots in a day
t	A time slot: $t \in T$
Γ	The set of operating rooms
R	The number of operating rooms: $R = \mid \Gamma \mid$
r	An operating room: $r \in \Gamma$
S	The set of surgeons
s	A surgeon: $s \in S$
A	The set of anesthetists
a	An anesthetist: $a \in A$
N	The set of nurses
n	A nurse: $n \in N$
Γ_s	The set of operating rooms allocated to surgeon s
Ω_s	The set of operations allocated to surgeon s
Ω_b	The set of operations with high priority that start earlier than the others
Ω_m	The set of operations with medium priorities
Ω_e	The set of operations with low priority that start later than the others
O_s	The number of operations allocated to surgeon s: $O_s = \mid \Omega_s \mid$
O_b	The number of operations taking place at the beginning of the day (high priority): $O_b = \mid \Omega_b \mid$
O_m	The number of operation with medium priorities : $O_m = \mid \Omega_m \mid$
O_e	The number of operations that take place at the end of the day (low priority): $O_e = \mid \Omega_e \mid$
d_o	The duration of operation o (in number of time slots)
ES_o	The earliest start of operation o (time slots)
LS_o	The latest start of operation o (time slots)
K^{ρ}	The set of renewable resources
K^{υ}	The set of non-renewable resources
k	A resource: $k \in \{ K^{\rho} \cup K^{\upsilon} \}$
m_{ok}^{ρ}	The quantity of the renewable resource k required by operation o
m_{ok}^{υ}	The quantity of the non-renewable resource k required by operation o
$M_k^{\rho}(t)$	The quantity of the renewable resource k, available at moment t
M_k^{υ}	The quantity of the non-renewable resource k, available for the day
$M^S(s, t)$	The availability of surgeon s at time t: $M^S(s, t) \in \{0, 1\}$
$M^N(n, t)$	The availability of nurse n at time t: $M^N(n, t) \in \{0, 1\}$
$M^A(a, t)$	The availability of anesthetist a at time t: $M^A(a, t) \in \{0, 1\}$
B	The set of recovery beds
b	A recovery bed: $b \in B$
db_o	The duration of the recovery after operation o (in number of time slots)
$B_{1,o}$	Beginning of operation o, in an operating room (time slots)
$B_{2,o}$	Beginning of recovery, in the recovery room, at the end of operation o (time slots)
$C_{2,o}$	End of recovery of the operation o, in the recovery room(time slots)
C_{max}	Makespan, end of recovery, in the recovery room, at the end of the latest operation o (time slots)
$AffSN(s,n)$	The affinity relationship between surgeon s and nurse n ($\in\{0,1\}$)
$AffSA(s,a)$	The affinity relationship between surgeon s and anesthetist a ($\in\{0,1\}$)
$AffNA(n,a)$	The affinity relationship between nurse n and anesthetist a ($\in\{0,1\}$)
$AffNN(n_1,n_2)$	The affinity relationship between two nurses n_1 and n_2 ($\in\{0,1\}$)

As discussed above, three sets of operations were created: Ω_b, containing all the operations that have to be carried out before the other operations; Ω_e, containing those that have to be carried out after the other operations; and Ω_m, containing those that have no special

Table 1
Example of matrix $M^S(s, t)$, giving the availability of Surgeons 1, 2 and 3 between 8.00 and 20.00.

	8.00	8.30	9.00	9.30	10.00	10.30	11.00	11.30	12.00	…	19.30
T	1	2	3	4	5	6	7	8	9	…	24
Surgeon 1	0	0	0	0	0	1	1	1	1	…	1
Surgeon 2	1	1	1	1	1	0	0	0	0	…	0
Surgeon 3	0	0	0	1	1	1	1	1	0	…	0

Table 2
Example of affinity matrix $Aff(p_1,p_2)$.

	Anes.1	Anes.2	…	Anes.A	Nurs.1	Nurs.2	…	Nurs.N
Surg.1	9	8	…	0	9	7	…	8
Surg.2	7	7	…	6	7	7	…	7
…	…	…	…	…	…	…	…	…
Surg.S	5	6	…	8	0	9	…	9
Nurs.1	8	8	…	8		5	…	8
Nurs.2	3	8	…	9	8		…	7
…	…	…	…	…	…	…		…
Nurs.N	9	8	…	9	8	3	…	

requirement. In addition, we can define the earliest/latest start times (ES_o and LS_o) for each operation when needed.

Operations are defined by a set of characteristics. We assume that the duration of a particular operation, d_o, can be predicted in advance. Furthermore, for the comfort and safety of the patient, we assume that the operation starts as soon as possible (ES_o) and no later than (LS_o) (expressed in time slots).

Renewable resources are available again once the operation terminates. For each operation, we know the quantity of renewable resource required by operation m^{ρ}_{ok}, and it will be compared to the amount of available resource $M^{\rho}_{k}(t)$ at time t. Unlike renewable resources, consumed non-renewable resources cannot be reused by other following operations. Only the total use of each resource $\sum m^{\nu}_{ok}$ is compared to the available quantity of the non-renewable resource M^{ν}_{k} for the day.

The availability of each surgeon is expressed in a matrix, $M^S(s, t)$. Table 1 gives an example of such a matrix, and shows that Surgeon 1 is available for operations from 10.30, Surgeon 2 between 8.00 and 10.30 and Surgeon 3 between 9.30 and 12.00. In the same way as for the surgeon, two availability matrices are given to describe respectively the availabilities of the anesthetist $M^A(a, t)$, and of the nurse $M^N(n, t)$.

The affinity relationship for teamwork is expressed through an affinity matrix $Aff(p_1,p_2)$, where each row and each column represents a person (a surgeon, an anesthetist, or a nurse). For each pair of persons, a score of 0–9 is assigned based on the preferences expressed by them (the lower of two unilateral notes). Mutual incompatibility is denoted by 0 and strong preference is denoted by 9. Several pretreatments are directly applied to this matrix before its use in the mathematical model through a Graphical User Interface. The full description of all these treatments lies outside of the scope of this paper. However, we may mention that their objective is to ensure the homogeneity of the values to a certain extent, and to ensure that there are sufficient non-null values to avoid any lack of teamwork with sufficient affinities.

It might be possible to use the integer values in Table 2 directly in the mathematical models. Then we need to ensure with appropriate constraints, that a team can be built if the average affinity value between the members is greater than a predefined threshold, which could be the median value5 between 0 and 9 for example. However, the average value may “hide” some dissatisfaction between members in the same team. In another solution, ensuring that the affinity criterion is linear, it might be possible to use a weighted sum of affinities $\Sigma(k_i \times Aff(p_1, p_2))$ between the members in the same team. However, this leads to the problem of choosing these weights k_i. In order to ensure a minimum affinity between all pairs of the members in each team, we have chosen to transform the affinities into binary values using a threshold. Thus, the integer values are of no use as the input data of our mathematical models since the affinity matrix is only used to answer a binary choice: whether two persons will work together in a same surgical team. Directly using integer values cannot enrich the results of the models, but increases the computational burden. If the affinity score is greater than or equal to the threshold, then the two persons p_1 and p_2 will be happy to work together. Otherwise, the two persons will refuse to work in a same team. Obviously a high threshold may hinder team-building due to incompatibility between members. Therefore, if, for example, the threshold is set to 5, the affinity matrix described above can be transformed as follows (Table 3). The variables $AffSA(s,a)$, $AffSN(s,n)$, $AffNA(n,a)$, $AffNN(n_1,n_2)$ obtain their values from the transformed matrix. However, the value of this threshold has to be defined. Several approaches are possible. One of them consists in starting from a value that is equal to the average value of the team members' affinities, and transforming the matrix into a binary matrix. If there is no solution due to a lack of surgical teams, then we should try again with a lower threshold value (using, for instance, a dichotomy approach to compute the new value). If there is a solution and there is enough time to restart a resolution, then we can increase the threshold value and try to solve the problem with this value, and so on. The resulting approach is a tradeoff between a set of pretreatments and mathematical models. All basic tests are implemented in the pretreatment phase to simplify the mathematical modelling.

2.2. Assumptions

A list of assumptions must be enumerated in order to complete the description of our framework:

- No surgeon can operate on more than one patient at the same time. Similarly, no recovery bed can be occupied by more than one patient at the same time.
- All the scheduled patients are ready for their surgery on the given day.
- All scheduled operations have to be performed during the current day, i.e. no operation can be postponed.
- The induction time for each operation, and the post-operation clean-up after the operation, are included in the operating time.
- The recovery beds in the recovery rooms are identical; that is, the patient can be transferred to any available recovery bed.
- Once a surgical case has started in an operating room, it cannot be interrupted, i.e. there is no pre-emption.
- The time needed at the start of every work day to clean the operating rooms before all operations is not taken into account. The post-operation clean-up time (cleaning time after each operation) is included in the operating time. These ‘setup’ times are independent of the operating schedule. However, operations may be sequenced in a particular order, as mentioned above. For example,

Table 3
Example of affinity matrices $AffSA(s,a)$, $AffSN(s,n)$, $AffNA(n,a)$ and $AffNN(n_1,n_2)$ after threshold applied.

	Anes.1	Anes.2	…	Anes.A	Nurs.1	Nurs.2	…	Nurs.N
Surg.1	1	1	…	0	1	1	…	1
Surg.2	1	1	…	1	1	1	…	1
⋮	⋮	⋮	…	⋮	⋮	⋮	…	⋮
Surg.S	1	1	…	1	0	1	…	1
Nurs.1	1	1	…	1		1	…	1
Nurs.2	0	1	…	1	1		…	1
⋮	⋮	⋮	…	⋮	⋮	⋮		⋮
Nurs.N	1	1	…	1	1	0	…	

⇨

	Anes. 1	Anes.2	…	Anes.A	Nurs.1	Nurs.2	…	Nurs.N
Surg.1 Surg.2 ⋮ Surg.S	$AffSA(s,a)$				$AffSN(s,n)$			
Nurs.1 Nurs.2 ⋮ Nurs.N	$AffNA(n,a)$				$AffNN(n_1,n_2)$			

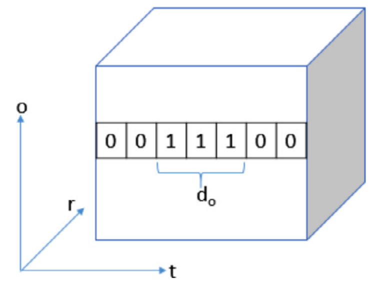

Fig. 1. The binary variables $OTR(o,t,r)$.

those which are particularly contaminating, such as iatrogenic infections, will be scheduled at the end of the day.

3. The MP model

The daily scheduling problem of the operating theatre can be mathematically formulated by a mixed integer programming (MP) model. The objective is to minimize the makespan of the generated work schedule.

The operations have to be placed individually; each one is described by the number of time-slots for which it will take up a room over the operating duration. Within a room r, the duration of each operation is given by d_o, a group of consecutive binary variables showing the assignment of operation o in this room. Only the binary variables relative to the operation for the room in question and for the sequence of time slots occupied by the operation will be set to 1; the others are all set to 0. In this way, a three-dimensional (O operations, T time slots and R rooms) binary variables matrix is created. This matrix is represented by $OTR(o, t, r)$. The use of such matrices is motivated by the fact that one of our objectives is to obtain the detailed schedule of each resource. As illustrated in Fig. 1, binary variables take the value 1 not only at the beginning of the operation but throughout its duration in a given room r.

A number of other matrices are described for the members of the team and the recovery room.

$OTR(o,t,r)$ =	1 if operation o is assigned at time t to operating room r
	0 otherwise
$OTB(o,t,b)$ =	1 if operation o is assigned at time t to recovery bed b
	0 otherwise
$STR(s,t,r)$ =	1 if surgeon s is assigned at time t to operating room r
	0 otherwise
$NTR(n,t,r)$ =	1 if nurse n is assigned at time t to operating room r
	0 otherwise
$ATR(a,t,r)$ =	1 if anesthetist a is assigned at time t to operating room r
	0 otherwise

The formulation of the model can now be written as:

$$\text{Minimize}\, C_{\max} \tag{1a}$$

s.t.

$$C_{2,o} = B_{2,o} + db_o - 1 \tag{1b}$$

$$C_{\max} \geq C_{2,o} \tag{1c}$$

$$B_{1,o} = \frac{\left(\sum_{r=1}^{R}\sum_{t=ES_o}^{LS_o+d_o-1} t.OTR(o,t,r)\right) - \left(\frac{d_o.(d_o-1)}{2}\right)}{d_o} \quad \forall o \in \Omega \tag{2}$$

$$B_{2,o} = \frac{\left(\sum_{b=1}^{B}\sum_{t=ES_o+d_o}^{LS_o+d_o+db_o-1} t.OTB(o,t,b)\right) - \left(\frac{db_o.(db_o-1)}{2}\right)}{db_o} \quad \forall o \in \Omega \tag{3}$$

$$\sum_{o=1}^{O} OTR(o,t,r) \le 1, \quad \forall r \in \Gamma, \forall t \in T \tag{4}$$

$$\sum_{r \in \Gamma_s} \sum_{o=1}^{O} OTR(o,t,r) \le 1, \quad \forall s \in S, \forall t \in T \tag{5}$$

$$\sum_{r=1}^{R} \sum_{t=ES_o}^{LS_o+d_o-1} OTR(o,t,r) = d_o, \quad \forall o \in \Omega \tag{6}$$

$$\sum_{r=1}^{R} \sum_{t=1}^{T-d_o+1} \left\lfloor \frac{\sum_{\tau=t}^{t+d_o-1} OTR(o,\tau,r)}{d_o} \right\rfloor = 1, \quad \forall o \in \Omega \tag{7}$$

$$\sum_{o=1}^{O} OTB(o,t,b) \le 1, \quad \forall b \in B, \forall t \in T \tag{8}$$

$$\sum_{b=1}^{B} \sum_{t=ES_o+d_o}^{LS_o+d_o+db_o-1} OTB(o,t,b) = db_o, \quad \forall o \in \Omega \tag{9}$$

$$\sum_{b=1}^{B} \sum_{t=1}^{T-db_o+1} \left\lfloor \frac{\sum_{\tau=t}^{t+db_o-1} OTB(o,\tau,b)}{db_o} \right\rfloor = 1, \quad \forall o \in \Omega \tag{10}$$

$$B_{1,o} + d_o = B_{2,o}, \quad \forall o \in \Omega \tag{11}$$

$$B_{1,o_1} \le B_{1,o_2} + \left(1 - \frac{\sum_{t=ES_{o_2}}^{LS_{o_2}+d_{o2}-1} OTR(o_2,t,r)}{d_{o2}} \right) T, \quad \forall o_1 \in \Omega_b, \forall o_2 \in \Omega_m, \forall r \in \Gamma \tag{12}$$

$$B_{1,o_2} \le B_{1,o_3} + \left(1 - \frac{\sum_{t=ES_{o_3}}^{LS_{o_3}+d_{o3}-1} OTR(o_3,t,r)}{d_{o3}} \right) T, \quad \forall o_2 \in \Omega_m, \forall o_3 \in \Omega_e, \forall r \in \Gamma \tag{13}$$

$$\sum_{r=1}^{R} \sum_{o=1}^{O} m_{ok}^{\rho} OTR(o,t,r) \le M_k^{\rho}(t), \quad \forall t \in T, \forall k \in K^{\rho} \tag{14}$$

$$\sum_{r=1}^{R} \sum_{o=1}^{O} \frac{m_{ok}^{\upsilon}}{d_o} \sum_{t=1}^{T} OTR(o,t,r) \le M_k^{\upsilon}, \quad \forall k \in K^{\upsilon} \tag{15}$$

$$\sum_{o \in \Omega_s} OTR(o,t,r) \le M^S(s,t), \quad \forall s \in S, \forall t \in T, \forall r \in \Gamma_s \tag{16}$$

$$STR(s,t,r) = \sum_{o \in \Omega_s} OTR(o,t,r), \quad \forall s \in S, \forall t \in T, \forall r \in \Gamma_s \tag{17}$$

$$\sum_{r=1}^{R} NTR(n,t,r) \le M^N(n,t), \quad \forall n \in N, \forall t \in T \tag{18}$$

$$2 \times STR(s,t,r) = \sum_{n=1}^{N} NTR(n,t,r), \quad \forall s \in S, \forall t \in T, \forall r \in \Gamma_s \tag{19}$$

$$\sum_{r=1}^{R} ATR(a,t,r) \le M^A(a,t), \quad \forall a \in A, \forall t \in T \tag{20}$$

$$STR(s,t,r) = \sum_{a=1}^{A} ATR(a,t,r), \quad \forall s \in S, \forall t \in T, \forall r \in \Gamma_s \tag{21}$$

$$STR(s,t,r) + NTR(n,t,r) - 1 \le AffSN(s,n) \quad \forall s \in S, \forall n \in N, \forall t \in T, \forall r \in \Gamma \tag{22}$$

$$STR(s,t,r) + ATR(a,t,r) - 1 \le AffSA(s,a) \quad \forall s \in S, \forall a \in A, \forall t \in T, \forall r \in \Gamma \tag{23}$$

$$NTR(n,t,r) + ATR(a,t,r) - 1 \le AffNA(n,a) \quad \forall n \in N, \forall a \in A, \forall t \in T, \forall r \in \Gamma \tag{24}$$

$$NTR(n_1,t,r) + NTR(n_2,t,r) - 1 \le AffNN(n_1,n_2) \quad \forall n_1, n_2 \in N, n_1 \ne n_2, \forall t \in T, \forall r \in \Gamma \tag{25}$$

$$OTR(o,t,r) = 0, \quad \forall o \in \Omega, \forall r \in \Gamma, \forall t \notin [ES_o, LS_o + d_o] \tag{26}$$

$$OTR(o,t,r) = 0, \quad \forall s \in S, \forall o \in \Omega_s, \forall r \notin \Gamma_s, \forall t \in T \tag{27}$$

$$OTB(o,t,b) = 0, \quad \forall o \in \Omega, \forall b \in B, \forall t \notin [ES_o + d_o, LS_o + d_o + db_o] \tag{28}$$

The objective function (1a) and constraints (1b) and (1c) ensure minimization of the makespan of the operating theatre.

The beginning time of an operation or a recovery is given by constraints (2) and (3). They allow us to locate the first value 1 of the variables *OTR* declared for the operation *o* in an operating room *r* or on a recovery bed *b*. Constraints (4) indicate that two operations cannot take place at the same time in the same operating room. Furthermore, there certainly exists an exact match between every operation and its surgeon. Thus, the operations allocated to each surgeon, selected from the set of surgeons *S*, are known in advance. Constraints (5) prevent any surgeon from conducting two operations at the same time in different operating rooms. Constraints (6) and (7) were introduced into the model to express the fact that an operation o has to take place over d_o consecutive time slots. Constraints (6) require the number of variables set to 1 to be equal to d_o, while constraints (7) specify that the variables set to 1 in this interval must be continuous, due to the fact that the integer division of the sum of consecutive 1 by d_o must be equal to 1. In this way, we ensure that there is only one string of consecutive 1, representing the operation (zero everywhere else). Constraints (8), (9) and (10) express the conditions at the recovery room stage, for instance the fact that there is only one patient at a time in a recovery bed, and that the variables *OTB* set to 1 have to be continuous throughout the interval of time db_o. Constraints (11) ensure continuity between two stages of the procedure; the expression on the left indicates the last *OTR* set to 1 at the operating stage of an operation, while that on the right represents the first *OTB* set to 1 at the recovery stage of the same operation.

Constraints (12) and (13) are precedence constraints. Similarly, they express the timing requirements for the operations included in the sets Ω_b and Ω_e. High priority operations should start before those with medium priorities, but low priority operations after those with medium priorities. Constraints (14) and (15) represent the limits on the renewable and non-renewable resources. The difference lies in the fact that we have to weight these variables by the inverse of the duration of the operation to obtain comparable values. Constraints (16) and (17) express the availability of surgeon s at time t by $M^S(s, t)$. Similarly, constraints (18) and (20) are derived from constraints (16) and express the availability of nurse and anesthetist at time t by $M^N(n, t)$ and $M^A(a, t)$. Constraints (19) and (21) express the fact that a team consists of a surgeon, an anesthetist and two nurses. Constraints from (22) to (25) describe the implementation of affinity relationship in the model. Team members can work together only when all combinations $Aff(p_1,p_2)$ are greater than or equal to the affinity threshold. Constraints (26) and (27) express that one operation performed by surgeon s should be placed in the time window between its earliest start time and latest end time, only in one of the operating rooms allocated to surgeon s. They also ensure that there is no operation planned when the surgeon cannot operate. Finally, constraints (28) ensure that post-operative recovery can only be placed in the time window between its earliest start time and latest end time.

Constraints (7) and (10) should be linearized to be used with Cplex solver. One of the solutions is to replace each of the two constraints with three linear constraints and new decision variables. The equivalent constraints to (7) are designed as following.

$$x(o,t,r) \geq OTR(o,t,r) - OTR(o,t-1,r) \quad \forall o \in \Omega,\ \forall t \in T \tag{7a}$$

$$\sum_{t=1}^{T}\sum_{r=1}^{R} x(o,t,r) = 1 \quad \forall o \in \Omega \tag{7b}$$

$$OTR(o,0,r) = 0 \quad \forall o \in \Omega,\ \forall r \in \Gamma \tag{7c}$$

In the same way, the equivalent constraints to (10) are designed as following.

$$y(o,t,b) = \begin{cases} 1 & \text{if recovery after operation o starts at time t in bed b} \\ 0 & \text{otherwise} \end{cases}$$

$$y(o,t,b) \geq OTB(o,t,b) - OTB(o,t-1,b) \quad \forall o \in \Omega,\ \forall t \in T \tag{10a}$$

$$\sum_{t=1}^{T}\sum_{b=1}^{B} y(o,t,b) = 1 \quad \forall o \in \Omega \tag{10b}$$

$$OTB(o,0,b) = 0 \quad \forall o \in \Omega,\ \forall b \in B \tag{10c}$$

An overview analysis of MP model is performed to estimate the size of the scheduling problem. The number of decision variables depends on $(o+s+n+a)*t*r+o*t*b$, and the number of constraints are given in Table 4 below. The size of the problem scales up quickly with the time-splitting scheme and the number of operations, surgeons, nurses, anesthetists, operating rooms, and recovery beds. A MP optimization engine will take a considerable amount of time to analyze the model's feasibility and to calculate the optimal solution in a fragmented solution space.

$$\text{Teams} = \left\{ < o, s, r, n_1, n_2, a > \middle| \begin{array}{l} \forall s \in S, \forall o \in \Omega_s, \forall r \in \Gamma_s, \forall n_1, n_2 \in N, n_1 \neq n_2, \forall a \in A, \\ AffSA(s,a) \times AffSN(s,n_1) \times AffNA(n_1,a) \times AffSN(s,n_2) \\ \times AffNA(n_2,a) \times AffNN(n_1,n_2) > 0 \end{array} \right\} \tag{30}$$

Due to the limits introduced by the use of linearization in solving scheduling problems, the effectiveness of MP models is apparently not guaranteed, especially for large-scale problems. In contrast, constraint programming offers suitable modeling techniques to solve time-tabling and sequencing problems usually characterized by non-linear constraints, logical constraints, and incompatibility constraints. The CP engine is often used as a fast generator of feasible solutions, but takes a considerable amount of time in terms of finding optimal solutions. Detailed analysis will be described in the Section 5.

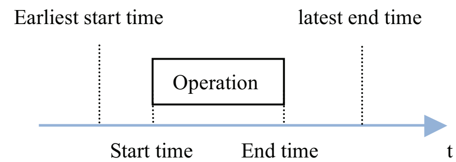

Fig. 2. An interval variable.

4. The CP model

Constraint programming deals not only with logical constraints used in constraint satisfaction problems (CSP), but also with mathematical constraints that are usually solved by classical approaches. A CP model is defined in three parts. The first part describes all decision variables; the second gives the domain for each of these variables, including discrete values; the third contains various sets of constraints, representing logical and mathematical relationships between variables.

4.1. Decision variables

The operations are represented by interval variables instead of binary variables in MP-MOD. An interval variable represents an interval of time during which an operation is performed. In general, it is defined (see Fig. 2) by its inherent attributes, such as earliest start time, latest end time and duration, but its position in time or its start time are unknowns in CP-MOD. In contrast, an operation's start time in MP-MOD should be deduced from output results, once the optimal solution is found. The definition of interval variables for operations is given as follows (definition 29).

$$\text{INTERVAL operation } (o \text{ in } \Omega) \text{ in } ES_o \ .. \ LS_o + d_o - 1 \text{ SIZE } d_o \tag{29}$$

4.2. Logical relationships

An operation can be performed only when all human and material resources are brought together. A six-tuple $<o,s,r,n_1,n_2,a>$ is proposed to represent team building for a given operation. The objective is to make a preselecting constraint that generates all possible surgical teams which satisfy affinity constraints. An instance of the six-tuple is composed of an operation o, the surgeon s who is capable to perform the operation o, an operating room r allocated to s, two nurses n_1, n_2, and an anesthetist a, with all ($Aff(p_1,p_2)$ > threshold) satisfied. The set of all possible surgical teams can be expressed in terms of the definition (30). In the same way, the set of all operation-recovery combinations is obtained by the definition (31). These two definitions also considerably reduce the search space of the scheduling problem.

$$\text{Recovery} = \{< o, b > | \forall b \in B, \forall o \in \Omega \} \tag{31}$$

In one of the solutions generated by the CP-MOD, only one six-tuple per operation will be present. The others, having not satisfied

Table 4
Overview analysis about number of constraints in MP-MOD.

Set of constraints	Decision matrix	Number of constraints	Set of constraints	Decision matrix	Number of constraints
4	*OTR*	$R*T$	17	*STR, OTR*	$\sum_{s=1}^{S} T*\lvert\Gamma_s\rvert$
5	*OTR*	$S*T$	18	*NTR*	$N*T$
6	*OTR*	O	19	*STR, NTR*	$S*T*R$
7	*OTR*	O	20	*ATR*	$A*T$
8	*OTB*	$B*T$	21	*STR, ATR*	$S*T*R$
9	*OTB*	O	22	*STR, NTR*	$S*T*R*N$
10	*OTB*	O	23	*STR, ATR*	$S*T*R*A$
11	*OTR, OTB*	O	24	*NTR, ATR*	$N*T*R*A$
12	*OTR*	$R*O_b*O_m$	25	*NTR*	N^2*T*R
13	*OTR*	$R*O_m*O_e$	26	*OTR*	$O*T*R$
14	*OTR*	$T*\lvert K^\rho\rvert$	27	*OTR*	$\sum_{s=1}^{S} O_s*T*(R-\lvert\Gamma_s\rvert)$
15	*OTR*	$\lvert K^\upsilon\rvert$	28	*OTB*	$O*T*B$
16	*OTR*	$\sum_{s=1}^{S} T*\lvert\Gamma_s\rvert$			

all constraints and the objective function, are absent. This can be expressed by a Boolean function (32).

$$\text{presenceOf}(x) = \begin{cases} \text{true}, \forall x \in \text{Teams}, x \text{ is present} \\ \text{false}, \forall x \in \text{Teams}, x \text{ is absent} \end{cases} \tag{32}$$

Any given resource, no matter whether it is a human resource, an operating room or a recovery bed, can be used at moment t by only one operation or one patient, so there is no question of overlap between two operations in the schedule of a resource above. However an overlap is allowed when two operations use different resources. Here, all individual schedules of these resources are defined as follows, but no-overlap constraints will be added in the model's section below. Definitions from (33) to (37) give different schedules for surgeon s, nurse n_1, n_2, anesthetist a, operating room r, and recovery bed b as soon as a feasible solution found.

$$\text{Sched}(s) = \left\{\text{operation}(x) \middle| \forall x \in \text{Teams}, \text{presenceOf}(x), x.s = s\right\} \quad \forall s \in S \tag{33}$$

$$\text{Sched }(n) = \left\{\text{operation }(x) \middle| \forall x \in \text{Teams}, \ \text{presenceOf }(x), x.n_1 = n \text{ or } x.n_2 = n\right\} \quad \forall n_1, n_2 \in N \tag{34}$$

$$\text{Sched}(a) = \left\{\text{operation}(x) \middle| \forall x \in \text{Teams}, \text{presenceOf}(x), x.a = a\right\} \quad \forall a \in A \tag{35}$$

$$\text{Sched}(r) = \left\{\text{operation}(x) \middle| \forall x \in \text{Teams}, \text{presenceOf}(x), x.r = r\right\} \quad \forall r \in \Gamma \tag{36}$$

$$\text{Sched}(b) = \left\{\text{operation}(y) \middle| \forall y \in \text{Recovery}, \text{presenceOf}(y), y.b = b\right\} \quad \forall b \in B \tag{37}$$

The availabilities of each member in a surgical team are expressed by a stepwise function. A typical stepwise function, denoted by stepwise(i in 1..n) {Value[i] → Time point[i]} is often used to model the availability or the use of a resource over time. An example of a stepwise function is illustrated in Fig. 3. Using its concept, the functions from (38) to (40) respectively express the availabilities of surgeon, nurse and anesthetist in time.

$$\text{Avail}(s \in S) = \text{stepwise}(t \in T)\left\{M^s(s,t) \to t; 0\right\} = \left\{ \begin{array}{l} M^s(s,1), t < 1 \\ M^s(s,i+1), \forall i \in [1..n-1], \forall t \in [t_i, t_{i+1}) \\ 0, t > t_n \end{array} \right\} \tag{38}$$

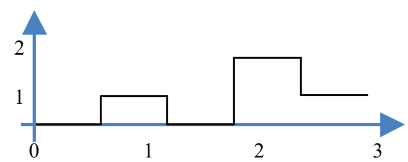

Fig. 3. Example of stepwise function stepwise{0→1, 1→2, 0→3, 2→4, 1→5}.

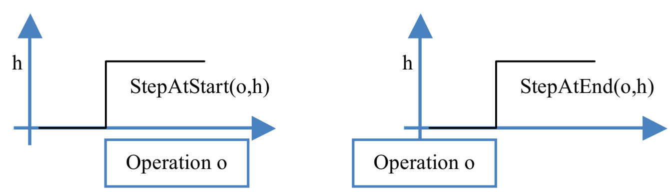

Fig. 4. Example of stepAtStart(o,h) and stepAtEnd(o,h).

$$\text{Avail}(n \in N) = \text{stepwise}(t \in T)\left\{M^n(n,t) \to t; 0\right\} \tag{39}$$

$$\text{Avail}(a \in A) = \text{stepwise}(t \in T)\left\{M^a(a,t) \to t; 0\right\} \tag{40}$$

However, the availabilities of renewable resources cannot be described by a simple stepwise function. The quantity of a renewable resource is reduced at the beginning of an operation and restored at the end of it. Two additional functions are introduced to model the consumption and the production of a cumulative resource. The value of function stepAtStart(o,h) changes to h at the start of an interval variable, an operation o in our case, while the value of stepAtEnd(o,h) changes at the end of the operation o. Therefore, a function Qty(k) is given in (41) to express the available quantity of renewable resource k over time (Fig. 4).

$$Qty(k \in K^\rho) = \text{stepwise}(t \in T)\left\{M_k^\rho(t) \to t; 0\right\} - \sum_{o \in \Omega} \text{stepAtStart}(o, m_{ok}^\rho) + \sum_{o \in \Omega} \text{stepAtEnd}(o, m_{ok}^\rho) \tag{41}$$

4.3. The constraint programming model

The CP-Model of the daily scheduling problem is introduced below.

$$\text{MinimizeMax}(\text{endOf}(\text{o})) \qquad \forall o \in \Omega \tag{42}$$

s.t.

$$\neg(\text{start}(\delta_1) < \text{start}(\delta_2) < \text{end}(\delta_1)) \Leftrightarrow \text{noOverlap}(\text{Sched}(\beta)) \quad \forall \delta_1, \delta_2 \in \text{Sched}(\beta), \forall \beta \in S \cup N \cup A \cup \Gamma \cup B \tag{43}$$

$$\text{presenceOf}(\forall x_1 \in \text{Teams}, x_1.o = o) \Rightarrow \neg(\exists x_2 \in \text{Teams}, x_2.o = o, x_2 \neq x_1, \text{presenceOf}(x_2)) \quad \forall o \in \Omega \tag{44}$$

$$\text{presenceOf}(\forall y_1 \in \text{Recovery}, y_1.o = o) \Rightarrow \neg(\exists y_2 \in \text{Recovery}, y_2.o = o, y_2 \neq y_1, \text{presenceOf}(y_2)) \quad \forall o \in \Omega \tag{45}$$

$$\text{End}(\text{operation}(x)) + 1 = \text{Start}(\text{recovery}(y)) \quad \forall x \in \text{Teams}, \forall y \in \text{Recovery}, x.o = y.o \tag{46}$$

$$\text{End}(\text{operation}(x_1)) \geq \text{Start}(\text{operation}(x_2)) \quad \forall x_1, x_2 \in \text{Teams}, x_1.r = x_2.r, (x_1.o \in \Omega_b, x_2.o \in \Omega_m) \; or \; (x_1.o \in \Omega_m, x_2.o \in \Omega_e) \tag{47}$$

$$\text{presenceOf}(x) \Rightarrow Avail(s) \wedge Avail(n_1) \wedge Avail(n_2) \wedge Avail(a) = 1 \quad \forall x \in \text{Teams}, x.s = s, x.n_1 = n_1, x.n_2 = n_2, x.a = a, \text{start}(\text{operation}(x)) \leq t \leq \text{end}(\text{operation}(x)) \tag{48}$$

$$Qty(k) \geq 0 \quad \forall k \in K^{\rho} \cup K^{\upsilon} \tag{49}$$

The objective function (42) ensures minimization of the makespan of the operating theatre. The function endOf(o) gives the end of recovery of the operation o, and we have endOf(o) $= C_{2,o}$. Constraints (43) are used to indicate that no overlap is allowed between two operations inside an individual schedule of any resource, such as surgeon, nurse, anesthetist, operating room, and recovery bed. Constraints (44) and (45) state that for a given operation, the presence of one surgical team and one recovery bed is exclusive in a generated solution, hence there does not exist another alternative that is admitted at the same time. Constraints (46) and (47) are precedence constraints. Constraints (46) ensure that recovery starts as soon as the operation finishes. Constraints (47) express that operations with higher priority should start before those with lower priority. Constraints (48) are used to ensure that a six-tuple can be chosen into a feasible solution only if all necessary human resources are available during the whole period of operation. Finally, Constraints (49) concern the available quantity of renewable and non-renewable resources.

4.4. Comparison with the MP model

There are various ways to model this daily scheduling problem, and the choice of modeling is crucial to the performance of the model. Our MP and CP models have been optimized but remain accurate to the real world constraints. The comparison between these two models is based on our choice of modeling that aims at testing their respective abilities.

First, the decision variables are defined in different ways. Binary variables are used in the MP model to describe each of the five decision variables in a three-dimensional matrix. The core element of this daily scheduling problem, the operations, is represented by a sequence of *OTR* (*o,t,r*). However in the CP model, an integrated interval variable is sufficient to describe an operation.

In order to ensure that an operation is correctly represented in output results, Constraints (6) and (7) should be introduced into the MP model. Furthermore, additional variables and constraints have been developed for the purpose of obtaining linear constraints. In the case of linear programming, where the solver accepts only linear constraint and linear objective function, linearization becomes a major obstacle to the modelling work at a conceptual level. There is no guarantee that equivalent and efficient linear constraints can always be found. None of these issues matter in the CP model.

Second, the constraint section of the CP model takes into account only some precedence and overlap constraints; all the others, like affinity constraints from (22) to (25) in the MP model, can be directly integrated into the definitions of team tuples (30). This can significantly reduce the search space of the daily scheduling problem. In addition, the start time of each operation, defined by (2), becomes an inherent unknown of the interval variables; the availability constraints (16) (18) and (20) can be replaced by (38), (39) and (40) in the CP model. Constraint programming using logical operators, variables, and expressions is much more intuitive for the modelling work of scheduling problems, compared with mathematical programming.

5. Experimental results

Once the two models have been properly formulated and specified, we are now able to compare them and assess which is the more successful. The models were both run on a Core2^{TM} Duo processor (2 GHz, 4 GB RAM, Operating System: Windows 7). The MP-MOD, using mixed-integer programming, was coded in Optimization Programming Language and solved by Cplex 12.5 (IBM ILOG, 2010).

The CP-MOD uses a constraint-programming method. It was solved by the Constraint Programming Optimizer included in IBM ILOG optimizers. This method possesses the descriptive power needed to model this kind of problem accurately. Built on an event-based propagation mechanism with back-tracking structure, constraint programming also offers the advantage of finding a feasible solution via a depth-first search algorithm, or a more sophisticated, branch-and-bound type of search can be carried out to find an optimal solution. However such searches tend to involve extremely long computing time (Fages, 1996).

5.1. Input data

In order to evaluate the proposed methods of improving the practical arrangement of surgical cases in the operating theatre, real life data from a Belgian University Hospital are used in this study. In this hospital, there are nine surgical specialties: stomatology, gynecology, urology, orthopedic surgery, ENT/otorhinolaryngology, ophthalmology, pediatric surgery, plastic surgery and abdominal surgery. The operating theatre in this hospital is composed of 4 operating rooms and one recovery room with 8 beds. Normally, all the operating rooms are open from 8 a.m. to 6 p.m., and can be extended to 8 p.m. if necessary. The recovery room opens simultaneously with the operating rooms and does not close until the last patient has left the operating theatre.

The main aim is to provide the operating theatre manager with a good solution, which satisfies all the constraints and can be computed in a short time. The data used in this study come from 6321 records from the operating theatre, collected over a one-year period. The data consist mainly of date of surgery, induction time, the start time and end time of surgery, the time the patient left the operating room, the surgeon and specialty for each surgical case, the reason for admittance and some personal information (the patient's birthday, gender, etc.). Overtime hours are not considered because we have defined the time slots for a working day as fixed. However they could be taken into account by allocating them a specific hourly cost, higher than that which applies during the normal day.

The same surgical cases undertaken by the same surgeons using the same resources were analyzed in each model, with the duration

Table 5
Data sets.

Data sets	Operations	Surgeons	Anesthetists	Nurses	Renewable resources	Non-renewable resources	Operating rooms	Recovery beds
D1	8	4	4	8	5	5	4	8
D2	15	7	4	8	5	5	4	8
D3	17	9	4	8	5	5	4	8

Table 6
Comparison of solutions obtained through MP-MOD.

		MP-MOD				
Data set	Affinity threshold	Number of constraints	Number of variables	Number of solutions	Makespan	Time (opt.)
D1	5	17061	6251	5	14	26.41
	6	17061	6251	4	14	26.22
	≥7	Infeasible				
D2	5	23350	10662	4	24	10.05
	6	23350	10662	2	24	30.59
	≥7	Infeasible				
D3	5	26620	12032	2	24	27.89
	6	26620	12032	1	24	32.74
	≥7	Infeasible				

Table 7
Comparison of solutions obtained through CP-MOD.

		CP-MOD					
Data set	Affinity threshold	Number of constraints	Number of variables	Number of solutions	Makespan	Time (best)	End time
D1	5	207929	1900	1	14	8.01	387.36
	6	28985	616	1	14	2.52	94.3
	≥7	Infeasible					
D2	5	715985	3541	1	23	12.24	728.85
	6	45591	928	1	23	1.51	115.87
	≥7	Infeasible					
D3	5	868313	4011	1	23	16.17	772.45
	6	49263	1029	1	23	2.18	133.98
	≥7	Infeasible					

of the time slots set to 30 minutes (the greatest common factor of operation durations). Besides the performance comparison between two models, their robustness is also evaluated using three datasets of various sizes (from D1 to D3 in Table 5), with all constraints taken into account. For example, the dataset 3 for this day consists of 17 operations, 9 surgeons, 8 nurses, 4 anesthetists, 5 renewable material resources and 5 non-renewable resources. The differences between datasets consist in the numbers of operations and of surgeons.

As mentioned earlier in Section 2.1, the integer affinity matrix is transformed into a binary matrix by using a threshold value. In the presented results the threshold value is set to 5, the median value. It will then be increased successively in a sequence of testing scenarios as shown in Table 6.

5.2. Computational results with the makespan objective

The objective function in both models is defined by a simple formulation that minimizes the makespan, that is, the maximum job completion time, as presented by the functions (1a) and (42). Tables 6 and 7 compare the computational results of the two models.

Table 6 gives the results obtained from experiments using the MP-MOD. All the three data sets have been introduced into the model successively, and for each of these data sets, different threshold values have been applied. Since the threshold value only affects the affinity matrix inside a data set, the numbers of constraints and variables do not change along with the threshold value for a given data set, but increase from a small data set to a larger one. In D1, eight surgical cases need to be scheduled; if the threshold value is set to 5 or 6, the objective function leads to an optimal solution to 14 time-slots within less than 27 seconds. Using D2 and D3, the optimal solution found - also within a reasonably short time - is that all surgical cases can be finished before the end of 24th time-slot. When the threshold value goes beyond 6, no any feasible solution can be found, because no surgical teams can be formed. The column 'Number of solutions' indicates that the number of feasible solutions found before the solver terminates. There are a few solutions which can satisfy all the constraints. This phenomenon, as mentioned at the beginning of the paper, corresponds to the main characteristic of highly constrained problems.

Table 7 shows the results generated by the CP-MOD. With regard to constraints, their number is much larger in CP-MOD than in MP-MOD, because the CP engine needs to generate more domain information about the variables from the compact formulation of the problem, in order to perform a filtering algorithm. In consequence, much more memory is required. However, fewer variables are employed since an operation is represented by only one interval variable, instead of a sequence of binary variables. An important phenomenon should be noted; the numbers of constraints and variables decrease when the threshold value increases from 5 to 6, as opposed to what

Table 8
Comparison of solutions obtained through MP-MOD.

		MP-MOD				
Data set	Affinity threshold	Number of constraints	Number of variables	Number of solutions	Makespan	Time (opt.)
D1	5	17052	6241	5	14	1.97
	6	17052	6241	3	14	1.64
	≥7	Infeasible				
D2	5	23334	10645	2	23	2.01
	6	23334	10645	1	23	1.96
	≥7	Infeasible				
D3	5	26602	12013	1	23	3.59
	6	26602	12013	1	23	3.95
	≥7	Infeasible				

Table 9
Comparison of solutions obtained through CP-MOD.

		CP-MOD					
Data set	Affinity threshold	Number of constraints	Number of variables	Number of solutions	Makespan	Time (best)	End time
D1	5	207929	1900	2	14	7.8	757.69
	6	28985	616	2	14	1.9	240.94
	≥7	Infeasible					
D2	5	715985	3541	1	23	25.16	2229.42
	6	45591	928	3	23	19.14	314.66
	≥7	Infeasible					
D3	5	868313	4011	1	23	44.14	1862.62
	6	49263	1029	12	23	133.78	407.98
	≥7	Infeasible					

we have indicated in Table 6. The problem size is then reduced. As a result, the solver can find a feasible solution much more rapidly. In other words, the CP-MOD is highly sensitive to restrictions, even when they are configured inside the input data. As for the solutions' quality, only one solution has been identified in each scenario, but this first and best solution was found slightly quicker than the corresponding computing time of the MP-MOD. In the cases where the threshold value is set to 6 in D2 and D3, the CP-MOD gives a better solution than the MP-MOD in less than 3 seconds. The column 'End time' indicates the time spent before the CP-MOD reaches the fail limits[1]. CP-MOD is obviously inefficient in identifying optimal solutions, due to long computing time. But it does not seem necessary to seek the optimum solution for a highly constrained and frequently encountered problem, like the daily scheduling problem.

5.3. Computational results with a weighted sum objective function

The makespan objective function used initially in the two models is a logical function rather than a mathematical one. Now that the CP-MOD with the makespan objective seems to be better than the MP-MOD, it becomes necessary to estimate whether the CP-MOD retains an advantage under a pure mathematical objective function. A second objective function is then designed as follows.

In the MP-MOD

$$\text{Minimize} \quad w_1 \sum_{o\in O_b} C_{2,o} + w_2 \sum_{o\in O_m} C_{2,o} + w_3 \sum_{o\in O_e} C_{2,o} \tag{50}$$

[1] Considering extremely long computing time before finding the optimal solution with CP optimizer, we set the fail limit to 200,000 in the CP-MOD as a stopping rule. That means 200,000 failures can occur before terminating the search.

And in the CP-MOD

$$\text{Minimize} \quad w_1 \sum_{o\in O_b} endOf(o) + w_2 \sum_{o\in O_m} endOf(o) + w_3 \sum_{o\in O_e} endOf(o) \tag{51}$$

where w_1, w_2, and w_3 are non-negative weights and $w_1>>w_2>>w_3$.

Tables 8 and 9 compare the computational results after implementing this new objective function.

Table 8 presents better results compared with those in Table 6. There is indeed a very small difference in terms of problem size after changing the objective function, while optimal solutions are found ten times faster than before. However, it should be noted that this weighted sum objective function remains a specific function specially designed for our daily surgical scheduling problem. In contrast, our initial objective function (minimizing the makespan) is a more general function fitting most scheduling problems.

Table 9 compares the results obtained from the CP-MOD after changing the objective function to a weighted sum. With regard to the problem size, none of the numbers of constraints and variables has been changed, compared to Table 7. Though the same best makespan can always be found in each experiment, the amount of time spent has increased. Especially, the 'End time' becomes very long. The CP solver takes much more time to examine the feasibility of one potential solution. In addition, when D3 is used and the threshold value is set to 6, the CP-MOD returned 12 feasible solutions, among which 10 refer to the same makespan (i.e. 23). That means, a lot of alternatives have been found which satisfy all the constraints including all the precedence constraints. Only the order of several surgical operations is different between two alternatives. In real life, any of these alternatives conforms to what we expect in an operating theater; nevertheless, having more alternatives is conducive to the management of an operating theater facing unexpected events.

6. Conclusions and perspectives

In this paper, we introduced two models, using mixed-integer linear programming and constraint programming respectively, to solve the daily surgical case scheduling problem at the operational level, taking into account human and material constraints. In particular, affinity relationships between surgical team members have been integrated into the problem formulation and modelling, an inclusion that makes this scheduling problem a highly constrained problem. Regarding the detailed formulation of the two models, the CP-MOD allows logical constraints that provide explicit formulation and intuitive comprehension, while the MP-MOD can only be solved once all the constraints are linearized. Our MP-MOD, designed for a real-life highly constrained problem, can be solved to optimality in a short computation time.

Using real-life data, three data sets having different sizes have been designed and used to generate a series of experimental scenarios, in each of which a different threshold value has been applied to transform the affinity matrix into a binary matrix. In addition, a weighted sum objective function which takes into account operations' priorities has been introduced to carry out an extended analysis. We have made a cross-comparison through experimental results. On the basis of the results, a few feasible solutions have been identified for most scenarios, with the MP-MOD giving better performance using the weighted sum objective function than the initial makespan objective function. In contrast, the CP-MOD works with the initial objective function rather than the weighted sum objective function.

In general, the mathematical programming model has the advantage that numerous methods of resolution already exist in terms of investigating the solution space. However it has to be composed of linear constraints, in order to be solved using MILP solver. Moreover, it does not allow the sophisticated constraints encountered in real life without complicating the model in such a way that it becomes difficult to build, explain, generalize and solve. In contrast, the constraint programming model uses in addition a declarative logic language, which has a higher descriptive power than classical mixed-integer linear programming languages. Logical and non-linear constraints can be included. It allows a large number of constraints to be taken into account. In particular, the more constraints there are, the better the CP model will perform, due to the fact that problem size decreases, as does the computation time required to find feasible solutions. Certainly, the CP model is usually slower than the MP model in finding optimal solutions, but it is more efficient in generating feasible solutions, especially for highly constrained and frequently encountered problems.

Future perspectives focus on integrating additional real-life constraints, such as size of surgical team according to specific operations, skill of surgical nurse, nurses' work schedule, and overtime hours. This will take us further away from "standard/academic" problems and toward real-life highly constrained problems. Matheuristics are surely one of our perspectives if larger instances occur and/or if the number of constraints continues to increase.

References

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

Baptiste, P., LePape, C., & Nuijten, W. (2001). *Constraint-Based scheduling – applying constraint programming to scheduling problems*. Kluwer Academic Publishers.

Beliën, J., & Demeulemeester, E. (2007). A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal Operational Research, 189*(3), 652–668.

Bourdais, S., Galinier, Ph., & Pesant, G. (2003). HIBISCUS: a constraint programming application to staff scheduling in health care. *Principles and Practice of Constraint Programming, LNCS, 2833*, 153–167.

Bulgarini, N., Di Lorenzo, D., Lori, A., Matarrese, D., & Schoen, F. (2014). Operating room joint planning and scheduling. In A. Matta, J. Li, E. Sahin, E. Lanzarone, & J. Fowler (Eds.), *Proceedings of the International Conference on Health Care Systems Engineering* (pp. 127–138). Springer Proceedings in Mathematics & Statistics.

Carney, B. T., West, P., Neily, J., Mills, P. D., & Bagian, J. P. (2010). Differences in nurse and surgeon perceptions of teamwork: implications for use of a briefing checklist in the OR. *AORN Journal, 91*(6), 722–729.

Cardoen, B., Demeulemeester, E., & Belien, J. (2009). Optimizing a multiple objective surgical case sequencing problem. *International Journal of Production Economics, 119*, 354–366.

Cardoen, B., Demeulemeester, E., & Belien, J. (2009). Sequencing surgical cases in a day-care environment: an exact branch-and-price approach. *Computers & Operations Research, 36*, 2660–2669.

Denton, B., Viapiano, J., & Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science, 10*, 13–24.

Durant, G. (2006). Financement des hôpitaux: Des divergences, mais surtout des convergencesˇ, Revue Hospitals.be, N°1 p. 7.

EHCI (2013). http://www.healthpowerhouse.com/files/ehci-2013/ehci-2013-be-fr.pdf.

El Khayat, G., Langevin, A., & Riopel, D. (2006). Integrated production and material handling scheduling using mathematical programming and constraint programming. *European Journal of Operational Research, 175*(3), 1818–1832.

Fages, F. (1996). Programmation par contraintes, Ellipses Marketing.

Fei, H., Chu, C., & Meskens, N. (2009). Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria. *Annals of Operations Research, 166*(1), 91–108.

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

Ghazalbash, S., Sepehri, M. M., Shadpour, P., & Atighehchian, A. (2012). Operating room scheduling in teaching hospitals. *Advances in Operations Research, 2012*, 1–16.

Guinet, A., & Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics, 85*, 69–81.

Hans, E., Wullink, G., Van Houdenhoven, M., & Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research, 185*, 1038–1050.

Harjunkoski, I., & Grossmann, I. E. (2002). Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Computers & Chemical Engineering, 26*(11), 1533–1552.

Hashemi, D., Rousseau, L., & Pesant, G. (2014). A constraint programming based column generation approach for operating room planning and scheduling. *Lecture Notes in Computer Science, 8451*, 455–463.

IBM (2006). Santé 2015: Gagnant-gagnant ou perdant-perdant. https://www-03.ibm.com/industries/ca/fr/healthcare/files/sante_2015_gagnant_gagnant_ou_perdant_perdant_sommaire.pdf.

IBM ILOG (2010). http://www-01.ibm.com/software/fr/websphere/ilog/.

Itinera Institute (2008). Quel est l'état de santé du système de soins en Belgique. http://www.itinerainstitute.org/fr/pressroom/communiques-de-presse/_press/hc/.

Kharraja, S., Hammami, S., & Abbou, R. (2004). Plan directeur d'allocation des plages horaires: approche globale. *Gestion et ingénierie des systèmes hospitaliers*. Mons, Belgium: GISEH'04 9-11 September.

Krzysztof, R. A. (2003). *Principles of constraint programming*. Cambridge: Cambridge University Press.

Kurmann, A., Tschan, F., Semmer, N. K., Seelandt, J., Candinas, D., & Beldi, G. (2010). Human factors in the operating room – The surgeon's view. *Trends in Anaesthesia and Critical Care, 2*(5), 224–227.

Lamiri, M. (2008). *Planification des blocs opératoires avec prises en compte des aléas*. France: University of St Etienne, Doctoral thesis.

Lamiri, M., Xie, X. L., Dolgui, A., & Grimaud, F. (2009). A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research, 185*, 1026–1037.

Leach, L. S., Myrtle, R. C., & Weaver, F. A. (2011). Surgical Team: role perspectives and role dynamics in the operating room. *Health Services Management Research, 24*(2), 81–90.

Marcon, E., Kharraja, S., & Simonnet, G. (2003). The operating theatre planning by the follow-up of the risk of no realization. *International Journal of Production Economics, 85*, 83–90.

Marques, I., Captivo, M. E., & Vaz Pato, M. (2012). An integer programming approach to elective surgery scheduling analysis and comparison based on a real case. *OR spectrum, 34*, 407–427.

Mazzocco, K., Petitti, D. B., Fong, K. T., Bonacum, D., Brookey, J., Graham, S., Lasky, R. E., Sexton, J. B., & Thomas, E. J. (2009). Surgical team behavior and patient outcomes. *The American Journal of Surgery, 197*(5), 678–685.

Meskens, N., Duvivier, D., & Hanset, A. (2013). Multiobjective scheduling of operating theatre considering desiderata of surgical team. *Decision Support Systems, May 2013, 55*(2), 650–659.

Min, D., & Yih, Y. (2010). Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research, 206*, 642–652.

Min, D., & Yih, Y. (2010). An elective surgery scheduling problem considering patient priority. *Computers & Operations Research, 37*, 1091–1099.

Novas, J. M., & Henning, G. P. (2014). Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming. *Expert Systems with Applications, April 2014, 41*(5), 2286–2299.

Pham, D.-N., & Kinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research, 185*, 1011–1025.
Roland, B., Di Martinelly, C., Riane, F., & Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering, 58*(2), 212–220.
Santibanez, C., Begen, M., & Atkins, D. (2007). Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a British Columbia health authority. *Health Care Management Science, 103*, 269–282.
Sekhar, L., & Mantovani, A. (2015). Teamwork mentality in neurosurgical teams to improve patient safety. *World Neurosurgery, 83*(1), 41–43.
Testi, A., & Tànfani, E. (2009). Tactical and operational decisions for operating room planning: efficiency and welfare implications. *Health Care Management Science, 12*(4), 363–373.
Tibbs, S. M., & Moss, J. (2014). Promoting teamwork and surgical optimization: combining team stepps with a specialty team protocol. *AORN Journal, 100*(5), 477–488.
Topaloglu, S., & Ozkarahan, I. (2011). A constraint programming-based solution approach for medical resident scheduling problems. *Computers & Operations Research, 38*(1), 246–255.
Trilling, L. (2006). *Aide à la décision pour le dimensionnement et le pilotage de ressources humaines mutualisées en milieu hospitalier*. Institut National des Sciences Appliquées de Lyon (INSA), Doctoral thesis.
Van Huele, C., & Vanhoucke, M. (2014). Analysis of the integration of the physician rostering problem and the surgery scheduling problem. *Journal of medical systems, 28*(43), 1–16.
Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., & Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research, 224*(3), 583–591.
Wang, Y., Tang, J., Pan, Z., & Yan, C. (2014). Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources. Soft Computing doi: 10.1007/s00500-014-1453-z.
Weaver, S. J., Rosen, M. A., Diaz Granados, D., Lazzara, E. H., Lyons, R., Salas, E., Knych, S. A., McKeever, M., Adler, L., & Barker, M. (2010). Does teamwork improve performance in the operating room? A multilevel evaluation. *Joint Commission. Journal on Quality and Patient Safety, 36*(3), 133–142.
Zhao, Z., & Li, X. (2014). Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming. *Operations Research for Health Care, 3*, 160–167.