



Solving Operating Room Scheduling Problems with Surgical Teams via Answer Set Programming

Carmine Dodaro¹, Giuseppe Galatà², Muhammad Kamran Khan³,
Marco Maratea³ (✉), and Ivan Porro²

¹ DEMACS, University of Calabria, Rende, Italy
dodaro@mat.unical.it

² SurgiQ srl, Genova, Italy

{giuseppe.galata,ivan.porro}@surgiq.com

³ DIBRIS, University of Genova, Genova, Italy

muhammad.kamrankhan@edu.unige.it,marco.maratea@unige.it

Abstract. The optimization of daily operating room surgery schedule can be problematic because of many constraints, like to determine the starting time of different surgeries and allocating the required resources, including the availability of surgical teams for complete surgical procedures. Recently, Answer Set Programming (ASP) has been successfully employed for addressing and solving real-life scheduling and planning problems in the healthcare domain. In this paper we present an enhanced solution using ASP for scheduling operating rooms taking explicitly into consideration availability of surgical teams, that include a surgeon and an anesthetist in different specialties for the entire duration of the surgery. We tested our solution on different benchmarks with realistic parameters for schedule's length up to the target 5-days planning. The results of our experiments show that ASP is a suitable methodology for solving also such enhanced problem.

1 Introduction

Hospitals, whose production output is service, often come across issues of long waiting times, surgeries cancellation for patients and even worst resource overload occur frequently. Within every Hospital, Operating Rooms (ORs) are an important unit. As indicated in [30], the ORs account for approximately 33% of the total hospital budget because it includes high staff costs (e.g., surgeons, anaesthetists, nurses) and material cost. Nowadays, in most modern Hospitals, long surgical waiting lists are present because of inefficient planning. Therefore, it is extremely important to improve the efficiency of ORs to enhance the survival rate and satisfaction of patients, thereby improving the overall quality of healthcare system. The Operating Room Scheduling (ORS) [1, 6, 29, 30] problem is the task of assigning patients to ORs by considering specialties, surgery durations, shift durations, beds availability and, most importantly the availability of

surgical teams for the entire duration of the surgery. Further, the solution must prioritise patients based on health urgency. In recent years a solution based on Answer Set Programming (ASP) [8, 21, 22] was proposed and is used for solving such problem [14, 15], together with other similar scheduling problems in this context, given its intuitive semantics [9] and the availability of efficient solvers put forward by ASP Competitions (see, e.g., [10, 18, 19]). We have recently enhanced the first solution by incorporating beds management [12]. However, the drawback with these solutions is that they do not consider availability of surgical teams which are an important part of the surgical process.

In this paper we improve our basic solution [14, 15] following another direction, and present an enhanced encoding that takes into explicit account the availability of surgical teams for planning surgical procedures. The problem is expressed in ASP as modular additions to previous, more limited, encoding of ASP rules implementing the surgical teams, and then efficient solvers like CLINGO [17] are used to solve the resulting ASP encoding. Results for planning horizons up to the target 5-days planning, obtained on different benchmarks and scenario with realistic parameters for a small-medium-sized Hospital, are positive and inline with Hospital needs, and further confirm that ASP is a suitable methodology for solving scheduling problems in this context.

The paper is structured as follows. Section 2 presents needed preliminary about ASP. Then, Sect. 3 describes the target problem in an informal way and as a mathematical formulation, whose ASP encoding is presented in Sect. 4. Section 5 shows the results of our experiments. The paper ends in Sect. 6 and 7 by discussing related work, and by showing conclusions and possible topics for further research.

2 Answer Set Programming

Answer Set Programming (ASP) is a programming paradigm developed in the field of non monotonic reasoning and logic programming. It is a form of declarative programming oriented towards difficult primarily NP-hard search problems and is based on the stable model (answer set) semantics. This section presents in the first paragraph the syntax of the ASP language, for easy the readability of the encoding, and then a widely used shortcut in the second paragraph. The semantics is presented informally while describing the encoding, and details can be found in [9].

Syntax. The syntax of ASP is similar to the one of Prolog. Variables are strings starting with uppercase letter and constants are non-negative integers or strings starting with lowercase letters. A *term* is either a variable or a constant. A *standard atom* is an expression $p(t_1, \dots, t_n)$, where p is a *predicate* of arity n and t_1, \dots, t_n are terms. An atom $p(t_1, \dots, t_n)$ is ground if t_1, \dots, t_n are constants. A *ground set* is a set of pairs of the form $\langle \text{consts} : \text{conj} \rangle$, where *consts* is a list of constants and *conj* is a conjunction of ground standard atoms. A *symbolic set* is a set specified syntactically as $\{Terms_1 : Conj_1; \dots ; Terms_t : Conj_t\}$, where

$t > 0$, and for all $i \in [1, t]$, each $Terms_i$ is a list of terms such that $|Terms_i| = k > 0$, and each $Conj_i$ is a conjunction of standard atoms. A *set term* is either a symbolic set or a ground set. Intuitively, a set term $\{X : a(X, c), p(X); Y : b(Y, m)\}$ stands for the union of two sets: the first one contains the X -values making the conjunction $a(X, c), p(X)$ true, and the second one contains the Y -values making the conjunction $b(Y, m)$ true. An *aggregate function* is of the form $f(S)$, where S is a set term, and f is an *aggregate function symbol*. Basically, aggregate functions map multisets of constants to a constant. The most common functions implemented in ASP systems are the following:

- *#count*, number of terms;
- *#sum*, sum of integers.

An *aggregate atom* is of the form $f(S) \prec T$, where $f(S)$ is an aggregate function, $\prec \in \{<, \leq, >, \geq, \neq, =\}$ is a comparison operator, and T is a term called guard. An aggregate atom $f(S) \prec T$ is ground if T is a constant and S is a ground set. An *atom* is either a standard atom or an aggregate atom. A *rule* r has the following form:

$$a_1 \vee \dots \vee a_n \text{ :- } b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m.$$

where a_1, \dots, a_n are standard atoms, b_1, \dots, b_k are atoms, b_{k+1}, \dots, b_m are standard atoms, and $n, k, m \geq 0$. A literal is either a standard atom a or its negation *not* a . The disjunction $a_1 \vee \dots \vee a_n$ is the *head* of r , while the conjunction $b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m$ is its *body*. Rules with empty body are called *facts*. Rules with empty head are called *constraints*. A variable that appears uniquely in set terms of a rule r is said to be *local* in r , otherwise it is a *global* variable of r . An ASP program is a set of *safe* rules, where a rule r is *safe* if both the following conditions hold: (i) for each global variable X of r there is a positive standard atom ℓ in the body of r such that X appears in ℓ ; and (ii) each local variable of r appearing in a symbolic set $\{Terms : Conj\}$ also appears in *Conj*.

A *weak constraint* ω is of the form:

$$:\sim b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m. [w@l]$$

where w and l are the weight and level of ω , respectively. (Intuitively, $[w@l]$ is read “as weight w at level l ”, where weight is the “cost” of violating the condition in the body of w , whereas levels can be specified for defining a priority among preference criteria). An ASP program P is a finite set of rules. An ASP program with weak constraints is $\Pi = \langle P, W \rangle$, where P is a program and W is a set of weak constraints.

A standard atom, a literal, a rule, a program or a weak constraint is *ground* if no variables appear in it.

Syntactic Shortcuts. We will also use *choice rules* of the form $\{p\}$, where p is an atom. Choice rules can be viewed as a syntactic shortcut for the rule $p \vee p'$, where p' is a fresh new standard atom not appearing elsewhere in the program.

3 Problem Description

This section provides the description and the requirements of our problem, both informally and as a mathematical formulation, in two separate subsections.

3.1 Informal Description

The elements of the waiting list are called registrations. Moreover, registrations are not all equal, as they can belong to different specialties and they can have different priorities and duration. All ORs are available for a specialty according to the Hospital Master Surgical Schedule (MSS), and for 5 consecutive hours (300 min) in a single shift, while a full day consists of two shifts. Of course, the assignments must guarantee that the sum of the predicted duration of surgeries assigned to a particular OR shift does not exceed the length of the shift itself. For each registration we consider three priority score P1, P2, and P3, where P1 is for high priority registrations or very urgent, P2 is for medium priority and P3 is for low priority. Since P1 gathers high priority registrations, they must be all assigned to an OR, followed by P2 registrations over the P3. Additionally, in each specialty (considered to be 5 as target in small-medium-sized Hospitals) surgical teams are allocated with number of surgeons and anaesthetists every day as shown in Table 1. Tables 2 and 3, instead, show how many surgeons/anaesthetists are available in each shift and for each specialty. However, surgeons assigned to a shift in a day are different from the ones assigned for the other shift of the same day, while the same anaesthetists cover both shifts of the same day. Every surgeon works specifically for a number of hours every day; also surgeons in each specialty are assigned only to a single shift in a day, i.e., they either work in the morning (represented as shift 1, 3, 5, 7 and 9) or in evening shift (represented as shift 2, 4, 6, 8 and 10) as shown in Table 2. The anaesthetists are also linked to specialty and they also work for a fixed number of hours every day, but they can work together with surgeons during any shift of the day as shown in Table 3. In our model, we also assume that once a surgery is started in an OR it cannot be interrupted. Further, surgeons cannot operate on more than one patient at the same time. The overall goal is to assign the maximum number of registrations to the ORs, respecting the priorities, and taking into account the availability of respective surgical teams in a particular specialty for the complete surgery duration.

3.2 Mathematical Formulation

In this subsection we proceed by expressing our ORS problem in a more rigorous mathematical formulation. The first step is to describe more rigorously the elements we are dealing with. Let

- R be a set of registrations,
- SP be a set of specialties,
- O be a set of operating rooms,

We are ready to define the functions that can help establish the relations between the elements of the ORS problem.

Definition 1. (ORS problem) *Let*

- $p : R \mapsto \{1, 2, 3\}$ *be a function associating each registration to a priority;*
- $\delta : R \mapsto [1, \text{shift_duration}]$ *be a function associating each registration to a duration;*
- $\sigma : R \mapsto SP$ *be a function associating each registration to a specialty;*
- $mss : O \times SH \times SP \times D \mapsto \{0, 1\}$ *be a function such that $mss(o, sh, sp, d) = 1$ if the OR o is reserved to the shift sh and the specialty sp during the day d , and 0 otherwise;*
- $surg : SU \times SP \times SH \mapsto \{0, 1\}$ *be a function such that $surg(su, sp, sh) = 1$ if the surgeon su is associated to the specialty sp during the shift sh , and 0 otherwise;*
- $an : A \times SP \times SH \mapsto \{0, 1\}$ *be a function such that $an(a, sp, sh) = 1$ if the anaesthetist a is associated to the specialty sp during the shift sh , and 0 otherwise;*
- $surgWT : SU \times D \mapsto \mathbb{N}$ *be a function associating each surgeon and day to a working time;*
- $anWT : A \times D \mapsto \mathbb{N}$ *be a function associating each anaesthetist and day to a working time.*

Let $x : R \times SU \times A \times O \times SH \times D \times ST \mapsto \{0, 1\}$ be a function such that $x(r, su, a, o, sh, d, st) = 1$ if the registration r is assigned to the surgeon su , the anaesthetist a and the operating room o during the shift sh of the day d and the time slot st , and 0 otherwise. Moreover, for a scheduling x let $A_x = \{(r, su, a, o, sh, d, st) \mid r \in R, su \in SU, a \in A, o \in O, sh \in SH, d \in D, st \in ST, x(r, su, a, o, sh, d, st) = 1\}$ and $R_x^* = \{r \mid (r, su, a, o, sh, d, st) \in A_x\}$. Then, given sets $R, SP, O, SU, A, SH, D, ST$ and functions $p, \delta, \sigma, surg, an, surgWT, anWT$, and $\text{shift_duration} = 300$ and $\text{slot_duration} \in \{10, 20, 40, 60\}$, the ORS problem is defined as the problem of finding a schedule x , such that

- (c1) $st \cdot \text{slot_duration} + \delta(r) \leq \text{shift_duration} \quad \forall (r, su, a, o, sh, d, st) \in A_x;$
- (c2) $mss(o, sh, \sigma(r), d) \cdot surg(su, \sigma(r), sh) \cdot an(a, \sigma(r), sh) = 1 \quad \forall (r, su, a, o, sh, d, st) \in A_x;$
- (c3) $|\{(su, a, o, sh, d, st) : (r, su, a, o, sh, d, st) \in A_x\}| \leq 1 \quad \forall r \in R;$
- (c4) $x(r_1, su_1, a_1, o, sh, d, st) \cdot x(r_2, su_2, a_2, o, sh, d, st) = 0 \quad \forall r_1, r_2 \in R : r_1 \neq r_2;$
- (c5) $|\{r : (r, su, a, o, sh, d, st) \in A_x, st \leq t < st + \delta(r) \div \text{slot_duration}\}| \leq 1 \quad \forall o \in O, sh \in SH, t \in ST : mss(o, sh, sp, d) = 1;$
- (c6) $|\{r : (r, su, a, o, sh, d, st) \in A_x\}| \leq 1 \quad \forall su \in SU : surg(su, sp, sh) = 1, sp \in SP;$
- (c7) $|\{r : (r, su, a, o, sh, d, st) \in A_x, st \leq t < st + \delta(r) \div \text{slot_duration}\}| \leq 1 \quad \forall su \in SU : surg(su, sp, sh) = 1, sp \in SP;$
- (c8) $|\{r : (r, su, a, o, sh, d, st) \in A_x\}| \leq 1 \quad \forall a \in A : an(a, sp, sh) = 1, sp \in SP;$
- (c9) $|\{r : (r, su, a, o, sh, d, st) \in A_x, st \leq t < st + \delta(r) \div \text{slot_duration}\}| \leq 1 \quad \forall a \in A : an(a, sp, sh) = 1, sp \in SP;$

$$(c10) \sum_{(r,su,a,o,sh,d,st) \in A_x} \delta(r) \leq \text{surgWT}(su, d) \quad \forall su \in SU, d \in D;$$

$$(c11) \sum_{(r,su,a,o,sh,d,st) \in A_x} \delta(r) \leq \text{anWT}(a, d) \quad \forall a \in A, d \in D;$$

$$(c12) (c12) \{r : r \in R, p(r) = 1\} \subseteq R_x^*;$$

Condition (c1) ensures that each registration is assigned to a time slot only if it does not exceed the shift duration.

Condition (c2) ensures that each registration is assigned to an OR, surgeon and anaesthetist that are in the same specialty of the registration.

Condition (c3) ensures that each registration is scheduled at most once.

Condition (c4) ensures that two different registrations cannot be scheduled in the same OR, shift and time slot.

Condition (c5) extends condition (c4) to take into account the duration of each registration.

Condition (c6) ensures that a surgeon cannot work at the same time slot and shift in different ORs.

Condition (c7) extends condition (c6) to take into account the duration of each registration.

Conditions (c8) and (c9) are similar to (c6) and (c7), respectively, by considering anaesthetists instead of surgeons.

Condition (c10) (resp. (c11)) ensures that surgeons (resp. anaesthetists) do not exceed their daily number of working hours.

Finally, condition (c12) imposes all priority 1 registrations to be assigned.

Definition 2. (Solution) A solution ψ is a schedule x that satisfies all conditions from (c1) to (c12).

Definition 3. (Unassigned registrations) Given a solution ψ , let $R_\psi^{pr} = \{r \mid r \in R, p(r) = pr, r \notin R_\psi^*\}$. Intuitively, R_ψ^{pr} represents the set of registrations of priority pr that were not assigned to any OR.

Definition 4. (Minimal scheduling solution) A solution ψ is said to dominate solution ψ' if $|R_\psi^2| < |R_{\psi'}^2|$, or if $|R_\psi^2| = |R_{\psi'}^2|$ and $|R_\psi^3| < |R_{\psi'}^3|$. A solution is minimal, if it is not dominated by any other scheduling solutions.

4 ASP Encoding for ORS with Surgical Teams

In this section we present the input predicates and our ASP encoding for representing data and our solution, in two different subsections.

4.1 Data Model

The input data to our model is specified by means of the following atoms:

- Instances of $\text{time}(\mathbf{S}, \mathbf{ST})$ show the time slots (\mathbf{ST}) for each shift (\mathbf{S}), i.e., each shift is divided it into a certain number of time slots, say n . In our case, we have exactly n instances of $\text{time}(\mathbf{S}, \mathbf{ST})$ for each shift, where ST ranges from 1 to n . Note that n is set to $\text{shift_duration} \div \text{slot_duration}$ as described in Sect. 3.

```

(r1) {x(R,P,SR,AN,O,S,D,ST): (ST+SU) <= slots} :- registration(R,P,SU,SP),
    mss(O,S,SP,D), surgeon(SR,SP,S), an(AN,SP,S), time(S,ST).
(r2) :- registration(R,_,_,_), #count{R,SR,AN,O,S,D,ST : x(R,P,SR,AN,O,S,D,ST)}>1.
(r3) :- x(R1,_,_,_,O,S,D,ST), x(R2,_,_,_,O,S,D,ST), R1 != R2.
(r4) :- #count{R:x(R,_,_,_,O,S,_,ST), registration(R,_,SU,_, T)>=ST, T<ST+SU}>1,
    mss(O,S,_,_), time(S,T).
(r5) :- #count{R:x(R,_,_,SR,_,_,S,_,ST)} > 1, surgeon(SR,_,S), time(S,ST).
(r6) :- #count{R:x(R,_,SR,_,_,S,_,ST), registration(R,_,SU,_, T)=ST, T<ST+SU}>1,
    surgeon(SR,_,S), time(S,T).
(r7) :- #count{R:x(R,_,_,AN,_,_,S,_,ST)} > 1, an(AN,_,S), time(S,ST).
(r8) :- #count{R:x(R,_,_,AN,_,_,S,_,ST), registration(R,_,SU,_, T)=ST, T<ST+SU}>1,
    an(AN,_,S), time(S,T).
(r9) :- #sum{SU,R:x(R,_,_,SR,_,_,D,_) } > SWT, surgWT(SWT,SR,D).
(r10) :- #sum{SU,R:x(R,_,_,AN,_,_,D,_) } > AWT, anWT(AWT,AN,D).
(r11) :- #count{R:x(R,1,_,_,_,_,_)} < totRegsP1.
(r12) :- M=#count{R:x(R,2,_,_,_,_,_)}, N=totRegsP2-M. [N@3]
(r13) :- M=#count{R:x(R,3,_,_,_,_,_)}, N=totRegsP3-M. [N@2]

```

Fig. 1. ASP encoding of the ORS problem with surgical teams.

- Instances of `registration(R,P,SU,SP)` represent the registrations, with an identifier (`R`), a priority score (`P`), the duration of the surgery expressed in terms of time slots (`SU`), and the id of the specialty it belongs to (`SP`).
- Instances of `mss(O,S,SP,D)` link each operating room (`O`) to a shift (`S`) for each specialty (`SP`) and planning day (`D`), as established by the MSS.
- Instances of `surgeon(SR,SP,S)` represent the surgeons with an id (`SR`) for each specialty (`SP`) and shift (`S`).
- Instances of `an(AN,SP,S)` show the anaesthetists with an id (`AN`) for each specialty (`SP`) and shift (`S`).
- Instances of `surgWT(SWT,SR,D)` represent the total work time (`SWT`) expressed in time slots for surgeons with id (`SR`) for each day (`D`).
- Instances of `anWT(AWT,AN,D)` represent the total work time (`AWT`) expressed in time slots for anaesthetists with id (`AN`) for each day (`D`).

The output is stored in an assignment to atom of the following form:

$$x(R,P,SR,AN,O,S,D,ST)$$

representing that the registration (`R`) with priority (`P`) is assigned with surgeon id (`SR`) and anaesthetist id (`AN`) to the operating room (`O`) during the shift (`S`) of the day (`D`) with a time slot (`ST`).

4.2 Encoding

The related ASP encoding is shown in Fig. 1, and is described in this subsection. The encoding is based on the Guess&Check programming methodology.

Rule (r_1) guesses an assignment for the registrations, surgeons and anaesthetists to an OR in a given day, shift and with a time slot among the ones permitted by the MSS for the particular specialty the registrations, surgeons and anaesthetists belongs to, such that the registrations assigned with a slot

Table 4. Total number of randomly generated registrations for each benchmark.

Specialty	Registrations				ORs
	5-day	3-day	2-day	1-day	
1	80	48	32	16	3
2	70	42	28	14	2
3	70	42	28	14	2
4	60	36	24	12	1
5	70	42	28	14	2
Total	350	210	140	70	10

time and surgery duration should be less than `slots` of OR, where `slots` represents the total number of slots in the shift. Note that (r_1) encodes conditions $(c1)$ and $(c2)$ thanks to the minimality property of the ASP semantics.

After guessing an assignment for the registrations, the encoding presents constraints from (r_2) to (r_{11}) to discard some unwanted assignments. Note that each constraint r_i ($i = 2..11$) encodes condition (cN) , $N = i+1$.

Finally, weak constraints (r_{12}) and (r_{13}) are used to give preference to registrations having priority 2 over those having priority 3, where `totRegsP2` and `totRegsP3` are constants representing the total number of registrations having priority 2 and 3, respectively.

5 Experimental Results

This section reports about the results of an empirical analysis of the ORS problem with surgical teams. In the first subsection we present the benchmarks we have employed, whose results coupled with our encoded are shown in the second subsection. The third subsection reports about a further analyses focused on anaesthetists WT efficiency.

5.1 Benchmarks

For each scenario, the characteristics of the tests are as follows:

- 4 scenarios for testing the dimension of the slot interval: A, B, C, and D for slot interval of 10, 20, 30, and 60 min, respectively.
- 4 different benchmarks, with a planning period of 1, 2, 3 and 5 working days;
- For each benchmark the total number of randomly generated registrations are 350 for 5 days, 210 for 3 days, 140 for 2 days and 70 for 1 day, respectively;
- 5 specialties;
- 20 surgeons assigned to the 5 specialties;
- 20 anaesthetists assigned to the 5 specialties;
- 4 h of work time in a day for each surgeon;
- 6 h of work time in a day for each anaesthetist;

- 10 ORs distributed among the specialties;
- 5 h morning and afternoon shifts for each OR summing up to 500, 300, 200 and 100 h of OR available time for the four benchmarks, resp..

Table 4 shows the distribution of the total number of randomly generated registrations for each benchmark of 5, 3, 2 and 1 day, for each specialty, together with the distribution of ORs for each specialty.

5.2 Results

Experiments have been run on a HP 630 Notebook with Intel(R) Core(TM) i3 CPU M380@2.53GHz. The ASP system used is CLINGO [17]. Results of the experiments are reported for scenario A in Table 5, for scenario B in Table 6, for scenario C in Table 7 and for scenario D in Table 8, respectively.

Each benchmark was tested 10 times with different randomly generated inputs. A time limit of 300s was set for each experiment. In each table averages for 10 instances for each benchmark are reported. The first three columns show the number of assigned registrations out of the generated ones for each priority P1, P2 and P3, the fourth column shows the cumulative assigned registrations, while the last three columns show a measure of the total time occupied by the assigned registrations as a percentage of the total OR time available (indicated as OR time Eff in the tables) and the total percentage of surgeons and anesthetists working time (indicated as Surg WT Eff and Anest WT Eff in the tables, respectively).

As we can see in scenario A (Table 5), with slot interval of 10 min, we obtain results only for schedules up to 3 days, while in the case of the 5-day benchmark the computation time exceeds our time limit on all instances. Scenario B (Table 6) details the scheduling results with slot interval of 20 min. It can be seen that OR efficiency is 75% while the Surgeons and Anesthetists WT efficiency remain greater than 90% and 60%, respectively, for all benchmarks in this scenario. In scenario C (Table 7) with a slot interval of 30 min, the OR efficiency is around 76% while the Surgeons and Anesthetists WT efficiency are enhanced up to 95% and 63% for all benchmarks, respectively. In scenario D (Table 8) with a slot interval of 60 min, OR efficiency is almost 79% while the Surgeons WT efficiency is further enhanced to more than 95%, and Anesthetists WT efficiency is up to 65%.

In all the evaluated benchmarks for different scenarios we observed that the OR efficiency and the anesthetists WT efficiency are limited by the fact that we reached the ceiling of the surgeons maximum working hours. Considering that in our setup we had one anesthetist for each surgeon, the ratio between anesthetist and surgeon efficiencies coincides to the ratio between their maximum working time of the surgeons and the anesthetists, i.e., $2/3$. In a real application, this would be a useful information for the Hospital manager to quantify the excess of anesthetists and reorganize their numbers or their working times.

Overall, we obtained satisfying results but for the 5-day schedule length for the more fine-grained slot interval of Scenario A. In order to further investigate

the issue, we moved on a different dimension and tested the Scenario A configuration with half the number of registrations (35 instead of 70 for each planning day), surgeons (10 instead of 20), anesthetists (10 instead of 20) and ORs (5 instead of 10). With these numbers we can reach acceptable solutions after 60s of computation time (see Table 9) for every benchmark, including the 5-day one.

Table 5. Averages of the results for 5, 3, 2 and 1 day benchmarks for Scenario A.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	–	–	–	–	–	–	–
3 days	43.1/43.1	53.0/81.6	26.2/85.3	122.3/210.0	73.5%	91.8%	61.2%
2 days	29.9/29.9	37.0/54.7	17.7/55.4	84.6/140.0	74.7%	93.4%	62.3%
1 day	13.4/13.4	22.8/28.0	8.9/28.6	46.1/70.0	75.6%	94.5%	62.9%

Table 6. Averages of the results for 5, 3, 2 and 1 day benchmarks for Scenario B.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	71.2/71.2	99.4/140.1	38.3/138.7	208.9/350.0	75.1%	93.8%	62.5%
3 days	40.9/40.9	61.6/85.3	25.2/83.8	127.7/210.0	75.3%	94.1%	62.8%
2 days	28.2/28.2	41.1/56.1	14.9/55.7	84.2/140.0	75.0%	93.7%	62.5%
1 day	12.5/12.5	23.1/29.7	8.9/27.8	43.5/70.0	75.5%	94.4%	62.9%

5.3 Extended Analysis

In order to improve the anesthetists WT efficiency, we further analysed our solution considering the new setting introduced at the end of the previous analysis, and introducing 3 alternative scenarios wrt number of surgeons and anesthetists:

- Scenario 1: 10 surgeons and 8 anesthetists
- Scenario 2: 10 surgeons and 7 anesthetists
- Scenario 3: 10 surgeons and 5 anesthetists

Overall, for each new scenario, the characteristics of the tests performed that were modified wrt the analysis in the previous subsection are:

- For each benchmark the total number of randomly generated registrations were 175 for 5 days, 105 for 3 days, 70 for 2 days and 35 for 1 day;

Table 7. Averages of the results for 5, 3, 2 and 1 day benchmarks for Scenario C.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	71.9/71.9	99.0/139.8	44.1/138.3	215.0/350.0	76.0%	95.2%	63.3%
3 days	41.7/41.7	66.9/84.8	21.6/83.5	130.2/210.0	76.1%	95.1%	63.5%
2 days	27.9/27.9	42.7/53.8	16.9/58.3	87.5/140.0	76.2%	95.2%	63.5%
1 day	14.2/14.2	23.0/29.4	6.7/26.4	43.9/70.0	76.2%	95.1%	63.5%

Table 8. Averages of the results for 5, 3, 2 and 1 day benchmarks for Scenario D.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	68.7/68.7	109.6/143.8	46.9/137.5	224.8/350.0	79.0%	98.8%	65.8%
3 days	41.8/41.8	65.1/ 81.9	25.5/86.3	132.4/210.0	78.7%	98.4%	65.6%
2 days	27.5/27.5	46.5/ 54.5	14.6/58.0	87.7/140.0	79.2%	98.9%	65.9%
1 day	13.3/13.3	23.1/27.5	8.3/29.2	44.7/70.0	78.3%	97.8%	65.2%

- 10 surgeons assigned to the 5 specialties;
- 10 anaesthetists assigned to the 5 specialties;
- 5 ORs, unevenly distributed among the specialties;

Results of the extended analysis are reported in this section for Scenario 1 in Table 10, for Scenario 2 in Table 11 and for Scenario 3 in Table 12, respectively, organized as Tables 5, 6, 7 and 8. Each benchmark was tested 10 times with different randomly generated inputs with a time limit of 60 s set for each experiment, and averages over 10 instances.

As we can see, in Scenario 1 (Table 10) the OR efficiency is greater than 64% while the Surgeons and Anesthetists WT efficiency remain greater than 80% and 67%, respectively. In Scenario 2 (Table 11) OR efficiency is again between 64% and 68%, with the Surgeons WT efficiency of around 84% and improved Anesthetists WT efficiency up to 80%. Finally, on Scenario 3 (Table 12) the OR efficiency decreases around 50%, while the Surgeons WT efficiency is still from 64% to 67%, and Anesthetists WT efficiency is further enhanced up to 90%.

6 Related Work

In this section we discuss some relevant works related to this research. Meskens et al. [30] considered the surgical teams in the computation of an OR schedule, and developed a model using Constraint Programming (CP) with multiple constraints such as availability, staff preferences and affinities among surgical teams. They optimize the use of ORs by minimizing makespan and maximizing affinities among surgical team members. The effectiveness of their proposed method for improving surgical cases was evaluated using real data from an Hospital. Hamid et al. [29] incorporated the decision-making styles (DMS) of the surgical team

Table 9. Averages of the results for 5, 3, 2 and 1 day benchmarks with slot interval 10 min and reduced number of registrations.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	35.5/35.5	52.3/68.3	17.0/71.2	104.8/175.0	74.4%	93.1%	62.0%
3 days	19.0/19.0	31.9/41.2	12.9/40.8	63.8/105.0	74.4%	93.0%	62.0%
2 days	14.2/14.2	20.8/28.6	6.9/26.3	40.9/70.0	73.0%	91.3%	60.8%
1 day	5.5/5.5	10.4/15.1	3.6/14.4	19.5/35.0	70.7%	89.5%	58.9%

Table 10. Averages of the results for 5, 3, 2 and 1 day benchmarks with 10 surgeons and 8 anesthetists for Scenario 1.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	35.5/35.5	47.0/68.3	13.5/71.2	93.0/175.0	67.6%	84.5%	70.2%
3 days	19.0/19.0	29.3/41.2	10.3/41.2	58.6/105.0	68.1%	85.1%	70.9%
2 days	13.9/13.9	18.8/28.2	5.3/26.9	38.0/70.0	66.8%	83.6%	69.7%
1 day	5.5/5.5	9.8/15.1	2.4/14.4	17.7/35.0	64.3%	80.4%	67.0%

Table 11. Averages of the results for 5, 3, 2 and 1 day benchmarks with 10 surgeons and 7 anesthetists for Scenario 2.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	35.5/35.5	45.2/68.3	14.8/71.2	95.5/175.0	67.3%	84.1%	80.2%
3 days	19.0/19.0	30.3/41.2	9.0/44.8	58.3/105.0	67.6%	84.5%	80.5%
2 days	13.9/13.9	19.1/28.2	4.8/26.9	37.9/70.0	67.2%	84.0%	80.0%
1 day	5.5/5.5	9.8/15.1	2.4/14.4	17.7/35.0	64.3%	80.4%	76.5%

Table 12. Averages of the results for 5, 3, 2 and 1 day benchmarks with 10 surgeons and 5 anesthetists for Scenario 3.

Bench.	P1	P2	P3	Total	OR time Eff.	Surg WT Eff.	Anest WT Eff.
5 days	35.5/35.5	35.3/68.3	9.4/71.2	80.2/175.0	53.6%	67.0%	89.4%
3 days	19.0/19.0	25.2/41.2	3.6/44.8	48.3/105.0	54.3%	67.8%	90.4%
2 days	13.9/13.9	14.6/28.2	2.8/26.9	31.0/70.0	53.4%	66.7%	88.0%
1 day	5.5/5.5	7.8/15.1	1.5/14.3	14.9/35.0	51.2%	64.0%	85.3%

to improve the compatibility level by considering constraints such as the availability of material resources, priorities of patients, and availability, skills, and competencies of the surgical team. They developed a multi-objective mathematical model to schedule surgeries. Two metaheuristics, namely Non-dominated Sorting Genetic Algorithm and Multi-Objective Particle Swarm Optimization,

were developed to find pareto-optimal solutions. Xiang et al. [34] proposed an Ant Colony Optimization (ACO) approach to surgical scheduling taking into account all resources in the entire process of a surgery. The problem was represented as an extended multi-resource constrained flexible job shop scheduling problem, which was solved using a two-level hierarchical graph to integrate sequencing job and allocating resources. To evaluate the efficiency of ACO, a Discrete Event System (DES) model of an OR system was developed in the simulation platform SIMIO. Monteiro et al. [31] developed a comprehensive multi-objective mathematical model using epsilon-constraint method coupled to the CPLEX solver. Vijayakumar et al. [33] used Mixed Integer Programming (MIP) model for multi-day, multi-resource, patient-priority-based surgery scheduling. A First Fit Decreasing algorithm was developed. From a solution time perspective, their model took hours and in most cases was unable to optimally solve the problem. Belkhamisa et al. [7] proposed two meta heuristics, an Iterative Local Search (ILS) approach and Hybrid Genetic Algorithm (HGA) to solve a daily surgery scheduling problem. Zhou et al. [35] developed an Integer Programming model for optimal surgery schedule of assigning patients to different resources in any surgical stage. They used Lagrangian Relaxation algorithm and solved the subproblem by using branch and bound. They verified their model using real data instances from an Hospital. A common issue with all such solutions seem to be computation time and scalability.

About, instead, other scheduling problems in which ASP have been proficiently employed: Nurse Scheduling Problem [3, 4, 16], where the goal is to create a scheduling for nurses working in Hospitals; Team Building Problem [32], where the goal is to allocate the available personnel of a seaport for serving the incoming ships; the Conference Paper Assignment Problem [5], which deals with the problem of assigning reviewers in the PC to submitted conference papers; and scheduling production materials between storage locations and assembly station [20].

Finally, this is an extended and revised version of a paper appearing in [13], with the following main improvements: (i) the mathematical formulation (Sect. 3.2), (ii) the precise definition of the problem (still Sect. 3.2), and (iii) the extended experimental analysis (Sect. 5.3).

7 Conclusions

In this paper we employed ASP for solving ORS problems with surgical teams. The results of our experiments confirm that ASP is a suitable methodology for addressing planning and scheduling problems in healthcare system. We presented the results of an experimental analysis on several directions to check scalability of our solution in terms of efficiency, considering shift duration, surgeons and anaesthetist working hours. This solution achieved satisfied ORs, surgeons' and anaesthetists' efficiency also for the planning length of 5 days. As a future work we would like first to analyze the performance of our solution on real data, that we only recently obtained. We also want to integrate the extension of the

ORS model with beds management with the one presented in this paper, in order to have a more complete unified solution. We also plan to compare to alternative methods, assuming this is possible (i.e., availability of alternative solutions), and viable (i.e., very same problem solved). Finally, through results are satisfying, we plan to work also on improving performance by both evaluating more solvers, e.g., WASP [2], other than Clingo actually used, and employing SAT techniques (e.g., [11,25–28]), given the strong existing relation between ASP and SAT [23,24].

References

1. Abedini, A., Ye, H., Li, W.: Operating room planning under surgery type and priority constraints. *Proc. Manuf.* **5**, 15–25 (2016)
2. Alviano, M., Amendola, G., Dodaro, C., Leone, N., Maratea, M., Ricca, F.: Evaluation of disjunctive programs in WASP. In: Balduccini, M., Lierler, Y., Woltran, S. (eds.) *LPNMR 2019*. LNCS, vol. 11481, pp. 241–255. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20528-7_18
3. Alviano, M., Dodaro, C., Maratea, M.: An advanced answer set programming encoding for nurse scheduling. In: Esposito, F., Basili, R., Ferilli, S., Lisi, F. (eds.) *AI*IA 2017*. LNCS, vol. 10640. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70169-1_35
4. Alviano, M., Dodaro, C., Maratea, M.: Nurse (re)scheduling via answer set programming. *Intelligenza Artificiale* **12**(2), 109–124 (2018)
5. Amendola, G., Dodaro, C., Leone, N., Ricca, F.: On the application of answer set programming to the conference paper assignment problem. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) *AI*IA 2016*. LNCS (LNAI), vol. 10037, pp. 164–178. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49130-1_13
6. Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., Testi, A.: A two level metaheuristic for the operating room scheduling and assignment problem. *Comput. Oper. Res.* **54**, 21–34 (2015)
7. Belkhamza, M., Jarboui, B., Masmoudi, M.: Two metaheuristics for solving no-wait operating room surgery scheduling problem under various resource constraints. *Comput. Ind. Eng.* **126**, 494–506 (2018)
8. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. *Commun. ACM* **54**(12), 92–103 (2011)
9. Calimeri, F., et al.: ASP-Core-2 input language format. *Theory Pract. Logic Program.* **20**(2), 294–309 (2020)
10. Calimeri, F., Gebser, M., Maratea, M., Ricca, F.: The design of the fifth answer set programming competition. *CoRR* abs/1405.3710 (2014). <http://arxiv.org/abs/1405.3710>
11. Di Rosa, E., Giunchiglia, E., Maratea, M.: A new approach for solving satisfiability problems with qualitative preferences. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N.M. (eds.) *ECAI*. *Frontiers in Artificial Intelligence and Applications*, vol. 178, pp. 510–514. IOS Press (2008)
12. Dodaro, C., Galatà, G., Khan, M.K., Maratea, M., Porro, I.: An ASP-based solution for operating room scheduling with beds management. In: Fodor, P., Montali, M., Calvanese, D., Roman, D. (eds.) *RuleML+RR 2019*. LNCS, vol. 11784, pp. 67–81. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31095-0_5

13. Dodaro, C., Galatà, G., Khan, M.K., Maratea, M., Porro, I.: An ASP based solution for operating room scheduling with surgical teams in hospital environments. In: Benedictis, R.D., et al. (eds.) Joint Proceedings of the 8th Italian Workshop on Planning and Scheduling and the 27th International Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion. CEUR Workshop Proceedings, vol. 2745. CEUR-WS.org (2020)
14. Dodaro, C., Galatà, G., Maratea, M., Porro, I.: Operating room scheduling via answer set programming. In: Ghidini, C., Magnini, B., Passerini, A., Traverso, P. (eds.) AI*IA 2018. LNCS (LNAI), vol. 11298, pp. 445–459. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03840-3_33
15. Dodaro, C., Galatà, G., Maratea, M., Porro, I.: An ASP-based framework for operating room scheduling. *Intelligenza Artificiale* **13**(1), 63–77 (2019)
16. Dodaro, C., Maratea, M.: Nurse scheduling via answer set programming. In: Balduccini, M., Janhunen, T. (eds.) LPNMR 2017. LNCS (LNAI), vol. 10377, pp. 301–307. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61660-5_27
17. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with Clingo 5. In: ICLP (Technical Communications). OASICS, vol. 52, pp. 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
18. Gebser, M., Maratea, M., Ricca, F.: The design of the seventh answer set programming competition. In: Balduccini, M., Janhunen, T. (eds.) LPNMR 2017. LNCS (LNAI), vol. 10377, pp. 3–9. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61660-5_1
19. Gebser, M., Maratea, M., Ricca, F.: The seventh answer set programming competition: design and results. *Theory Pract. Log. Program.* **20**(2), 176–204 (2020)
20. Gebser, M., Obermeier, P., Schaub, T., Ratsch-Heitmann, M., Runge, M.: Routing driverless transport vehicles in car assembly with answer set programming. *Theory Pract. Log. Program.* **18**(3–4), 520–534 (2018)
21. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, 15–19 August 1988 (2 Volumes), pp. 1070–1080. MIT Press (1988)
22. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *New Gener. Comput.* **9**(3/4), 365–386 (1991). <https://doi.org/10.1007/BF03037169>
23. Giunchiglia, E., Leone, N., Maratea, M.: On the relation among answer set solvers. *Ann. Math. Artif. Intell.* **53**(1–4), 169–204 (2008). <https://doi.org/10.1007/s10472-009-9113-1>
24. Giunchiglia, E., Maratea, M.: On the relation between answer set and SAT procedures (or, between CMODELS and SMODELS). In: Gabbriellini, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 37–51. Springer, Heidelberg (2005). https://doi.org/10.1007/11562931_6
25. Giunchiglia, E., Maratea, M.: Solving optimization problems with DLL. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 141, pp. 377–381. IOS Press (2006)
26. Giunchiglia, E., Maratea, M., Tacchella, A.: Dependent and independent variables in propositional satisfiability. In: Flesca, S., Greco, S., Ianni, G., Leone, N. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 296–307. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45757-7_25
27. Giunchiglia, E., Maratea, M., Tacchella, A.: (In)Effectiveness of look-ahead techniques in a modern SAT solver. In: Rossi, F. (ed.) CP 2003. LNCS, vol. 2833, pp.

- 842–846. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45193-8_64
28. Giunchiglia, E., Maratea, M., Tacchella, A., Zambonin, D.: Evaluating search heuristics and optimization techniques in propositional satisfiability. In: Goré, R., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS, vol. 2083, pp. 347–363. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45744-5_26
 29. Hamid, M., Nasiri, M.M., Werner, F., Sheikahmadi, F., Zhalechian, M.: Operating room scheduling by considering the decision-making styles of surgical team members: a comprehensive approach. *Comput. Oper. Res.* **108**, 166–181 (2019)
 30. Meskens, N., Duvivier, D., Hanset, A.: Multi-objective operating room scheduling considering desiderata of the surgical team. *Decis. Support Syst.* **55**(2), 650–659 (2013). <https://doi.org/10.1016/j.dss.2012.10.019>
 31. Monteiro, T., Meskens, N., Wang, T.: Surgical scheduling with antagonistic human resource objectives. *Int. J. Prod. Res.* **53**(24), 7434–7449 (2015)
 32. Ricca, F., et al.: Team-building with answer set programming in the Gioia-Tauro seaport. *Theory Pract. Logic Program.* **12**(3), 361–381 (2012)
 33. Vijayakumar, B., Parikh, P.J., Scott, R., Barnes, A., Gallimore, J.: A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *Eur. J. Oper. Res.* **224**(3), 583–591 (2013)
 34. Xiang, W., Yin, J., Lim, G.: An ant colony optimization approach for solving an operating room surgery scheduling problem. *Comput. Ind. Eng.* **85**, 335–345 (2015)
 35. Zhou, B., Yin, M., Lu, Z.: An improved Lagrangian relaxation heuristic for the scheduling problem of operating theatres. *Comput. Ind. Eng.* **101**, 490–503 (2016)