Master Surgical Scheduling via Answer Set Programming Tested on Real Data

Giuseppe Galatà[1], Marco Maratea[2] and Marco Mochi[3]

[1] SurgiQ srl, Genova, Italy
[2] University of Calabria, Rende, Italy
[3] University of Genoa, Genova, Italy

Abstract

The problem of finding a Master Surgical Schedule (MSS) consists of scheduling different specialties to the operating rooms of a hospital clinic. To produce a proper MSS, each specialty must be assigned to some operating rooms. The number of assignments is different for each specialty and can vary during the considered planning horizon. Realizing a satisfying schedule is of upmost importance for a hospital clinic. Recently, a compact solution based on Answer Set Programming (ASP) to the MSS problem has been introduced and tested with satisfying results, but only on synthetic data. In this paper, we also adapt the encoding and test our overall solution on real data from ASL1 Liguria in Italy. The experiments show that our ASP solution provides satisfying results also when tested on real data.

Keywords

Healthcare, Scheduling, Answer Set Programming

1. Introduction

Digital Health, defined as the usage of information and communication technologies in medicine and in the management processes of healthcare, arose several years ago, but has gained increasing importance in recent years. Thanks to new technologies and also due to new challenges such as an aging society, the COVID-19 pandemic and the need to reduce high costs. One of the major problems related to modern hospitals are long waiting lists that reduce patients' satisfaction and the level of care offered to them. The Master Surgical Schedule (MSS) represents which specialty is assigned to each operating room in a particular day and session. The administrative practices of surgical departments on this task can have a large impact on hospital costs, patient outcomes and on the overall efficiency of a hospital. Many papers have analyzed this problem (see for example [1, 2, 3, 4]); in particular, the introduction of an effective MSS led to efficiency gains at the operating room department: At Beatrix hospital the annual budget for operating room hours is reduced from 12,848 hours to 9,972 hours (22.4% reduction) while the patients operated increased by 7.7% in 2007 respect to 2006, using the same capacity as at the same time surgery duration decreases by 9.0% [5]. The MSS is often considered as an already available input in

giuseppe.galata@surgiq.com (G. Galatà); marco.maratea@unical.it (M. Maratea); marco.mochi@edu.unige.it (M. Mochi)
http://www.star.dist.unige.it/~marco/ (M. Maratea); https://www.marcomochi.me (M. Mochi)
0000-0002-1948-4469 (G. Galatà); 0000-0002-9034-2527 (M. Maratea); 0000-0002-5849-3667 (M. Mochi)

many healthcare problem solutions but, due to the different aspects that need to be taken into account for computing a valid schedule, the MSS is an interesting combinatorial problem that deserves its own interest. Going in some more details, the MSS problem is the task of assigning the specialties to the available operating rooms in the different days and sessions, taking into account that not all the specialties need to be assigned the same amount of time and that, during the considered days, the amount of time each specialty should be assigned can vary. The aim of the MSS is to support the hospital to organize the resources and plan the different specialties in the next weeks/months. In particular, by developing a MSS early a hospital can properly manage the personnel and the resources, thus leading to a reduction of the costs. Moreover, by helping the hospital to manage the surgeries and reducing the surgery waiting list, a proper solution to the MSS problem is vital to improve the degree of patients' satisfaction. Complex combinatorial problems, possibly involving optimizations, such as the MSS problem, are usually the target applications of AI languages such as Answer Set Programming (ASP). Indeed ASP, thanks to its readability and the availability of efficient solvers, e.g., CLINGO [6] and WASP [7], that originates from the relation between answer set and propositional satisfiability procedures [8, 9], and driven by ASP competitions (see, e.g., [10]) has been successfully employed for solving hard combinatorial problems in several research areas, and it has been also employed to solve many scheduling problems [11, 12, 13, 14, 15], also in industrial contexts (see, e.g., [16, 17, 18] for detailed descriptions of ASP applications).

In this paper we first present a mathematical formulation of the MSS problem. We then apply ASP to solve the MSS problem, by presenting a compact ASP encoding obtained by modularly representing input specifications in ASP, and then running an experimental analysis on randomly generated MSS benchmarks, obtained by varying the number of days and trying different scenarios [19]. Further, as the main contribution of this work, we adapt the encoding and test our solution on real data from ASL1 Liguria in Italy. Results using the state-of-the-art ASP solver CLINGO show that ASP is a suitable solving methodology also for the MSS problem, since we are able to solve optimally instances of the MSS problem in few seconds even when considering long planning horizon.

The paper is structured as follows. Sections 2 and 3 present an informal description of the MSS problem, and its precise mathematical formulation, respectively. Then, Section 4 shows our ASP encoding, whose experimental evaluation is presented in Section 5. Section 6 adapts and test the solution on real data. The paper ends by discussing related work and conclusions in Section 7 and 8, respectively.

2. Problem Description

With the computation of an MSS, a hospital can see in which days, sessions and operating rooms (ORs) each specialty will do the surgeries. This is important since by looking at the MSS the hospital can manage the personnel and the resources in advance. The MSS is thus often scheduled for long periods of time and as soon as possible, to be able to assign the surgery to the patients in time and to properly organize the personnel. To schedule the MSS a hospital should evaluate the percentage of time that needs to be assigned to each specialty and the allowed errors for such a period of time, in order to better respond to the patients' needs. The percentage of assignments is evaluated as the number of times each specialty is assigned divided by the total number of sessions available in the period considered. To produce a proper schedule, the solution must assign the specialties taking into account the percentage targets and the allowed errors of each specialty. At most n sessions are associated to each day, where n is equal to the maximum number of sessions that could be assigned to an OR. Each session is identified by an id. For example, in a hospital with the maximum number of sessions equal to 2, day 1 will be linked to sessions 1 and 2, while day 2 will be linked to sessions 3 and 4, and so on for all the remaining days. Each session is then linked to all the ORs and the scheduler must assign a specialty to each session. Since the MSS is planned for a long period of time, hospitals could desire that the target assignment of each specialty is respected not for all the considered days, but may vary, e.g., on a monthly or weekly basis. Another aspect that could change during the considered period and between the ORs are the sessions. The usage of each OR is often splitted in two sessions for each day but, sometimes, some ORs can be split in a different number of sessions, higher or used even for just one session. In particular, the single-session solution could be used when a specialty requires particular resources and the time to prepare them is long enough that changing the specialty at mid day would be a waste of time. Moreover, some ORs could be unavailable in some days and the scheduler must be able to consider these unavailability.

Overall, the MSS problem takes as input the number of ORs and specialties, the number of days to consider for the scheduling, the number of sessions for each day, and the different target values for each specialty, and computes the assignment of the different specialties to the available ORs of a hospital in the considered planning horizon. An optimal solution minimizes the difference between the percentage of usage of each specialty and the target value of each period. An example of MSS is presented in Table 1. In particular, the table is the result obtained by our solution, that we will show later in the paper, considering 90 days and fixed target value for each month. Moreover, we considered a hospital with 10 ORs, each splitted in 2 sessions in each day, and 5 specialties (these numbers corresponding to hospitals of small-medium size in Italy) SP1 … SP5 : The table shows the MSS for the first 7 days of the solution. In particular, each row represents a day and the sessions linked to that day, the columns report the ORs, and the intersection shows the specialty assigned to the OR in that day and session.

3. Mathematical formulation of the MSS problem

In this section, we provide a mathematical formulation of the basic version of the problem (called Scenario A later).

Definition 1. *Let*

- *day be a constant that is equal to the number of days considered;*
- *$max\_session$ be a constant that is equal to the maximum number of session associated to an operating room in a day;*
- *s_count be a constant that is equal to $day \times max\_session$ and represents the number of sessions that must be assigned to each operating room;*
- *$D = \{t : t \in [1..day]\}$ be the set of all days;*

Table 1
Example of MSS generated by our solution.

Day	Session	OR1	OR2	OR3	OR4	OR5	OR6	OR7	OR8	OR9	OR10
1	1	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
	2	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
2	3	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
	4	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
3	5	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
	6	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
4	7	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
	8	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
5	9	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP3	SP1	SP4
	10	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP2	SP1	SP4
6	11	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP2	SP1	SP4
	12	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP2	SP1	SP4
7	13	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP2	SP1	SP4
	14	SP5	SP5	SP3	SP3	SP2	SP5	SP4	SP2	SP1	SP4

- $DD = \{(d_1, d_2)_1, \ldots, (d_1, d_2)_n\}$ *be a set of n pair of days such that for every pair* d_2 *is greater than* d_1*;*
- $OR = \{o_1, \ldots, o_m\}$ *be a set of m operating rooms;*
- $SP = \{sp_1, \ldots, sp_k\}$ *be a set of k specialties;*
- $S = \{s_1, \ldots, s_{s\_count}\}$ *be a set of s_count sessions id;*
- $\delta : OR \times SP \mapsto \{0, 1\}$ *be a function associating an operating room to a specialty such that* $\delta(o, sp) = 1$ *if the operating room o can be assigned to the specialty sp, and* 0 *otherwise;*
- $\rho : OR \times D \mapsto S$ *be a function associating an operating room and a day to a session id such that* $\rho(o_n, d_m) \geq max\_session * d_m$ - $(max\_session$-$1)$ *and* $\rho(o_n, d_m) \leq max\_session * d_m$*;*
- $\varepsilon : SP \times D \times D \mapsto \mathbb{N}$ *be a function associating a specialty, a starting day and an ending day to a value representing the percentage target to reach from the starting day to the ending day;*
- $\omega : SP \times D \times D \mapsto \mathbb{N}$ *be a function associating a specialty, a starting day and an ending day to a value representing the maximum error that is allowed from the starting day to the ending day;*
- $\zeta : SP \times D \times D \mapsto \mathbb{N}$ *be a function associating a specialty, a starting day and an ending day to a value representing the percentage of times that a session has been assigned to the specialty.*

Let mss $: OR \times S \times SP\ D \mapsto \{0, 1\}$ *be a function such that* $mss(o, s, sp, d) = 1$ *if the session s in the day d and in the operating room o is assigned to the specialty sp, and* 0 *otherwise. Moreover, for a given mss, let* $A_{mss} = \{(o, s, sp, d) : o \in OR, s \in S, sp \in SP, d \in D, mss(o, s, sp, d) = 1\}$.

Then, given sets OR, SP, S, D, DD and functions δ, ρ, ε, ω, ζ, *the MSS problem is defined as the problem of finding a schedule x, such that*

(c_1*)* $|\{\rho(o, d) = s\}|$=1 $\quad \forall o \in OR, \forall d \in D, \forall s \in S$;

(c_2) $|\{sp : mss(o,s,sp,d) = 1\}|{=}1 \quad \forall o \in OR, \forall s \in S, \forall sp \in SP, \forall d \in D, \rho(o,d) = s;$

(c_3) $|\{mss(o,s,sp,d) = 1\}|{=}0 \quad \forall o \in OR, \forall s \in S, \forall sp \in SP, \forall d \in D, \rho(o,d) = s, \delta(or,sp) = 0;$

(c_4) $|\{mss(o,s,sp,d) = 1\}|{=}0 \quad \forall o \in OR, \forall s \in S, \forall sp \in SP, \forall d \in D, \rho(o,d) \neq s;$

(c_5) $\zeta(sp,d_1,d_2) > 0 \quad \forall sp \in O, \forall (d_1,d_2) \in DD;$

(c_6) $|\varepsilon(sp,d_1,d_2) - \zeta(sp,d_1,d_2)| \leq \omega(sp,d_1,d_2) \quad \forall sp \in O, \forall (d_1,d_2) \in DD;$

Condition (c_1) ensures that at each operating room is assigned to a session s_{count} times. Condition (c_2) ensures that each operating room, in each day and session is assigned to exactly one specialty. Condition (c_3) ensures that no operating room is assigned to a not allowed specialty. Condition (c_4) ensures that each specialty is assigned to an operating room in the right session and day. Condition (c_5) ensures that the percentage of times a specialty is assigned is bigger than 0 in every range of days required. Condition (c_6) ensures that the percentage target of time a specialty is assigned minus the actual percentage is less than the allowed error.

Definition 2 (Distance target percentage). *Given a solution mss, let* $t_{mss} = \sum_{sp \in SP, (d_1,d_2) \in DD;} \mid \varepsilon(sp,d_1,d_2) - \zeta(sp,d_1,d_2) \mid$. *Intuitively,* t_{mss} *represents the sum of the distance between the target percentage and the actual percentage of times each specialty is assigned to the operating rooms in the range between* d_1 *and* d_2*.*

Definition 3 (Optimal solution). *A solution mss is said to dominate a solution mss' if* $|t_{mss}| < |t_{mss'}|$*. A solution is optimal if it is not dominated by any other solution.*

4. ASP Encoding for the MSS problem

We assume the reader is familiar with syntax and semantics of ASP. Starting from the specifications in the previous section, here we present our compact and efficient ASP solution for the MSS problem, organized in two paragraphs containing input and output data model, and the ASP encoding, respectively. The ASP encoding is based on the input language of CLINGO [20]. For details about syntax and semantics of ASP programs we refer the reader to [21].

Data Model. The input data is specified by means of the following atoms:

- Instances of `sessionN(OR,N,DAY)` represent the number of sessions (N) in which the operating room identified by an id (OR) is split in the day (DAY).
- Instances of `operatingRoom(OR,SP)` represent which specialty (SP) can be assigned to the operating room identified by an id (OR).
- Instances of `specialty(SP)` represent the different specialties identified by their id (SP).
- Instances of `targetShare(SP,TARGET,ERROR,START,END)` represent for each specialty (SP) the target percentage (TARGET) of utilization and the maximum distance allowed to the target value (ERROR) in the range of days between START and END.

```
1 session(SID,DAY,OR) :- operatingRoom(OR,_), sessionN(OR,N,DAY), SID=1..s_count, SID >= ((max_session*DAY)-(max_session-1)), SID<=((max_session*DAY)-(max_session-N)), not inactive(OR,DAY).
2 n_session(N,START,END) :- N = #count{SID,OR,DAY : session(SID,OR,DAY), DAY >= START, DAY < END}, targetShare(_,_,_START,END).
3 {mss(OR,SID,SP,DAY) : operatingRoom(OR, SP)} == 1 :- session(SID,DAY,OR).
4 effectiveShare(SP,PERCENTAGE,START,END) :- SESSION = #count{ OR,SID,DAY : mss(OR,SID,SP,DAY), D >= START, D < END}, n_session(N,START,END), specialty(SP), PERCENTAGE = ((SESSION*100) / N).
5 :- effectiveShare(SP,PERCENTAGE,START,END), targetShare(SP,TARGET,ERROR,START,END), PERCENTAGE < (TARGET-ERROR).
6 :- effectiveShare(SP,PERCENTAGE,START,END), targetShare(SP,TARGET,ERROR,START,END), PERCENTAGE > (TARGET+ERROR).
7 :- effectiveShare(SP,PERCENTAGE,START,END), PERCENTAGE <= 0.
8 :~ effectiveShare(SP,ES,START,END), targetShare(SP,TS,ERR,START,END). [|ES-TS|@1,SP,START]
```

Figure 1: ASP encoding of the MSS problem.

- Instances of `day(DAY)` represent the available days.

The output is an assignment represented by an atom of the form `mss(OR,SID,SP,DAY)`, where the intuitive meaning is that the operating room with id `OR` in the session with id `SID` and in the day `DAY` is assigned the specialty `SP`.

Encoding. The related encoding is shown in Figure 1, and is described next. To simplify the description, we denote as r_i the rule appearing at line i of Figure 1.

Auxiliary atoms in the heads of rules r_1, r_2 and, r_4 are derived by the encoder to simplify the other rules. In particular, rule r_1 assigns the correct session ids to each operating room for all the days considered. The assignment is made assigning an id such that the number of ids assigned in each active day is equal to the number of sessions in which the operating room is splitted. Rule r_2 evaluates the total number of sessions available in the range of days between the values start and end. This value is then used to evaluate the percentage of assignment of each specialty. Rule r_3 assigns one of the possible specialties to a session of every operating room. Rule r_4 derives an atom that represents the assignment percentage of each specialty. In particular, it counts the number of sessions linked to each specialty and divides it by the total number of sessions that are available in that period. Then, rules r_5 and r_6 check that the percentage of each specialty is compatible with the target values and the allowed errors. Rule r_7 ensures that the percentage of each specialty is bigger than 0. Finally, weak constraint r_8 minimizes the difference between the assigned and target percentage of each specialty in each period of time.

5. Experimental Results

In this section, we report the results of an empirical analysis of the MSS problem via ASP (second paragraph). For the problem, data have been randomly generated using parameters inspired by literature and real world data (first paragraph).The experiments were run on a AMD Ryzen 5 2600 CPU @ 3.40GHz with 16 GB of physical RAM. The ASP system used was CLINGO [20] 5.4.0, using parameters *--opt-strategy=usc* for faster optimization and *--parallel-mode 4* for parallel execution. This setting is the result of a preliminary analysis done also with other parameters, i.e., the default configuration and the one having *--restart-on-model* for optimization. The time limit was set to 30 seconds. Encodings and benchmarks employed in this section can be found at: https://github.com/MarcoMochi/HC2023mss .

MSS benchmarks. Data are based on the sizes and parameters of a typical middle sized hospital, with 5 different specialties and 10 ORs. Each specialty is associated with a target value for each month and an error, that is equal to 10 for all the specialities. Each specialty can be assigned to just some randomly selected ORs and the target value is assigned by dividing the number of ORs in which the specialty can be assigned to the total number of ORs, and adding to the result a random value in the range between -5 and 5. To test our solution we considered four different scenarios. In the first scenario, that we will call Scenario A, we considered to have the constant *max_session* equal to 2, while the constant *d_count* has values from 30 to 180. Moreover, in this scenario the target value for each specialty is equal for each month. For this scenario, we considered 10 instances, each with different target values for all the specialties, for each range of days considered. In particular, we tested the scalability of the scheduler by considering an increasing number of days: 30, 60, 90, 120, 150 and, 180.

Then, we generated a second scenario, that we will call Scenario B, that is based on the Scenario A considering 90 days. The difference with Scenario A is that for each month the target value is increased or decreased by a random value between -2 and 2, thus for each specialty there are three different target values. Changes in the target values could be done by the hospital manager because of different availability of doctors or due to the increase of the surgeries of some specialty.

For the third and fourth scenario, named Scenario C and D, respectively, we again considered a planning horizon fixed to 90 days. The constant *max_session* is equal to 2 for the Scenario C, while for the Scenario D is equal to 3. This means that, in the fourth scenario, one randomly selected operating room is splitted in three sessions. The difference between the Scenario C and the others is that, for 5 days, three ORs are unavailable, meaning that no session can be assigned to them during that days. The scenarios C and D aim thus at evaluating what is the impact of limiting the usage of the ORs, or changing the number of sessions, respectively.

Results of our MSS solution. First, we tested the performances of the scheduler in the basic scenario (Scenario A). The results for this scenario are shown in Figure 2, which represents the range of seconds required to reach the optimal solution in all the 10 instances tested with the different number of days considered, identified by the minimum and maximum times for solving the instances in the set, together with the mean and the median time. From the figure it can be seen that the scheduler is able to optimally schedule the MSS in a mean time of less than 10 seconds even considering 180 days of planning horizon, which is a remarkable result. Moreover, besides being able to reach an optimal solution in less than 10 seconds on average, from the figure it can be noted that even in the worst case, the scheduler is able to find the optimal solution in less than 30 seconds.

Then, we tested the performance of the scheduler in the Scenario B. Testing the scheduler with the 10 instances with 90 days in this scenario we found that the scheduler was able to reach the

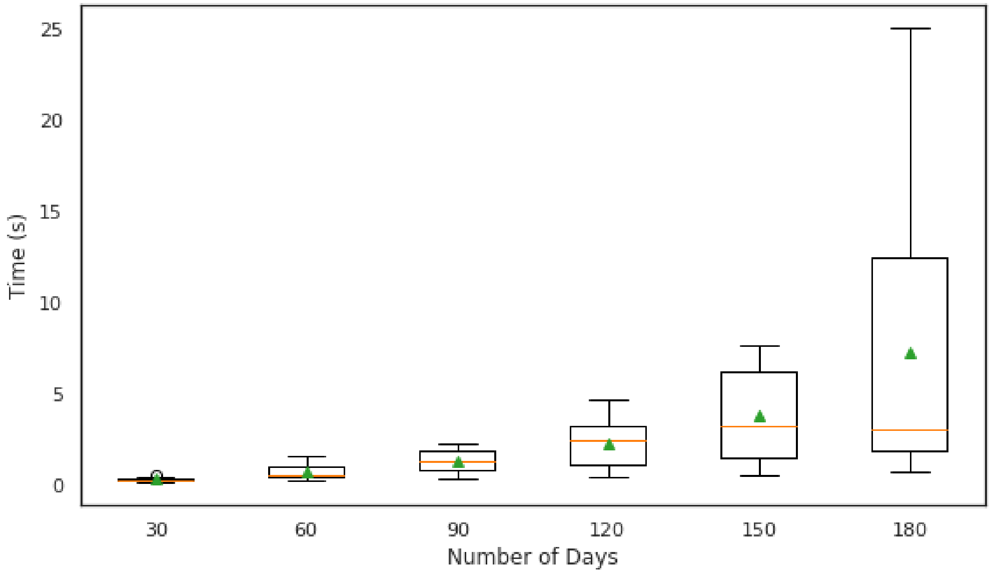

Figure 2: Results obtained by solving 10 instances per group of days in Scenario A. The box starts from the first quartile and ends at the third quartile. The mean time is represented by the (green) triangle, while the (orange) line represents the median value.

optimal solution on average in 3 seconds, that is a time that is very near to the time required in Scenario A. Thus, this analysis reveals that even changing the target values in each month for all the specialties, our solution maintains very good performance.

Having evaluated now the performance in Scenario A and B, we then tested the scheduler in Scenario C and D. Table 2 reports the time required by each instance in Scenario A and in these more constrained scenarios, on 90 days planning horizon.

From the table we can see that the timing obtained by Scenario C is almost equal to the original one. So, even if three ORs are unavailable for 5 days, the scheduler is able to compute the optimal solution in the same time required in the Scenario A. In the Scenario D, the scheduler obtained the optimal solution almost in the same time as in the Scenario A for all but one instance: Indeed, the third instance requires 4 seconds instead of 2 seconds to reach the optimal solution (from a preliminary analysis, this harder instance corresponds to a setting in which a higher number of sessions is set to an OR assigned to only one specialty with low target).

Overall, we can say that also in Scenario C and D the scheduler is able to reach highly satisfying results, also when compared to the basic Scenario A.

6. Adaptation to and results on real data

In this section, we present the results obtained by using real data and the modification done to use the solution with the new data. The section is split into two parts: in the first one we present the results obtained by trying to replicate the same MSS used by the real hospital. In the second one, we present a new encoder that tries to increase the quality of the MSS. Producing an MSS

Table 2
Time required for each instances in the different Scenarios and considering 90 days.

Instance #	Time (s) Scenario A	Time (s) Scenario C	Time (s) Scenario D
1	1.7	1.7	1.7
2	0.3	0.2	0.3
3	1.8	1.1	4.3
4	2.0	3.0	2.0
5	0.9	2.2	0.9
6	1.9	1.0	2.1
7	0.8	0.6	0.6
8	2.2	1.6	2.3
9	0.9	0.8	0.9
10	0.9	5.2	0.8
Mean	1.3	1.7	1.5

```
specialty("IMPERIA CARDIOLOGIA").
operatingRoom("SALA D","IMPERIA CARDIOLOGIA").
operatingRoom("SALA EP","IMPERIA CARDIOLOGIA").
specialty("IMPERIA CHIRURGIA GENERALE").
operatingRoom("SALA A","IMPERIA CHIRURGIA GENERALE").
operatingRoom("SALA B","IMPERIA CHIRURGIA GENERALE").
operatingRoom("SALA C","IMPERIA CHIRURGIA GENERALE").
operatingRoom("SALA D","IMPERIA CHIRURGIA GENERALE").
operatingRoom("SALA E","IMPERIA CHIRURGIA GENERALE").
specialty("IMPERIA CHIRURGIA GENERALE DH SURGERY").
operatingRoom("SALA A","IMPERIA CHIRURGIA GENERALE DH SURGERY").
 ...
```

Figure 3: Example of input derived from the data for Imperia hospital.

that reduces the shift between specialties is crucial to reducing the costs of a hospital since it allows for a reduction in the work needed to prepare the ORs for different specialties. Moreover, keeping the specialties to the same percentage of assignments, if possible, eases the organization of the personnel turnover.

Replicate the results with real data. After having tested our solution in different scenarios with synthetic data in Section 5, we wanted to test it with real data. To accomplish this, we used data from ASL1 Liguria, Italy, already used in [22]. ASL1 is a local health authority consisting of three hospitals: Bordighera, Sanremo, and Imperia. Each hospital has between 2 and 5 ORs and the number of patients visited in a typical month ranges between 100 patients of Bordighera to 500 patients of Imperia. We have made the decision to proceed with the test using data from the hospital of Imperia since it is the hospital with more ORs, specialties, and patients.

In this case, we wanted to assign the ORs to the different specialties of the hospital with the same percentages of assignments. In order to complete these tests, we derived the number of ORs, number of specialties, possible assignments of specialties to ORs, and percentage of assignments of each specialty from the real data. Then, we used this information as input, of which an excerpt obtained from the Imperia hospital can be seen in Figure 3. We tested the solution considering 90

```
9 :- effectiveShare(SP, PERCENTAGE, START, END), needed(SP,TARGET,START,END), PERCENTAGE < TARGET.
10 slack(SP, PERCENTAGE-TARGET, START) :- effectiveShare(SP, PERCENTAGE, START, _), needed(SP,
     TARGET, START, _), PERCENTAGE > TARGET + 10.
11 :~ mss(OR, SID1, SP1, DAY), mss(OR, SID2, SP2, DAY), SID2 > SID1, SP1 != SP2. [1@4, DAY, OR]
12 :~ mss(OR, _, SP, DAY),not mss(OR, _, SP, DAY+7), session( _, DAY+7, OR). [1@3, OR, DAY]
13 :~ effectiveShare(SP, P1, START, _), effectiveShare(SP, P2, START+30, _). [|P1-P2|@2,SP, START]
14 :~ slack(SP, SL, ST). [SL@1, SL, ST]
```

Figure 4: Added rules to the MSS encoding to deal with real data.

days in a different, real, setting where, differently from the synthetic data, the number of ORs is larger than the number of specialties. The solution was able to find an optimal solution in less than a second. This means that the solution managed to assign the different ORs as requested by the hospitals very rapidly and, moreover, the error was equal to 0. This allows us to confirm the goodness of our solution, even on real data.

Additions to consider real data. In the following, we present an addition to the already presented solution that we did to increase the quality of the MSS produced by the hospital and that allows us to derive the percentage of assignments to assign to each specialty. Indeed, in the rest of the work, we consider the target of assignment of the specialties as an input. Here, we want to consider the case in which a hospital wants to derive the assignments considering the number of expected patients and caring about the quality of the MSS. To produce a better MSS, we want to obtain an MSS in which: the ORs are assigned, as much as possible, to the same specialty for all the sessions in a day, every week the specialties are assigned to the same day, and the percentage of time assigned to each specialty every month does not change. Indeed, analyzing the real data, we found that many ORs were assigned to different specialties on consecutive days and, even if not needed, looking at the number of patients, the percentage of time assigned to the different specialties varies a lot during the months.

Starting from the ASL1 data, we derived the number of patients assigned each month for every specialty, considering it as a prediction of future needs, and used this information as a new input for a modified version of the encoder in Figure 1.

Data Model. The input data is the same as the one presented in Section 4 but for the atom `targetShare(SP,TARGET,ERROR,START,END)` that is not used and is replaced by an atom `needed(SP, EXPECTED, START, END)` that represents for each specialty (SP) the expected percentage (EXPECTED) of utilization needed for patients of that specialty (ERROR) in the range of days between START and END. The output is the same as in Section 4.

Encoding. The related encoding is the same as presented in Figure 1 but without the rules appearing in lines 5, 6, and, 7 and the weak constraint in line 8, plus the rules in Figure 4. In the following, we describe such additional rules.

In particular, rule r_9 ensures that the percentage of sessions assigned to each specialty is more than the expected percentage. Rule r_{10} is used to derive the difference between the percentage of sessions assigned to a specialty and the expected percentage for the specialty in which the difference is bigger than a value equal to 10. Weak constraint r_{11} minimizes the assignments of two different specialties to the same OR and on the same day in different sessions. Weak constraint r_{12} minimizes the assignments of two different specialties to the same OR in a one-week time gap. Weak constraint r_{13} minimizes the difference in the percentage of sessions assigned to each specialty each month. Finally, weak constraint r_{14} minimizes the value obtained with rule r_{10}, thus, it reduces the unnecessary assignments.

Testing the solution with the new encoder with the data of the Imperia hospital, we obtained an optimal solution in 5.8 seconds. Thus, the solution is able to obtain an MSS in which for every day, each OR has just one specialty assigned, and every week the specialties are assigned to the same day. Moreover, the difference in the percentage of sessions assigned to each specialty every month is 0, while keeping the value higher than the expected need.

Upon testing the new solution with real data, we can affirm its viability as a valid option for generating a MSS, especially when the hospital prioritizes enhancing the overall quality of the MSS over achieving specific assignments for each specialty.

7. Related Work

The section is organized in two paragraphs: the first presents works that highlights the importance of solving the MSS problem and alternative methods for solving the problem, with a focus on the works using real data. The second paragraph, instead, mentions works in which ASP has been already successfully employed to closely related scheduling problems.

Solving the MSS problem. In [4] is presented a literature review on how different Operations Research techniques can be applied to the surgical planning. Presenting the different approaches to the MSS problem, the authors pointed out that a more efficient MSS can improve the usage of the different resources involved (such as wards, that we do not take into account). Some works were able to use real data to test their solutions; among them, [5] shows the benefit of implementing an effective MSS in a regional hospital in the Netherlands. In particular, thanks to the suggestion of the solution proposed, the hospital was able to reduce the budget while increasing the number of patients operated. In this work, the MSS is evaluated as a cyclic schedule composed of different individual surgical case types. Thus, the MSS is composed by a sequence of surgeries instead of blocks of specialties. Moreover, the MSS is planned for 3 weeks only. In [3], the authors proposed a solution to the MSS problem and the surgical case assignments problem formulating it using a mixed integer nonlinear programming approach. They compared their solutions to the historical data of an Australian public hospital. Differently from our work, the solution proposed by the authors maximizes the number of patients operated instead of focusing on target values required by the hospital. [23] used a mixed integer linear programming model to address the problem. They used the required surgeries of the week to assign the ORs to the different specialties and considered a fixed (two) number of sessions for each day. The tests done in this work are conducted on real data provided by a medium-sized Portuguese private hospital. In [24], the authors addressed the MSS problem by proposing a cyclic schedule for the frequently performed surgical procedures, maximizing the operating room utilization. In this work, the solution was tested with data from the Erasmus Medical Center in

Rotterdam, The Netherlands.

The work in [25] used a simulation-optimization approach to solve the MSS problem. In particular, they used a two-stage stochastic optimization model and a discrete-event simulation model to handle uncertainty such as the surgery duration. They tested the solution with generated synthetic data; further, they did not consider a target value for the different specialties.

Solving scheduling problems in Healthcare with ASP. ASP has been successfully used for solving hard combinatorial and application scheduling problems in several research areas. In the healthcare domain, the first solved problem was the *Nurse Scheduling Problem* [26, 27, 14], where the goal is to create a scheduling for nurses working in hospital units. Then, the problem of assigning ORs to patients, denoted as *Operating Room Scheduling*, has been treated [28], and further extended to include bed management [13]. More recent problems include the *Chemotherepy Treatment Scheduling* problem [29], in which patients are assigned a chair or a bed for their treatments, and the *Rehabilitation Scheduling Problem* [15], which assigns patients to operators in rehabilitation sessions. In both works, real data were used to test the solutions. Often problems in which an MSS needs to be computed, including those dealing with the Operating Room Scheduling problem mentioned above, consider the MSS as an input of the problem; however, as we have seen in this paper and by the presence of a number of works at the state of the art dealing uniquely with the problem, the MSS is per se of interest and deserves devoted solutions, to be possibly integrated with other problem solutions building on it. In [30] and [31], it is proposed a solution to a problem split into two phases. In the former, the problem consists to assign a date to the patients in the first phase and the time for the exams in the second phase. In the latter, the problem consists of assigning a date to a visit or a therapy for multiple recurrent exams to chronic patients. The problem is split into two sub-problems to increase the performance of the solution using Benders' decomposition method.

8. Conclusion

In this paper, we have presented an analysis of the MSS problem modeled and solved with ASP. We started from an informal description of the problem, formulated it in precise mathematical terms, and then presented our ASP solution. Results on synthetic benchmarks show that the ASP solution is able to optimally solve the MSS problem even when considering large planning horizons, up to 6 months. Moreover, solving more difficult scenarios, in which, e.g., targets and number of sessions change within the planning horizon, reduce just slightly the performance of the scheduler. Further, we adapted the encoding and tested our solution on real data from ASL1 Liguria in Italy, still obtaining satisfying results. Results also demonstrate that our solution is more targeted towards increasing the quality of the MSS compared to the original schedules on the real data. For what concerns future works, we plan to investigate rescheduling solutions, that may come into play when the MSS scheduling can not be implemented for some reasons, e.g., sudden unavailability of ORs. Finally, we plan to compare our solution to other logic-based formalisms: Preliminary analysis in [32] has shown that the ASP solution described in Section 4, compared to other logic-based formalisms via automatic translations of ASP instances, was able to reach the optimal solution in less time than the other formalisms such as MaxSAT solvers. We plan to confirm these results on the improved solution presented in Section 6 concerning real data.

References

[1] C. Van Riet, E. Demeulemeester, Trade-offs in operating room planning for electives and emergencies: A review, Operations Research for Health Care 7 (2015) 52–69. doi:https://doi.org/10.1016/j.orhc.2015.05.005, proc. of ORAHS 2014.

[2] Y. B. Ferrand, M. J. Magazine, U. S. Rao, Managing operating room efficiency and responsiveness for emergency and elective surgeries—a literature survey, IIE Transactions on Healthcare Systems Engineering 4 (2014) 49–64. arXiv:https://doi.org/10.1080/19488300.2014.881440.

[3] B. Spratt, E. Kozan, Waiting list management through master surgical schedules: A case study, Operations Research for Health Care 10 (2016) 49–64. URL: https://www.sciencedirect.com/science/article/pii/S2211692316300042. doi:https://doi.org/10.1016/j.orhc.2016.07.002.

[4] G. Francesca, R. Guido, Operational research in the management of the operating theatre: a survey., Health care management science 14,1 (2001) 89–114. doi:doi:10.1007/s10729-010-9143-6.

[5] van Oostrum, Jeroen, Applying Mathematical Models to Surgical Patient Planning, Ph.D. thesis, E, 2009. URL: http://hdl.handle.net/1765/16728.

[6] M. Gebser, B. Kaufmann, T. Schaub, Conflict-driven answer set solving: From theory to practice, Artificial Intelligence 187 (2012) 52–89.

[7] M. Alviano, G. Amendola, C. Dodaro, N. Leone, M. Maratea, F. Ricca, Evaluation of disjunctive programs in WASP, in: M. Balduccini, Y. Lierler, S. Woltran (Eds.), LPNMR, volume 11481 of *LNCS*, Springer, 2019, pp. 241–255.

[8] E. Giunchiglia, M. Maratea, On the Relation Between Answer Set and SAT Procedures (or, Between cmodels and smodels), in: ICLP, volume 3668 of *LNCS*, Springer, 2005, pp. 37–51.

[9] E. Giunchiglia, N. Leone, M. Maratea, On the relation among answer set solvers, Ann. Math. Artif. Intell. 53 (2008) 169–204.

[10] F. Calimeri, M. Gebser, M. Maratea, F. Ricca, The design of the fifth answer set programming competition, CoRR abs/1405.3710 (2014). URL: http://arxiv.org/abs/1405.3710. arXiv:1405.3710.

[11] F. Ricca, G. Grasso, M. Alviano, M. Manna, V. Lio, S. Iiritano, N. Leone, Team-building with answer set programming in the Gioia-Tauro seaport, Theory and Practice of Logic Programming 12 (2012) 361–381.

[12] D. Abels, J. Jordi, M. Ostrowski, T. Schaub, A. Toletti, P. Wanko, Train scheduling with hybrid ASP, in: LPNMR, volume 11481 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 3–17.

[13] C. Dodaro, G. Galatà, M. K. Khan, M. Maratea, I. Porro, An ASP-based solution for operating room scheduling with beds management, in: P. Fodor, M. Montali, D. Calvanese, D. Roman (Eds.), Proceedings of the Third International Joint Conference on Rules and

Reasoning (RuleML+RR 2019), volume 11784 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 67–81.
[14] M. Alviano, C. Dodaro, M. Maratea, Nurse (re)scheduling via answer set programming, Intelligenza Artificiale 12 (2018) 109–124.
[15] M. Cardellini, P. D. Nardi, C. Dodaro, G. Galatà, A. Giardini, M. Maratea, I. Porro, A two-phase ASP encoding for solving rehabilitation scheduling, in: S. Moschoyiannis, R. Peñaloza, J. Vanthienen, A. Soylu, D. Roman (Eds.), Proceedings of the 5th International Joint Conference on Rules and Reasoning (RuleML+RR 2021), volume 12851 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 111–125.
[16] E. Erdem, M. Gelfond, N. Leone, Applications of answer set programming, AI Magazine 37 (2016) 53–68.
[17] A. A. Falkner, G. Friedrich, K. Schekotihin, R. Taupe, E. C. Teppan, Industrial applications of answer set programming, Künstliche Intelligenz 32 (2018) 165–176.
[18] P. Schüller, Answer set programming in linguistics, Künstliche Intelligence 32 (2018) 151–155.
[19] L. Cadermatori, G. Galatà, C. L. Monaco, M. Maratea, M. Mochi, M. Schouten, An asp-based approach to master surgical scheduling, in: R. Calegari, G. Ciatto, A. Omicini (Eds.), Proceedings of the 37th Italian Conference on Computational Logic, Bologna, Italy, June 29 - July 1, 2022, volume 3204 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 313–328. URL: https://ceur-ws.org/Vol-3204/paper_33.pdf.
[20] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, P. Wanko, Theory solving made easy with clingo 5, in: ICLP (Technical Communications), volume 52 of *OASICS*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, pp. 2:1–2:15.
[21] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, T. Schaub, ASP-Core-2 input language format, Theory and Practice of Logic Programming 20 (2020) 294–309.
[22] M. Scanu, M. Mochi, C. Dodaro, G. Galatà, M. Maratea, Operating room scheduling via answer set programming: The case of ASL1 liguria, in: CILC 2023, volume 3428 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023.
[23] I. Marques, M. E. Captivo, N. Barros, Optimizing the master surgery schedule in a private hospital, Operations Research for Health Care 20 (2019) 11–24. URL: https://www.sciencedirect.com/science/article/pii/S2211692318300225.
[24] J. van Oostrum, M. van Houdenhoven, J. Hurink, E. Hans, G. Wullink, G. Kazemier, A master surgical scheduling approach for cyclic scheduling in operating room departments, OR Spectrum = OR Spektrum 30 (2008) 355–374. doi:10.1007/s00291-006-0068-x.
[25] T. R. Bovim, M. Christiansen, A. N. Gullhav, T. M. Range, L. Hellemo, Stochastic master surgery scheduling, European Journal of Operational Research 285 (2020) 695–711.
[26] C. Dodaro, M. Maratea, Nurse scheduling via answer set programming, in: LPNMR, volume 10377 of *LNCS*, Springer, 2017, pp. 301–307.
[27] M. Alviano, C. Dodaro, M. Maratea, An advanced answer set programming encoding for nurse scheduling, in: AI*IA, volume 10640 of *LNCS*, Springer, 2017, pp. 468–482.
[28] C. Dodaro, G. Galatà, M. Maratea, I. Porro, Operating room scheduling via answer set programming, in: AI*IA, volume 11298 of *LNCS*, Springer, 2018, pp. 445–459.
[29] C. Dodaro, G. Galatà, A. Grioni, M. Maratea, M. Mochi, I. Porro, An ASP-based solu-

tion to the chemotherapy treatment scheduling problem, Theory and Practice of Logic Programming 21 (2021) 835–851.
[30] S. Caruso, G. Galatà, M. Maratea, M. Mochi, I. Porro, Scheduling pre-operative assessment clinic with answer set programming, Journal of Logic and Computation (2023). URL: https://doi.org/10.1093/logcom/exad017. doi:10.1093/logcom/exad017, exad017.
[31] P. Cappanera, M. Gavanelli, M. Nonato, M. Roma, Decomposition approaches for scheduling chronic outpatients' clinical pathways in Answer Set Programming, Journal of Logic and Computation (2023) exad038. URL: https://doi.org/10.1093/logcom/exad038. doi:10.1093/logcom/exad038.
[32] M. Mochi, G. Galatà, M. Maratea, Master Surgical Scheduling via Answer Set Programming, Journal of Logic and Computation (2023) exad035. doi:10.1093/logcom/exad035.