Answer Set Programming in Healthcare: Extended Overview*

Mario Alviano[1], Riccardo Bertolucci[2], Matteo Cardellini[2], Carmine Dodaro[1], Giuseppe Galatà[3]**, Muhammad Kamran Khan[2], Marco Maratea[2], Marco Mochi[2,3], Victoria Morozan[2], Ivan Porro[3], and Marco Schouten[2]

[1] DEMACS, University of Calabria, Rende, Italy,
{alviano, dodaro}@mat.unical.it
[2] DIBRIS, University of Genova, Genova, Italy
riccardo.bertolucci@unige.it, matteo.cardellini@edu.unige.it, muhammad.kamrankhan@unige.it,marco.mochi@edu.unige.it,marco.maratea@unige.it, victoria.morozan@edu.unige.it,marco.shouten@edu.unige.it
[3] SurgiQ srl, Italy
{name.surname}@surgiq.com

Abstract. The ASP methodology has been recognized to be a viable solution to many practical applications, including scheduling problems, and in particular in the healthcare domain, of high interest in this period given the COVID-19 pandemic, where ASP proved to be an effective solution to some interesting problems.
In this paper we present an overview of scheduling problems in the healthcare domain that have been successfully solved via ASP in the last two years. Starting from two "basic" problems, i.e., referred to the scheduling of operating rooms in presence of scarce resources (e.g., ICU beds) and the Nurse Scheduling problem, we show how we have improved such solutions, and mention further problems we have dealt with, including the Chemotherapy Scheduling problem. On top of all problems, we are working on providing explainability features, of utmost importance in this field, that for the moment are being instantiated on a single problem.

1 Introduction

Answer Set Programming (ASP) [36, 37] has been recognized to be a viable methodology for solving many practical applications, e.g., Artificial Intelligence [9], Bioinformatics [27, 43], Hydroinformatics [29], and Databases [48]; more recently, ASP has been applied to solve industrial applications [3, 24, 25]. Indeed, the simple syntax [12] and the intuitive semantics [37], combined with the availability of robust implementations (see, e.g. [5, 30, 47, 46]) put forward by the ASP Competition series (see, e.g. [13, 32, 33, 14, 35, 34, 45, 31]), make ASP an ideal candidate for addressing combinatorial problems that naturally arise in these contexts.

** Corresponding author.

Recently, ASP has been also applied to solving scheduling problems, and in particular problems in the healthcare domain, of high interest in this period given the COVID-19 pandemic.

In this paper we present an overview of scheduling problems in the healthcare domain that have been successfully solved via ASP in the last two years (mainly) in Genova, and from this year in the context of the Scheduling and Resource Management topic of the CLAIRE COVID-19 Task Force[4]. The starting point are two basic problems that we have overviewed in [22], namely the Nurse Scheduling problem [10, 16] and the Operating Room Scheduling (ORS) problem [2, 8, 44, 49] . The first refers to the organization of the working shifts of nurses in an hospital unit, while the second refers to the scheduling of operating rooms in presence of scarce resources (e.g., ICU beds). We describe both the improvements we have designed and implemented on these two basic problems, as well as other scheduling problems we have solved, i.e., the Chemotherapy Scheduling problem. The description of the problems takes into account real requirements for small-medium sized Italian Hospitals or private institutes, and the resulting ASP encodings have been tested on instances with realistic sizes and parameters. On top of all problems, we are working on providing explainability features, of utmost importance in this field, that for the moment are being instantiated on the ORS problem.

The paper is structured as follows. Section 2 introduces our starting problems, while Section 3 describes the improvements we made over these problems. Then, Section 4.2 outlines further problems we have solved, and Section 5 concludes the paper and envisages directions for future research.

2 Description of the two basic problems

In this section we briefly describe our two starting problems, outlined in [22].

2.1 Nurse scheduling problem

The Nurse Scheduling problem (NSP) consists of generating a schedule of working and rest days for nurses working in hospital units. The schedule should determine the shift assignments of nurses for a predetermined window of time, and must satisfy requirements imposed by the Rules of Procedure of hospitals. Our NSP definition contains requirements from the International Nurse Rostering Competition [15]. A proper solution to the NSP is crucial to guarantee the high level of quality of healthcare, to improve the degree of satisfaction of nurses and the recruitment of qualified personnel.

In particular, three different types of requirements have been considered, namely *hospital*, *nurses*, and *balance requirements*.

Hospital requirements include the different types of shifts that can be considered, namely *morning* (7 A.M. – 2 P.M.), *afternoon* (2 P.M. – 9 P.M.), and *night*

[4] `https://covid19.claire-ai.org/`.

(9 P.M. – 7 A.M.). In order to ensure the best assistance program for patients, each shift is associated with a minimum and a maximum number of nurses that must be present in the hospital.

Nurses requirements are expressed to guarantee a fair workload between nurses. Therefore, a limit on the minimum and maximum number of working hours per year is imposed. Moreover, additional requirements are imposed to ensure an adequate rest period to each nurse: *(a)* nurses are legally guaranteed 30 days of paid vacation, *(b)* the starting time of a shift must be at least 24 hours later than the starting time of the previous shift, and *(c)* each nurse has at least two rest days each fourteen days window. In addition, after two consecutive working nights there is one special rest day which is not included in the rest days of *(c)*.

Finally, balance requirements ensure that the number of times a nurse can be assigned to morning, afternoon and night shifts is fixed.

Solutions for this problem can be found in [1, 26, 6].

2.2 Operating room scheduling problem

In the following we will call a registration each single planned surgery inserted in the hospital waiting list. A registration is associated to a patient and is characterized by a predicted surgery duration and length of stay (LOS) in the hospital ward, it is assigned to a specialty (e.g. General Surgery, Orthopedics, etc.) and has a priority score, which takes into account two different factors: the surgical procedure urgency and the time already spent in the waiting list. We have classified each registration according to three different priority categories, namely P_1, P_2 and P_3. The first one gathers either very urgent registrations or the ones that have been longer in the waiting list; these registrations must be assigned to the OR schedule. The P_2 registrations should be assigned but can be postponed in case of necessity, while the last category collects the registrations that may be used to fill any possible hole left in the schedule.

The schedule is organized in a series of OR time blocks, uniquely identified by the OR id and the time and date when the block is scheduled. The number and distribution of the OR blocks available for each specialty during the whole planning period is given by the cyclic timetable of the hospital, referred to as the Master Surgical Schedule (MSS) and set beforehand by the hospital management.

The overall goal of our formulation of the ORS problem is to assign the maximum number of registrations, subject to the following constraints:

- each registration assigned to a OR block must belong to the same specialty associated to the block,
- each registration must be assigned to a single block and to a bed of the specialty associated to the block by the MSS for all the LOS,
- the sum of the surgery durations of the registrations assigned to a OR block must not exceed the length of the block itself,

- the registrations belonging to the priority category P_1 must all be assigned, while the number of unassigned registrations belonging to the other categories must be minimized, prioritizing the P_2 ones.

A first ASP solution for this problem can be found in [21].

3 Improvements to the basic problems

In this section we will describe, in two subsections, the improvements that have been made, or are currently undertaken, to the problems outlined in Section 2 in the last couple of years.

3.1 Improvements to NSP

Employee scheduling problem. The NSP has been generalized to consider every employee type needed to operate an hospital or a clinic, such as kitchen staff, cleaning staff, nurses and unlicensed assistive personnel. All staff should be appropriately allocated to provide high quality and efficient health services. Effective employee scheduling considers factors such as patient needs, staff needs, organisational needs, the workforce and skills required to deliver services, and workforce availability.

The automatically generated staff schedule must take into consideration employees working restrictions, the required skill-mix, safe staffing hours, employees agreed clinical unavailability and, when feasible, staff preferences and personalised working patterns. In particular, according to the constraints we considered, each employee:

- is entitled to one day off every N days (where N is customizable for each employee type),
- can only be qualified to perform certain shifts (e.g., only in the morning and afternoon but not at night),
- cannot work more than two consecutive night shifts,
- after each night shift, he/she can be assigned either another night shift or a rest,
- can express preferences on the shifts to be assigned (e.g., an operator can be qualified for all shifts but would prefer to avoid the nights),
- can request holidays, that may be expressed either as preferences, with no guarantee that the requested dates are fully respected, or as constraints.

In order to cover the situations where the employees of more than one facility needs to be scheduled, and some of these employees can be moved between different facilities depending on their daily needs, our solution supports a two-phase scheduling procedure. The first phase of scheduling deals with the allocation of shifts of employees working permanently in a single structure, covering one month (but several months can be scheduled in sequence). Typically not all shifts are expected to be covered at this stage. In the second phase, the shifts left uncovered in the first phase are assigned to the most mobile operators, who can move from facility to facility as required. For each facility the classic shifts in the morning, afternoon and night are considered, with some variations based on the different possible contracts.

Rescheduling procedure for NSP. The Nurse Rescheduling Problem (NRP) addresses situations where an already computed schedule is not usable because of sudden absences of some nurses. Nurses are following a previously computed schedule, and some of them report impossibility to work on some future days, for example because of health problems or personal issues. The previously computed schedule has to be changed starting from a future day that must not follow any reported absence.

The new schedule needs to satisfy all requirements of the NSP with the exception that any absence due to health problems must not be rescheduled. Additionally, the new schedule minimizes the differences with the previous schedule, and such a minimization has priority over any other optimality NSP requirements.

A solution for the NRP can be found in Section 6 of [7], where in particular the situation mentioned earlier is represented with a new atom representing that a nurse is absent for an interval of days for a given reason.

3.2 Improvements to ORS

Solution to the MSS problem. In Section 2.2 we considered the MSS as set beforehand by the hospital management, so among the inputs of the problem and encoding. However, one may want to have the control over it (the hospital management) or to have the possibility to tune and modify it. For these reasons, we have defined an encoding for computing the MSS. Starting from the set of available ORs, the available sessions and their durations, the specialties and a planning horizon, we have defined an encoding for computing a MSS for that planning horizon. The ASP encoding devoted to the MSS computation can be unified to the ORS encoding in a modular way, given that the output (predicate) of the MSS encoding has been designed for corresponding to the input predicate of the ORS encoding.

ORS with ward and ICU beds. The ORS problem is extended with the task of assigning maximum patients to operating rooms, taking into account different specialties, surgery durations, and the availability of beds for the entire length of stay (LOS) both in the Intensive Care Unit (ICU) and in the specialty wards. Given that patients may have priorities, the solution has to find an accommodation for the patients with highest priorities, and then to the other with lower priorities, if space is still available, at the same time taking into proper account beds availability.
The overall goal of the ORS with beds management is to assign the maximum number of registrations to the ORs, taking explicitly into consideration the availability of beds in the wards and in the ICU. We have ensured that a particular registration can only be assigned to an OR only if there is an available bed for

the patient in the ward for the entire LOS.
In our model, a patient's LOS has been subdivided in the following phases:

- a LOS in the ward before surgery: It is possible that a patient is admitted to the ward a day (or more) before the surgery takes place;
- the LOS after surgery, It considers the duration of patient stay after surgery which can be further subdivided into the ICU LOS (duration of stay in ICU) and the following ward LOS (duration of stay in the specialty ward).

Following two requirements related to the beds management are considered:

- *ward bed requirement.* Here we consider a situation where each specialty is linked to a ward with a variable number of available beds exclusively dedicated to the patients associated to that specialty.
- *ICU bed requirement.* Here we consider ICU as a particular type of ward that is accessible to patients from any specialty. However, only a small percentage of patients is expected to need to stay in the ICU.

A solution to this problem can be found in [19].

ORS with PACU beds. Details on the treatment of beds have been explained in the previous paragraph. In this paragraph we introduce another type of beds we have analyzed. Similarly as above, we have also extended the ORS encoding in order to take into account also Post Anesthetic Care Unit (PACU) beds. After a surgery, (fortunately) not all patients need an ICU bed, but a (vast) majority need a PACU bed, where they can recover from the anesthetic. Patients that need a bed go either in PACU or in ICU. These PACU beds are separated to those of the ward and the ICU, and the characteristic of their usage is that they are occupied by a patients for a limited (in minutes) amount of time, often related to a percentage of the surgery duration. The requirements and constraints entailed by the presence of PACU are similar to those of ICU, and have been added to the previous encoding in a modular way, while ensuring that a patient may go in PACU or ICU alternatively.

Rescheduling procedure for ORS. Similarly to the NSP problem, but for slightly different reasons, it is important to design a rescheduling procedure also in case of the ORS problem. Also in this case, the rescheduling procedure is applied to a previously planned schedule, i.e., we start from an already created schedule that could not be executed fully till the end due to some reasons, e.g., some patients could not be operated in their assigned slots or the patients may delete their registration. In such situation all those postponed registrations (or surgeries) must be reallocated to one of the next slots in the remaining part of the original planning period. Once planned, a speciality schedule does not influence other specialties so it makes sense to reschedule one specialty at a time. Since we already have the initial schedule for the planning period, we assume that in day 2 (out of a planning period of 5 days) a number of registrations from a specialty had to be postponed to a next day. In order to insert the postponed registrations in the new schedule we have to make sure that the start of the schedule leaves enough available OR time by automatically dropping the necessary registrations from the old schedule. The encoding for the rescheduling procedure has been defined starting from the ORS encoding, substituting the old schedule (considered now as input of the problem) with the new to be computed, and modularly adding the necessary rules for representing the wanted rescheduling behavior.

4 New problems and explainability

In this section we present, in two separate subsections, a new scheduling problem we have solved with ASP, and explainability features we are adding to one of our solutions.

4.1 Chemotherapy Scheduling problem

The Chemotherapy Treatment Scheduling (CTS) problem consists of generating a calendar of treatment following a chemotherapy regimen for every patient.
We call registration every single treatment inserted on the waiting list. A registration is associated with a patient and a sequential number, since every patient will follow a chemotherapy regimen and the sequential number represents the order to follow. The registration is characterized by the drugs that the patient needs, the days to wait for the next treatment, i.e., the registration associated with the same patient and the next sequential number, and has a priority score with values from 1 to 3.
Each drug has a different time of administration, represented as time slot in our solution, and dose. Every treatment needs a nurse and a chair; the number of nurses, chairs, and, the times slot in which the treatments can take place are set beforehand according to the hospital or clinic availability.
The goal of our solution to the CTS problem is to schedule all registrations following some constraints:

- each treatment must be separated from the previous treatment of the same person by the number of days indicated in the registration;
- each registration must be assigned to an available chair and to an available nurse;
- each patient to which more drugs are assigned in a treatment has to stay in the same chair during all the time of the treatment;
- the sum of a drug used during a day must not exceed the available quantity of that drug;
- each patient with a priority 1 must have the first treatment before a specific day n, such as the one with priority 2 that must have the first treatment before a specific day $n + m$ and the one with priority 3 must have the first treatment before the end of the planning days considered.

A solution based on ASP can be found in [20], and it is quite general to as to possibly include other treatment with relatively low effort, like the radiology treatment [11, 17, 28].

4.2 Explainability

Explainable Artificial Intelligence (AI) refers to methods and techniques in the application of AI methodology such that the results of the solution can be understood by humans, also specifically for the medical domain [42]. We are working on adding explainability features to our ORS solution. As an example, it may happen that no feasible solutions are available for the given set of inputs. Therefore, there is the necessity to:

- investigate which elements of the input set led to the unsatisfiability of the problem (UIE),
- explain in human language these causes in order to make them comprehensible to a large amount of users.

In order to isolate those elements we analyze the registrations in the waiting list with the highest priority, since the lower priority registrations can not lead to unsatisfiability, the availability of the beds and the OR time blocks. Once the UIEs are isolated, it is possible to track the inconsistencies that led to the unsatisfiability, e.g., if it is found that the issue is due to the number of beds of a certain speciality we may state that the issue is generated because the number of available beds in that speciality is too low or the registrations for that speciality is too high. Once the UIEs and their causes have been identified it is possible to proceed with the explanation in human language of the causes of the unsatisfiability. Moreover, we use the gathered data to isolate elements of interest (EI) that are used as suggestions for the user to remove the UIE; e.g., if the issue is due to the lack of beds in a certain speciality, we can give as EI the list of all the registrations with high priority belonging to that speciality. Furthermore, it is given to the user the possibility to remove from the input set any EI making, the user an integral part of the error identification process. Thus, the user perceive the explanation as more reliable since he took part to the identification process.

5 Conclusions and future research

In this document we have overviewed the extended and new contributions to solvoling scheduling problems in the healthcare domain we carried out in the last couple of years mainly in Genova, by employing the ASP methodology. As a result, we can say that in all studied problems up to now ASP led to satisfying solutions, both on the scientific and application view-points.

In the future we would like to continue the effort in the following four directions: (*i*) add new problems, e.g., at the moment we are working on planning and scheduling the pre-surgery phase, (*ii*) deepen the integration of the solutions for the various problems in order to be able to solve bigger and more concrete problems, (*iii*) confront our solutions to others available in the state of the art, whenever possible and meaningful, and (*iv*) turn our ASP encodings into usable applications, by allowing, e.g., to set parameters and run the encoding via a web interface, as we have already done for the basic ORS problem [23]. Moreover, even if we did not give details of the performance of our solution, there are overall satisfying from a computational and application viewpoint. Nonetheless, we plan to work also on this direction by both evaluating more solvers, e.g., WASP [4], other than Clingo actually used, and employing SAT techniques (e.g., [40, 41, 18]), given the strong relation between ASP and SAT [39, 38].

References

1. Abdennadher, S., Schlenker, H.: Nurse scheduling using constraint logic programming. In: Hendler, J., Subramanian, D. (eds.) Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA. pp. 838–843. AAAI Press / The MIT Press (1999)
2. Abedini, A., Ye, H., Li, W.: Operating Room Planning under Surgery Type and Priority Constraints. Procedia Manufacturing **5**, 15–25 (2016)
3. Abseher, M., Gebser, M., Musliu, N., Schaub, T., Woltran, S.: Shift design with answer set programming. Fundamenta Informaticae **147**(1), 1–25 (2016)
4. 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. Lecture Notes in Computer Science, vol. 11481, pp. 241–255. Springer (2019)
5. Alviano, M., Dodaro, C.: Anytime answer set optimization via unsatisfiable core shrinking. Theory and Practice of Logic Programming **16**(5-6), 533–551 (2016)
6. Alviano, M., Dodaro, C., Maratea, M.: An advanced answer set programming encoding for nurse scheduling. In: AI*IA. LNCS, vol. 10640, pp. 468–482. Springer (2017)
7. Alviano, M., Dodaro, C., Maratea, M.: Nurse (re)scheduling via answer set programming. Intelligenza Artificiale **12**(2), 109–124 (2018)
8. Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., Testi, A.: A two level metaheuristic for the operating room scheduling and assignment problem. Computers & Operations Research **54**, 21–34 (2015)
9. Balduccini, M., Gelfond, M., Watson, R., Nogueira, M.: The USA-Advisor: A case study in answer set planning. In: LPNMR. LNCS, vol. 2173, pp. 439–442. Springer (2001)
10. Burke, E.K., Causmaecker, P.D., Berghe, G.V., Landeghem, H.V.: The state of the art of nurse rostering. J. Scheduling **7**(6), 441–499 (2004)
11. Burke, E.K., Rocha, P.L., Petrovic, S.: An integer linear programming model for the radiotherapy treatment scheduling problem. CoRR **abs/1103.3391** (2011), `http://arxiv.org/abs/1103.3391`
12. Calimeri, F., Faber, W., Gebser, M., Ianni, G., Kaminski, R., Krennwallner, T., Leone, N., Maratea, M., Ricca, F., Schaub, T.: ASP-Core-2 input language format. Theory and Practice of Logic Programming **20**(2), 294–309 (2020)
13. 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`
14. Calimeri, F., Gebser, M., Maratea, M., Ricca, F.: Design and results of the Fifth Answer Set Programming Competition. Artificial Intelligence **231**, 151–181 (2016)

15. Ceschia, S., Dang, N., Causmaecker, P.D., Haspeslagh, S., Schaerf, A.: The second international nurse rostering competition. Ann. Oper. Res. **274**(1-2), 171–186 (2019)
16. Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems - a bibliographic survey. European Journal of Operational Research **151**(3), 447–460 (2003)
17. Conforti, D., Guerriero, F., Guido, R.: Non-block scheduling with priority for radiotherapy treatments. Eur. J. Oper. Res. **201**(1), 289–296 (2010)
18. 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)
19. Dodaro, C., Galatà, G., Khan, M.K., Maratea, M., Porro, I.: An asp-based solution for operating room scheduling with beds management. In: RuleML+RR. Lecture Notes in Computer Science, vol. 11784, pp. 67–81. Springer (2019)
20. Dodaro, C., Galatà, G., Maratea, M., Mochi, M., Porro, I.: Chemotherapy treatment scheduling via answer set programming. In: Calimeri, F., Perri, S., Zumpano, E. (eds.) Proceedings of the 35th Italian Conference on Computational Logic (CILC 2020). CEUR Workshop Proceedings, vol. 2710, pp. 342–356. CEUR-WS.org (2020)
21. Dodaro, C., Galatà, G., Maratea, M., Porro, I.: Operating room scheduling via answer set programming. In: AI*IA. LNCS, vol. 11298, pp. 445–459. Springer (2018)
22. Dodaro, C., Galatà, G., Maratea, M., Porro, I.: An overview of ASP applications in the health-care domain. In: Proceedings of the RiCeRcA Workshop co-located with the 17th International Conference of the Italian Association for Artificial Intelligence, (RiCeRcA@Ai*iA 2018). CEUR Workshop Proceedings, vol. 2272. CEUR-WS.org (2018)
23. Dodaro, C., Galatà, G., Maratea, M., Porro, I.: An asp-based framework for operating room scheduling. Intelligenza Artificiale **13**(1), 63–77 (2019)
24. Dodaro, C., Gasteiger, P., Leone, N., Musitsch, B., Ricca, F., Schekotihin, K.: Combining answer set programming and domain heuristics for solving hard industrial problems (application paper). Theory and Practice of Logic Programming **16**(5-6), 653–669 (2016)
25. Dodaro, C., Leone, N., Nardi, B., Ricca, F.: Allotment problem in travel industry: A solution based on ASP. In: RR. LNCS, vol. 9209, pp. 77–92. Springer (2015)
26. Dodaro, C., Maratea, M.: Nurse scheduling via answer set programming. In: LPNMR. LNCS, vol. 10377, pp. 301–307. Springer (2017)
27. Erdem, E., Öztok, U.: Generating explanations for biomedical queries. Theory and Practice of Logic Programming **15**(1), 35–78 (2015)
28. Gatta, R., Vallati, M., Mazzini, N., Kitchin, D.E., Bonisoli, A., Gerevini, A., Valentini, V.: On the efficient allocation of diagnostic activities in modern imaging departments. In: Progress in Artificial Intelligence - 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings. pp. 103–109 (2015)
29. Gavanelli, M., Nonato, M., Peano, A.: An ASP approach for the valves positioning optimization in a water distribution system. Journal of Logic and Computation **25**(6), 1351–1369 (2015)
30. Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., Schneider, M.T.: Potassco: The Potsdam answer set solving collection. AI Communications **24**(2), 107–124 (2011)
31. Gebser, M., Leone, N., Maratea, M., Perri, S., Ricca, F., Schaub, T.: Evaluation techniques and systems for answer set programming: a survey. In: Lang, J. (ed.)

Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018. pp. 5450–5456. ijcai.org (2018)
32. Gebser, M., Maratea, M., Ricca, F.: The Design of the Sixth Answer Set Programming Competition. In: LPNMR. LNCS, vol. 9345, pp. 531–544. Springer (2015)
33. Gebser, M., Maratea, M., Ricca, F.: What's Hot in the Answer Set Programming Competition. In: AAAI. pp. 4327–4329. AAAI Press (2016)
34. Gebser, M., Maratea, M., Ricca, F.: The design of the seventh answer set programming competition. In: Balduccini, M., Janhunen, T. (eds.) LPNMR. Lecture Notes in Computer Science, vol. 10377, pp. 3–9. Springer (2017)
35. Gebser, M., Maratea, M., Ricca, F.: The sixth answer set programming competition. Journal of Artificial Intelligence Research **60**, 41–95 (2017)
36. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the Fifth International Conference and Symposium , Seattle, Washington, August 15-19, 1988 (2 Volumes). pp. 1070–1080. MIT Press (1988)
37. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Comput. **9**(3/4), 365–386 (1991)
38. Giunchiglia, E., Leone, N., Maratea, M.: On the relation among answer set solvers. Ann. Math. Artif. Intell. **53**(1-4), 169–204 (2008)
39. Giunchiglia, E., Maratea, M.: On the Relation Between Answer Set and SAT Procedures (or, Between cmodels and smodels). In: ICLP. LNCS, vol. 3668, pp. 37–51. Springer (2005)
40. Giunchiglia, E., Maratea, M., Tacchella, A.: Dependent and independent variables in propositional satisfiability. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA. Lecture Notes in Computer Science, vol. 2424, pp. 296–307. Springer (2002)
41. Giunchiglia, E., Maratea, M., Tacchella, A.: (In)Effectiveness of look-ahead techniques in a modern SAT solver. In: Rossi, F. (ed.) CP. Lecture Notes in Computer Science, vol. 2833, pp. 842–846. Springer (2003)
42. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable AI systems for the medical domain? CoRR **abs/1712.09923** (2017), `http://arxiv.org/abs/1712.09923`
43. Koponen, L., Oikarinen, E., Janhunen, T., Säilä, L.: Optimizing phylogenetic supertrees using answer set programming. TPLP **15**(4-5), 604–619 (2015)
44. Landa, P., Aringhieri, R., Soriano, P., Tànfani, E., Testi, A.: A hybrid optimization algorithm for surgeries scheduling. Operations Research for Health Care **8**, 103–114 (2016)
45. Lierler, Y., Maratea, M., Ricca, F.: Systems, engineering environments, and competitions. AI Magazine **37**(3), 45–52 (2016)
46. Maratea, M., Pulina, L., Ricca, F.: A multi-engine approach to answer-set programming. Theory and Practice of Logic Programming **14**(6), 841–868 (2014)
47. Maratea, M., Ricca, F., Faber, W., Leone, N.: Look-back techniques and heuristics in DLV: Implementation, evaluation, and comparison to QBF solvers. J. Algorithms **63**(1-3), 70–89 (2008)
48. Marileo, M.C., Bertossi, L.E.: The consistency extractor system: Answer set programs for consistent query answering in databases. Data Knowl. Eng. **69**(6), 545–572 (2010)
49. Molina-Pariente, J.M., Hans, E.W., Framinan, J.M., Gomez-Cia, T.: New heuristics for planning operating rooms. Computers & Industrial Engineering **90**, 429–443 (2015)