An Overview of ASP Applications in the Health-care Domain

Carmine Dodaro[1], Giuseppe Galatà[2], Marco Maratea[1], and Ivan Porro[2]

[1] DIBRIS, University of Genova, Genova, Italy
{dodaro,marco}@dibris.unige.it,
[2] SurgiQ srl, Genova, Italy
{name.surname}@surgiq.com

Abstract. Answer Set Programming is a (nonmonotic) knowledge representation and reasoning framework that has been employed in a number of applications. In this paper, we focus on two recent applications in the health-care domain.

1 Introduction

Modern hospitals operate 24 hours a day, 7 days for week, and they frequently have to deal with several complex combinatorial problems. Such problems are usually the target for the application of logic formalisms such as Answer Set Programming (ASP). Indeed, the simple syntax [18] and the intuitive semantics [34], combined with the availability of robust implementations (see, e.g. [4, 28, 41, 40]), also results of the ASP Competition series (see, e.g. [31, 32, 19, 33, 39, 30]), make ASP an ideal candidate for addressing such problems. As a matter of fact, ASP has been successfully used in several research areas, including Artificial Intelligence [14, 9, 10], Bioinformatics [25, 37], Hydroinformatics [26], and Databases [42]; more recently ASP has been applied to solve industrial applications [2, 22, 23]. In this paper, we report a brief survey of two problems in the health-care domain which have been recently solved by means of an ASP encoding, namely the Nurse Scheduling problem (NSP) and the Operating Room Scheduling (ORS) problem.

The NSP [17, 20] 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. A proper solution to the NSP is crucial to guarantee the high level of quality of health-care, to improve the degree of satisfaction of nurses and the recruitment of qualified personnel.

The ORS [1, 11, 38, 43] problem consists of assigning patients to operating rooms, taking into account different specialties, surgery durations, and operating room shift durations. 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. A proper solution to the ORS problem is important for improving the whole quality of the health-care and the satisfaction of patients. Indeed, modern hospitals are often characterized by long surgical waiting lists, which are caused by inefficiencies in operating room planning, leading to an obvious dissatisfaction of patients.

2 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. A proper solution to the NSP is crucial to guarantee the high level of quality of health-care, to improve the degree of satisfaction of nurses and the recruitment of qualified personnel.

In recent years, several approaches to solve NSP have been proposed. They usually differ from each other since different requirements are considered. In particular, the main differences concern (i) the planning periods; (ii) the different type of shifts; (iii) the requirements related to the coverage of shifts, i.e. the number of personnel needed for every shift; and (iv) other restrictions on the rules of nurses (see [17] for more detailed information). Other requirements depend on the different policies of the considered hospitals. Thus, this makes the different strategies not directly comparable.

Solving technologies reported in the literature range from mathematical to meta-heuristics approaches, including solutions based on integer programming [13, 15], genetic algorithms [3], fuzzy approaches [45], and ant colony optimization algorithms [35], to mention a few. Detailed and comprehensive surveys on NSP can be found in [17, 20].

The ASP approaches have been described in [7, 24]. They are based on a formalization of the NSP as described by an Italian hospital. 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* (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 in a range.

The ASP encodings have been tested on an instance provided by an Italian hospital which includes 41 nurses and several fixed holidays for each nurse. The results of the experiment were reported in [7, 8], where the ASP systems CLINGO [27, 29] and WASP [6, 21] configured with the core-based algorithms [4, 5] were compared to

SAT solvers LINGELING [16], GLUCOSE [12] and the mathematical programming tool GUROBI[3]. The best result overall was obtained by CLINGO which is able to find a schedule in 43 seconds, while GUROBI found a schedule in approximately 17 minutes, and other tested solvers were not able to find a schedule in less than 1 hour. A scalability analysis has been also performed, which confirmed the mentioned relative results.

3 Operating Room Scheduling

According to a study carried on by National Health Service (NHS) England, NHS hospitals could carry out 280,000 more non-emergency operations a year by organizing operating room schedules (ORS) better[4]. On the one hand OR resources are often allocated on a historical basis and time allocation can bear little relation to the need associated with the procedure being performed, on the other hand bed shortages remain a major issue for surgery and there is a need of having protected beds and OR time for planned surgery and separate emergency OR lists[5].

One way this problem can be addressed is by optimizing ORS. The ORS problem can be broken down into two segments: production of an initial schedule for a given time unit (e.g. day, week, or month) and generation of altered schedules based on complications or conflicts that require changes in the initial schedule.

Previous works in the ORS problem include for the initial schedule production those by Aringhieri et al. [11, 38], Abedini et al. [1], Molina-Pariente et al. [43], Zhang et al. [46] and for the production of the altered schedules Shu et al. [44].

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) 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 chosen to classify 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 ORS. 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:

[3] http://www.gurobi.com
[4] http://www.bbc.com/news/health-41727535
[5] https://www.rcseng.ac.uk/news-and-events/media-centre/press-releases/nhs-improvement-theatre-improvement/

- 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 number of occupied beds for each specialty and each day must not exceed the available beds,
- 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.

Moreover, the user can interact with the algorithm through an interface that allows to add so called "customizable constraints", that express user requirements and preferences such as imposing or forbidding any number of OR blocks to a set of user selected registrations.

The result of the process described above is an optimized initial schedule which could be later put into practice. However, in hospital units it is frequent that one planned assignment of ORs cannot be fulfilled due to complications or conflicts that may occur either during the surgery or before. In particular, some surgeries may be postponed due to medical reasons or to delays provoked by the previous ones. In order to not defer the postponed registrations for too long, it is often necessary to reallocate them to an already scheduled OR block, thus potentially disrupting the initial schedule. Therefore, in such cases it is required to compute a new schedule which reallocates every surgical procedure not yet executed (i.e. the ones in OR blocks not started yet and the postponed ones) to the OR blocks and, at the same time, minimizes the differences with the previously computed initial schedule, obviously excluding the part of the schedule already passed at the reassignment time. This problem is usually referred to as *rescheduling*.

Most of the constraints defined above for the production of the initial schedule remain valid in the rescheduling case, with the exception of the priority related one: since we aim to reallocate every registration, there is no need to use their priority category during the rescheduling. In the occurrence that this is not possible because the overall duration of the registrations to be reallocated is superior to the available time in the OR blocks, one or more can be excluded from the rescheduling, manually by the user or by a separate algorithm that starts from the registrations placed in the last OR block of the initial schedule and had the lowest priorities.

The other difference between initial schedule production and the rescheduling is the objective function. In the latter case the goal is to minimize the difference in days between the new and old assignments for each registration, as opposed to the minimization of the unassigned registrations in the former case.

We performed an empirical analysis of our ASP solutions for the scheduling and rescheduling problems. For the initial scheduling problem, data have been randomly generated but having parameters and sizes inspired by real data, then a part of the results of the planning has been used as input for the rescheduling, simulating a disrupted schedule. Both experiments were run on a Intel Core i7-7500U CPU @ 2.70GHz with 7.6 GB of physical RAM. The ASP system used was CLINGO [27], version 5.5.2. Our tests show that our scheduling solution obtains around 95% of efficiency after few seconds of computation on planning length of 5 days usually used in Italian hospitals. Our solution also enjoys good scalability property, having an efficiency over or equal to 90% for planning periods up to 10 days, i.e. double w.r.t. the target period. Also our rescheduling solution reached positive results.

4 Conclusions

In this paper we have shown two applications in the health-care domain where ASP has been proficiently employed. Another related application, we did not focus on, is presented in [36], where authors represented and evaluated the Outpatient Day Service Operations with ASP methodology.

About future works, we first plan to compare our ORS solution to other approaches. We further plan to evaluate ASP in other problems in the health-care domain, e.g. the generation of optimized schedules for night shift hospital operators, in order to assign in (possibly) a few seconds to each operator the most effective list of tasks taking into account constraints such as the type, priority and duration of the tasks to be assigned and the number of required operators.

References

1. Abedini, A., Ye, H., Li, W.: Operating Room Planning under Surgery Type and Priority Constraints. Procedia Manufacturing **5**, 15–25 (2016)
2. Abseher, M., Gebser, M., Musliu, N., Schaub, T., Woltran, S.: Shift design with answer set programming. Fundam. Inform. **147**(1), 1–25 (2016)
3. Aickelin, U., Dowsland, K.A.: An indirect genetic algorithm for a nurse-scheduling problem. Computers & OR **31**(5), 761–778 (2004)
4. Alviano, M., Dodaro, C.: Anytime answer set optimization via unsatisfiable core shrinking. TPLP **16**(5-6), 533–551 (2016)
5. Alviano, M., Dodaro, C.: Unsatisfiable core shrinking for anytime answer set optimization. In: Sierra, C. (ed.) IJCAI. pp. 4781–4785. ijcai.org (2017)
6. Alviano, M., Dodaro, C., Leone, N., Ricca, F.: Advances in WASP. In: LPNMR. LNCS, vol. 9345, pp. 40–54. Springer (2015)
7. 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)
8. Alviano, M., Dodaro, C., Maratea, M.: Nurse (re)scheduling via answer set programming. Intelligenza Artificiale (2018 To appear)
9. Amendola, G., Dodaro, C., Leone, N., Ricca, F.: On the application of answer set programming to the conference paper assignment problem. In: AI*IA. Lecture Notes in Computer Science, vol. 10037, pp. 164–178. Springer (2016)
10. Amendola, G., Greco, G., Leone, N., Veltri, P.: Modeling and reasoning about NTU games via answer set programming. In: IJCAI 2016. pp. 38–45 (2016)
11. Aringhieri, R., Landa, P., Soriano, P., Tnfani, E., Testi, A.: A two level metaheuristic for the operating room scheduling and assignment problem. Computers & Operations Research **54**, 21–34 (2015)
12. Audemard, G., Simon, L.: Extreme cases in SAT problems. In: Creignou, N., Berre, D.L. (eds.) SAT. LNCS, vol. 9710, pp. 87–103. Springer (2016)

13. Azaiez, M.N., Sharif, S.S.A.: A 0-1 goal programming model for nurse scheduling. Computers & OR **32**, 491–507 (2005)
14. 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)
15. Bard, J.F., Purnomo, H.W.: Preference scheduling for nurses using column generation. European Journal of Operational Research **164**(2), 510–534 (2005)
16. Biere, A., Fröhlich, A.: Evaluating CDCL variable scoring schemes. In: SAT. LNCS, vol. 9340, pp. 405–422. Springer (2015)
17. 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)
18. Calimeri, F., Faber, W., Gebser, M., Ianni, G., Kaminski, R., Krennwallner, T., Leone, N., Ricca, F., Schaub, T.: ASP-Core-2 Input Language Format (2013), `https://www.mat.unical.it/aspcomp2013/files/ASP-CORE-2.01c.pdf`
19. Calimeri, F., Gebser, M., Maratea, M., Ricca, F.: Design and results of the Fifth Answer Set Programming Competition. Artificial Intelligence **231**, 151–181 (2016)
20. Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems - a bibliographic survey. European Journal of Operational Research **151**(3), 447–460 (2003)
21. Dodaro, C., Alviano, M., Faber, W., Leone, N., Ricca, F., Sirianni, M.: The birth of a WASP: preliminary report on a new ASP solver. In: CILC. CEUR Workshop Proceedings, vol. 810, pp. 99–113. CEUR-WS.org (2011)
22. 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). TPLP **16**(5-6), 653–669 (2016)
23. 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)
24. Dodaro, C., Maratea, M.: Nurse scheduling via answer set programming. In: LPNMR. LNCS, vol. 10377, pp. 301–307. Springer (2017)
25. Erdem, E., Öztok, U.: Generating explanations for biomedical queries. TPLP **15**(1), 35–78 (2015)
26. Gavanelli, M., Nonato, M., Peano, A.: An ASP approach for the valves positioning optimization in a water distribution system. J. Log. Comput. **25**(6), 1351–1369 (2015)
27. 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)
28. Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., Schneider, M.T.: Potassco: The potsdam answer set solving collection. AI Commun. **24**(2), 107–124 (2011)
29. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Artif. Intell. **187**, 52–89 (2012)
30. 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)
31. 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)
32. Gebser, M., Maratea, M., Ricca, F.: What's hot in the answer set programming competition. In: AAAI. pp. 4327–4329. AAAI Press (2016)
33. Gebser, M., Maratea, M., Ricca, F.: The sixth answer set programming competition. Journal of Artificial Intelligence Research **60**, 41–95 (2017)
34. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Comput. **9**(3/4), 365–386 (1991)

35. Gutjahr, W.J., Rauner, M.S.: An ACO algorithm for a dynamic regional nurse-scheduling problem in austria. Computers & OR **34**(3), 642–666 (2007)
36. Ielpa, G., Guido, R., Conforti, D.: Outpatient day service operations: A case study within rheumatology diseases management. In: Cappanera, P., Li, J., Matta, A., Sahin, E., Vandaele, N., Visintin, F. (eds.) Health Care Systems Engineering: Proceedings of the International Conference on Health Care Systems Engineering (ICHCSE 2017). Proceedings in Mathematics & Statistics, vol. 210, pp. 269–279. Springer (2018)
37. Koponen, L., Oikarinen, E., Janhunen, T., Säilä, L.: Optimizing phylogenetic supertrees using answer set programming. TPLP **15**(4-5), 604–619 (2015)
38. Landa, P., Aringhieri, R., Soriano, P., Tnfani, E., Testi, A.: A hybrid optimization algorithm for surgeries scheduling. Operations Research for Health Care **8**, 103–114 (2016)
39. Lierler, Y., Maratea, M., Ricca, F.: Systems, engineering environments, and competitions. AI Magazine **37**(3), 45–52 (2016)
40. 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)
41. Maratea, M., Ricca, F., Faber, W., Leone, N.: Look-back techniques and heuristics in DLV: Implementation, evaluation, and comparison to QBF solvers. Journal of Algorithms **63**(1-3), 70–89 (2008)
42. 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)
43. 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)
44. Shu, A.C., Subbaraj, I., Phan, L.: Operating Room Rescheduler (2015)
45. Topaloglu, S., Selim, H.: Nurse scheduling using fuzzy modeling approach. Fuzzy Sets and Systems **161**(11), 1543–1563 (2010)
46. Zhang, J., Dridi, M., El Moudni, A.: A stochastic shortest-path MDP model with dead ends for operating rooms planning. pp. 1–6 (Sep 2017)