Specification languages and ASP-based solutions for scheduling problems in Healthcare*

Simone Caruso[1]

[1]University of Genoa, Genova, Italy

Abstract

In the Digital Health domain, there are many scheduling problems that are important to solve to improve the quality of the medical cares offered and patients' satisfaction. Following this need, we want to contribute to the resolution of these problems. Especially, in recent years, the AI language Answer Set Programming (ASP) has been successfully applied in scheduling problems such as the ones we want to solve, proving to be an efficient solution. However, despite the success of ASP, solutions of this type are not easy to implement and to be accepted by people without a background in logic programming, making it difficult to use and deploy in real-world contexts. Hence, the aim is also to define a specification language, i.e. a controlled natural language (CNL), and to develop a tool called CNL2ASP that allows to convert such language into AI languages and, in particular, into ASP as first target language.

Keywords

Healthcare, Answer Set Programming, Logic Programming, Controlled Natural Language

1. Introduction and problem description

In this section, two healthcare problems are first presented: The Pre-Operative Assessment Clinic (PAC) scheduling problem and The Scheduling Periodic Treatments problem, after which it is presented the problem of converting a CNL into ASP. Complex combinatorial problems, involving optimizations, such as the two mentioned, in fact, are usually the target applications of AI languages such as ASP, that has proven to be a suitable and efficient solution for many healthcare problems. The Pre-Operative Assessment Clinic scheduling problem is the task of assigning patients to a day, in which the patient will be examined and prepared to a surgical operation, taking into account, e.g. patients with different priority levels, due dates, and operators availability. This is the phase in which the patients enter the facility to perform laboratory tests, radiological and cardiological examination, counseling for anesthesiological suitability for surgery, and any other examination that the medical specialists deem appropriate for the definition of the pre-intervention diagnostic pathway. This phase of the surgical pathway is very important, because it ensures a proper preparation of patients to their surgical operation allowing them to stay at home until the morning of the surgery, instead of being admitted to the hospital one or two days before the scheduled operation; moreover, this allows to reduce

AIxIA'23: 22nd International Conference of the Italian Association for Artificial Intelligence, November 06–09, 2023, Rome, Italy

*Doctoral project supervised by Marco Maratea (University of Calabria).

simone.caruso@edu.unige.it (S. Caruso)
0000-0002-2724-4342 (S. Caruso)

waiting time between the exams, thus increasing patients' satisfaction [1], and to avoid the cancellation of the surgery [2].

Instead, the problem of scheduling periodic treatments consists of planning a care path over a period of several weeks, in which patients have to perform different treatments respecting a certain periodicity. This problem is very recurrent in healthcare as periodic appointments are often required: this is the case, for example, of rehabilitation or planning a cycle of drug therapies. A typical use case is that, after a medical examination, the patient comes with a precise list of treatments, e.g. 10 sessions of physiotherapy followed up by 10 minutes of cardio training, and asks for the scheduling of a certain number of appointments based on their availability and on the availability of the care facility. Thus, an optimal solution to this problem assigns the patients' appointments in their desired days of the week, as close as possible from a given target week, i.e. the week prescribed by the doctor and minimizes waiting time between treatments, if more than one is requested in the same day, while the operators' working time is daily maximized and weekly balanced among the available operators.

Moreover, we want to address these scheduling problems by creating reliable systems that can be applied in real-world scenarios. In particular for problems in healthcare the solution must, certainly, ensure optimized scheduling but, being indeed applications used in real-world contexts, it is very important to be able to deal with unforeseen issues. This is the case, for example, of the rescheduling problem, where a new schedule assignment must be found with as few changes as possible but, obviously, still respecting all the constraints of the problem.

Despite the success of ASP in solving complex combinatorial problems, its usage is not straightforward for people without mathematical or logical backgrounds. The idea of CNL2ASP is to define a specification language and develop a tool that converts such language into ASP. This tool would lead to different advantages as CNL specifications are easy to read and write, even by users without mathematical and logical backgrounds, but also defining problems in CNL specifications should be easier and faster than encoding knowledge in a formal KR language like ASP and this can be used as a starting point for further optimization made by ASP experts.

2. Overview of the state of the art and contribution

2.1. Overview of the state of the art

In this section it is presented the state of the art of the problems presented before. As already mentioned, 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 problem of scheduling nurses working in hospital units [3, 4]. Then, the problem of assigning operating rooms to patients [5] was considered. More recent problems include the Chemotherapy Treatment Scheduling problem [6], and the Rehabilitation Scheduling Problem [7], which assigns patients to operators in rehabilitation sessions.

Regarding the PAC problem, [8] used two simulation models to analyse the difficulties of planning in the context of PAC and to determine the resources needed to reduce waiting times and long access times. The authors of [1, 2] studied the importance of implementing the PAC problem and the positive results obtained by having less waiting time between the exams and for the visit to the hospital. In particular, while different clinics follow different guidelines, implementing PAC has proved to be an important tool to avoid the cancellation of surgeries and to significantly reduce the risk associated with the surgery. Despite the importance of this problem, none of the works above mentioned proposes an actual scheduling of the PAC with real requirements, making a practical system able to solve the PAC scheduling problems. Instead, in our work we propose a practical solution to this problem.

For the scheduling of periodic treatments, in [9] a review of the literature on multi-appointment scheduling problems in hospitals, i.e. problems in which patients need appointments for different resources, is presented and shows how research interest in these problems has grown only in recent years.
Problems similar to the one we dealt with and presented before can be found only in particular use cases such as rehabilitation, chemotherapy [10] and radiotherapy [11]; however, there is not a general solution for these problems as the one we aim for. We faced, in fact, a problem flexible and able to adapt to different contexts where different appointments, involving different operators and resources, are required with a certain frequency and in a time horizon of several weeks.

Finally, for the CNL2ASP problem, in the following it is presented an overview of the main CNLs proposed in the area of logic programming. One of the first attempts of encoding expressed in a CNL as logic programs was presented by Attempto CNL [12], whose idea was to convert sentences expressed in a CNL as Prolog clauses. More recently, [13] defined the language PENG^{ASP}, a CNL that is automatically converted into ASP. Albeit some aspects of PENG^{ASP}'s grammar rules are present in the grammar of our CNL, the latter is geared more towards the formal definition of combinatorial problems in a natural-feeling and unambiguous way that is also reliably predictable in its translation to ASP, choosing words that would stand out easily during reading and with an easily deducible meaning from the given context; this meant sacrificing some of the naturalness of PENG^{ASP}.

2.2. Contribution

For the PAC scheduling problem, we dealt with the scheduling and rescheduling problem [14]. Then, we developed a web application in order to make the solution we provided accessible to users with no or low knowledge of ASP, allowing hospital operators to interact with our solution and understand the results through a graphical interface. In figure 1, an example of the schedule that we want to compute, by assigning the starting times (x-axis) to each patient (y-axis) in a particular day. As can be seen in figure 1, in this case the scheduler is able to assign all patients optimally; indeed, all patients have no waiting time among the exams and thus the time spent in the hospital is reduced to the minimum.

For the scheduling periodic treatments [15] problem, we developed a solution and conducted a preliminary experimental analysis. Despite the complexity of the encoding realized, involving several rules and optimization criteria and levels, we achieved good results.

In the work on CNL2ASP, we have reached a good state, where different propositions and constructs are supported that allow different problems to be represented. For this purpose, we have tested the CNL we made to encode different problems coming from different domains. In particular, we encoded a scheduling problem in healthcare domain, one in the robotics domain and different problems related to graphs such as the graph coloring shown in figure 2. Moreover,

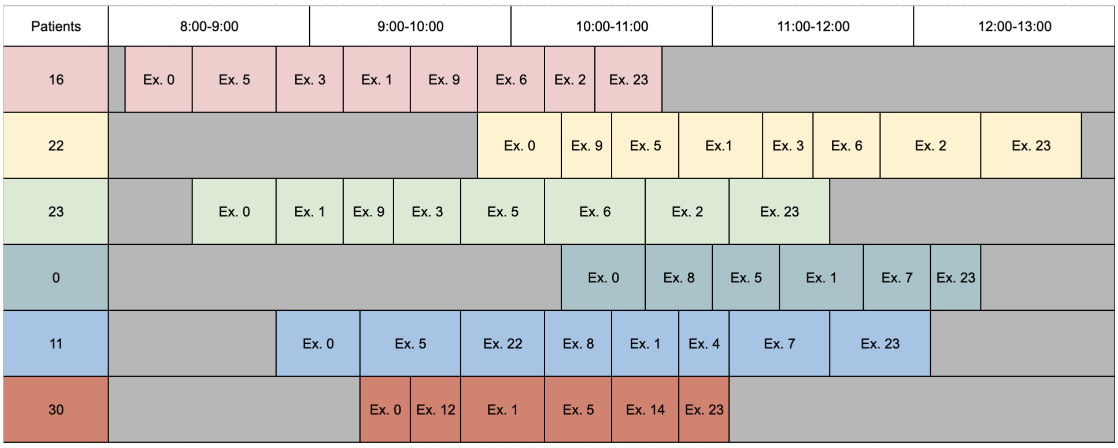

Figure 1: Schedule example: assignments of the starting time of the different exam locations of each patient in a single day

```
A node goes from 1 to 3.
A color is one of red, green, blue.
Node 1 is connected to node X, where X is one of 2, 3.
Node 2 is connected to node X, where X is one of 1, 3.
Node 3 is connected to node X, where X is one of 1, 2.
Every node can be assigned to exactly 1 color.
It is required that when node X is connected to node Y then node X is not assigned
    to color C and also node Y is not assigned to color C.
```

Figure 2: Graph coloring problem via CNL

the tool is open source and publicly available at https://github.com/dodaro/cnl2asp.

3. Ongoing and future work

As mentioned, one of the goals is to contribute to the resolution of scheduling problems, in particular the two presented before. The results obtained present good performance for the realized solutions, however, a possible idea to improve performance is to decompose the scheduling problems [16]. Improving performance as much as possible is very important since, in a real context, the operator and the patient, who has to schedule a new appointment, cannot be asked to wait too long. It is, therefore, required that a solution is computed in the shortest time possible. For example, in the scheduling periodic treatments problem, we have a target week to start from and as the highest priority optimization criterion we have to depart from this as little as possible. One could, therefore, decompose the problem into N (where N is the total number of weeks considered) sub-problems in which we start from the target week and gradually move away from it. This would probably decrease the time spent in the grounding phase, that is the phase where all possible assignments that meet the constraints of the problem are computed, because, instead of having a possible schedule over N weeks, we have N sub-problems that can also be solved in parallel. Then, one could still decompose these sub-problems into two other sub-problems, in which, in the first it is assigned a time and day to the patient's treatments and, then, in the second sub-problem it is assigned the necessary resources. This, again, to reduce grounding. Each assignment or rather choice rule in terms of ASP, in fact, complicates the search of the solution as we add new possible combinations. Finally, the goal is to extend the CNL to support a language as close as possible to a natural language, so that it is easier for less experienced users to use. Other possible extension that we would like to support is, certainly, a conversion from ASP into CNL so that it is possible to have greater readability of models also encoded directly in ASP. Then, an other feature can be to support other languages on either sides. One could support other controlled natural languages, e.g. SBVR, in addition to our developed CNL and, on the other side, other logic programming languages in addition to ASP, the latter feature would also favor the possibility of having comparisons between different AI languages (which is a problem that often arises in this field of research and making these comparisons is not always simple and fair, since one may be an expert in one language but not in another). Moreover, an interesting application could be to do high-level explainability. It is, indeed, necessary for the success of ASP, but in general for any system, to be able to explain the reasons behind a decision taken, e.g. why one solution is better than another, why something is or is not part of a solution. Taking advantage of the readability of the CNL, one could, in fact, give explanations of the solutions computed by the solvers, which at the moment are rather difficult to explain, especially, to the less experienced users in AI languages, bringing an important step in the usage of ASP.

References

[1] M. P. Harnett, D. Correll, S. Hurwitz, A. Bader, D. Hepner, Improving Efficiency and Patient Satisfaction in a Tertiary Teaching Hospital Preoperative Clinic, Anesthesiology 112 (2010) 66–72.

[2] M. Ferschl, A. Tung, B. Sweitzer, D. Huo, D. Glick, Preoperative Clinic Visits Reduce Operating Room Cancellations and Delays, Anesthesiology 103 (2005) 855–859.

[3] C. Dodaro, M. Maratea, Nurse scheduling via answer set programming, in: LPNMR, volume 10377 of *LNCS*, Springer, 2017, pp. 301–307.

[4] M. Alviano, C. Dodaro, M. Maratea, Nurse (re)scheduling via answer set programming, Intelligenza Artificiale 12 (2018) 109–124.

[5] C. Dodaro, G. Galatà, M. Maratea, I. Porro, An ASP-based framework for operating room scheduling, Intelligenza Artificiale 13 (2019) 63–77.

[6] C. Dodaro, G. Galatà, A. Grioni, M. Maratea, M. Mochi, I. Porro, An ASP-based solution to the chemotherapy treatment scheduling problem, Theory and Practice of Logic Programming 21 (2021) 835–851.

[7] 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.

[8] G. M. Edward, S. F. Das, S. G. Elkhuizen, P. J. M. Bakker, J. A. M. Hontelez, M. W. Hollmann, B. Preckel, L. C. Lemaire, Simulation to analyse planning difficulties at the preoperative assessment clinic, BJA: British Journal of Anaesthesia 100 (2008) 195–202.

[9] J. Marynissen, E. Demeulemeester, Literature review on multi-appointment scheduling problems in hospitals, European Journal of Operational Research 272 (2019) 407–419. URL: https://www.sciencedirect.com/science/article/pii/S0377221718302108. doi:https://doi.org/10.1016/j.ejor.2018.03.001.

[10] A. Condotta, N. Shakhlevich, Scheduling patient appointments via multilevel template: A case study in chemotherapy, Operations Research for Health Care 3 (2014) 129–144. URL: https://www.sciencedirect.com/science/article/pii/S221169231400006X. doi:https://doi.org/10.1016/j.orhc.2014.02.002.

[11] D. Conforti, F. Guerriero, R. Guido, Optimization models for radiotherapy patient scheduling, 4or 6 (2008) 263–278.

[12] N. E. Fuchs, Knowledge representation and reasoning in (controlled) natural language, in: Proceedings of ICCS, volume 3596 of *LNCS*, Springer, 2005, pp. 51–51. URL: https://doi.org/10.1007/11524564_3. doi:10.1007/11524564_3.

[13] R. Schwitter, Specifying and verbalising answer set programs in controlled natural language, Theory Pract. Log. Program. 18 (2018) 691–705. doi:10.1017/S1471068418000327.

[14] S. Caruso, G. Galatà, M. Maratea, M. Mochi, I. Porro, Scheduling preoperative assessment clinic with answer set programming, Journal of Logic and Computation (2023) exad017. URL: https://doi.org/10.1093/logcom/exad017. doi:10.1093/logcom/exad017. arXiv:https://academic.oup.com/logcom/advance-article-pdf/doi/10.1093/logcom/exad017/49857006/exad017.pdf.

[15] S. Caruso, C. Dodaro, G. Galatà, M. Maratea, Scheduling periodic treatments via answer set programming, in: A. Dovier, A. Formisano (Eds.), Proceedings of the 38th Italian Conference on Computational Logic, Udine, Italy, June 21-23, 2023, volume 3428 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3428/paper7.pdf.

[16] P. Cappanera, M. Gavanelli, M. Nonato, M. Roma, Logic-based benders decomposition in answer set programming for chronic outpatients scheduling, CoRR abs/2305.11969 (2023). URL: https://doi.org/10.48550/arXiv.2305.11969. doi:10.48550/arXiv.2305.11969. arXiv:2305.11969.