Nurse Scheduling via Answer Set Programming

Carmine Dodaro (✉) and Marco Maratea

DIBRIS, University of Genova, Italy
dodaro@dibris.unige.it, marco@dibris.unige.it

Abstract. The Nurse Scheduling problem (NSP) is a combinatorial problem that consists of assigning nurses to shifts according to given practical constraints. In previous years, several approaches have been proposed to solve different variants of the NSP. In this paper, an ASP encoding for one of these variants is presented, whose requirements have been provided by an Italian hospital. We also design a second encoding for the computation of "optimal" schedules. Finally, an experimental analysis has been conducted on real data provided by the Italian hospital using both encodings. Results are very positive: the state-of-the-art ASP system CLINGO is able to compute one year schedules in few minutes, and it scales well even when more than one hundred nurses are considered.

Keywords: Answer Set Programming, Scheduling, Nurse Scheduling problem

1 Introduction

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. For these reasons, several approaches to solve the NSP are reported in the literature, including those based on integer programming [1, 2], genetic algorithms [3], fuzzy approaches [4], and ant colony optimization algorithms [5], to mention a few. However, such approaches are not directly comparable with each other, since the requirements usually depend on the policy of the specific hospitals [6]. Detailed surveys on the NSP can be found in [7, 8].

Complex combinatorial problems, such as the NSP, are usually the target for the application of logic formalisms such as Answer Set Programming (ASP). Indeed, the simple syntax [9] and the intuitive semantics [10], combined with the availability of robust implementations (see, e.g. [11, 12]), 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 [13], Bio-informatics [14, 15], and Databases [16]; more recently ASP has been applied to solve industrial applications [17–19]. However, to the best of our knowledge, no ASP encoding has presented to solve the NSP.

In this paper, we report an ASP encoding to address a variant of the NSP (see Section 3), whose requirements, presented in Section 2, have been provided by an Italian hospital. We also present a variant of the encoding for the computation of "optimal" schedules. The encoding is natural and intuitive, in the sense that it was obtained by applying the standard modeling methodology, and it is easy to understand. Moreover, an experimental analysis has been conducted on real data provided by an Italian hospital using both encodings. Results are very positive: the state-of-the-art ASP system CLINGO [12] is able to compute one year schedules in few minutes. Moreover, scalability analysis shows that CLINGO scales well even when more than one hundred nurses are considered (see Section 4).

2 Nurse Scheduling Problem

The NSP involves the generation of schedules for nurses consisting of working and rest days over a predetermined period of time, which is fixed to one year in this paper. Moreover, the schedules must satisfy a set of requirements. In this section, we informally describe the ones used in the paper as specified by an Italian hospital.

Hospital requirements. We consider three different shifts: *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. Concerning nurses, the schedules have to guarantee a fair workload. Thus, 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)*.

Balance requirements. Finally, the number of times a nurse can be assigned to morning, afternoon and night shifts is fixed. However, schedules where this number ranges over a set of acceptable values are also valid. Thus, we identify two variants of the encoding that will be discussed in the next section.

3 Answer Set Programming Encoding

The NSP described in the previous section has been solved by means of an ASP encoding. Answer sets of the logic program presented in this section correspond to the solutions of the NSP. In the following, we assume that the reader is familiar with Answer Set Programming and ASP-CORE-2 input language specification [9].

The encoding has been created following the *Guess&Check* programming methodology. In particular, the following choice rule guesses an assignment to exactly one shift for each nurse and for each day.

$$1 \leq \{assign(N,S,D) : shift(S,H)\} \leq 1 \text{ :- } day(D),\ nurse(N). \tag{1}$$

Note that the rule also filters out assignments where a nurse works twice during the same day. Instances of the predicate *assign(N, S, D)* are used to store the shift assignment *S* for a nurse *N* in a specific day *D*. Instances of the predicate *shift(S, H)* are used to represent the shifts, where *S* is a shift and *H* is the number of working hours associated to the shift. In our setting, we consider the following instances of the predicate: *shift(″1−mor″, 7)*, *shift(″2−aft″, 7)*, *shift(″3−nig″, 10)*, *shift(″4−specres″, 0)*, *shift(″5−rest″, 0)*, and *shift(″6−vac″, 0)*. Actually, the latest three instances are used in our encoding to model the nurses days off. Note that the number used in the name of the shift will be used in the following to compactly encoding constraints in (5).

Hospital requirements. Hospitals need to guarantee that a minimum and a maximum number of nurses are present in the hospital during a specific shift:

$$\begin{aligned} &:\!-\ day(D),\ \#count\{N : assign(N,''1{-}mor'',D)\} > K,\ maxNurseMorning(K).\\ &:\!-\ day(D),\ \#count\{N : assign(N,''1-mor'',D)\} < K,\ minNurseMorning(K).\end{aligned} \tag{2}$$

In particular, constraints reported in (2) refers to the morning shift. The ones related to afternoon and the night shifts are similar and are not reported here for space constraints.

Nurse requirements. First of all, requirements to guarantee a fair workload of nurses are accomplished as follows:

$$\begin{aligned} &:\!-\ nurse(N),\ maxHours(M),\ \#sum\{H,D : assign(N,S,D),\ shift(S,H)\} > M.\\ &:\!-\ nurse(N),\ minHours(M),\ \#sum\{H,D : assign(N,S,D),\ shift(S,H)\} < M.\end{aligned} \tag{3}$$

In particular, for each nurse the number of working hours is bounded by a minimum and a maximum number provided as input by the hospital. Moreover, nurses are guaranteed exactly 30 days of vacation. Thus, the following constraint filters out assignments where the number of vacation days is different from 30:

$$:\!-\ nurse(N),\ \#count\{D : assign(N,''6{-}vac'',D)\} \neq 30. \tag{4}$$

Then, the starting time of a shift must be at least 24 hours later than the starting time of the previous shift. In other words, a nurse assigned to the afternoon shift cannot be assigned to the morning shift of the day after; and a nurse assigned to the night shift cannot be assigned to the morning and to the afternoon shifts of the day after. These requirements are compactly expressed by means of the following constraint:

$$:\!-\ nurse(N),\ assign(N,S1,D),\ assign(N,S2,D+1),\ S2 < S1 \leq\ ''3-nig''. \tag{5}$$

The correctness of this constraint is guaranteed by the following ordering on shifts: *″1−mor″* < *″2−aft″* < *″3−nig″*. As implementation note, the lexicographic order of strings is not always guaranteed by current systems, thus in the tested encoding only integers are used, e.g. *shift(″1−mor″, 7* is replaced by *shift(1,7)*.

Moreover, each nurse is guaranteed at least two rest days for each fourteen days. Assignments violating such requirement are filtered out by the following constraint (where YD is the number of days in a year):

$$\begin{aligned} &:\!-\ nurse(N),\ day(D),\ days(YD),\ D \leq YD-13,\\ &\qquad \#count\{D1 : assign(N,''5{-}rest'',D1),\ D \leq D1 \leq D+13\} < 2.\end{aligned} \tag{6}$$

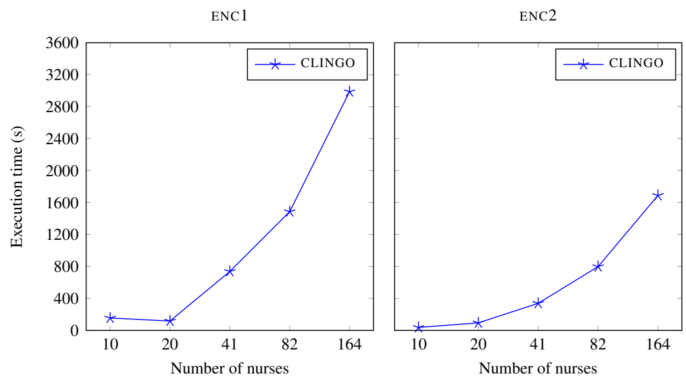

Fig. 1. Scalability analysis with 10, 20, 41, 82 and 164 nurses.

Finally, after two consecutive working nights one special rest day is guaranteed.

$$\begin{aligned}
&{:}-\ assign(N,''3{-}nig'',D-2),\ assign(N,''3{-}nig'',D-1),\\
&\qquad not\ assign(N,''4{-}specrest'',D).\\
&{:}-\ assign(N,''4{-}specrest'',D),\ not\ assign(N,''3{-}nig'',D-2).\\
&{:}-\ assign(N,''4{-}specrest'',D),\ not\ assign(N,''3{-}nig'',D-1).
\end{aligned} \tag{7}$$

Balance requirements. We report here the two variants considered in the paper. The first variant is expressed by means of the following constraints:

$$\begin{aligned}
&{:}-\ nurse(N),\ \#count\{D : assign(N,''1{-}mor'',D)\} > M,\ maxMorning(M).\\
&{:}-\ nurse(N),\ \#count\{D : assign(N,''1{-}mor'',D)\} < M,\ minMorning(M).
\end{aligned} \tag{8}$$

The idea of (8) is to filter out only assignments where shifts assigned to nurses are out of a fixed range of acceptable values.

The second variant of the encoding considers also the following weak constraint:

$$\begin{aligned}
&{:}\sim\ nurse(N),\ nbMorning(M),\ \#count\{D : assign(N,''1{-}mor'',D)\} = X,\\
&minMorning(M_1),\ maxMorning(M_2),\ M_1 \leq X \leq M_2,\ Y = \mid X - M \mid .\ [\,Y@1,\ N\,]
\end{aligned} \tag{9}$$

In this case the weak constraint is used to assign a cost for each assignment measuring the distance between the assignment and the target M. The optimum assignment is the one with the minimum cost. Note that in (8) and in (9) we reported only the constraints related to the morning shift since the ones related to other shifts are similar.

4 Empirical evaluation

In this section we report about the results of an empirical analysis conducted on real data provided by an Italian hospital used as reference in this paper. The scheduling has been created for a one year time window and the maximum (resp. minimum) number of hours per year is set to 1692 (resp. 1687). The number of nurses working to the considered hospital unit is 41, and the number of nurses working during the morning and the afternoon shifts ranges from 6 to 9, whereas the number of nurses working during the night shift ranges from 4 to 7. Concerning the holidays, we considered 15 days of vacation chosen by the nurses, according to the vacations selected in the year 2015, whereas the other 15 days of vacation are assigned according to the needs of the hospital. The desired number of working mornings and afternoons is equal to 78, whereas the desired number of working nights is equal to 60. In addition, the hospital can accept schedules where the number of working mornings and afternoons per year ranges from 74 to 82, whereas the number of working nights per year ranges from 58 to 61. We tested two variants of the encoding. The first one, reported as ENC1, considers constraints from (1) to (8), whereas the second variant, reported as ENC2, considers also the weak constraint (9). The experiments were run on an Intel Xeon 2.4 GHz. Time and memory were limited to 1 hour and 15 GB, respectively. Concerning the ASP systems, we used CLINGO [12] and WASP [20]. However, the performance of the latter are not reported since it is slower than CLINGO in all tested instances.

Concerning encoding ENC1, CLINGO computes a schedule in 12 minutes with a peak of memory usage of 300 MB. The performance of CLINGO executed on encoding ENC2 are even better. In fact, CLINGO configured with the option `--opt-strategy=usc` is able to find the optimum solution in 6 minutes with a peak of memory usage of 224 MB. The difference in performance can be explained by the strategy employed by CLINGO in presence of weak constraints. In particular, weak constraints are first considered as hard constraint and then relaxed when they cannot be satisfied.

Scalability. We also performed an analysis about the scalability of the encoding, considering different numbers of nurses. In particular, we considered different test cases containing 10, 20, 41, 82 and 164 nurses, respectively. For each test case, we proportionally scaled the number of working nurses during each shift, whereas other requirements are not modified. Results are reported in Figure 1.

Concerning the encoding ENC1, we note that all instances have been solved within the allotted time. In particular, less than 3 minutes are needed to compute the schedule with 10 nurses and approximately 100 MB of memory consumption. Similar results have been obtained for computing the schedule with 20 nurses. Concerning the scheduling with 82 and 164 nurses, CLINGO computes the solution within 25 and 50 minutes, respectively, whereas its memory usage is 500 and 990 MB, respectively.

Results obtained with the encoding ENC2 are also very positive: CLINGO finds the optimal assignment within 60 minutes and 1024 MB for all considered test cases, with a peak of 28 minutes and 838 MB of memory usage when 164 nurses are considered.

5 Conclusion

In this paper we described two ASP encodings for addressing a variant of the NSP. Since ASP programs are executable specifications, we obtained a practical tool for supporting the head nurse in producing the schedule for the next year. We experimented with our implementation on real data provided by an Italian hospital, and all instances are solved within one hour with both encodings, even with more that 100 nurses.

Acknowledgments. We would like to thank Nextage srl for providing partial funding for this work. The funding has been provided in the framework of a research grant by the Liguria POR-FESR 2014-2020 programme.

References

1. Azaiez, M.N., Sharif, S.S.A.: A 0-1 goal programming model for nurse scheduling. Computers & OR 32, 491–507 (2005)
2. Bard, J.F., Purnomo, H.W.: Preference scheduling for nurses using column generation. European Journal of Operational Research 164(2), 510–534 (2005)
3. Aickelin, U., Dowsland, K.A.: An indirect genetic algorithm for a nurse-scheduling problem. Computers & OR 31(5), 761–778 (2004)
4. Topaloglu, S., Selim, H.: Nurse scheduling using fuzzy modeling approach. Fuzzy Sets and Systems 161(11), 1543–1563 (2010), http://dx.doi.org/10.1016/j.fss.2009.10.003
5. 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)
6. Millar, H.H., Kiragu, M.: Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. European Journal of Operational Research 104(3), 582 – 592 (1998)
7. 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)
8. Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems - a bibliographic survey. European Journal of Operational Research 151(3), 447–460 (2003)
9. 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
10. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Comput. 9(3/4), 365–386 (1991)
11. Alviano, M., Dodaro, C.: Anytime answer set optimization via unsatisfiable core shrinking. TPLP 16(5-6), 533–551 (2016)
12. 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)
13. Balduccini, M., Gelfond, M., Watson, R., Nogueira, M.: The usa-advisor: A case study in answer set planning. In: Eiter, T., Faber, W., Truszczynski, M. (eds.) LPNMR. LNCS, vol. 2173, pp. 439–442. Springer (2001)
14. Erdem, E., Öztok, U.: Generating explanations for biomedical queries. TPLP 15(1), 35–78 (2015)
15. Koponen, L., Oikarinen, E., Janhunen, T., Säilä, L.: Optimizing phylogenetic supertrees using answer set programming. TPLP 15(4-5), 604–619 (2015)
16. 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)
17. Abseher, M., Gebser, M., Musliu, N., Schaub, T., Woltran, S.: Shift design with answer set programming. Fundam. Inform. 147(1), 1–25 (2016)
18. 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)
19. 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)
20. Alviano, M., Dodaro, C., Leone, N., Ricca, F.: Advances in WASP. In: LPNMR. LNCS, vol. 9345, pp. 40–54. Springer (2015)