A proposition of a flexible framework for generating nurse rostering instances

Nguyen Dang · Christopher Stone · Ian Miguel

Abstract We propose ideas towards a framework for generating benchmark nurse rostering instances in an automated fashion. The framework makes use of a newly developed automated instance generation approach based on high-level constraint modelling and automated algorithm configuration. It allows a modeller to change the kind of instances produced simply by imposing constraints on the properties expected to be satisfied by the instances generated.

Keywords nurse rostering, constraint modelling · instance generation

1 Introduction

Nurse rostering problem [9,5,20] is one of the most extensively studied problems in operation research [18]. The problem consists of assigning nurses with a certain skill to a working shift of a day. The assignments spread over a planning horizon, normally from a week to a month. The number of nurses required for each triple of days, shifts and skills are given. There are often several hard constraints that need to be satisfied, such as avoiding under-staffing and impractical shift patterns (e.g., a nurse cannot work a Night shift followed by an Early shift). Besides, constraints that can be violated, but should be satisfied as much as possible are expressed as soft constraints. Some examples of soft constraints include the minimum and maximum number of consecutive working days that a nurse can be assigned to. There are several variants of nurse rostering problems, depending on how the hard and soft constraints are stated. We refer to [5,9,20] for an overview of the problem and its variants.

Several algorithmic approaches have been proposed for solving nurse rostering problems during the last three decades. They include mathematical programming [19], constraint programming [17], metaheuristics [8], hyper-heuristics [3] and hybrid approaches [6]. Each approach often has their own

School of Computer Science, University of St Andrews, UK

strength and weakness. For example, constraint programming approaches offer the advantage of being flexible, as new constraints can be easily supported without interfering into solvers' implementation. However, they may suffer from the issue of scalability, and in some cases cannot solve large-size instances effectively as problem-specific solving approaches [23]. Metaheuristic approaches often find good-quality solutions in a reasonable amount of time and have good scalability, but they lack of the ability to prove optimality of solutions found. Even approaches within the same group can have different performance depending on the problem variants and instance distributions [21].

Having benchmark instances with different levels of difficulties is crucial for assessing performance of different solving approaches, and for understanding their strength and weakness [22]. A variety of benchmark instances have been proposed and made publicly available for nurse rostering problems. The KAHO dataset[1] [20] is constructed using real-world data from two Belgian hospitals where for each ward three different scenarios are considered: normal, high amount of unforeseen work and unexpected absence of nurses. The Nottingham dataset[2] [4] consist of artificial and real instances collected worldwide. NSPLib[3] [24] provides an instance generator and large sets of nurse rostering instances. The First [12] and Second International Nurse Rostering Competitions [7] propose artificial benchmark instance sets derived from real-world data. Recently a unified data format, namely `XESTT` [4], for representing several nurse rostering benchmark datasets has been proposed [15].

The necessity of having rich and diverse benchmark instance sets for nurse rostering makes the idea of an automated instance generator plausible. Another motivation comes from the successful applications of automated algorithm configuration in recent years [13]. Imagine that we have a highly configurable nurse rostering solver. When a practitioner wants to adapt a solver to a specific nurse rostering problem in a certain context of a hospital, the first step is to have instances that reflect the pattern of the local context. An automated algorithm configuration tool, such as SMAC [14] or irace [16], can then be used to find the best parameter setting of the solver on the given instance distribution.

In [24], Vanhoucke and Maenhout proposed an instance generator that characterizes an instance through various complexity indicators. They included problem sizes, preference distribution measures, coverage distribution measures, and time related constraints. They implemented a dedicated procedure for generating instances with properties corresponding to specific indicators' values as parameters.

The generator in [24] covers a wide range of nurse rostering instances' aspects. However, due to the complexity and diversity of nurse rostering problems in practice, there can be additional characteristics on the instances that

[1] `https://people.cs.kuleuven.be/~pieter.smet/nurserostering.html`

[2] `http://www.schedulingbenchmarks.org/nurse.html`

[3] `https://www.projectmanagement.ugent.be/research/personnel_scheduling/nsp`

[4] `http://jeffreykingston.id.au/xestt/`

a practitioner might wish to add to reflect the local context. For example, in a normal department of a hospital, the demand for highly skilled nurses can be very high during the middle of the week, while being low during the beginning and the end of the week. On the other hand, in an intensive care department, the demand for highly skilled nurses are generally much higher on most days compared to the demand for less skilled nurses.

In this work, we propose an alternative approach for automated nurse rostering instance generation based on constraint modelling and automated algorithm configuration. The high-level constraint modelling language used by our approach, namely Essence [10], allows for the flexible specification of additional characteristics on instances generated. Moreover, the combination of constraint programming and automated configuration ensures feasibility and diversity of instances generated.

2 Methodology

The flexible framework for generating nurse rostering instances proposed in this paper is an extension of the automated instance generation system for constraint programming proposed in [1,2]. The system makes use of Essence, a modelling language designed for the specification of problems in combinatorial decision and optimisation [10]. This system takes an abstract specification of a problem in Essence and automatically converts it into a parametrised instance generator. Desirable properties of the instances being generated can be incorporated into the generator as constraints. The system comes paired with a constraint programming solver (`minion` [11]) that generates an instance by solving the generator specification, and an automated algorithm configurator (irace [16]) that allows efficient search for generator's parameter settings that cover *feasible instances* of desired properties.

Essence is a high-level constraint modelling language that supports several abstract data types. The language can capture the structure of a problem above the level of abstraction at which modelling decisions are made. In our context, this enables the modeller to easily add or remove specific constraints on properties the instances must satisfy, which consequently change the kind of instances produced by the (automatically created) instance generator.

The nurse rostering instance generation framework will include two components. The first one is a nurse rostering specification in Essence that incorporates all the indicators proposed in [24] as constraints. We will also add a number of additional properties observed on real-world instances. One example is the distribution of demand for highly-skilled nurses during a week. The properties encoded in the specification can be easily deactivated if not needed. As a first step, we will model the specification proposed in the Second International Nurse Rostering Competition [7]. The specification can also be modified to comply with other nurse rostering variants. The second component is the automated instance generation system, which receives as input the spec-

ification from the first component, and automatically generate a diverse set of feasible instances following the distribution indicated in the specification.

This is a work in progress and we plan to make the framework and the instances generated publicly available once they are ready. We believe that this approach will increase the diversity of scenarios covered by nurse rostering's benchmarks and help in the development of new algorithms applicable in real world healthcare.

Acknowledgements

We thank Guilherme Redeker and Juliana Bowles for their help on identifying patterns in real-world nurse rostering instances. This work was supported by the EPSRC grant EP/P015638/1 and used the Cirrus UK National Tier-2 HPC Service at EPCC (`http://www.cirrus.ac.uk`) funded by the University of Edinburgh and EPSRC (EP/P020267/1). Nguyen Dang is a Leverhulme Early Career Fellow.

References

1. Akgün, Ö., Dang, N., Miguel, I., Salamon, A.Z., Spracklen, P., Stone, C.: Discriminating instance generation from abstract specifications: A case study with cp and mip. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pp. 41–51. Springer (2020)
2. Akgün, Ö., Dang, N., Miguel, I., Salamon, A.Z., Stone, C.: Instance generation via generator instances. In: International Conference on Principles and Practice of Constraint Programming, pp. 3–19. Springer (2019)
3. Bilgin, B., Demeester, P., Misir, M., Vancroonenburg, W., Vanden Berghe, G., Wauters, T.: A hyper-heuristic combined with a greedy shuffle approach to the nurse rostering competition. In: the 8th International Conference on the Practice and Theory of Automated Timetabling (2010)
4. Burke, E.K., Curtois, T., Post, G., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. European journal of operational research **188**(2), 330–341 (2008)
5. Burke, E.K., De Causmaecker, P., Berghe, G.V., Van Landeghem, H.: The state of the art of nurse rostering. Journal of scheduling **7**(6), 441–499 (2004)
6. Burke, E.K., Li, J., Qu, R.: A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. European Journal of Operational Research **203**(2), 484–493 (2010)
7. Ceschia, S., Dang, N., De Causmaecker, P., Haspeslagh, S., Schaerf, A.: The second international nurse rostering competition. Annals of Operations Research **274**(1-2), 171–186 (2019)
8. De Causmaecker, P., Berghe, G.V.: Novel meta-heuristic approaches to nurse rostering problems in belgian hospitals (2004)
9. De Causmaecker, P., Berghe, G.V.: A categorisation of nurse rostering problems. Journal of Scheduling **14**(1), 3–16 (2011)
10. Frisch, A.M., Harvey, W., Jefferson, C., Martínez-Hernández, B., Miguel, I.: E ssence: A constraint language for specifying combinatorial problems. Constraints **13**(3), 268–306 (2008)
11. Gent, I.P., Jefferson, C., Miguel, I.: Minion: A fast scalable constraint solver. In: ECAI, vol. 141, pp. 98–102 (2006)

12. Haspeslagh, S., De Causmaecker, P., Stølevik, M., Schaerf, A.: First international nurse rostering competition 2010 (august 10-13, 2010, belfast, uk). In: PATAT 2010-Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, Belfast, Northern-Ireland, UK (2010)
13. Hoos, H.H.: Automated algorithm configuration and parameter tuning. In: Autonomous search, pp. 37–71. Springer (2011)
14. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration (extended version)
15. Kingston, J.H., Post, G., Vanden Berghe, G.: A unifed nurse rostering model based on xhstt. In: Proceedings of the 12th International Conference of the Practice and Theory of Automated Timetabling, vol. 12, pp. 81–96. EWG PATAT; Vienna (2018)
16. López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives **3**, 43–58 (2016). DOI 10.1016/j.orp.2016.09.002. URL `http://iridia.ulb.ac.be/irace/`
17. Nonobe, K.: Inrc2010: An approach using a general constraint optimization solver. The First International Nurse Rostering Competition (INRC 2010) (2010)
18. Rais, A., Viana, A.: Operations research in healthcare: a survey. International transactions in operational research **18**(1), 1–31 (2011)
19. Santos, H.G., Toffolo, T.A., Gomes, R.A., Ribas, S.: Integer programming techniques for the nurse rostering problem. Annals of Operations Research **239**(1), 225–251 (2016)
20. Smet, P., De Causmaecker, P., Bilgin, B., Berghe, G.V.: Nurse rostering: a complex example of personnel scheduling with perspectives. In: Automated Scheduling and Planning, pp. 129–153. Springer (2013)
21. Smith-Miles, K., Baatar, D., Wreford, B., Lewis, R.: Towards objective measures of algorithm performance across instance space. Computers & Operations Research **45**, 12–24 (2014)
22. Smith-Miles, K., Lopes, L.: Measuring instance difficulty for combinatorial optimization problems. Computers & Operations Research **39**(5), 875–889 (2012)
23. Teuschl, S.: Casual employee scheduling with constraint programming and metaheuristics. Ph.D. thesis, Wien (2020)
24. Vanhoucke, M., Maenhout, B.: On the characterization and generation of nurse scheduling problem instances. European Journal of Operational Research **196**(2), 457–467 (2009)