ALASPO: An Adaptive Large-Neighbourhood ASP Optimiser (Extended Abstract)

Thomas Eiter[1], Tobias Geibinger[1], Nelson Higuera[1], Nysret Musliu[1,2], Johannes Oetsch[1], and Daria Stepanova[3]

[1] Institute of Logic and Computation, TU Wien,
Favoritenstraße 9-11, 1040 Vienna, Austria,
[2] Christian Doppler Laboratory for Artificial Intelligence in Planning and Scheduling
{thomas.eiter,tobias.geibinger,nelson.ruiz,
nysret.musliu,johannes.oetsch}@tuwien.ac.at
[3] Bosch Center for AI, Robert Bosch Campus 1, 71272 Renningen, Germany
daria.stepanova@de.bosch.com

Abstract. While answer-set programming (ASP) is a successful approach to declarative problem solving, optimisation can still be a challenge for it. Large-neighbourhood search (LNS) is a metaheuristic technique where parts of a solution are alternately destroyed and reconstructed, which has high but untapped potential for ASP solving. We present the system ALASPO which implements Adaptive LNS for ASP Optimisation. ALASPO currently supports the ASP solver clingo, as well as its extensions clingo-dl and clingcon for difference and full integer constraints. Neighbourhoods can be defined in code or declaratively as part of the ASP encoding. Furthermore, ALASPO incorporates portfolios for the LNS operators along with self-adaptive selection strategies. This improves usability considerably at no loss of solution quality, but on the contrary often yields benefits. To demonstrate this, we evaluate ALASPO on different optimisation benchmarks.

Introduction. This extended abstract encompasses two recent publications [4,3]. In [4], we have introduced a framework for ASP optimisation that leverages *large-neighbourhood search* (LNS) [8,7], a powerful meta-heuristic where parts of a solution are destroyed and reconstructed in an attempt to improve an overall objective. In [3], relying on this framework, we have developed a system for ASP optimisation which utilises adaptive LNS [5,6] based on general and tailored destruction and reconstruction operators as well as different learning strategies.

The system is available at https://gitlab.tuwien.ac.at/kbs/BAI/alaspo.

Architecture & Functionality. ALASPO is a system for ASP optimisation with support for different ASP solvers, search configurations, and neighbourhood definitions. Figure 1 gives an overview of its components and their interaction.

At the heart of ALASPO lies an LNS loop, where an incumbent solution is repeatedly relaxed and reconstructed by an ASP solver to continuously obtain better objective values for the optimisation problem at hand.

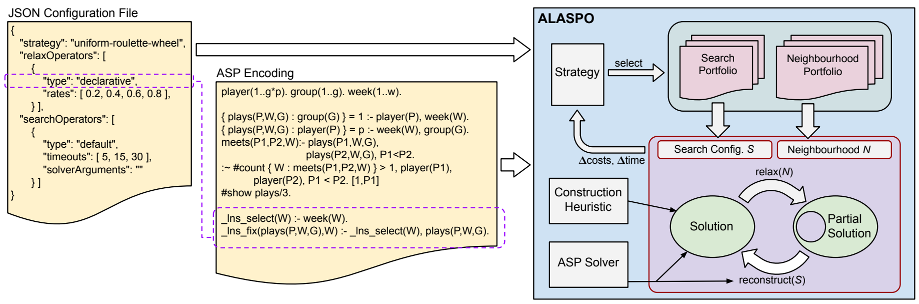

Fig. 1. Adaptive LNS for ASP in the system ALASPO.

An *initial solution* is generated by the ASP solver. Alternatively, it can be obtained by a custom procedure using a construction heuristic, which, however, is problem specific and must be provided via a Python 3 implementation.

In each iteration of the LNS loop, the currently best solution I is relaxed using a *neighbourhood operator* N, which is a procedure to select a subset of the atoms in I. For instance, N could pick 20% of the visible atoms at random. Then, the resulting partial solution is reconstructed using the ASP solver with a constraint to obtain a better objective value than I. This reconstruction depends on a search configuration S which defines solver options and a time limit. If a better solution is found within the time limit, it becomes the new incumbent, otherwise, I remains the best known solution. Which operators are chosen at each iteration, is based on a—potentially self-adaptive—*strategy*. ALASPO includes a simple learning strategy from the literature [5] as well as a novel strategy that attempts to escape a stuck search by varying relaxation rates and time limits.

The optimisation problem is formulated in ASP and stored in one or multiple input files. The currently supported ASP solvers are clingo, clingo-dl, and clingcon from the Potassco family.[4]

Evaluation & Conclusion. In the AAAI paper [4], experiments showed that the LNS framework with carefully selected neighborhoods and parameters, improves upon plain ASP optimisation on several benchmark problems like *Social Golfer*, *Travelling Salesman*, generating smallest sets of clues for *Sudoku*, an optimisation variant of the *Strategic Companies* problem, *Shift Design* [1] and a parallel machine scheduling problem [2].

In the KR paper [3], we have further demonstrated that ALASPO using adaptive strategies with reasonable portfolios achieves results that are competitive with the more tailored LNS approach used in previous work [4].

(Adaptive) LNS, and in particular ALASPO, has thus indeed the potential to enhance ASP optimisation in many applications; for some, this has already been demonstrated, others are planned for future work.

[4] https://potassco.org/.

Acknowledgements. This work was supported by funding from the Bosch Center for AI at Renningen, Germany.

Tobias Geibinger is a recipient of a DOC Fellowship of the Austrian Academy of Sciences at the Institute of Logic and Computation at the TU Wien.

References

1. Abseher, M., Gebser, M., Musliu, N., Schaub, T., Woltran, S.: Shift design with answer set programming. Fundamenta Informaticae **147**(1), 1–25 (2016)
2. Eiter, T., Geibinger, T., Musliu, N., Oetsch, J., Skocovsky, P., Stepanova, D.: Answer-set programming for lexicographical makespan optimisation in parallel machine scheduling. In: Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021) (2021)
3. Eiter, T., Geibinger, T., Ruiz, N.H., Musliu, N., Oetsch, J., Stepanova, D.: ALASPO: an adaptive large-neighbourhood ASP optimiser. In: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR 2022) (2022)
4. Eiter, T., Geibinger, T., Ruiz, N.H., Musliu, N., Oetsch, J., Stepanova, D.: Large-neighbourhood search for optimisation in answer-set solving. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022). pp. 5616–5625. AAAI Press (2022)
5. Laborie, P., Godard, D.: Self-adapting large neighborhood search: Application to single-mode scheduling problems. In Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2007) (2007)
6. Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Computers & Operations Research **34**(8), 2403–2435 (2007)
7. Pisinger, D., Ropke, S.: Large neighborhood search. In: Handbook of Metaheuristics, pp. 399–419. Springer (2010)
8. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming (CP 1998). LNCS, vol. 1520, pp. 417–431. Springer (1998)