Towards Inductive Learning of Domain-Specific Heuristics for ASP
(Extended Abstract)*

Richard Comploi-Taupe[1]

Siemens AG Österreich, Vienna, Austria
richard.taupe@siemens.com

1 Introduction

Answer Set Programming (ASP) [8,11] is a declarative problem-solving approach applied successfully in many industrial and scientific domains. For large and complex problems, however, domain-specific heuristics may be needed to achieve satisfactory performance [3,4].

Therefore, state-of-the-art ASP systems offer ways to integrate domain-specific heuristics in the solving process. An extension for WASP facilitates external *procedural* heuristics consulted at specific points during the solving process via an API [3]. *Declarative* specifications of domain-specific heuristics in the form of so-called *heuristic directives* are supported by CLINGO [6,7] and ALPHA [1,12,13].

However, suitable domain-specific heuristics must be established manually so far. Both human domain experts and ASP experts are required for their design. We present a first step toward the automatic learning of declarative heuristics.

Our core idea is to use Inductive Logic Programming (ILP) to learn declarative domain-specific heuristics from examples stemming from (near-)optimal answer sets of small but representative problem instances. These heuristics can then be used to improve solving performance and solution quality for larger, harder problem instances. Our experimental results are promising, indicating that this goal can be achieved.

2 Inductive Learning of Domain-Specific Heuristics

We now present our approach to the inductive learning of domain-specific heuristics for ASP.

We build on Inductive Logic Programming (ILP), which is an approach to learning a program that explains a set of examples given some background knowledge. ILASP [9,10] is a system capable of learning Answer Set Programs. ILASP operates on a *learning task*, which consists of three components: The *background knowledge* (an ASP program already known before learning), the *mode bias* (that

* This is an extended abstract of research already presented at HYDRA 2022 [2].

expresses which ASP programs can be learned), and the *examples* (which specify properties the learned program must satisfy).

The basic idea of our approach is to solve a small but representative instance of a problem, use the resulting answer set as a positive example for inductive learning, learn a set of definite rules, and transform the learned rules into declarative heuristic directives. These heuristics can then be used to speed up solving larger/harder instances of the same problem.

The rule space for ILASP is defined as follows in our approach:

– All predicates the solver will use for nondeterministic choices can be used in the heads of learned rules.
– All (other) predicates appearing in the original program can be used in the bodies of learned rules.
– The same placeholder is used several times, wherever a variable denotes the same real-world concept.

The background knowledge is the original program without any instance. Choice rules are not included, however, because we observed that ILASP only learns anything in our example domain when choice rules are not part of the background knowledge. We presume this is because we need to abstract away from the complete problem specification a bit to learn part of the missing information. Constraints are also not included because the rules we want to learn don't have to satisfy all constraints of the program. When used as heuristics, it suffices for them to give a general indication of what decisions might be useful during solving, even if some of these decisions will have to be backtracked.

As a positive example for learning, one answer set for a small but representative problem instance is used. In case the underlying problem is an optimisation problem, we propose to use a (near-)optimal answer set for this process, assuming that learning from better answer sets yields better heuristics.

We use context-dependent examples; the context is given by the problem instance. The set of inclusions corresponds to the whole answer set, and the set of exclusions is empty.

3 Results, Conclusions, and Future Work

For our experiments, we used the House Reconfiguration Problem (HRP) [5], an abstracted version of industrial (re)configuration problems. Heuristic directives for this problem were learned as described above. Since the HRP is an optimisation problem, we studied the solution quality achieved within 10 minutes of solving time, with and without the learned heuristics.

Experimental results are promising: Some instances could be solved only using the learned heuristics, and solution quality improved considerably on average. The fact that so far, we have only learned very simple heuristics and those already led to significant improvements is encouraging. Future work will show whether our method can be extended to learn more complex heuristics that can improve solving performance and solution quality even further.

References

1. Comploi-Taupe, R.: Speeding Up Lazy-Grounding Answer Set Solving. Ph.D. thesis, Alpen-Adria-Universität Klagenfurt (2021), https://digital.obvsg.at/urn/urn:nbn:at:at-ubk:1-41351
2. Comploi-Taupe, R.: Towards inductive learning of domain-specific heuristics for ASP. In: Bruno, P., Calimeri, F., Cauteruccio, F., Maratea, M., Terracina, G., Vallati, M. (eds.) Joint Proceedings of the 1st International Workshop on HYbrid Models for Coupling Deductive and Inductive ReAsoning (HYDRA 2022) and the 29th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion (RCRA 2022). CEUR Workshop Proceedings, vol. 3281, pp. 21–33. CEUR-WS.org (2022), https://ceur-ws.org/Vol-3281/paper3.pdf
3. 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). Theory Pract. Log. Program. **16**(5-6), 653–669 (2016). https://doi.org/10.1017/S1471068416000284
4. Falkner, A.A., Friedrich, G., Schekotihin, K., Taupe, R., Teppan, E.C.: Industrial applications of answer set programming. Künstliche Intell. **32**(2-3), 165–176 (2018). https://doi.org/10.1007/s13218-018-0548-6
5. Friedrich, G., Ryabokon, A., Falkner, A.A., Haselböck, A., Schenner, G., Schreiner, H.: (Re)configuration using answer set programming. In: Shchekotykhin, K.M., Jannach, D., Zanker, M. (eds.) Proceedings of the IJCAI 2011 Workshop on Configuration, Barcelona, Spain, July 16, 2011. CEUR Workshop Proceedings, vol. 755, pp. 17–24. CEUR-WS.org (2011), http://ceur-ws.org/Vol-755/paper03.pdf
6. Gebser, M., Kaminski, R., Kaufmann, B., Lindauer, M., Ostrowski, M., Romero, J., Schaub, T., Thiele, S., Wanko, P.: Potassco guide version 2.2.0 (2019), https://github.com/potassco/guide/releases/tag/v2.2.0
7. Gebser, M., Kaufmann, B., Romero, J., Otero, R., Schaub, T., Wanko, P.: Domain-specific heuristics in answer set programming. In: desJardins, M., Littman, M.L. (eds.) Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA. pp. 350–356. AAAI Press (2013), http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6278
8. Gelfond, M., Kahl, Y.: Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge University Press, New York, NY, USA (2014)
9. Law, M.: Conflict-driven inductive logic programming. Theory and Practice of Logic Programming (2022). https://doi.org/10.1017/S1471068422000011
10. Law, M., Russo, A., Broda, K.: The ILASP system for inductive learning of answer set programs. CoRR **abs/2005.00904** (2020)
11. Lifschitz, V.: Answer Set Programming. Springer (2019). https://doi.org/10.1007/978-3-030-24658-7
12. Taupe, R., Friedrich, G., Schekotihin, K., Weinzierl, A.: Solving configuration problems with ASP and declarative domain-specific heuristics. In: Aldanondo, M., Falkner, A.A., Felfernig, A., Stettinger, M. (eds.) Proceedings of the 23rd International Configuration Workshop (CWS/ConfWS 2021), Vienna, Austria, 16-17 September, 2021. CEUR Workshop Proceedings, vol. 2945, pp. 13–20. CEUR-WS.org (2021), http://ceur-ws.org/Vol-2945/21-RT-ConfWS21_paper_4.pdf
13. Taupe, R., Schekotihin, K., Schüller, P., Weinzierl, A., Friedrich, G.: Exploiting partial knowledge in declarative domain-specific heuristics for ASP. In: Bogaerts,

B., Erdem, E., Fodor, P., Formisano, A., Ianni, G., Inclezan, D., Vidal, G., Villanueva, A., Vos, M.D., Yang, F. (eds.) Proceedings 35th International Conference on Logic Programming (Technical Communications), ICLP 2019 Technical Communications, Las Cruces, NM, USA, September 20-25, 2019. EPTCS, vol. 306, pp. 22–35 (2019). https://doi.org/10.4204/EPTCS.306.9