Optimization problems in answer set programming

(invited talk)

Mario Alviano

DEMACS, University of Calabria, 87036 Rende (CS), Italy
alviano@mat.unical.it

Abstract. Answer set programming (ASP) is a declarative language for nonmonotonic reasoning based on stable model semantics, where stable models are classical models of the input program satisfying a stability condition: only necessary information is included in each of these models under the assumptions provided by the model itself for the unknown knowledge in the program, where unknown knowledge is encoded by means of default negation. Reasoning in presence of unknown knowledge is common for rational agents acting in the real world. It is also common that real world agents cannot meet all their desiderata, and therefore ASP programs may come with soft literals for representing numerical preferences over jointly incompatible conditions. Stable models are therefore associated with a cost given by the number of the unsatisfied soft literals, so that stable models of minimum cost are preferred. Algorithms and strategies for computing optimal stable models are reported in this paper, together with a brief discussion of their properties. Finally, the paper hints on how these algorithms can be extended to handle some qualitative preferences.

Keywords: answer set programming, boolean optimization, unsatisfiable core analysis

1 Extended abstract

Answer set programming (ASP) is a declarative language for nonmonotonic reasoning based on stable model semantics [1–5], and implemented by very efficient systems [6–8]. A stable models is a classical model of the input program that satisfies an additional condition, referred as *stability condition*. Specifically, a logic program may refer to *unknown knowledge* via default negation, whose interpretation is then fixed by the stable model candidate; stated differently, the stable model candidate provides an assumption on the truth or falsity of relevant unknown knowledge. On this new logic program, often referred in the literature as *program reduct*, the stability condition requires that only necessary information is included in the stable model candidate, which therefore must be subset-minimal for the program reduct in order to be a stable model of the original program.

Such an intuitive notion of stable model easily extends to expressive language constructs, among them aggregation functions [9, 10], a convenient linguistic extension for which the complexity of some reasoning tasks may raise [11], but fully supported in modern ASP systems [12, 13].

Several real world applications take advantage of ASP for fast prototyping and efficient evaluation. For example, USA-Advisor [14] is a decision support system for the Space Shuttle that was used for managing unpredicted failures of the reaction control system; in this setting, a collection of ASP programs were used to represent possible actions and their effects, so that several plans for restoring the reaction control system could be computed by an ASP solver. As another example, ASP was recently applied to nurse scheduling [15], where working hours of nurses of an hospital have to be scheduled subject to several constraints. A similar scheduling problem was also solved thanks to ASP in the Gioia Tauro sea port [16]. And the list of applications would go on with combinatorial auctions [17], phylogenetic supertrees [18], consistent query answering [19], and automatic configuration [20], just to mention a few of them.

Reasoning in presence of unknown knowledge is common for rational agents acting in the real world. It is also common that real world agents cannot meet all their desiderata, and therefore ASP programs may come with soft literals for representing numerical preferences over jointly incompatible conditions. Stable models are therefore associated with a cost given by the number of the unsatisfied soft literals, so that stable models of minimum cost are *preferred*. In fact, any stable model describes a plausible scenario for the knowledge represented in the input program, even if it may be only an admissible solution of non optimum cost. In fact, many rational agents would still accept suboptimal solutions, possibly with an estimate on the maximum distance to the optimum cost. This flexibility is also justified by the intrinsic complexity of the problem: the computation of an optimum stable model requires in general at least linearly many calls to a Σ_2^P oracle [21], and it is therefore practically unfeasible for the hardest instances.

Taking into account such a high computational complexity, a good algorithm for answer set optimization should produce better and better stable models during the computation of an optimum stable model. Algorithms having this property are called *anytime* in the literature [22, 23]. Unfortunately, the most efficient algorithms are not anytime by themselves: they are based on *unsatisfiable core* analysis [24], which means that they try to satisfy all soft literals, possibly replacing those in the input program with less restricting constraints until an optimum stable model is found. Prominent examples of these algorithms are OLL [25], PMRES [26], ONE and K [27]. However, anytime variants of these algorithms are obtained thanks to a simple observation [28, 29]: Unsatisfiable cores are often non-minimal, and their sizes can be significantly reduced by a few additional oracle calls, where each call may either return a smaller core, or a stable model possibly improving the current overestimate. Specifically, two strategies, referred to as *linear* and *reiterated progression based shrinking*, proved to provide significant performance gains to the ASP solver WASP [30, 31].

Finally, it turns out that unsatisfiable core analysis can be efficiently used also for the enumeration of models of circumscribed theories [32], which are essentially logic programs subject to a subset-minimality preference on a set of atoms. The idea is simple and intuitive: cardinality optimal stable models are also subset optimal stable models; these models are computed, and the theory extended by constraints that discard these models and those less preferred, so that the enumeration procedure can continue the search until all optimal models are computed. In order to obtain an efficient enumeration algorithm, some internal properties of the specific unsatisfaible core analysis procedure employed have to be taken into account. Specifically, unsatisfiable core analysis modifies the original propositional theory, which has to be restored several times during the computation; auxiliary atoms introduced by ONE and K can be used for this purpose [33]. Addressing the computation of models of circumscribed theories provides concrete strategies for solving several computational tasks of abstract argumentation frameworks [34, 35].

Acknowledgement

This research has been partially supported by the Italian Ministry for Economic Development (MISE) under project "PIUCultura – Paradigmi Innovativi per l'Utilizzo della Cultura" (n. F/020016/01-02/X27), and under project "Smarter Solutions in the Big Data World (S2BDW)" (n. F/050389/01-03/X32) funded within the call "HORIZON2020" PON I&C 2014-2020, and by Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM).

References

1. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. **9**(3/4) (1991) 365–386
2. Niemelä, I.: Logic programming with stable model semantics as constraint programming paradigm. Annals of Mathematics and Artificial Intelligence **25**(3–4) (1999) 241–273
3. Marek, V.W., Truszczyński, M.: Stable models and an alternative logic programming paradigm. In Apt, K.R., Marek, V.W., Truszczyński, M., Warren, D.S., eds.: The Logic Programming Paradigm – A 25-Year Perspective. Springer Verlag (1999) 375–398
4. Lifschitz, V.: Answer set programming and plan generation. Artificial Intelligence **138** (2002) 39–54
5. Gelfond, M., Kahl, Y.: Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge University Press (2014)
6. Gebser, M., Kaminski, R., Kaufmann, B., Romero, J., Schaub, T.: Progress in clasp series 3. In Calimeri, F., Ianni, G., Truszczynski, M., eds.: Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings. Volume 9345 of Lecture Notes in Computer Science., Springer (2015) 368–383

7. Alviano, M., Calimeri, F., Dodaro, C., Fuscà, D., Leone, N., Perri, S., Ricca, F., Veltri, P., Zangari, J.: The ASP system DLV2. In Balduccini, M., Janhunen, T., eds.: Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings. Volume 10377 of Lecture Notes in Computer Science., Springer (2017) 215–221
8. Lierler, Y., Maratea, M.: Cmodels-2: Sat-based answer set solver enhanced to non-tight programs. In Lifschitz, V., Niemelä, I., eds.: Logic Programming and Nonmonotonic Reasoning, 7th International Conference, LPNMR 2004, Fort Lauderdale, FL, USA, January 6-8, 2004, Proceedings. Volume 2923 of Lecture Notes in Computer Science., Springer (2004) 346–350
9. Faber, W., Pfeifer, G., Leone, N.: Semantics and complexity of recursive aggregates in answer set programming. Artif. Intell. **175**(1) (2011) 278–298
10. Ferraris, P.: Logic programs with propositional connectives and aggregates. ACM Trans. Comput. Log. **12**(4) (2011) 25:1–25:40
11. Alviano, M., Faber, W.: The complexity boundary of answer set programming with generalized atoms under the FLP semantics. In Cabalar, P., Son, T.C., eds.: Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings. Volume 8148 of Lecture Notes in Computer Science., Springer (2013) 67–72
12. Alviano, M., Faber, W., Gebser, M.: Rewriting recursive aggregates in answer set programming: back to monotonicity. TPLP **15**(4-5) (2015) 559–573
13. Alviano, M.: Evaluating answer set programming with non-convex recursive aggregates. Fundam. Inform. **149**(1-2) (2016) 1–34
14. 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.: Logic Programming and Nonmonotonic Reasoning, 6th International Conference, LPNMR 2001, Vienna, Austria, September 17-19, 2001, Proceedings. Volume 2173 of Lecture Notes in Computer Science., Springer (2001) 439–442
15. Alviano, M., Dodaro, C., Maratea, M.: An advanced answer set programming encoding for nurse scheduling. In: AI*IA 2017: Advances in Artificial Intelligence - XVIth International Conference of the Italian Association for Artificial Intelligence, Bari, Italy, November 14-17, 2017, Proceedings. Lecture Notes in Computer Science, Springer (2017) To appear.
16. Ricca, F., Grasso, G., Alviano, M., Manna, M., Lio, V., Iiritano, S., Leone, N.: Team-building with answer set programming in the gioia-tauro seaport. TPLP **12**(3) (2012) 361–381
17. Baral, C., Uyan, C.: Declarative specification and solution of combinatorial auctions using logic programming. In Eiter, T., Faber, W., Truszczynski, M., eds.: Logic Programming and Nonmonotonic Reasoning, 6th International Conference, LPNMR 2001, Vienna, Austria, September 17-19, 2001, Proceedings. Volume 2173 of Lecture Notes in Computer Science., Springer (2001) 186–199
18. Koponen, L., Oikarinen, E., Janhunen, T., Säilä, L.: Optimizing phylogenetic supertrees using answer set programming. TPLP **15**(4-5) (2015) 604–619
19. Marileo, M.C., Bertossi, L.E.: The consistency extractor system: Answer set programs for consistent query answering in databases. Data Knowl. Eng. **69**(6) (2010) 545–572
20. Gebser, M., Ryabokon, A., Schenner, G.: Combining heuristics for configuration problems using answer set programming. In Calimeri, F., Ianni, G., Truszczynski, M., eds.: Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings. Volume 9345 of Lecture Notes in Computer Science., Springer (2015) 384–397

21. Buccafurri, F., Leone, N., Rullo, P.: Enhancing Disjunctive Datalog by Constraints. IEEE Trans. Knowl. Data Eng. **12**(5) (2000) 845–860
22. Alviano, M., Dodaro, C., Ricca, F.: Anytime Computation of Cautious Consequences in Answer Set Programming. TPLP **14**(4-5) (2014) 755–770
23. Bliem, B., Kaufmann, B., Schaub, T., Woltran, S.: ASP for Anytime Dynamic Programming on Tree Decompositions. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, AAAI Press (2016)
24. Alviano, M., Dodaro, C., Marques-Silva, J., Ricca, F.: Optimum stable model search: algorithms and implementation. Journal of Logic and Computation (2015)
25. Andres, B., Kaufmann, B., Matheis, O., Schaub, T.: Unsatisfiability-based optimization in clasp. In Dovier, A., Costa, V.S., eds.: Technical Communications of the 28th International Conference on Logic Programming, ICLP 2012. Volume 17 of LIPIcs., Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012) 211–221
26. Narodytska, N., Bacchus, F.: Maximum satisfiability using core-guided maxsat resolution. In Brodley, C.E., Stone, P., eds.: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada., AAAI Press (2014) 2717–2723
27. Alviano, M., Dodaro, C., Ricca, F.: A maxsat algorithm using cardinality constraints of bounded size. In Yang, Q., Wooldridge, M., eds.: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press (2015) 2677–2683
28. Alviano, M., Dodaro, C.: Anytime answer set optimization via unsatisfiable core shrinking. TPLP **16**(5-6) (2016) 533–551
29. Alviano, M., Dodaro, C.: Unsatisfiable core shrinking for anytime answer set optimization. In Sierra, C., ed.: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ijcai.org (2017) 4781–4785
30. Alviano, M., Dodaro, C., Leone, N., Ricca, F.: Advances in WASP. In Calimeri, F., Ianni, G., Truszczynski, M., eds.: Proceedings of Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015. Volume 9345 of LNCS., Springer (2015) 40–54
31. Alviano, M., Dodaro, C.: Answer set enumeration via assumption literals. In Adorni, G., Cagnoni, S., Gori, M., Maratea, M., eds.: AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Proceedings. Volume 10037 of LNCS., Springer (2016) 149–163
32. Lifschitz, V.: On the satisfiability of circumscription. Artif. Intell. **28**(1) (1986) 17–27
33. Alviano, M.: Model enumeration in propositional circumscription via unsatisfiable core analysis. CoRR **abs/1707.01423** (2017)
34. Alviano, M.: The ingredients of the argumentation reasoner pyglaf: python, circumscription, and glucose to taste. In: Proceedings of the 24th RCRA International Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion 2017 (RCRA 2017) A workshop of the XVI International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 14, 2017. (2017) To appear.
35. Alviano, M., ed.: Technical Communications of the 33rd International Conference on Logic Programming, ICLP 2017 TCs, August 28-September 1, 2017, Melbourne, Australia. (2017) To appear.