AAAI-13 Author Response Open (paper 209)

1 message

AAAI-13 <aaai13@easychair.org> Mon, Mar 11, 2013 at 11:28 PM
To: Torsten Schaub <torsten@cs.uni-potsdam.de>

Dear Torsten,

Thank you for your submission to the AAAI-13. The AAAI-13 review response period will be between NOW and March 13 at 5:00PM PDT.

During this time, you will have access to the current state of your reviews and have the opportunity to submit a response of up to 500 words. Please keep in mind the following during this process:

* The response must focus on any factual errors in the reviews and any questions posed by the reviewers. It must not provide new research results or reformulate the presentation. Try to be as concise and to the point as possible.

* The review response period is an opportunity to react to the reviews, but not a requirement to do so. Thus, if you feel the reviews are accurate and the reviewers have not asked any questions, then you should not respond.

* The reviews are as submitted by the PC members, without any coordination between them. Thus, there may be inconsistencies. Furthermore, these are not the final versions of the reviews. The reviews will be updated to take into account the discussions at the program committee meeting, and we may find it necessary to solicit other outside reviews after the review response period.

* The program committee will read your responses carefully and take this information into account during the discussions. On the other hand, the program committee will not directly respond to your responses, either before the program committee meeting or in the final versions of the reviews.

* Your response will be seen by all PC members who have access to the discussion of your paper, so please try to be polite and constructive.

The reviews on your paper are attached to this letter. To submit your response you should log on the EasyChair Web site (https://www.easychair.org/account/signin.cgi?conf=aaai13) and select your submission on the menu. The author response period ends promptly at 5:00pm PDT on Wednesday, March 13 and no author feedback will be allowed after that time.

Thank you,

Marie desJardins and Michael Littman
AAAI-13 Program Cochairs

----------------------- REVIEW 1 ---------------------
PAPER: 209
TITLE: Domain-specific Heuristics in Answer Set Programming
AUTHORS: (anonymous)

1. Technical quality: 3 (adequate, not a basis for accepting or rejecting the paper)
2. Experimental analysis: 3 (adequate, not a basis for accepting or rejecting the paper (or not applicable))
3. Formal analysis: 3 (adequate, not a basis for accepting or rejecting the paper (or not applicable))
4. Clarity/presentation: 3 (adequate, not a basis for accepting or rejecting the paper)
5. Novelty/innovation of question addressed: 3 (adequate, not a basis for accepting or rejecting the paper)
6. Novelty/innovation of solution proposed: 4 (positive, a factor in accepting the paper)
7. Breadth of interest to the AI community: 3 (adequate, not a basis for accepting or rejecting the paper (or not applicable))
8. Potential for impact to practical applications: 3 (adequate, not a basis for accepting or rejecting the paper

(or not applicable))
SUMMARY RATING: 1 (Weak acceptance. No 1, 5 or 6 in any category, overall above 3.)

----------- QUALITY JUSTIFICATION -----------
The paper presents an approach for integrating domain-specific heuristics into ASP solving. The approach is based on the use of heuristic atoms and rules in logic programs, and is implemented by modifying the heuristic functions of an existing algorithm (CDCL). The authors present clearly and in detail how the new heuristic predicate that they introduce are used in the non-deterministic assignment of literals. WIth some small examples, they describe the use of each of the four different heuristic modifiers. There are, though, some questions, for which the answers are not clear: how can someone select the appropriate heuristic rules for a specific problem? what is the scope of problems/programs that this approach is appropriate for? overall, it would help if the authors could give a bigger running example to describe their approach, instead of the small examples that they give for each different modifier.

The results of the experiments show that the proposed approach improves the performance of ASP reasoning in most cases. The authors comment on the results of the experiments, without, though, explaining, why some of the heuristics perform better in some problems and worse in others. Without a formal analysis on the selection of heuristics or some explanation of the results of the experiments, it is not clear how one could select the appropriate heuristics for a given problem - and by randomly selecting heuristics, someone may end up with a worse performance. The authors admit in the end that they just provide indications on the prospect of this approach, and that more systematic studies are needed to exploit the full power of the approach.

----------- INNOVATION JUSTIFICATION -----------
The problem of optimising algorithms for ASP solving is not new. It is, however, very challenging, and there are many different approaches to solve it, many of which are cited in the paper. Other similar approaches that the authors do not mention are:

Wolfgang Faber, Nicola Leone, Gerald Pfeifer: Experimenting with Heuristics for Answer Set Programming. IJCAI 2001: 635-640

Wolfgang Faber, Nicola Leone, Marco Maratea, Francesco Ricca: Experimenting with Look-Back Heuristics for Hard ASP Programs. LPNMR 2007: 110-122

Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. A Comparison of Heuristics for Answer Set Programming. In Proceedings of the 5th Dutch-German Workshop on Nonmonotonic Reasoning Techniques and their Applications (DGNMR 2001), pages 64-75, April 2001.

The novelty of the proposed approach is that it is not based on the modification of an ASP solver, but rather on extending the logic program with appropriate rules that contain the heuristic predicate. Although the idea has not been fully analyzed and evaluated yet, it is interesting and the initial results seem promising.

----------- IMPACT JUSTIFICATION -----------
Optimising ASP solvers is of interest to the AI community, because of this challenging characteristics but also because of the application of ASP. Although the results of the experiments show that the proposed approach is promising, it should be further analysed before its potential is actually proved and fully exploited. The authors have to explain how one can select the appropriate heuristic rules for a given problem, either by further analysing their approach or by appropriately interpreting and analysing the results of the experiments. They also have to define clearly the scope of problems that they can solve, and the range of applications that could benefit from this approach. For example, is it only relevant to planning problems? If not, how could someone adapt it to other types of problems?

----------- SUMMARY OF REVIEW FOR AUTHORS -----------
Overall, the paper presents an approach for optimising ASP solving using domain-specific heuristics. The proposed methodology is novel in ASP and the initial results are promising. However, the authors should better present it through one big example, so that the scope becomes more clear. Also, they should further analyse their approach, so that it's also clear how one can select the appropriate heuristics for a given problem, and convince in this way about the real potential of the approach.

---------------------- REVIEW 2 --------------------
PAPER: 209
TITLE: Domain-specific Heuristics in Answer Set Programming
AUTHORS: (anonymous)

1. Technical quality: 4 (positive, a factor in accepting the paper)
2. Experimental analysis: 2 (problematic, a factor in rejecting the paper)
3. Formal analysis: 4 (positive, a factor in accepting the paper)
4. Clarity/presentation: 2 (problematic, a factor in rejecting the paper)
5. Novelty/innovation of question addressed: 4 (positive, a factor in accepting the paper)
6. Novelty/innovation of solution proposed: 4 (positive, a factor in accepting the paper)
7. Breadth of interest to the AI community: 4 (positive, a factor in accepting the paper)
8. Potential for impact to practical applications: 3 (adequate, not a basis for accepting or rejecting the paper (or not applicable))

SUMMARY RATING: 1 (Weak acceptance. No 1, 5 or 6 in any category, overall above 3.)

----------- QUALITY JUSTIFICATION -----------
The paper was very difficult to follow. Descriptions of previous work, and even the problem being solved, were exceptionally brief. The notation used was also assumed to be understood, with a reference to another paper provided in place of an actual explanation. There are several issues with the clarity of the writing with the work presented in the paper as well. Phrases like "Thus, while the atom occ(climb,1) is preferably made true, false should rather be assigned to occ(climb,1)" appear with some frequency. Basically, things that seem contradictory at first glance and take a bit of puzzling to figure out the meaning of the authors.

The authors start of the paper by making the bold claim that domain specific solvers never need to be made again. That's a really strong claim, and it's just not supported by the empirical evaluation. Specifically, we never see a comparison to a domain specific solver in the evaluation. Further, we don't see a proof that the language provided here can provide any desirable heuristics, nor are we presented with any discussion of the kinds of heuristic information used by domain-specific solvers.

That said, the discussion of the heuristics provided by this technique, how and why they work, and what kinds of expressive power they give you is very thorough. You just have to work for it a bit harder than maybe is needed.

That sum of all runtimes are presented rather than mean runtimes seems a bit odd to me. I understand the interest in the time required to solve an entire suite of problems, but I've always been interested in runtimes on single instances. It would be nice to see both.

----------- INNOVATION JUSTIFICATION -----------
The idea of not wanting to write a large set of special-purpose solvers is basically the corner stone of AI. So, the problem being addressed isn't novel on its own. However, it feels like people are focusing on ever narrower aspects of particular domains and problems, and to see a paper come in and try to go in the other direction, towards the more general, is refreshing.

Trying to extrapolate heuristic information from the search space during solving is also not new. Heuristic search planners have been doing it for the last few years. However, doing it by application of rules like these is something I've not seen the like of before, and it's very interesting.

----------- IMPACT JUSTIFICATION -----------
People in the automatic configuration of algorithms community may be very excited about this work.
 Tweaking the various parameters of solvers to get better run-times is one thing, but tuning the heuristic information for solving a family of problems based on performance will be a new and exciting avenue for them to follow.

----------- SUMMARY OF REVIEW FOR AUTHORS -----------
This paper presents a new technique for encoding domain-specific heuristic information in a way where it can be used by a non-domain-specific solver. The description of the technique is hard to follow for anyone not already intimately familiar with answer set programming, and beyond that several bits of the paper require multiple reads as the writing itself is sometimes unclear. The paper makes some very strong claims, namely the end of domain-specific solvers, which it doesn't really deliver on. What it does present, a language for encoding heuristic information in answer set programming, is very interesting and is likely to open up new avenues of research in both answer-set programming as well as automatic algorithm configuration.

---------------------- REVIEW 3 --------------------
PAPER: 209
TITLE: Domain-specific Heuristics in Answer Set Programming
AUTHORS: (anonymous)

1. Technical quality: 3 (adequate, not a basis for accepting or rejecting the paper)

2. Experimental analysis: 4 (positive, a factor in accepting the paper)
3. Formal analysis: 2 (problematic, a factor in rejecting the paper)
4. Clarity/presentation: 3 (adequate, not a basis for accepting or rejecting the paper)
5. Novelty/innovation of question addressed: 3 (adequate, not a basis for accepting or rejecting the paper)
6. Novelty/innovation of solution proposed: 4 (positive, a factor in accepting the paper)
7. Breadth of interest to the AI community: 4 (positive, a factor in accepting the paper)
8. Potential for impact to practical applications: 4 (positive, a factor in accepting the paper)
SUMMARY RATING: 1 (Weak acceptance. No 1, 5 or 6 in any category, overall above 3.)

----------- QUALITY JUSTIFICATION -----------
The usefulness of heuristics to guide search of Clasp sometimes, is illustrated by various experiments. The results are promising for development/improvement of ASP solvers. Although, it is not clear how to obtain heuristics that will lead to better computational efficiency.

The authors mention, just before the experimental evaluation, some quarantees about preserving answer sets for the main program, as heuristic rules are added to it. However, no theoretical result is provided about that.

----------- INNOVATION JUSTIFICATION -----------
It is interesting that the heuristic functions be defined in ASP as well, in addition to the main ASP program describing the problem, and use these heuristic functions to modify the search of the ASP solver.

In that sense, the presented work is relevant to the following study where some externally defined functions are used to guide the search:

Cakmak et al.: Computing weighted solutions in ASP: representation-based method vs. search-based method. Ann. Math. Artif. Intell. 62(3-4): 219-258 (2011)

----------- IMPACT JUSTIFICATION -----------
ASP provides a powerful knowledge representation and reasoning tool for various real-world applications, includig planning, diagnosis, natural language understanding, robotics, etc.

----------- SUMMARY OF REVIEW FOR AUTHORS -----------
It is interesting to define heuristic function in ASP and use them to modify the search of an ASP solver for better computational efficiency results. Considering the usefulness of the proposed method illustrated by some examples and also various applications of ASP that may make use of such an ASP solver, this study is important to ASP and other areas as well.

On the other hand, since ASP provides a formal framework for representaton and reasoning, it would be good to provide some technical results about the requirements on heuristic functions (e.g., whatif the rules defining heuristic functions are contradictory with each other, can we use aggregates to defined heuristic functions, can we use heuristic functions in the problem description), and some properties of heuristic functions (e.g., similar to admisssibility/consistency of heuristic functions used for search).

It will be good to discuss under what conditions and for what kind of problems it is a good idea to use the proposed method, compared to 1) encoding domain-specific heuristics as part of the main problem description, maybe utilizing preferences, weak cnstraints, consistency restoring rules, etc. and 2) defining domain-specific heuristics externally (not in ASP) and using them to compute weighted solutions.

It will be good to emphasize the usefulness of defining heuristic functions in ASP (for that some more sophisticated definitions would be useful, is possible).

Most of the examples and discussion are around planning problems. It may be good to discuss the formulation and use of heuristic functions used in planning, as part of the proposed approach to solve planning problems.

The title is misleading: The presented work is more about the use of some domain-specific heuristics in ASP solvers (rather than ASP representations of problems).

---------------------- REVIEW 4 ---------------------
PAPER: 209
TITLE: Domain-specific Heuristics in Answer Set Programming
AUTHORS: (anonymous)

1. Technical quality: 4 (positive, a factor in accepting the paper)
2. Experimental analysis: 4 (positive, a factor in accepting the paper)
3. Formal analysis: 4 (positive, a factor in accepting the paper)
4. Clarity/presentation: 4 (positive, a factor in accepting the paper)
5. Novelty/innovation of question addressed: 3 (adequate, not a basis for accepting or rejecting the paper)
6. Novelty/innovation of solution proposed: 4 (positive, a factor in accepting the paper)
7. Breadth of interest to the AI community: 3 (adequate, not a basis for accepting or rejecting the paper (or not applicable))
8. Potential for impact to practical applications: 4 (positive, a factor in accepting the paper)

SUMMARY RATING: 1 (Weak acceptance. No 1, 5 or 6 in any category, overall above 3.)

----------- QUALITY JUSTIFICATION -----------
TECHNICAL QUALITY: The addition of domain-dependent heuristics in ASP is an interesting result and the technical extensions are adequately described in the paper. I particularly like the connection to planning heuristics, especially since the ability to model planning domains seemed to be of interest to the authors. While the main technical details of the paper appear to be correct, it does raise a couple questions.

- The authors primarily focus on look-back heuristics while look-ahead heuristics have proven to be quite successful, especially in the planning community. My guess is that these are possible but would probably require more support at the solver level. Is this the case?

- I was wondering if the authors were familiar with the work of "Baier, Fritz, and McIlraith (2007). Exploiting procedural domain control knowledge in state-of-the-art planners. Proc. of ICAPS 2007"? Are any of these techniques applicable to your approach?

- While the present paper focuses on domain-specific heuristics, what are the challenges of using domain-independent heuristics, or are such methods better served by changes at the solver level? With the number of planning examples in the paper I can't help but think about the domain independent heuristic methods used in the planning community (namely something like FF's relaxed plan heuristics). Are there any opportunities for adapting these methods in an approach like the one in the paper?

EXPERIMENTAL ANALYSIS: The paper presents a series of experiments using planning problems from the IPC and abductive problems with optimisation. Especially in the case of the IPC domains, a variety of heuristics are tested and compared across the problem domains. The use of real PDDL domains (converted to ASP) and planning-inspired heuristics is an interesting result. My one complaint here, however, is the presentation of data in Table 3. You appear to be testing on multiple problem instances of varying plan lengths, however, only the average runtime is given. Is this a useful measure given that the solution times may vary widely between different problem instances (i.e., different problem sizes)? Also, I assume that the average runtime only covers successful instances?

FORMAL ANALYSIS: The paper includes a formal description of the proposed heuristic extension and how it is incorporated into the language. A discussion of some of the tested heuristics is also given. Given the comparison with planning heuristics (especially, those of Rintanen 2011, 2012), I would have liked to have seen a few more details of these approaches, especially since some of them differ from more mainstream (domain-independent) planning heuristics. A more formal account of the limitations of lookback heuristics in this setting (or the problems of using lookahead heuristics at the declarative level) might also be useful for the reader.

CLARITY/PRESENTATION: Overall, the paper is well written and is understandable. The balance between theory and experimentation is about right. My only small gripe from a presentation point of view was the description of the "decide" process on page 2 and page 4. I would have preferred if this was set apart as a separate algorithm rather than taken out of context in this way. One other small gripe: I'm not sure why the word "the" is referenced as Footnote 6.

----------- INNOVATION JUSTIFICATION -----------
In some sense, the question addressed in the paper is an obvious one, given the popularity of heuristic approaches in many fields of AI, and at first glance the solution doesn't appear to be particularly novel. However, I was impressed by the breadth of heuristics that could be modelled with this approach, especially those based on planning approaches like (Rintanen 2011), which help demonstrate the potential of this extension.

----------- IMPACT JUSTIFICATION -----------
This paper is primarily of interest to answer set programming researchers, however, its focus on incorporating domain-specific heuristics (in particular, using planning examples) into ASP encodings may widen its appeal. Heuristic approaches have been an active area of research in many AI-related fields and in some cases, these techniques have had significant impacts (e.g., heuristic planners). As such, the approach in the paper has the potential to improve practical applications based on ASP techniques.

----------- SUMMARY OF REVIEW FOR AUTHORS -----------
This paper addresses the problem of introducing domain-specific heuristics into ASP solvers, by defining a language for specifying heuristics at the declarative level. The new facility is expressive enough to accommodate a range of heuristics, including those inspired by heuristic approaches in the planning community. A description of the required extensions is given at the formal and algorithmic levels and a set of empirical results are presented using common benchmarks, abductive domains, and IPC problems.

Although the heuristic extension is somewhat straightforward, the empirical results are promising and indicate a useful step forward. I particularly liked the references to heuristics arising in the planning community (i.e., the work of Rintanen). However, the approach does have its limitations since the heuristic must be defined at the declarative level (rather than within the solver), and in the present paper restricts its attention to lookback heuristics. Nevertheless, the paper is well written and the result is potentially useful to the wider ASP community.