Quo Vadis, IPC-4? — Proposals for the Classical Part of the 4th International Planning Competition

Stefan Edelkamp
Fachbereich Informatik
Baroper Str. 301, GB IV
44221 Dortmund, Germany
stefan.edelkamp@cs.uni-dortmund.de
+49 (231) 755-5809

Jörg Hoffmann
Institut für Informatik
Georges-Köhler-Allee, Geb. 52
79110 Freiburg, Germany
hoffmann@informatik.uni-freiburg.de
+49 (761) 203-8229

Abstract

The 4th International Planning Competition, IPC-4, will take place alongside ICAPS'04 as a continuation of the previous competition events. The intention is to divide the competition into three separate parts, one for *classical* planning, one for *probabilistic* planning, and one for *non-deterministic* planning. As the co-chairs of the classical part, we give an outline of our proposals for that part's organization. Concerning further language extensions, our standpoint is that PDDL 2.1 still provides a lot of challenges, and no or at most moderate extensions are appropriate at this point in time. Concerning benchmark domains, we want to lay stress on a more careful choice of these, specifically we want to move towards real-world problems, and diverse problem structures. Concerning evaluation, we want to lay more emphasis on plan quality, in particular on optimality guarantees. Concerning organizational aspects, our main focus is to make the competition results more widely perceivable and understandable at conference-time.

Introduction

The International Planning Competition has been a biannual event, hosted at the Artificial Intelligence Planning and Scheduling Conference series. The objectives of the competition are to provide a forum for empirical comparison of planning systems, to highlight challenges to the community in the form of problems at the edge of current capabilities, to propose new directions for research and to provide a core of common benchmark problems and a representation formalism that can aid in the comparison and evaluation of planning systems. Although the series has a competitive style (individual systems are identified for exceptional performance at the event itself), the focus is on data-collection and presentation, with interpretation of results being understated. The real goal of the competition is to make as much data as possible available to the community.

The competition event began in 1998 when Drew McDermott and a committee created a common specification language and a collection of problems forming a first benchmark. Five systems participated in the competition. In 2000, Fahiem Bacchus continued this work, and the event attracted 16 competitors. The event was extended to include both fully automatic and hand-tailored planning systems. Both STRIPS and ADL domains were used but no further extensions were made to the language. In 2002, the competition was run by Derek Long and Maria Fox. The event attracted 14 competitors, and focussed on planning in temporal and metric domains. McDermott's PDDL was not expressive enough to enable the modelling of durative actions and continuous resource consumption, so it was extended to enable the modelling of these features. The resulting competition language PDDL2.1 consisted of three levels of expressive power. Level 1 was ADL planning as before, level 2 added numerical variables, level 3 added durational constructs.

The 4th International Planning Competition, IPC-4, will be hosted at ICAPS'04. IPC-4 will build on the previous efforts, in particular the language PDDL2.1. The competition event will be extended and revised in several respects. The intention is to divide the competition into three separate parts: in addition to a continuation of the previous, *classical*, competitions, there shall be parts for *probabilistic* and *non-deterministic* planning. In the latter two parts, the main objective of the event will be to introduce a common representation language for the respective fields, and to establish some first benchmarks and results. The probabilistic part will be organized by Michael Littman, for the non-deterministic part an organizer has yet to be found. The classical part of IPC-4 will be organized by the authors. Several revisions to the previous classical competitions are planned. Most importantly, we propose a more careful choice of benchmark domains; in particular, a serious effort will be made towards including a number of real-world problems as testing domains in the competition. Also, revisions to the way of results presentation (in a more directly perceivable way) and evaluation (with more emphasis on plan quality) are intended. Language extensions (to the classical setting) we propose to handle more conservatively: the competition domains should best stay within the features of PDDL2.1, or moderate extensions thereof.

In what follows, we describe our proposals for the classical part of IPC-4 in more detail. The sections of the paper are, in turn, concerned with language extensions, benchmark domains, competition results evaluation, and organizational aspects of IPC-4 (including data collection and presentation). We close the paper with a section containing a few summarizing remarks.

Language Extensions

We interpret PDDL2.1 as an agreed fundament for all further language extensions. As said above, our standpoint is that IPC-3 has already brought sufficiently many new features for the time being, and that the planning community should be given time to catch up. To name a few things that we consider worth doing before making further significant language extensions:

- Consolidate the existing techniques – the algorithms used by the IPC-3 participants have only just scratched the surface of what can be done within PDDL2.1, e.g. by applying and / or combining existing approaches in the new context.
- Understand the results – IPC-3 provides the first publicly available data for a large-scale experiment in numerical and durational planning, and it will take time to make sense of these, e.g. by analyses such as those given for classical benchmark domains by Helmert (2003) and Hoffmann (2002).
- Develop techniques for handling features that have yet hardly been adressed – like, e.g., flexible optimization criteria, which are taken into account by only very few existing systems (to the best of the authors' knowledge, LPG (Gerevini & Serina 2003), Mips (Edelkamp 2003c), and Metric-FF (Hoffmann 2003)), and for which there is *no* system in existence that can find optimal solutions.

That said, let us state that we are not categorically excluding extensions to PDDL2.1 for the classical part of IPC-4. As explained in more detail below, an effort will be made to include real-world problems as benchmarks into the competition. So if a new language feature is practically motivated in the sense that it can arguably facilitate the modelling of phenomena relevant in real-world applications, and moderate in the sense that the feature seems reasonably easy to deal with in the more wide-spread planning approaches, then including the language feature into the 2004 competition language is an option worth considering. There are various ideas for such language features. Some possibilities that might be worth considering are:

- Axioms (aka derived predicates). These have been a part of original PDDL, but have never been used. They provide a natural means for modelling, e.g., updates on the transitive closure of a relation, as occur in structures such as paths or flows. We have come up with a formal semantics for axioms, easily checkable conditions for them to be well-defined, as well as a proof that they can not be compiled away without significant costs (Thiebaux, Hoffmann, & Nebel 2003a; 2003b). Integrating an explicit treatment of axioms is trivial within a forward search (but might be more complicated in other approaches).
- References to the current time, i.e. a "look on the clock" in operator or goal specifications. This feature is a natural way of modelling the correct optimization function in one of the domains we are considering as a competition benchmark. The feature can be realized by a predefined numerical variable ("current-time"), and is easy to deal with in any approach that fixes action execution times at plan-time (in least-commitment approaches the feature might be more problematic). An idea for a moderate use of the feature is to restrict the use of "current-time" such that it can only influence the value of the optimization expression, not the truth value of any pre- or goal-condition.
- Deterministic exogenous events, i.e. statements of the sort "literal l will become true at time point t, independently of the actions taken by the planner". Such statements are necessary to model the Satellite and Rovers domains, versions of which were already used at IPC-3, more adequately (by stating the time windows within which certain phenomena in space, e.g. a moon, become observable). There seem to be straightforward (and inexpensive) ways of compiling deterministic exogenous events down into the PDDL2.1 language features (imagine an action that must immediately be applied, that has duration t, and that makes literal l true), but nevertheless it might make sense to talk about such events explicitly.
- Dynamic object creation. It is often desirable to create and destroy objects during the planning process. As an example take the *Settlers* domain used in IPC-3, where vehicles (carts, trains, ships) have to be created to provide the necessary transport infrastructure. In a dynamically extended PDDL one would have additional effects that are of the form `(new (object - type) (atomic-formula*))` and `(delete ?a)`. The former construct would generate a new object and set initial predicates for it, while the latter construct deletes an existing one. Obviously, this language extension would affect planners that fix the problem representation during pre-processing. For more flexible approaches (like TLPlan which does not pre-instantiate actions), it might be less harmful. Assuming that a superset of the set of created objects can be fixed, a compilation to standard PDDL is available by using additional activation predicates for the objects. Dynamic creation of objects can appear in Stefan Edelkamp's protocol domain (Edelkamp 2003a; 2003b), when new processes (threads) are invoked from already existing ones. For general software verification problems, dynamic state descriptions and object creation are crucial.

Let us state once more that we intend to be conservative about language extensions. The above list is to be understood as a pool of ideas for discussion. For actual use in the competition, introducing at most one of the list items would seem appropriate. The final decision about language extensions to the classical setting, if any, will be taken by the authors and the IPC-4 advisory committee, taking account of feedback from the community.

As in the previous competitions, in order to enable broad participation, of all benchmark domains there will be versions that stay within simpler language requirements. In particular, for all domains that make use of new language features (if any), there will be versions of the domains without these features. In difference to the previous competitions, however, we will try to avoid using simpler versions of the domains in the simpler language classes: as far as possible, we intend to *compile the full semantics* of the domains into the simpler languages. Examples of language features that can be compiled away are axioms or deterministic exogenous events (see above), as well as all ADL constructs. For one thing, we expect that compiled, rather than simplified, problems will yield more interesting (STRIPS, e.g.) benchmarks. For another thing, by making (careful) comparisons between different language expressivity levels, this way the competition data will make a point on the value of language extensions.

Compiling domain semantics may result in PDDL files hard to tackle for existing systems (as the compiled files might differ from what manually designed domains usually look like). We will make test files available in time. In that context, one thing we already want to make potential competition participants aware of is this. In some real-world benchmarks we are currently considering, ADL formulae are needed for the (most natural) domain formulation. So there is a good chance that ADL capabilities will be of value in the competition. As has been done by various researchers, ADL formulae can be compiled away. We will provide domain versions where this has been done along the lines of (Gazen & Knoblock 1997). Also, we will provide domain versions in "SIMPLE-ADL": this has first appeared in the 2000 competition, and avoids the blow-up implied by compiling conditional effects away. It is identical to STRIPS except that actions can have conditional effects, where the conditions are fact sets (i.e., conjunctions of facts). We encourage researchers to make their systems capable of handling, at least, SIMPLE-ADL.

Benchmark Domains

The main point that we as the organizers want to make in the continuation of the classical part of the competition is a more deliberate set of benchmark domains. It is a trivial insight that the benchmarks we use have an important impact on the field, i.e. on the algorithmic techniques that are developed. It is much less trivial to think about what "good" benchmarks would be. We propose a kind of hybrid answer to the question, exploring three different philosophies of benchmarking. The keywords might be given as "real-world problems", "problems with characteristic structures", and "realistic testing problems". The competition domains would then be split up across these three topics. More detailed descriptions of our ideas are below. We do not claim that our ideas are entirely new; most of what we say underlies the way in which people have designed the benchmarks so far, and the borderlines between the different topics aren't strict. The new thing is probably to state these topics explicitly.

1. Real-world problems. Candidates for this class would be real-world domains in the literal sense of the word: problems that are being tackled in real-world applications, modelled closely enough to be assumed real, run with instances that occur in the real-world scenario. It is obviously desirable for the community to have such benchmarks (even if it were only to improve the image of the field), and we do believe there is hope in getting some together:
   - Jana Koehler's Miconic domain (Koehler & Schuster 2000) (in the known ADL version as well as in a purely numerical version), which can hopefully be run with real data from a skyscraper.
   - An airport-ground-traffic-control domain (Hatzack & Nebel 2001) from a company of Wolfgang Hatzack. If planners scale up that far, the domain can be run with data from Frankfurt airport, otherwise smaller airports (Zürich, e.g.) can be used.
   - A deterministic version of Sylvie Thiebaux's PSR domain (Bertoli *et al.* 2002; Bonet & Thiebaux 2003).
   - Stefan Edelkamp's protocol domain derived from the automatic translation of Promela specifications to PDDL 2.1 (Edelkamp 2003a; 2003b). The domain can be run with instances that are models of real world-systems like a public old telephone system (POTS) and parts of the Corba Architecture (GIOP).
   - Richard Goodwin at IBM is currently looking into planning formulations of some industrial scheduling problems he has dealt with.
   - It might be possible to use some of the IRST verification applications as classical planning benchmarks.
   - Celcorp got a domain about communicating data between business processes.
   - We are in contact to Salim Khan who is working on a domain formulating "Biological Pathway Discovery" as a planning problem (Khan *et al.* 2003).
   - The *Pipesworld* domain by Milidiu et al. (2003) promises to yield a very interesting real-world benchmark.
   - It might be possible to create a realistic domain about mixed planning and scheduling problems based on Bartak's ideas (Bartak 2003).

   Ideally we want realistic domains within the language features of PDDL 2.1. Our intention is not to try and run every application domain we can lay a hand on, but to focus on a smaller number of real-world domains, 3 to 5, which can be modelled most realistically with no, or as cautious as possible, language extensions.

2. Problems with characteristic structures. What we also consider a rather strong point is that at the base research level one is not interested in the exact form of the real-world application itself, but in the different forms of characteristic structures that cause the nature of the different applications. So the benchmarks should be domains that each capture a very clean characteristic structural phenomenon, and that are very much different from each other, spanning a wide range of interesting characteristica. Evaluating planners in such domains yields insights into the kinds of domains that favor one exploration engine or another. Known candidate domains here we see as the Blocksworld and Logistics domains, maybe Towers-of-Hanoi, maybe Mystery. Hitherto not used candidates might be found in the search community, like e.g. the (n^2-1)-puzzle, Sokoban, Atomix, or Quasi-group completion. Criterions for a good domain collection might

be: to cover the relevant complexity classes (at least, P, NP, and PSPACE) for deciding (bounded) plan existence within a domain; and to cover the relevant distinctions in Hoffmann's planning domain taxonomy (Hoffmann 2002).

3. Realistic testing problems. Here we got manually designed domains which aim to describe realistic situations, and test planners on different kinds of structures. This topic is, so to speak, to ensure downwards compatibility. As participants in this class we would see interesting planning problems like *Settlers*, *Driverlog*, etc., much in the sense of the IPC-3 competition. The idea would be to re-use some challenging domains, like Settlers, and perhaps also to introduce some new ones, trying to make more extensive use of the PDDL 2.1 language features (like Settlers does with the numerical variables).

Of course, not arbitrarily many domains can be used in the competition so it will be a careful decision exactly how much of the above ideas, and in which form, will be realized at IPC-4. In benchmark class 1., as said we aim at a set of 3 to 5 real-world testing domains. In benchmark class 2., to cover a sufficiently broad range of characteristic structures it appears one will need around 4 to 6 domains. The testing domains in benchmark class 3. can be chosen last to make for an appropriate size of the event (in particular one might think about skipping that class of domains altogether).

Evaluation

As has been done before, we intend to evaluate the competition data by "soft" criteria such as judging runtime and solution quality by looking at figures, and counting numbers of solved example problems. Our reason for this choice is that the abilities of individual planners, and the performance criteria relevant in individual domains, are so diverse it seems impossible to come up with adequate "hard" (i.e. mathematically precise) evaluation criteria. Indeed, according to some private discussions we had there seems to be wide disagreement in the planning community as to what good "hard" evaluation criteria (for plan quality performance, e.g.) would be. It appears best to allow people to draw their own conclusions based upon quickly perceivable detailed data (see next section). Nevertheless, as has been done in the previous competitions we intend to identify competition "winners" based on our own observations. The awarding of prizes, after all, is one of the main sources of excitement in the event.

In *difference* to what has been done previously, in the results evaluation we would like to explicitly take into account the following issues:

- Optimal vs. Suboptimal Planning. Considering recent developments and results, it seems worthwhile to think about evaluating "optimal planners" separately from "sub-optimal planners". "Optimal planners" could be defined as planners that give a guarantee on the quality of the found solution. In many benchmark domains optimal planning is computationally harder than sub-optimal planning (Helmert 2003), and indeed optimal planners tend to be far less efficient than their sub-optimal counterparts. This, probably enforced by the fact that the competitions did yet not honor optimality guarantees at all, has lead to a tendency away from optimal planning as done by, e.g., SAT-based planners. As a way out we propose to award (a) separate competition prize(s) for the best (i.e., fastest) optimal planner(s).
- Sub-optimal plan quality. We intend to provide some measure of how good the plans of the sub-optimal competitors are. In domains where it is possible, we intend to generate the optimal solution quality values using domain-specific optimal solvers. This data will be included into the results presentation (as optimal planners are rare, and usually don't scale up very far, their data will likely not provide a likewise good measure of plan quality performance). Also, to honor good plan quality behavior, one might consider awarding a prize for the sub-optimal planner with the best quality plans, specifically with the plans that best meet the specified flexible optimization criteria.
- The effort of hand-coding. The whole point of hand-tailored planners is to reduce the effort of coding. So one should find a way of assessing the coding effort that the teams invest into their solution methods. The question is, how? In the 2000 competition, there was a restricted amount of time (a week, roughly) between availability of testing problems and data deadline. But of course the amount of work done within the available time might differ considerably between the individual planners. Ideas for explicitly measuring coding effort are to have the participating teams provide the number of man-hours they spent on the various domains (distinguishing between students and lecturers, maybe), and/or to measure the length of the files that contain the final domain-specific information. We are aware that these are very coarse measurements of coding effort. But they would at least provide hints. Another interesting idea (Kambhampati 2003) is to let outsiders, people other than the respective planners' makers, generate the control information for the hand-tailored planners. Then the data would provide a measure of how easy it is to generate good control information, rather than a measure of how much the planner can be tuned by experts.

Organization

The organizational issues shall, ideally, be co-ordinated across the different parts of IPC-4. A rough time table is this:

- 06/2003: Advertisement at ICAPS-2003, assisted with a workshop.
- 07/2003: Language extensions fixed, testing problems available.
- 12/2003: Submission deadline for planners.
- 01/2004 – 02/2004: Competition data collection phase.
- 03/2004: Data evaluation.

- 04/2004: Conference with detailed results presentation and participants handout.
- Summer 2005: Participants publication.

This time table differs from what has been done previously in that we want to finish the data collection phase well ahead – one month – of the conference. This involves a stricter way of handling deadlines, and an earlier starting point for the data collection phase. Our reasons for this choice are the following. First, it makes evaluation a lot easier for the organizers if they actually got some time to do it. Second, the competition has grown too large to present or even try to summarize the results within a 30 minute presentation. Instead, we propose to set up an extra room at the conference where all results are presented (e.g., in the form of posters assisted with a visualization tool) throughout the conference, so the conference participants and the competitors can take their time to see what has happened. We understand that this takes some of the excitement out of the event, at least if the data is made available before the award ceremony. An idea would be to have the competition award ceremony early at the conference, and set up the room with detailed data presentation afterwards. In any case, we believe that the main aim of the competitions is to provide a large scale evaluation of the state-of-the-art, and that this aim would be served much better by making the precise results more quickly and widely perceivable. In the same spirit, we plan to distribute, at the conference, a hand-out with extended abstracts describing the competing systems. In the previous competitions, information about the algorithms used by the competitors was not made publicly available at the time of the competition event. An article about the 1998 competition appeared in AIMag summer 2000, an article about the 2000 competition appeared in AIMag fall 2001, and a JAIR special issue about the 2002 competition has yet to appear. As a consequence, at conference time there has so far not been much chance of knowing what the single planners do except trying to catch their makers – this holds true at least for the newer systems on which yet no conference or journal publication was out independently of the competition. It would be great to have a hand-out for each conference participant, containing brief descriptions of the individual systems, even if it's only extended abstracts of 1-3 pages. This way people could get a picture of what is happening, especially in connection with the possibility to have a close look at the results.

As in the former years, there shall eventually be a competition publication or the possibility thereof. We are yet unclear whether JAIR or AIMag is the better choice. If we succeed in providing a handout at the conference, then AIMag appears to be better: it seems a more adequate forum for the competition publications than a full-scale top journal article, and the disadvantage of time delay is not as serious when there has already been a handout anyway.

Finally, there is a choice to be made about how the data for domain-independent (fully automated) planners is collected. So far the planners have been run by their makers, on the domains which were made available one-by-one during the data collection phase. This allowed developers time to work on their planners on the fly, and thus invest certain tuning efforts. An alternative would be to have the developers provide executables by a deadline, then run the planners automatically. While this alternative would serve the spirit of domain-independent planning better, it is questionable whether it is practically feasible (planners might fail in unseen domains due to trivial bugs, parsing problems, etc.). An idea is to subdivide the data collection phase into a testing phase, and an execution phase. In the testing phase developers would be given a few example instances from all domains so they can make sure their planners work. Then there would be a deadline for executables, and data would be collected automatically in the execution phase. In any case, the deadline for the end of the data collection phase must be early enough to allow the developers to provide extended abstracts describing their systems, and to turn these abstracts into a handout.

Summary

To summarize, our main proposals for the classical part of IPC-4 are these:

- No, or at most one moderate, language extension, e.g. the re-introduction of PDDL axioms, or the introduction of dynamically created objects.
- A careful choice of structurally different benchmark domains, including a number of domains that are as close as possible to real-world applications.
- More emphasis on plan quality, in particular on planners that give optimality guarantees, in the awarding of prices.
- An earlier end of the data collection phase, enabling a detailed and widely accessible results presentation at conference time, in connection with a handout describing the competitors.

References

Bartak, R. 2003. Real-life manufacturing problems: A challenge. In *Proceedings ICAPS-03 Workshop on the Competition*.

Bertoli, P.; Cimatti, A.; Slaney, J.; and Thiebaux, S. 2002. Solving power supply restoration problems with planning via symbolic model checking. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02)*, 576–80. Lyon, France: Wiley.

Bonet, B., and Thiebaux, S. 2003. GPT meets PSR. In ICAPS-03 (2003). Accepted for publication.

Edelkamp, S. 2003a. Limits and possibilities of PDDL for model checking software. In *Proceedings ICAPS-03 Workshop on the Competition*.

Edelkamp, S. 2003b. Promela planning. In *Proceedings SPIN-03*. Accepted for publication.

Edelkamp, S. 2003c. Taming numbers and durations in the model checking integrated planning system. *Journal of Artificial Intelligence Research*. Special issue on the 3rd International Planning Competition, to appear.

Gazen, B. C., and Knoblock, C. 1997. Combining the expressiveness of UCPOP with the efficiency of Graphplan. In Steel, S., and Alami, R., eds., *Recent Advances in AI Planning. 4th European Conference on Planning (ECP'97)*, volume 1348 of *Lecture Notes in Artificial Intelligence*, 221–233. Toulouse, France: Springer-Verlag.

Gerevini, A., and Serina, I. 2003. Fast planning through local search and action graphs. *Journal of Artificial Intelligence Research*. Special issue on the 3rd International Planning Competition, to appear.

Hatzack, W., and Nebel, B. 2001. The operational traffic control problem: Computational complexity and solutions. In Cesta, A., and Borrajo, D., eds., *Recent Advances in AI Planning. 6th European Conference on Planning (ECP'01)*, 49–60. Toledo, Spain: Springer-Verlag.

Helmert, M. 2003. Complexity results for standard benchmark domains in planning. *Artificial Intelligence*. Accepted for publication.

Hoffmann, J. 2002. Local search topology in planning benchmarks: A theoretical analysis. In Ghallab, M.; Hertzberg, J.; and Traverso, P., eds., *Proceedings of the 6th International Conference on Artificial Intelligence Planning and Scheduling (AIPS-02)*, 92–100. Toulouse, France: Morgan Kaufmann.

Hoffmann, J. 2003. The Metric-FF planning system: Translating "ignoring delete lists" to numerical state variables. *Journal of Artificial Intelligence Research*. Special issue on the 3rd International Planning Competition, to appear.

2003. *Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS-03)*, Trento, Italy: Morgan Kaufmann.

Kambhampati, S. 2003. Are we comparing Dana and Fahiem or SHOP and TLPlan?: A critique of the knowledge-based planning track at IPC. In *Proceedings ICAPS-03 Workshop on the Competition*.

Khan, S.; Gillis, W.; Schmidt, C.; and Decker, K. 2003. A multi-agent system-driven ai planning approach to biological pathway discovery. In ICAPS-03 (2003). Accepted for publication.

Koehler, J., and Schuster, K. 2000. Elevator control as a planning problem. In Chien, S.; Kambhampati, R.; and Knoblock, C., eds., *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems (AIPS-00)*, 331–338. AAAI Press, Menlo Park.

Milidiu, R. L.; dos Santos Liporace, F.; and de Lucena, C. J. 2003. Pipesworld: Planning pipeline transportation of petroleum derivatives. In *Proceedings ICAPS-03 Workshop on the Competition*.

Thiebaux, S.; Hoffmann, J.; and Nebel, B. 2003a. In defence of PDDL axioms. In *Proceedings ICAPS-03 Workshop on the Competition*.

Thiebaux, S.; Hoffmann, J.; and Nebel, B. 2003b. In defence of PDDL axioms. In Gottlob, G., ed., *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*. Acapulco, Mexico: Morgan Kaufmann. accepted for publication.