IPC-4 Probabilistic Planning Track: FAQ 1.01
November 1, 2003

Michael L. Littman
Department of Computer Science
Rutgers University
Piscataway, NJ 08854 USA
mlittman@cs.rutgers.edu

Håkan L. S. Younes
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
lorens@cs.cmu.edu

Abstract

The 2004 International Planning Competition, IPC-4, will include a probabilistic planning track for the first time. This document provides some of the high level decisions that have been made concerning how the competition will be run.

The 2004 International Planning Competition, IPC-4, will include a probabilistic planning track for the first time. This document lays out some of the high level decisions that have been made concerning how the competition will be run. The details are still in flux at the time of this writing, and we hope to get feedback from the participants to spelled them out more precisely during the fall.

The overriding goal of the probabilistic planning track is to bring together two communities converging on a similar set of research issues and aid them in creating comparable tools and approaches. One community consists of Markov decision process (MDP) researchers interested in developing algorithms that apply to powerfully expressive representations of environments. The other consists of planning researchers incorporating probabilistic and decision theoretic concepts into their planning algorithms. Cross fertilization has begun, but the probabilistic planning track promises a set of shared benchmarks and evaluation metrics that could crystallize efforts in this domain of study.

This document represents a snapshot of the ongoing development of the IPC-4 probabilistic track. For the latest developments, please visit: `http://www.cs.rutgers.edu/~mlittman/topics/ipc04-pt.html`.

Frequently Asked Questions

There are many issues to be ironed out as part of establishing the new track, but there are some issues that are mostly decided at this point. Here are some clarifying questions and their answers.

What domain description language will be used to represent probabilistic domains?

That's a great first question. We are creating a new domain description language modeled on PDDL 2.1, the domain description language for deterministic domains that has been used in the IPC in the past. Syntactically, this language has a STRIPS-like flavor, but includes probabilistic constructs. The current version is PPDDL 1.0 (Younes & Littman 2003).

By basing the domain description language on PDDL, we remain in the spirit of the existing programming competition, which we expect to help further bring the communities together.

But, I like DBN representations. Is there some way for me to participate?

Do not fear. The representation is sufficiently powerful to support a direct translation from the kind of conditional probability tables used in dynamic Bayesian network (DBN) representations (Dean & Kanazawa 1989), even if they include context-specific independence (Boutilier *et al.* 1996). We are working towards constructing such a translation, but could definitely use some help. Optimizing the translation process seems to be a research topic in and of itself.

Since the domain description language is based on PDDL, does this mean the representation is relational?

Yes, although representations with explicit objects are not a traditional feature of MDP-based domain description languages, algorithms that exploit these features have begun to appear.

We expect that many groups will propositionalize the domains because they cannot directly plan with parameterized operators. Most of the test domains will allow for relatively straightforward propositionalization, so the relational representation should not be seen as an impediment to entry for interested groups. We simply feel that relational MDPs are an exciting direction worth supporting and want to give researchers interested in relational issues an opportunity to explore this type of representation.

What resources will be available to support parsing and converting PPDDL into other representations?

We have made available software in C++ for a plan validator and are developing a very simple planner to use as an example. We are willing to work closely with groups to produce conversion tools to other representations. Note that anything we help develop will be made available to any interested participants.

Wait a second. Doesn't PDDL 2.1 support numbers? How can we propositionalize when there are numbers?

Numbers will only be used in a very limited way to express rewards; further details are provided by Younes & Littman (2003). None of the domains for the probabilistic track will have numeric variables as part of the state space, so the domains we use will propositionalize.

Will problems have a single initial state, or a probability distribution over possible initial states?

We don't think this matters conceptually, since the initial state can always be set so that it produces a probabilistic transition to a set of states immediately following the first action. However, it is syntactically convenient to allow explicit initial distributions, so we include this feature.

Will there be continuous variables or simulated physics?

Not at this time, no. Although these would be critical for representing many important domains (like billiards, say), we are not aware of any planners that can exploit representations of this kind. We hope the community is able to move in this direction in the future.

I haven't seen anything about partial observability. Will it be supported?

Planning in partially observable domains is very important and it is a direction we believe the community should pursue. At this time, however, there are a greater number of planning algorithms that can make use of complete information, so the IPC-4 will feature complete observability exclusively (MDPs, not POMDPs).

Some research groups have planners that cannot make use of any observation information at all. We hope to include one or more domains for which straightline plans can perform well.

Do the planners need to support the full flexibility of the PPDDL specification language to compete?

For the probabilistic track the absolute minimum is ":strips" and ":probabilistic-effects". It is also highly recommended to support ":typing", ":equality", ":negative-preconditions", and ":conditional-effects". Support for ":disjunctive-preconditions" and ":quantified-preconditions" is also good to have. Of course, if you want to do MDP planning you should also support the ":rewards" requirement. The probabilistic track will not make use of the ":fluents" requirement.

How will the competition be run?

The probabilistic track will follow the same procedure as the classical track of IPC-4 (see `http://ipc.icaps-conference.org/`). The current plan is as follows:

- Test domains will be distributed and all experiments will be run by competitors prior to the ICAPS 2004 conference, not on site.
- Our intention is to provide an extra room at the conference in which the results can be viewed in detail throughout the conference, after a ceremony announcing results and recognizing the "winners".
- In addition, we plan to distribute a handout containing abstracts describing the competing planners.
- These events might be supported by a separate competitors' workshop.

What will we use for test problems?

The organizers of the classical track are moving toward more practical problems, but this first instantiation of the probabilistic planning track will be more about realistic expectations than realistic problems.

There will definitely be a noisy blocks world and a noisy logistics problem. Other problems that showcase the probabilistic representation will also be included, but the details are not yet available. We hope participants can contribute some example problems of their own.

In future years, once the foundation has been laid, problems of practical interest should be introduced, for example planning in a Mars rover with continuous resource management.

Do you have any idea yet how big the domains will be? Will they be at least large enough so that they can't just be solved optimally using modified policy iteration?

We will include problems of several sizes, including one small enough to be solved by enumerating the state space (we're working on a planner like this right now). However, most will require a more clever approach than naive enumeration and solving the resulting MDP.

How will plans be represented?

In the classical track, a plan is a series of operators. A successful plan is one that, when applied to the initial state, achieves the goal.

Life is not so straightforward in the probabilistic track. While there are many proposals for plan representations in non-deterministic environments (straightline plans, plan trees, policy graphs, triangle tables, etc.), none is considered a widely accepted standard. In addition, even simple plans are challenging to evaluate exactly in a non-deterministic environment, as all possible outcomes need to checked and combined.

For these reasons, we will evaluate planners by simulation. That is, the plan validator will be a server and individual planning algorithms clients. Planners connect to the validator, receive an initial state, and return an operator. This dialog continues until a terminating condition is reached at which point the validator evaluates the performance of the planner. This entire process is repeated several times and results averaged over the multiple runs.

How will plans be scored, then?

Evaluating plans using a client–server model means that the distinction between a planner and an executor has significantly blurred. It also means that computation is no longer a one-time preprocessing cost, but something integrated with action selection itself.

Planning quality, therefore, needs to be a combination of expected utility and running time. Each domain will have a time limit. In goal-oriented domains, planners will be evaluated by the number of trials in which a goal was reached before the time limit. In reward-oriented domains, planners will be evaluated by the total reward achieved before the time limit. Note that we will restrict even the reward-oriented domains to have some sort of terminating state, so we do not expect there to be an advantage for planners that can simply produce more actions within the time limit.

Will the domains focus on a more MDP-like decision theoretic reward criterion or a more AI-planning-like goal satisfaction criterion?

In many ways, this is a false distinction. The probability of reaching a goal is equivalent to expected reward if a reward of $+1$ is issued upon goal achievement and all other transitions have $+0$ rewards.

Can total expected reward be simulated by a **discounted** expected reward MDP?

Well, they are the same if the discount factor is 1. And they continue to behave the same if you make the discount factor ever so slightly less than 1. From a practical standpoint, you've got a pretty good approximation if you use a discount factor of $1 - \epsilon$. Condon (1992) showed that the ϵ you'd use to ensure preserving the optimal policy can be specified with polynomially many bits.

In the competition, will there be a discount factor (less than one) for scoring the accumulated reward? Will there will be a given planning horizon?

No, there won't be an explicit discount factor or horizon in any of the problems. In fact, PPDDL doesn't currently have direct support for either of these.

But, what if you have more general rewards being accumulated during execution? Would a goal-oriented planner still be able to do something interesting?

Mathematically, general reward problems can be cast as goal-achievement problems. Essentially, each transition with a reward can be viewed as a probabilistic transition to a goal state (proportional to the reward), a probabilistic transition to a non-goal sink state (proportional to the difference between the reward and the maximum possible reward), and a probabilistic transition to the next state (proportional to the original transition probability). Majercik & Littman (2003) provide this argument in more depth and give citations for papers on this topic.

Based on this mapping, it ought to be possible to write a converter that creates a goal-oriented problem from a reward-based problem. Whether this results in a competitive planner is an open question, however.

Note that one or more of the test domains will use only a goal-type performance objective. So, even if no convenient conversion can be created, goal-oriented planners will be welcome in IPC-4. A description of the `(reward)` fluent appears in our companion document (Younes & Littman 2003).

What about nondeterministic planning?

As of this time, we have not been contacted by any groups interested in nondeterministic planning. We will not be supporting nondeterministic planning in IPC-4.

Is there room for reinforcement-learning methods?

We welcome reinforcement-learning approaches to the domains we will use in the competition. We are not planning a special subtrack in reinforcement learning (no domain model provided). There is a good chance that the line separating learning from planning will blur considerably in the next decade, so it is our hope that the competition will move the community in this direction.

Will there be opportunities to use other kinds of learning?

Yes, because we will provide the formal descriptions for some of the domains in advance, there will be an opportunity for groups to learn about these domains in advance. We have a blocksworld domain with an explicit parameterized generator for problems. This is available to competing groups and we hope this will allow some of the groups to learn successful strategies for this class of domains. Approaches that try to generalize planning strategies from solving small instances will have an opportunity to benefit from this information.

What is the competition timeline?

- We have received more than 20 statements of interest and have released two versions of the client-server validator.
- December 15, 2003, we will be accepting example domains from the competitors to build a base of interesting test problems.
- February 1, 2004, we will hold a dry run to help work out the kinks in the client-server code in the context of a few example domains. It is not required that participants have a complete planner at this time, but those that don't will miss an exceptional opportunity to test their system under realistic conditions.
- In June 2004, IPC-4 will be held in Vancouver. Planners will be run on the competition problems at least one week prior to the conference, with analysis and results to be announced at the conference. Results and test domains will be released publicly.

Where can I learn more about PPDDL 1.0?

We have prepared an introduction to the language, which can be downloaded from the project site: `http://www.cs.rutgers.edu/~mlittman/topics/ipc04-pt.html`.

How can I get a copy of the validator software?

At this time, we are only distributing the software to participants in the competition. However, we will consider all requests. Send mail to `mlittman@cs.rutgers.edu`.

Acknowledgments

David Andre, Hector Geffner, Bob Givan, Judy Goldsmith, Carlos Guestrin, Eldar Karabaev, Terran Lane, Nicolas Meuleau, James Park, Ron Parr, Sailesh Ramakrishnan, Christian Shelton, Olga Skvortsova, Kurt Steinkraus, and others asked some of the questions.

References

Boutilier, C.; Friedman, N.; Goldszmidt, M.; and Koller, D. 1996. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI 96)*, 115–123.

Condon, A. 1992. The complexity of stochastic games. *Information and Computation* 96(2):203–224.

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.

Majercik, S. M., and Littman, M. L. 2003. Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence* 147(1–2):119–162.

Younes, H. L. S., and Littman, M. L. 2003. PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. Available at `http://www.cs.cmu.edu/ lorens/papers/ppddl.pdf`.