Introduction to the Probabilistic Planning Track

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, includes a probabilistic planning track for the first time. We briefly summarize the design of the track.

Introduction

Domain-independent planners seek to synthesize plans that achieve goals as cheaply as possible. While classical planning is concerned with domains in which operators have deterministic effects—the planner can predict with certainty how its decisions will change the environment—work on probabilistic planning expands the field to include operators with uncertain effects. The inclusion of probabilistic effects extends domain description languages to a more realistic class of applications. However, this increased generality comes with the price of increased computational complexity of planners and plan evaluation (Littman, Goldsmith, & Mundhenk 1998).

The 2004 International Planning Competition, IPC-4, introduces a probabilistic planning track for the first time. The goal of the track is to provide a forum for the evaluation and comparison of approaches to probabilistic planning. At the time of this writing, most of the logistical decisions have been made, but the competition and evaluation have not yet taken place. This document summarizes the status of the competition as of April 2004. For the latest developments, please visit: `http://www.cs.rutgers.edu/~mlittman/topics/ipc04-pt/`.

The probabilistic track was organized by the authors, Michael L. Littman and Håkan L. S. Younes, and a team at Rutgers consisting of John Asmuth, David Weissman, and Paul Batchis.

Calendar

Planning for the probabilistic track dates back to shortly after IPC-1. However, it was Sven Koenig and Shlomo Zilberstein's idea to specifically create a probabilistic track for IPC-4. Initial attempts to drum up support for the competition in 2002 led to the creation of a mailing list with addresses of 87 interested researchers. As the form of the competition itself took shape, potential participants were asked to register in September 2003. Representatives from 22 groups (spread over 4 continents) signed up to receive the first version of the PPDDL validation software.

In April 2004, we held a "mock competition" as a way of identifying the most committed groups and for testing our evaluation procedure. Six groups participated (groups C (UMass), E (Dresden), G (ANU), J (Purdue), P (Simon Bolivar) and, D (Bowdoin)). Several other groups expressed regrets that their planners were not yet ready. As of this writing, several groups have explicitly pulled out of the competition and 15 groups remain signed up. We're expecting between 5 and 10 groups to participate in the competition within the next three weeks.

Domain Description Language

We intended the competition to be accessible to researchers studying "factored" or first-order Markov decision processes (extensions of MDPs to predicate-based state representations) and decision-theoretic planning (extensions of classical planning to uncertain effects and utilities). The state of the art in evaluating classical planners is the IPC and their choice of domain description languages is PDDL (Fox & Long 2001). We sought to introduce a minimal set of extensions to PDDL2.1 to support probabilistic effects. The probabilistic planning domain description language (PPDDL1.0) we developed is described in the following paper.

PPDDL1.0 extends PDDL2.1 to support the succinct representation of Markov decision processes. However, for this first competition, we decided to restrict the set of language features that participants would need to support. Specifically, the evaluation domains included neither numeric state variables nor hidden propositions. As such, there is a direct conversion from the provided PPDDL specifications to finite (though perhaps enormous) MDPs.

To support the programming efforts of the participants, we provided C++ code for parsing PPDDL domains and problems and an mtbdd-based converter from PPDDL to a propositionalized MDP representation. We believe several participants wrote their own parsers and converters and others used our initial code to varying degrees.

Objectives

Each domain used in the competition came in one of two possible styles. In *goal-only* domains, a goal specification was provided and the objective of the planner was to reach a goal state. Planners in these domains are evaluated by estimating the probability that they will reach a goal state. Such domains can be viewed as a type of MDP in which a unit reward value is provided upon arrival in a goal state and all other transitions result in zero reward.

The second, and more common, style of domain in the competition was "reward goal" problems. These domains include operators with state-independent cost, a goal specification, and a goal-reward value issued upon arrival in a goal state. Although PPDDL supports positive and negative state-dependent rewards as well as continuing tasks with no terminating goal state, we thought restricting objectives as described kept them as close as possible in spirit to the kinds of objectives supported in the classical track.[1] By assigning goal rewards, each execution of a planner on a problem terminates with a total reward value, with early termination preferred to longer execution traces. Planners are compared according to their total expected reward, computed as the sum of the goal reward (if obtained) minus any action costs.

We also planned to support evaluation of "nondeterministic" domains. However, as no groups stepped up to participate in such a track, we did not pursue it.

Evaluation

In classical planning, a plan is a series of operators. A valid plan is one that, when applied to the initial state, achieves the goal. Because of the uncertainty in state transitions, straight-line plans are often not appropriate in probabilistic domains. Although several groups have expressed an intention to synthesize only unconditional plans, we did not want to impose any particular plan representation on participants.

We decided to evaluate planners by sampling or simulation. That is, our plan validator is a server, and individual planning/execution algorithms connect to the evaluator as clients. They initiate a session by providing an agreed upon domain id, receive an initial state, and return an operator. The server-client 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 with results averaged over the multiple runs.

Source code for a server ("mdpsim") was provided to all participants and updated as changes were made to the domain description language and evaluation procedure. For official evaluations runs, a server was run at Rutgers with participants connecting via the Internet. In trial runs, participants reported communication times ranging from 20ms (CMU) to 76ms (South America) to 230ms (Australia) roundtrip. To compensate for the wide range of communication times, participants were offered the option of temporary accounts at CMU to install and run their clients.

Based on feedback from the mock competition, we decided to evaluate each planner in each domain in a 15-minute block. During this block, planners can carry out any computation, pre-processing, or plan generation that they choose to do. However, they must also execute 30 runs from an initial state to a goal state (voluntary premature termination is also an option). The average reward obtained over these 30 runs (with zero reward for any runs that were not taken) is the planner's evaluation score.

We chose 30 runs because this number may provide sufficient statistical confidence to distinguish between planners. We did not subdivide the 15 minutes into 30-second blocks to allow participants to amortize planning effort over multiple runs. We suspect that most planners will use the majority of the 15 minutes to construct a plan and the remainder to evaluate the plan 30 times. However, the evaluation procedure supports a wide variety of strategies.

Domains

In the mock competition, we included 19 test problems: blocksworld (5 5-block problems, 5 25-block problems, and 5 125-block problems), one colored blocksworld problem, one fileworld problem, a variation of the coffee domain (Dearden & Boutilier 1997), and a variation of the sandcastle problem (Majercik & Littman 1998). These include problems with and without functions and both goal-only and reward-goal domains.

The blocksworld problems were created using a blocksworld problem generator that we developed. It will be available after the conference on the competition website. We have also released a logistics domain generator we call "boxworld". Problems generated from the blocksworld and boxworld generators will be included in the competition. Because these generators were released in advance, participants have the option of learning or hand-tuning rules for their planners to exploit structure in these domains.

Several other domains will be included in the competition, to be distributed immediately prior to evaluation. All domains we used for evaluation will be made available to interested researchers. Visit our web site or contact us by email for more information.

Acknowledgements

This work was supported in part by NSF grants IIS-0329153 and IIS-0315909. We thank the ICAPS and IPC organizers for their support and encouragement and the participants for their enthusiasm and creativity.

References

Dearden, R., and Boutilier, C. 1997. Abstraction and approximate decision-theoretic planning. *Artificial Intelligence* 89(1–2):219–283.

Fox, M., and Long, D. 2001. PDDL2.1: An extension to PDDL for expressing temporal planning domains. Technical report, University of Durham, UK.

Littman, M. L.; Goldsmith, J.; and Mundhenk, M. 1998. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research* 9:1–36.

Majercik, S. M., and Littman, M. L. 1998. MAXPLAN: A new approach to probabilistic planning. In Simmons, R.; Veloso, M.; and Smith, S., eds., *Proceedings of the Fourth International Conference on Artificial Intelligence Planning*, 86–93. AAAI Press.

---

[1] Thanks to Héctor Geffner for sharing this observation.