

Challenges for Theory and Practice in Planning

Thomas Dean* and Jak Kirman

Department of Computer Science

Box 1910, Brown University

Providence, RI 02906, USA

Abstract

This paper reports on recent progress addressing two issues, one of importance to theoretical and experimental research in generative planning and the second of importance to the practical application of probabilistic planning methods. The first issue concerns what makes planning problems hard from a computational perspective. We report on research investigating threshold phenomena in deterministic planning problems. The observed phenomena are similar to those observed in boolean satisfiability problems, but raise interesting issues concerning the random generation of planning instances. The second issue concerns the problem of learning models for stochastic domains. Developing planning systems currently requires the tedious hand construction of domain models. We survey Bayesian methods for learning such models from data, augmented with expert knowledge when available.

Introduction

For the past ten years, we have studied methods for representing and solving planning and scheduling problems that involve dynamic and uncertain domains. We have been primarily concerned with how to go about solving a problem given a representation of the domain in a particular format.

The format that we have chosen uses Bayesian networks to compactly represent the probability distributions governing change and the consequences of acting. These representations are able to model quite general dynamics and they simplify the construction of abstractions corresponding to statistical summaries that can be used to reduce the complexity of prediction and decision making. Our representations handle as a special case the propositional STRIPS formulation for planning, *i.e.*, the dynamics of the domain is encoded as a set of STRIPS operators.

*This work was supported by the Air Force and the Advanced Research Projects Agency of the Department of Defense under Contract No. F30602-91-C-0041, and by the National Science Foundation in conjunction with the Advanced Research Projects Agency under Contract No. IRI-8905436.

Our recent research has focused on two issues: (1) building better general-purpose algorithms for solving planning and scheduling problems cast as Markov decision processes, and (2) investigating applications to convince ourselves and others that we can both represent and solve interesting problems. Regarding (1) we are developing abstraction and decomposition methods for coping with high-dimensional state spaces (Dean & Lin 1995; Dean *et al.* 1995; Lin & Dean 1995). Regarding (2) we are attempting to encode a portion of the air campaign planning problem faced by U.S. Air Force.

These issues have forced us to address some even more basic problems. The first such problem concerns understanding what makes planning problems difficult. Are the problems that we encounter in practice inherently difficult or are we just looking at them from the wrong perspective? In particular, are our computational difficulties caused by poor problem formulation? The second problem concerns how are we are going to get applications people to adopt our technology given that most of them are not trained in probability, statistics, and the decision sciences. In the following two sections, we summarize our progress so far in addressing these two problems.

Hard Problems in Planning

What constitutes a hard instance for a given planning problem? We don't even know the answer to this question for the simplest of planning problems: planning for conjunctive goals in domains specified as a set of deterministic propositional STRIPS operators. NP-completeness results have done little to influence experimental and practical research one way or the other (Bäckström & Klein 1991; Bylander 1994; Gupta & Nau 1991). In the following, we motivate theoretical and experimental studies designed to provide an answer to the above question that will ultimately have a much stronger impact on practice than results based on asymptotic, worst-case complexity.

Fikes and Nilsson (1971) proposed the *STRIPS* formulation for planning problems. Occasionally derided as an inadequate formulation for planning problems,

this formulation is surprisingly expressive and serves as the semantic foundations on which to build a very general theory of planning.

The STRIPS formulation employs propositional logic as a language for describing states of the world, a simple but general format for specifying operators, and a clear semantics for change over time. In the STRIPS formulation, there is a set of *conditions*, e.g., a set of boolean variables, used to describe states. A *complete assignment* maps the set of conditions to possible values, e.g., truth values, and a *partial assignment* maps a subset of conditions to values. A *state* is a complete assignment. An operator consists of two partial assignments; the first, called the *preconditions*, determines the states in which the operator can be applied, and the second, called the *postconditions*, determines the state that results from applying the operator to a particular starting state. Implicit *frame axioms* specify that the value of any condition not mentioned in an operator's postconditions is unchanged by the application of the operator.

A planning problem (instance) is specified by a set of operators, an *initial state*, and a *goal* specified by a partial assignment. Generally speaking, a solution is a partially ordered multiset of operators such that any total ordering consistent with the given partial order transforms the assignment specified by the initial state into an assignment that satisfies the goal formula. The STRIPS formulation can be extended to handle external events, multiple agents, probabilistic transformations, and variants that allow the agent to observe aspects of the current state and choose actions conditioned on observations.

The STRIPS formulation has a number of desirable properties that we wish to preserve in extending it to handle planning problems in stochastic domains. First of all, states are described compactly in terms of conditions that roughly correspond to the state variables used in control theory. The resulting model of the underlying dynamics is said to be *factored* or *factorial*. Another desirable property of the STRIPS formulation concerns the exploitation of *locality* in specifying how actions precipitate change over time.

In the case of the STRIPS formulation, the dynamical model is discrete time and finite state and hence results from automata theory and Markov chain theory apply directly. The factored method of description constitutes a departure from the standard formulations for finite state machines and their stochastic counterparts which generally assume a flat, unstructured state space. The representation for change over time is locally specified in the sense that each operator can be described in terms of a small number of conditions. This property of locality means that the dynamics can be encoded in space which is linear in the number of state variables assuming a constant bound on the number of conditions appearing in an operator specification. This local linearity is possible despite the fact

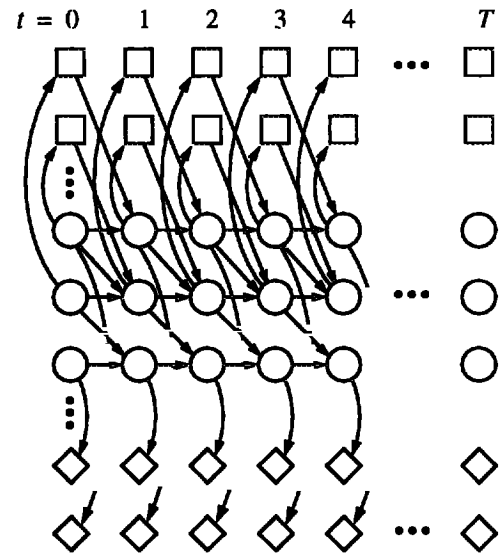


Figure 1: An operator dependency diagram.

that the total number of states is exponential in the number of conditions.

Figure 1 provides a general representation for the dynamics in discrete-time planning problems. In Figure 1, circles (○) indicate conditions, boxes (□) indicate operators, and diamonds (◇) indicate clauses in the conjunctive normal form of the boolean formula specifying the goal. An arc from a condition to an operator indicates a precondition, an arc from an operator to a condition indicates a postcondition, an arc from a condition to a condition a frame axiom, and an arc from a condition to a clause indicates that the corresponding condition is present as a literal in the clause. Figure 1, referred to as an *operator dependency graph*, allows us to reason about different instances of the same operator invoked at different times and identify dependencies involving operator sequences of length T or less.

Mitchell *et al.* (1992) have shown that the problem of satisfiability for 3CNF formulas has an associated phase transition with respect to the fraction of satisfiable instances. As the ratio of clauses to variables increases, at first the fraction of satisfiable instances is asymptotically zero and then, at a threshold of about 4.3, the fraction abruptly jumps to one. What is even more interesting is that these instances exhibits a definite hard/easy pattern with respect to one of the best-known algorithms for satisfiability, the Davis-Putnam procedure. In a rather narrow interval about 4.3, the Davis-Putnam procedure takes a long time on average and outside this interval it takes very little time.

Bylander (1996) provides empirical evidence that propositional STRIPS planning exhibits threshold phenomena similar to those observed in the case of boolean satisfiability. Bylander used the ratio of the number of

operators to the number of propositional conditions as an analog of the ratio of clauses to variables. He did not obtain a plot of the fraction of solvable instances as a function of the ratio of operators to conditions, but indirectly he estimated this function using a computationally inexpensive oracle for solving easy instances.

Bylander was primarily interested in investigating the performance of two very simple—conceptually and computationally—algorithms. The first algorithm, called POSTS-COVER-GOALS, returns false if there is a conjunct in the goal that does not appear in the postcondition of any operator, otherwise it returns “don’t know.” The second algorithm, called PLAN-FORWARD, performs a greedy search starting from the initial state and at each point chooses the operator that results in the state with the largest number of satisfied goal conjuncts. If at any time, no operator increases the number of satisfied conjuncts, the search halts and the algorithm returns “don’t know”. If the search reaches a goal state, it returns “true”. For a particular distribution governing random instances of propositional STRIPS planning, Bylander was able to show that the combination of POSTS-COVER-GOALS and PLAN-FORWARD can solve (find a solution if it exists and otherwise report that no solution exists) most planning instances.

Bylander’s distribution is similar in some respects to the fixed-clause-length distribution used in Mitchell *et al.* (1992). In Bylander’s fixed model, each operator has exactly r preconditions and s postconditions. In order to generate instances for a problem that is asymptotically as hard as the general propositional STRIPS problem, *i.e.*, PSPACE-complete, it is sufficient to choose $r = s = 2$ and allow at least the postconditions to appear both positively and negatively (Bylander 1994). Fixing the number of conditions and operators, generate an instance by choosing each operator independently in the obvious manner.

Bylander shows a plot of effectiveness (fraction of instances solved) versus the number of operators (for a fixed number of conditions). This plot depicts a distinct hard/easy pattern reminiscent of the plots of the running time of the Davis-Putnam procedure versus the ratio of clauses to variables for boolean satisfiability. We thought it might be revealing to run experiments employing more interesting analogs of the Davis-Putnam procedure for planning, *e.g.*, SNLP (McAllester & Rosenblitt 1991), or GRAPH-PLAN (Blum & Furst 1995).

We devised an algorithm that is at least as effective as Bylander’s. The algorithm first uses Bylander’s POSTS-COVER-GOALS and PLAN-FORWARD to determine solvability, and in the case where both of these algorithms return “don’t know,” applies a slightly modified version of Blum and Furst’s GRAPH-PLAN. The modification to GRAPH-PLAN was the imposition of a fixed limit on the length of the plans it considers; this version of GRAPH-PLAN

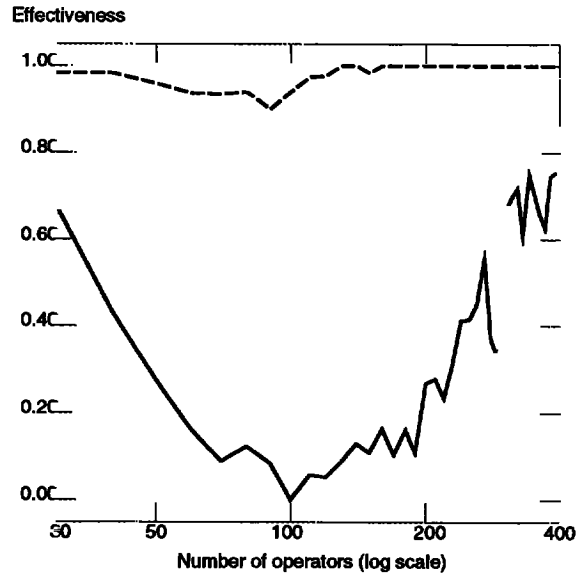


Figure 2: Effectiveness of Bylanders’s POSTS-COVER-GOALS plus PLAN-FORWARD (solid line) and the effectiveness of a combination of these two simple algorithms plus Blum and Furst’s GRAPH-PLAN (dotted line).

is therefore not complete: it may report “don’t know.”

Figure 2 illustrates some preliminary results of this approach on STRIPS problems with 30 conditions. The graph plots the effectiveness of the two algorithms as a function of the number of operators in the STRIPS problem. The solid line shows the effectiveness of POSTS-COVER-GOALS plus PLAN-FORWARD, and the dotted line shows the effectiveness of POSTS-COVER-GOALS plus PLAN-FORWARD plus GRAPH-PLAN. Effectiveness is measured as the proportion of trials in which the algorithm returns “yes” or “no”, as opposed to “don’t know.” The two simple algorithms solve a significant proportion of the problems for both small numbers of operators and large numbers of operators, but solve very few in the intermediate range. The addition of GRAPH-PLAN provides solutions to almost all the problems, but exhibits the same hard/easy pattern as Bylander’s combination. It would also be interesting to run experiments involving distributions based on other hard planning problems reported in the literature, *e.g.*, Gupta and Nau (1991) or Bäckström and Klein (1991).

One major limitation of the above approach is that randomly chosen operators do not result in random graphs, making it difficult to apply the results of the theory of random graphs. In “natural” problems, one would expect the preconditions of one operator to depend on postconditions of the same and other operators. A significant part of our current research involves running experiments using distributions that account

for the “natural” dependencies that arise in operator dependency graphs.

Community Involvement

To encourage the participation of the planning community at large in these investigations, we have put together two workshops, one in the U.S. and the other in Europe. The first will be held at AAAI-96 in Portland, Oregon and was organized in collaboration with Christer Bäckström of Linköping University, Rina Dechter of UC Irvine, and David McAllester from AT&T. This workshop is aimed at exploring the structural properties of planning and temporal reasoning problems in an attempt to tie theory and practice closer together.

The second workshop will be held at ECAI-96 in Budapest, Hungary and is being organized in conjunction with Susanne Biundo of DFKI and Richard Waldinger of SRI. Here again the theme of the workshop concerns the cross fertilization of ideas from theory and practice.

Learning Dynamical Systems

In the classical STRIPS formulation, the dynamics are assumed to be deterministic and only the actions of the agent serve to change the state of the world. We are interested in stochastic domains in which the consequences of actions are governed by probability distributions, the agent is required to choose some action at each stage (tick of the clock), and changes can result from processes outside (or only indirectly under) the control of the agent. Such domains can be represented qualitatively in terms of the sort of graph shown in Figure 1.

The complete specification of a stochastic domain requires quantifying the dependencies between conditions. A conditional probability distribution is required for each node in the graph. This distribution records the conditional probability of the corresponding condition given the conditions corresponding to its predecessors (parents) in the graph. The theory that relates to manipulating such graphical models algebraically and numerically is extensive but Pearl (1988) and Neapolitan (1990) provide good starting places. See (Dean & Kanazawa 1989; Dean & Wellman 1991; Boutilier, Dean, & Hanks 1995) for discussions on using graphical models to represent problems in planning and control.

If we are now able to compactly represent interesting real-world problems, why, you might ask, are there not droves of people begging for our algorithms. A partial answer comes from reflecting on the use of technologies developed in the field of operations research. In management and information sciences (MIS) groups throughout industry, there are legions of practitioners who know how to represent (approximately if need be) real world problems as linear programs. These folks are ready consumers of algorithms (or at least

systems that implement algorithms) to solve such linear programs. Many of these consumers were schooled in operations research methods and taught the art of representation using linear programs and the more expressive but combinatorially problematic extensions in the form of mixed-integer-and-linear programs.

Before you can solve a planning problem you have to understand it and translate it into an appropriate language for expressing operators, goals, and initial conditions. In order to make a practical impact, we need to teach students to represent planning problems in our formalisms. Unfortunately, this may not be sufficient to make a practical impact. Stochastic domains require not only that you specify the sort of operator graph specification shown in Figure 1 but also the marginal and conditional probability distributions that are needed to quantify the dependencies shown in Figure 1. Such quantitative stochastic dependencies are often hard to assess, and, for this reason, we are interested in automating the construction of (*i.e.*, learning) dynamical models for stochastic domains.

The basic idea behind learning dynamical systems is as follows. We take as input a schematic (qualitative) model of the dynamics such as that shown in Figure 1; two stages are sufficient for stationary domains in which the probabilities do not change over time. This model should capture many but not necessarily all of the dependencies in the target domain and may introduce dependencies (hopefully only a few) that are not manifest in the dynamics governing the target domain. In addition to the qualitative model, we require actual data acquired by observing the domain dynamics and, in some cases, probing the domain by executing actions to determine their (stochastic) consequences.

Given the qualitative model which serves as a prior to bias search and the data which provides a representative sample of the actual dynamics, we search for a model that serves to explain the data and, hopefully, generalizes to situations that are not explicitly represented in the data. Our investigations have taken us into diverse areas of statistics, nonlinear dynamics, adaptive control, and approximation theory. The best introduction to this work is available in the form of a web-based tutorial complete with exercises and mini-tutorials on related subjects in statistics and signal processing. The URL is <http://www.cs.brown.edu/research/ai/dynamics/>. This tutorial can be used by an individual or can form the basis for a course on learning dynamical systems.

Community Involvement

We have organized a 1996 AAAI Spring Symposium in Stanford, CA to address the issues of learning complex dynamical systems. The symposium is entitled “Computational Issues in Learning and Using Models of Dynamical Systems” and was organized jointly with James Crutchfield of UC Berkeley, Thomas Dietterich

of Oregon State University, Leslie Kaelbling of Brown University, Michael Kearns of AT&T, Melanie Mitchell of the Santa Fe Institute, David Wolpert of the Santa Fe Institute, and Brian Williams of NASA Ames Research Center.

Conclusions

It seems unlikely that we will ever completely eliminate the gap between theory and practice. The mathematical tools used to represent and analyze problems are likely to lag behind the needs of experimentalists and practitioners. Nevertheless, there are plenty of opportunities for theory to influence practice and vice versa.

In this paper, we have sketched a research strategy that deviates from asymptotic, worst-case approaches and focuses instead on particular algorithms and instances of relatively small size. The strategy borrows from successful methods that relate threshold phenomena to run-time performance in other combinatorial optimization problems.

This paper also addresses a practical problem of technology transfer concerned with constructing models of planning domains. For our planning technology to gain wider acceptance, we must provide knowledge acquisition technology to simplify the construction of domain models. Here again we outline our research strategy and provide pointers for further information and opportunities for wider community participation.

References

- Bäckström, C., and Klein, I. 1991. Parallel non-binary planning in polynomial time. In *Proceedings IJCAI 12*, 268–273. IJCAI.
- Blum, A. L., and Furst, M. L. 1995. Fast planning through planning graph analysis. In *Proceedings IJCAI 14*, 1636–1642. IJCAI.
- Boutilier, C.; Dean, T.; and Hanks, S. 1995. Planning under uncertainty: Structural assumptions and computational leverage. In *Proceedings of the Second European Workshop on Planning*.
- Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69:161–204.
- Bylander, T. 1996. A probabilistic analysis of STRIPS planning. *Artificial Intelligence*.
- Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.
- Dean, T., and Lin, S.-H. 1995. Decomposition techniques for planning in stochastic domains. In *Proceedings IJCAI 14*. IJCAI.
- Dean, T., and Wellman, M. 1991. *Planning and Control*. San Francisco, California: Morgan Kaufmann.
- Dean, T.; Kaelbling, L.; Kirman, J.; and Nicholson, A. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76(1-2):35–74.
- Fikes, R., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.
- Gupta, N., and Nau, D. S. 1991. Complexity results for blocks-world planning. In *Proceedings AAAI-91*, 629–633. AAAI.
- Lin, S.-H., and Dean, T. 1995. Generating optimal policies for high-level plans with conditional branches and loops. In *Proceedings of the Second European Workshop on Planning*.
- McAllester, D. A., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings AAAI-91*, 634–639. AAAI.
- Mitchell, D.; Selman, B.; and Levesque, H. 1992. Hard and easy distributions of SAT problems. In *Proceedings AAAI-92*, 459–465. AAAI.
- Neapolitan, R. E. 1990. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. New York: John Wiley and Sons.
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, California: Morgan Kaufmann.