Large-Scale Planning Under Uncertainty: A Survey

Michael L. Littman and Stephen M. Majercik
Dept. of Computer Science
Duke University
Durham, NC 27708-0129
{mlittman,majercik}@cs.duke.edu

Abstract

Our research area is planning under uncertainty, that is, making sequences of decisions in the face of imperfect information. We are particularly concerned with developing planning algorithms that perform well in large, real-world domains. This paper is a brief introduction to this area of research, which draws upon results from operations research (Markov decision processes), machine learning (reinforcement learning), and artificial intelligence (planning). Although techniques for planning under uncertainty are extremely promising for tackling real-world problems, there is a real need at this stage to look at large-scale applications to provide direction to future development and analysis.

INTRODUCTION

Planning—making a sequence of choices to achieve a goal—has been a mainstay of artificial intelligence (AI) research for many years. Traditionally, the decision-making models that have been studied admit no uncertainty: every aspect of the world that is relevant to the generation and execution of a plan is known in advance. In contrast, work in operations research (OR) has focussed on the uncertainty of the effects of actions, but uses an impoverished representation that does not capture relationships among states for specifying and solving planning problems.

The area of planning under uncertainty in large domains explores a middle ground between these two well-studied extremes with the hope of developing systems that can reason efficiently about plans in complex, uncertain applications. This paper reviews some of the most recently developed techniques that may scale well to large, real-world domains.

We feel that the potential benefit of applying such methods to planning problems in the space program is significant. We suggest two NASA activities in particular that would benefit greatly from the development of successful techniques for large-scale planning under uncertainty: scheduling the Deep Space Network (DSN), and on-board planning for autonomous spacecraft.

The purpose of the DSN is to support both unpiloted interplanetary spacecraft missions and radio and radar astronomy observations in the exploration of space (Chien, Lam, & Vu 1997). Planning to fulfill DSN service requests from tens of projects with varying priorities requires the consideration of thousands of possible tracks using tens of antenna resources and hundreds of subsystem configurations. The basic scheduling task is enormously complicated and made more difficult by three characteristics of DSN scheduling described by Chien, Lam, & Vu (1997):

1. The priority of a tracking request is dynamic and can change depending on the amount of tracking a project has received so far in a given time period.
2. Subsystems needed for the execution of tracks are shared by each Signal Processing Center.
3. Projects may request more tracking time than absolutely necessary and specify the absolute minimum; in these cases, the planner has the option of reducing the tracking time allotted to the request if doing so will remove a resource conflict.

There are also significant elements of uncertainty in the scheduling task (Chien *et al.* 1996; 1997):

- The needs of projects—and, thus, plan goals—change over time. Before (or even during) a track, a project may submit a request to add various services to the track.
- Equipment availability may change due to failure, maintenance or recalibration requirements, or preemption by a track with higher priority.
- Changing weather may require changes in an existing tracking schedule.

The system must respond to changing circumstances quickly and efficiently with minimal disruption.

This problem contains many elements targeted by research in planning under uncertainty in large domains. First, the sheer size and complexity means that algorithms designed to work on "toy" problems may not be adequate for this real-world problem. Second, the elements of uncertainty cited above mean that a successful plan will not be a fixed list of events, but will include contingencies for dealing with various eventualities; for example, the schedule could include alternate assignments that should be made if a new high-priority job were to appear at various points in the future—this would make it unnecessary to begin to reschedule from scratch in this circumstance (Drummond, Bresina, & Swanson 1994). For these reasons and others, a successful planner must take uncertainty into account.

The importance of on-board planning for autonomous spacecraft is emphasized by NASA's New Millennium program, the objective of which is to develop and validate new technologies that will both reduce mission costs and increase mission quality (Muscettola *et al.* 1997). The first New Millennium mission—Deep Space 1—features a Remote Agent autonomy architecture, one of whose components is a Planner/Scheduler, which translates a set of high-level mission goals specified by scientists and engineers into a sequence of low-level commands (Muscettola *et al.* 1997).

The necessity of autonomous spacecraft for deep space missions is made apparent by the extreme distance of the missions' targets, the impossibility of hands-on troubleshooting or maintenance, and the difficulties imposed by light-time delayed communication (Chien *et al.* 1997). The task of the planning component for the required autonomy technology is complicated by limited sensing capabilities, uncertain effects of actions (due to system failures or unanticipated environmental factors), and the fact that the system must construct plans without knowing with certainty the future state of the spacecraft (when the plan will begin executing). Furthermore, an autonomous spacecraft needs to be able to handle unexpected events intelligently and without lengthy deliberation. We will address these issues further in the section on partially observable Markov decision processes.

In the remainder of the paper, we describe recent developments that we feel may have an impact on solving this type of problem. The following section describes work on algorithms for planning problems formalized as Markov decision processes (MDPs), a model developed in the OR community. While MDP-related algorithms are quite mature and have been used in some fielded systems, standard treatments of MDPs use an unstructured representation of states and actions that cannot scale to many large real-world problems. In later sections, we describe some recent work geared to planning using richer state representations, and we describe planning algorithms that use richer representations of both states and actions. We also describe algorithms that address another limitation of MDPs, namely their assumption of "complete observability" during decision making. We conclude with a description of our current research efforts.

MARKOV DECISION PROCESSES

The MDP model is a formal specification for planning under uncertainty originally developed in the OR community in the late 50s and early 60s. The foundational work was done by Bellman (1957) and Howard (1960), and included a formal description of the model and basic results such as the existence of optimal solutions and algorithms for finding them. These algorithms still form the basis of nearly all the current approaches to solving MDPs.

In MDPs, an *agent* has the task of making decisions to solve some planning problem. The agent is embedded in its task environment and must decide on an action to take based on the current state of the environment. The actions it selects result in some immediate cost or benefit to the agent depending on the environment's state and can also change the environment according to its *transition model*. An intelligent agent will select actions that maximize its net benefit over a sequence of interactions with the environment.

This model presents the problem of intelligent behavior in a microcosm; the agent has a concrete measure of the success of its actions and must plan ahead to maximize its success. Compared to the larger problem of creating a broadly intelligent agent, the model includes various simplifications that make it tractable: the state space of the environment is typically finite, as is the set of possible actions the agent can choose from; time is discrete; costs and benefits are additive and not time dependent; and everything that is relevant to the decision-making problem is evident to the agent (i.e., there is complete observability—the Markov property holds for observations).

For MDPs with up to a million states or so, efficient implementations of the classic algorithms—value iteration, policy iteration, linear programming, and modified policy iteration—can be used to find optimal plans (or, policies, more precisely). Many practical problems in OR have been formalized this way and are being solved in commercial settings; Puterman (1994) describes the basic algorithms and cites applications in blast-furnace maintenance, bus-engine replacement, queuing, scheduling, fisheries management, and many others.

The DSN scheduling problem can be modeled as an MDP by creating a state for each combination of current project needs, available resources, and existing assignments. The actions create new assignments of tasks to resources and the stochastic transition model captures the arrival of new requirements and achieved needs. Similarly, planning problems for autonomous spacecraft can be modeled as MDPs in which the states capture all relevant aspects of the environment, the long and short-term goals of the mission, and the status of the spacecraft itself. The actions are the moment-to-moment decisions that the spacecraft can make for itself: firing thrusters, extending antennae, radioing home for help. The transition model describes the "physics" of the interaction between spacecraft and environment, using randomness to capture aspects of the environment that cannot be reliably predicted given the state representation of the system.

Even a modest-size version of these MDPs would likely contain trillions and trillions of states, making classical algorithms practically useless. In the following sections, we present a number of approaches that have been suggested for attacking MDP problems of this scale.

One concept that is central to both classical and recent MDP algorithms is that of a *value function.* Value functions map states of the MDP to a measure of how good it is for the agent to be in that state; they have several attributes that make them invaluable in planning under uncertainty. First, like evaluation functions in game-tree search, they can be used to guide the agent's choice of action using a simple one-step look-ahead scheme; the agent considers each possible action and chooses the one that leads to states with the highest expected value according to the value function. In fact, a standard result is that there is always an *optimal* value function that will guide the agent to making the best possible choice. Second, unlike a simple plan (fixed sequence of actions), behaving according to a value function is robust under uncertainty—at each moment, the agent chooses the best action for the state that it is actually facing. And, third, approximately correct value functions can be improved iteratively, making them easy to use in algorithms. Nearly every algorithm for planning under uncertainty uses a value function in some form.

Several AI researchers have addressed the issue of large-scale MDPs by creating variants of the classic algorithms that use heuristics to make more efficient use of computational resources. Prioritized Sweeping (Moore & Atkeson 1993) maintains an estimate of the optimal value function and uses a rule of thumb to predict when updating the value function for a particular state is likely to be important for improving the approximation. Real-time dynamic programming (Barto, Bradtke, & Singh 1995), or RTDP, attempts to find a good approximate policy quickly by focusing value-function updates on states that are likely to be visited; this approach is sometimes also called the *reinforcement-learning* approach (Kaelbling, Littman, & Moore 1996). *Envelope* methods produce a good partial policy by explicitly identifying states that are likely to be visited and solving a smaller MDP (Drummond & Bresina 1990; Tash & Russell 1994; Dean *et al.* 1995).

VALUE-FUNCTION APPROXIMATION

In the standard MDP approaches, states are represented as being completely unrelated objects. In many domains, states can be described in such a way that "similar" states have similar representations. In the autonomous spacecraft example, it is clear that a natural state representation would be one that explictly notes spacecraft location as distinct from the current objective. Therefore, simply on the basis of the "name" of the state, it is apparent that the two states could be treated similarly for planning.

This insight can be exploited by exchanging the classical table-based method for representing value functions in MDP algorithms for one that uses a function approximator (for example, a neural net) to map state-description vectors to values. A wildly successful example of this is TD-Gammon (Tesauro 1995); this work uses gradient descent and temporal-difference learning (Sutton 1988) (roughly a variant of RTDP) to train a neural-network value function to play backgammon at the level of the best human players. Playing a good game of backgammon is an interesting example of planning under uncertainty because of the impact of the dice rolls and the other player's moves on state transitions.

Several other applications have been developed using this same basic approach, including a controller for a bank of elevators (Crites & Barto 1996), a system for making cellular-phone-channel assignments (Singh & Bertsekas 1996), and a job-shop scheduler for space-shuttle payload processing (Zhang & Dietterich 1995). These commercially relevant applications exhibit a great deal of uncertainty, an astronomically huge state space (10^{20} and beyond), and have been studied closely enough that human-engineered policies are available for comparison. In each case, the automatic planning systems based on value-function approximation result in policies superior to the prior state of the art. We see no fundamental obstacles to applying value-function approximation to a broader array of problems including additional space-related applications.

Recent projects have begun to shed light on the practical and theoretical guarantees that can be made when using value-function approximation. Boyan & Moore (1995) show that representing value functions by neural networks need not behave well. Sutton (1996), however, provides a counterpoint that shows that an appropriate choice of training methodology can improve behavior significantly. Positive and negative theoretical results have been derived for gradient-descent methods (e.g., neural networks) and averaging methods (e.g., nearest neighbors) (Baird 1995; Gordon 1995; Tsitsiklis & Van Roy 1996b; Tsitsiklis & Van Roy 1996a). At present, this class of algorithms has been the most successful for solving large, practical planning problems and work continues in this area.

PROBABILISTIC PLANNING

Value-function approximation attempts to exploit structure in the state space (and the value function), but treats actions as black-box transformations from states to probability distributions over states. A promising alternative is to use symbolic descriptions of the actions to reason about entire classes of state-to-state transitions all at once. This is the approach taken in classical AI planning (McAllester & Rosenblitt 1991), and it can be a good deal more direct, and therefore more computationally efficient, than RTDP plus value-function approximation.

The classical view of planning ignores uncertainty. In the STRIPS representation (Fikes & Nilsson 1971), for example, an operator has a list of preconditions that must be satisfied before the operator can be applied. But, when these conditions are met and the operator is applied, the effects of the operator—specified as a list—take place with certainty. Uncertainty can be introduced gently into the STRIPS representation by assuming a deterministic domain with a small amount of "external" randomness (Blythe 1994). A number of researchers have explored the problem of planning given more general representations of stochastic operators (Goldman & Boddy 1994; Kushmerick, Hanks, & Weld 1995; Boutilier, Dearden, & Goldszmidt 1995).

The BURIDAN system (Kushmerick, Hanks, & Weld 1995) exploits a general representation for stochastic STRIPS operators and extends partial-order planning to stochastic domains. Its representation can express arbitrary MDPs, sometimes logarithmically more compactly than traditional OR representations. Similar to traditional deterministic planners, the plans found by BURIDAN are simple sequences of actions. C-BURIDAN (Draper, Hanks, & Weld 1994) extends the BURIDAN system so that the plan representation is more powerful; it can express contigent execution of plan actions, although it is still less powerful than a policy-type representation.

Another area of interest is in solving MDPs expressed in a compact STRIPS-like representation using adaptations of classic MDP algorithms. Boutilier, Dearden, & Goldszmidt (1995) show how the policy-iteration and value-iteration algorithms can be adapted to manipulate compact representations; Boutilier & Dearden (1996) extend this to deal with approximations of the value function. Dearden & Boutilier (1997) adopt the view that value-function approximation is a type of "abstraction," the form of which can be derived automatically from a propositional representation of the planning problem.

Although some promising algorithms have been described in the past few years, we are only just beginning to understand the computational properties of this class of problems. Littman (1997) provides some basic complexity results for probabilistic planning and also shows that most natural compact representation schemes are equivalently compact (to within polynomial factors). Goldsmith, Littman, & Mundhenk (1997) directly address the problem of finding compact representations of plans in stochastic domains; although most problems are computationally intractable, there are some ways of formalizing the problem that may be amenable to heuristic approaches. In a later section, we sketch some of our early work in using these insights to design a new type of planner for large-scale stochastic domains.

PARTIALLY OBSERVABLE MDPs

In real applications, especially those that involve physical devices whose effects on the agent's environment are uncertain, it is often impossible for the decision-making agent to base its choices on the true state of the world; in general, there will always be aspects of the world that are not directly or instantaneously accessible to the agent's sensors. Autonomous spacecraft, for example, are situated in unknown environments, possess limited sensing capabilities, and have a repertoire of actions whose effects are uncertain due to possible system malfunctions or unanticipated environmental factors. In this type of situation, value-function-based algorithms for planning do not work properly. Partially observable Markov decision processes (POMDPs) model the situation in which the agent must cope with uncertainty in its estimate of the current state (Lovejoy 1991; Cassandra, Kaelbling, & Littman 1994).

In general, the POMDP framework extends MDPs to a much wider range of potential applications. However, the resulting problems are often considerably more difficult to solve. New exact algorithms have been designed that can solve larger and more complex problems than those described in the OR literature (Littman, Cassandra, & Kaelbling 1996; Cassandra, Littman, & Zhang 1997); however, even these algorithms can run into difficulty finding optimal solutions to problems with more than a dozen or so states.

Algorithms dependent on heuristics (Washington 1996; Hansen 1994), approximations (Littman,

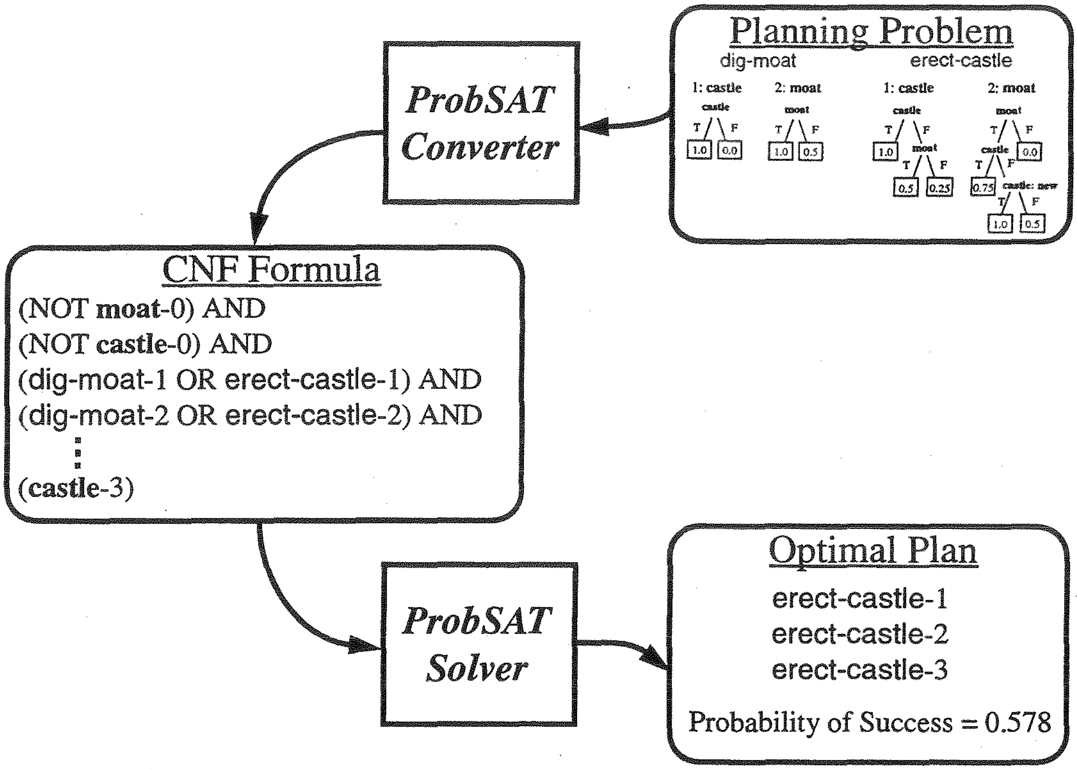

Figure 1: Block Diagram Illustrating the Steps of Our Planner

Cassandra, & Kaelbling 1995; Parr & Russell 1995; Simmons & Koenig 1995), and propositional representations (Draper, Hanks, & Weld 1994; Boutilier & Poole 1996) are being developed and have exhibited improvements in problem size and solution speed over exact approaches. This seems to be an area of intense interest, and many new approaches are being devised, although few of these have been rigorously tested at this time.

CURRENT RESEARCH

Presently, we are exploring an alternate approach to planning under uncertainty that attempts to combine the positive attributes of the various approaches described above. Our system is applicable to probabilistic planning problems, both MDPs and POMDPs, specified in a compact form. It is designed to find plans that maximize the probability of achieving a goal, but under constraints on the size and representation of the plan.

Our approach is inspired by the SATPlan planner of Kautz & Selman (1996). SATPlan uses a propositional logic problem encoding and stochastic search to solve very hard deterministic planning problems as much as an order of magnitude faster than the next best planning system. Briefly, SATPlan converts a deterministic planning problem to a Boolean satisfiability problem by constructing a CNF Boolean formula that has the property that any satisfying assignment to the variables in the formula corresponds to a valid plan—one that results in achieving the goal. This is done by making satisfiability equivalent to the enforcement of the following conditions:

- the initial conditions and goal conditions hold at the appropriate times,
- the existence of an effect at time t implies the disjunction of all actions that can produce that effect at time $t-1$,
- the use of an operator at time t implies the existence of its preconditions at time $t-1$, and
- conflicting actions are mutually exclusive (Kautz & Selman 1996).

The satisfiability of the resulting CNF formula is determined using WalkSAT, a generic satisfiability algorithm based on random hill-climbing.

To adapt this method to probabilistic planning, we make a distinction between *choice variables*, which encode the possible plans, and *chance variables*, which encode the uncertainty in the probabilistic planning problem. Choice variables, like all the variables in a SATPlan encoding, can be arbitrarily set to `True` or `False` in the search for a satisfiable assignment. The truth assignment of chance variables cannot be arbitrarily set; each such variable is True with a certain probability and, together, the chance variables determine the probability that a given truth assignment for the choice variables (i.e., a given plan) will actually lead to a satisfying assignment (i.e., reach a goal).

Given a CNF formula with chance variables, we wish to determine, not merely the satisfiability of the formula (as in SATPlan), but rather the assignment of choice variables that has the highest probability of producing an overall satisfying assignment. Such an assignment to the choice variables maximizes the probability of reaching a goal. This means that, for each setting of the choice variables (each plan), we must find all possible "assignments" to the chance variables that produce an overall satisfying assignment, and sum the probabilities of these probabilistic assignments to determine the probability of success for that plan. The computational complexity of this problem is NP^{PP}-complete (Goldsmith, Littman, & Mundhenk 1997); thus, the problem of finding the optimal restricted-size plan for a finite-horizon POMDP or MDP in compact representation is likely more difficult than an NP-complete problem like finding the optimal finite-horizon deterministic plan, but likely easier than an EXPSPACE-complete problem like finding the optimal *unrestricted* plan for a finite-horizon POMDP in compact representation (Goldsmith, Lusena, & Mundhenk 1996).

Our technique for solving these probability-extended satisfiability problems is based on the Davis-Putnam procedure for determining satisfiability (Davis, Logemann, & Loveland 1962) and can be envisioned as constructing a binary tree in which each node represents a choice variable or a chance variable, and the two subtrees represent the two possible remaining subproblems given the two possible assignments to the parent variable (if the parent is a choice variable) or the two possible outcomes (if the parent is a chance variable). Clearly, it is critical to construct an efficient tree to avoid evaluating an exponential number of assignments. As in the Davis-Putnam procedure, we do this by selecting for the next variable in the tree, whenever possible, a variable that appears by itself in a clause, or a variable that appears in only one sense (always negated or always not negated) in the formula. We also remove variables which become irrelevant as the tree is constructed. When no such obvious choice is possible, we choose the variable that satisfies the most clauses in the remaining subproblem.

We summarize our current approach to planning under uncertainty in Figure 1, which shows a solution to a problem described by Goldsmith, Littman, & Mundhenk (1997).

The basic insight of our approach—solve a probabilistic planning problem by converting it to an equivalent problem on a Boolean formula—has a number of attractive properties. By casting the problem in a generic format we can take advantage of proven algorithmic techniques from a number of areas of research. Our current solver already incorporates successful ideas from dynamic programming and Boolean satisfiability; future versions will also draw on the areas of AI planning, belief networks, reinforcement learning, and stochastic search.

CONCLUSIONS

Over the past few years, powerful formal models of planning under uncertainty have been explored and scores of new algorithms for planning with these models have been devised. The models appear to be useful for expressing real-world, practical problems and the algorithms show a great deal of promise for solving these problems.

One component that is sorely lacking in this picture is substantive contact with real applications; many algorithms have been tested only on tiny "proof-of-concept" problems and no information is available on how the algorithms scale up. It is critical at this stage to make a serious effort to apply these nascent technologies to real-world problems; only through contact with applications can we hope to focus our research effort in the most promising and productive directions. We believe that planning problems in the space industry would provide an excellent test bed for algorithms for planning under uncertainty that aspire to be effective in real-world domains. The interaction between emerging techniques and real problems will spur the development of more efficient and practical systems as well as contribute to the solution of some important problems.

References

Baird, L. 1995. Residual algorithms: Reinforcement learning with function approximation. In Prieditis, A., and Russell, S., eds., *Proceedings of the Twelfth International Conference on Machine Learning*, 30–37. San Francisco, CA: Morgan Kaufmann.

Barto, A. G.; Bradtke, S. J.; and Singh, S. P. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 72(1):81–138.

Bellman, R. 1957. *Dynamic Programming.* Princeton, NJ: Princeton University Press.

Blythe, J. 1994. Planning with external events. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 94–101.

Boutilier, C., and Dearden, R. 1996. Approximating value trees in structured dynamic programming. In Saitta, L., ed., *Proceedings of the*

Thirteenth International Conference on Machine Learning.

Boutilier, C., and Poole, D. 1996. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1168–1175. AAAI Press/The MIT Press.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting structure in policy construction. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1104–1113.

Boyan, J. A., and Moore, A. W. 1995. Generalization in reinforcement learning: Safely approximating the value function. In Tesauro, G.; Touretzky, D. S.; and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7*, 369–376. Cambridge, MA: The MIT Press.

Cassandra, A. R.; Kaelbling, L. P.; and Littman, M. L. 1994. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1023–1028.

Cassandra, A.; Littman, M. L.; and Zhang, N. L. 1997. Incremental Pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, 54–61. San Francisco, CA: Morgan Kaufmann Publishers.

Chien, S.; Govindjee, A.; Wang, X.; Estlin, T.; and Hill, Jr., R. 1996. Integrating hierarchical task-network and operator-based planning techniques to automate operations of communications antennas. *IEEE Expert* 11(6):9–11.

Chien, S.; DeCoste, D.; Doyle, R.; and Stolorz, P. 1997. Making an impact: artificial intelligence at the Jet Propulsion Laboratory. *AI Magazine* 18(1):103–122.

Chien, S.; Lam, R.; and Vu, Q. 1997. Resource scheduling for a network of communications antennas. In *IEEE Aerospace Conference Proceedings*, 361–373.

Crites, R. H., and Barto, A. G. 1996. Improving elevator performance using reinforcement learning. In Touretzky, D. S.; Mozer, M. C.; and Hasselmo, M. E., eds., *Advances in Neural Information Processing Systems 8.* Cambridge, MA: The MIT Press.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem proving. *Communications of the ACM* 5:394–397.

Dean, T.; Kaelbling, L. P.; Kirman, J.; and Nicholson, A. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76(1-2):35–74.

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

Draper, D.; Hanks, S.; and Weld, D. 1994. Probabilistic planning with information gathering and contingent execution. In *Proceedings of the AAAI Spring Symposium on Decision Theoretic Planning*, 76–82.

Drummond, M., and Bresina, J. 1990. Anytime synthetic projection: Maximizing the probability of goal satisfaction. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 138–144. Morgan Kaufmann.

Drummond, M.; Bresina, J.; and Swanson, K. 1994. Just-in-case scheduling. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 1098–1104. Seattle, WA: AAAI Press.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208. Reprinted in *Readings in Planning*, J. Allen, J. Hendler, and A. Tate, eds., Morgan Kaufmann, 1990.

Goldman, R. P., and Boddy, M. S. 1994. Epsilon-safe planning. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI94)*, 253–261.

Goldsmith, J.; Littman, M. L.; and Mundhenk, M. 1997. The complexity of plan existence and evaluation in probabilistic domains. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, 182–189. San Francisco, CA: Morgan Kaufmann Publishers.

Goldsmith, J.; Lusena, C.; and Mundhenk, M. 1996. The complexity of deterministically observable finite-horizon Markov decision processes. Technical Report 268-96, Department of Computer Science, University of Kentucky.

Gordon, G. J. 1995. Stable function approximation in dynamic programming. In Prieditis, A., and Russell, S., eds., *Proceedings of the Twelfth International Conference on Machine Learning*, 261–268. San Francisco, CA: Morgan Kaufmann.

Hansen, E. A. 1994. Cost-effective sensing during plan execution. In *Proceedings of the Twelfth National Conference on Artificial Intelligence.* AAAI Press/The MIT Press. 1029–1035.

Howard, R. A. 1960. *Dynamic Programming and Markov Processes.* Cambridge, Massachusetts: The MIT Press.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Kautz, H., and Selman, B. 1996. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1194–1201. AAAI Press/The MIT Press.

Kushmerick, N.; Hanks, S.; and Weld, D. S. 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76(1-2):239–286.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning policies for partially observable environments: Scaling up. In Prieditis, A., and Russell, S., eds., *Proceedings of the Twelfth International Conference on Machine Learning*, 362–370. San Francisco, CA: Morgan Kaufmann.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1996. Efficient dynamic-programming updates in partially observable Markov decision processes. Technical Report CS-95-19, Brown University, Providence, RI.

Littman, M. L. 1997. Probabilistic propositional planning: Representations and complexity. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 748–754. AAAI Press/The MIT Press.

Lovejoy, W. S. 1991. A survey of algorithmic methods for partially observable Markov decision processes. *Annals of Operations Research* 28(1):47–65.

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings of the 9th National Conference on Artificial Intelligence*, 634–639.

Moore, A. W., and Atkeson, C. G. 1993. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning* 13:103–130.

Muscettola, N.; Fry, C.; Rajan, K.; Smith, B.; Chien, S.; Rabideau, G.; and Yan, D. 1997. Onboard planning for New Millennium Deep Space One autonomy. In *IEEE Aerospace Conference Proceedings*, 303–318.

Parr, R., and Russell, S. 1995. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the International Joint Conference on Artificial Intelligence.*

Puterman, M. L. 1994. *Markov Decision Processes—Discrete Stochastic Dynamic Programming.* New York, NY: John Wiley & Sons, Inc.

Simmons, R., and Koenig, S. 1995. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1080–1087.

Singh, S., and Bertsekas, D. 1996. Reinforcement learning for dynamic channel allocation in cellular telephone systems. To appear in *Advances in Neural Information Processing Systems.*

Sutton, R. S. 1988. Learning to predict by the method of temporal differences. *Machine Learning* 3(1):9–44.

Sutton, R. S. 1996. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Touretzky, D. S.; Mozer, M. C.; and Hasselmo, M. E., eds., *Advances in Neural Information Processing Systems 8.* Cambridge, MA: The MIT Press.

Tash, J., and Russell, S. 1994. Control strategies for a stochastic planner. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 1079–1085.

Tesauro, G. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM* 58–67.

Tsitsiklis, J. N., and Van Roy, B. 1996a. An analysis of temporal-difference learning with function approximation. Technical Report LIDS-P-2322, Massachusetts Institute of Technology. Available through URL `http://web.mit.edu/bvr/www/td.ps`. To appear in *IEEE Transactions on Automatic Control.*

Tsitsiklis, J. N., and Van Roy, B. 1996b. Feature-based methods for large scale dynamic programming. *Machine Learning* 22(1/2/3):59–94.

Washington, R. 1996. Incremental Markov-model planning. In *Proceedings of TAI-96, Eighth IEEE International Conference on Tools With Artificial Intelligence*, 41–47.

Zhang, W., and Dietterich, T. G. 1995. A reinforcement learning approach to job-shop scheduling. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intellience*, 1114–1120.