Decision Making Under Uncertainty: Operations Research Meets AI (Again)

Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, ON M5S 3H5
cebly@cs.toronto.edu

Abstract

Models for sequential decision making under uncertainty (e.g., Markov decision processes, or MDPs) have been studied in operations research for decades. The recent incorporation of ideas from many areas of AI, including planning, probabilistic modeling, machine learning, and knowledge representation) have made these models much more widely applicable. I briefly survey recent advances within AI in the use of fully- and partially-observable MDPs as a modeling tool, and the development of computationally-manageable solution methods. I will place special emphasis on factored problem representations such as Bayesian networks and algorithms that exploit the structure inherent in these representations.

1 AI Meets OR

When one is reminded of the crossroads where artificial intelligence (AI) meets operations research (OR), the vital and active area of combinatorial optimization immediately springs to mind. The interaction between researchers in the two disciplines has been lively and fruitful. Linear and integer programming, constraint-based optimization, stochastic local search, all have broken from their "home communities" and spurred interdisciplinary advances to the extent that it is impossible to classify much of this research as strictly AI or strictly OR (except by the affiliations of the researchers). As a result, optimization problems of incredible scale are being solved on a daily basis, and our understanding of the relative strengths of various methods and how they can be combined has advanced considerably.

AI and OR meet at another, less-traveled, less-visible crossroads, passing through the area of sequential decision making under uncertainty. Sequential decision making—that is, *planning*—has been at the core of AI since its inception. Yet only in the last half-dozen years has the planning community started to seriously investigate stochastic models. *Decision-theoretic planning* (DTP) has allowed us to move beyond the classical (deterministic, goal-based) model to tackle problems with uncertainty in action effects, uncertainty in knowledge of the system state, and multiple, conflicting objectives. In this short time, *Markov decision processes* (MDPs) have become the *de facto* conceptual model for DTP.

MDPs were introduced in the OR community in the 1950s [2] and have studied and applied in OR and stochastic optimal control for decades. Both the fully-observable and partially-observable variants (FOMDPs and POMDPs, respectively) have proven to be very effective for capturing stochastic decision problems; in fact, one might view POMDPs as offering a general model in which most sequential decision problems can be cast.[1] A number of algorithms have been developed for constructing optimal policies, for both FOMDPs [2, 27] and POMDPs [45], generally based on Bellman's dynamic programming principle.

The generality of these models and algorithms comes at a price: the modeling of any specific problem as an MDP can be tedious, and the computational cost of applying a general-purpose solution algorithm is typically very high, often too high to be practical. Thus, special problem structure must generally be exploited in order to render the models practically solvable. Examples of models with special structure include linear-quadratic control problems (using Kalman filters) for POMDPs, or queuing models for FOMDPs.

AI planning problems also exhibit considerable structure. System states are composed of a number of different features (or variables or propositions). In classical planning, this fact has been exploited to great effect in the representation of deterministic actions using STRIPS [21], the situation calculus [35, 42], and a host of other formalisms. Furthermore, classical planning techniques such as regression [47] and partial-order planning [33] have exploited the structure inherent in such representations to construct plans much more effectively (in many cases) than one can through explicit, state-based methods. Other techniques such as hierarchical abstraction, decomposition, and so on, have rendered certain types of planning problems tractable as well.

The use of MDPs for DTP requires than analogous insights be applied to the decision-theoretic generalizations of classical planning problems. Two crucial tasks are therefore:

[1] For example, the exploration/exploitation tradeoff in reinforcement learning and bandit problems is best formulated as a POMDPs [3]. General models of sequential games are often formulated as (multiagent extensions of) POMDPs [37]. Models from control theory such as Kalman filters [29] are also forms of POMDPs.

(1) the development of natural and concise representations for stochastic, dynamic systems and utility functions, so that MDPs can be specified in a convenient form that exploits regularity in the domain and problem structure; and (2) the development of algorithms for policy construction that exploit this structure computationally. Fortunately, some good progress has been made in both of these directions. This is a key area where AI has much to offer in making MDPs more easily solved and, therefore, more widely applicable as a model for DTP.

In this talk I will survey a few of the techniques that have been developed recently for solving both FOMDPs and POMDPs. I will focus on a specific primarily on techniques that exploit concise system representations, much as regression or partial-order planning use (say) STRIPS action representations and propositional goal descriptions to discover structure in the set of plans they construct. There are, of course, many more concepts from various areas of AI (planning, learning, etc.) that can be—and have been—brought to bear on the effective solution of MDPs. Many techniques developed in the reinforcement learning community, for example, discover various forms of problem structure without being provided with a concise system representation: essentially structure in the representation of the *solution* is discovered without explicitly requiring that the problem be represented this way. The use of function approximation to represent value functions in reinforcement learning is a prime example of this [4]. Other techniques involve using search [1, 23], sampling [30, 46], and region-based problem decomposition [18, 25, 41, 38].

2 Action Representation

To represent stochastic actions and systems, a number of researchers have adapted a tool used for the representation of probability distributions, namely *Bayesian networks* [39], to the problem of action representation. Bayesian networks provide a formal, graphical way of decomposing a probability distribution by exploiting probabilistic independence relationships. Bayesian networks can also be augmented to represent *actions*, for instance, using the methods of *influence diagrams* [43, 39], or representations such as *two-stage* or *dynamic* Bayesian networks (DBNs) [17]. The use of DBNs has not only provided a natural and concise means of representing stochastic dynamic systems, but has also given rise to a number of computationally effective techniques for inference tasks such as monitoring, prediction and decision making.

In the talk I will briefly review DBNs and discuss why they are suitable representations for MDPs as applied to DTP. Roughly, DBNs exploit the fact that the (stochastic) effect of an action on different system variables often exhibits great probabilistic independence; furthermore, the effect on one variable may depend on the state of only a subset of system variables. For instance, the action of moving five meters in a certain direction may stochastically influence the state of a robot's battery, as well as its location; but the probabilities of various changes in these variables maybe independent. Furthermore, the robot's ending location may depend on its starting location, but not on the variable denoting the load it is carrying. These facts imply that the transition probabilities for this action—the probabilities of moving from state to state when the action is executed—exhibit a certain regularity. This regularity is exploited in the DBN representation, and we can often specify and represent system dynamics in time and space polynomial in the number of system variables (in contrast to the exponential space required by transition matrices used in the "standard" treatment of discrete-space MDPs). DBNs can be augmented with the structured representation of conditional probability tables for individual variables (e.g., using decision trees [11], algebraic decision diagrams [26], or Horn rules [40]), giving additional space savings and allowing even more natural specification. Similar remarks can be made for other components of MDPs such as reward function and observation probabilities. For a survey of these issues, see [9].

3 Abstraction

When solving an FOMDP, our aim is to produce a policy, or a mapping that associates an action to be executed with each state of the system. An optimal policy is one that has greatest expected value, where value is generally measured using some simple function of the rewards associated with the states the agent passes through as it executes the policy. For POMDPs, the goal is similar, except that we do not assume that the agent knows the true state of the system: it only obtains noisy observations of the system state. Instead, we consider a mapping from *belief states*—or probability distributions over system states—into actions. An agent's belief state reflects its uncertainty about the true state of the system when it is forced to act. For both FOMDPs and POMDPs, we generally produce optimal policies indirectly by first constructing a *value function* that measures the expected value of acting optimally at any state (or belief state), and then choosing actions "greedily" with respect to this optimal value function. Value functions are typically produced using dynamic programming algorithms.

Once again, the "standard" MDP model is intractable when we consider feature-based problems, as the number of states grows exponentially with the number of variables needed to describe the problem. Value functions and policies, which map system states (or belief states) into values or actions, cannot be represented or computed in an explicit fashion.[2] However, given that such problems generally exhibit regularity in their transition probabilities and reward functions, we might hope that value functions and policies would exhibit regularity as well. If so, then these mappings may be represented compactly.

For instance, it might be that the robot should always move to the mailroom when mail is ready to be picked up unless it's battery is low. This component of the policy mapping can be represented very concisely using a rule such as $MailReady \wedge$

[2] In fact, the space of belief states is continuous; however, value functions and policies have a nice structure that allows them to be represented finitely [45] using a set of system state-based mappings, as we discuss below.

$\neg \mathrm{L}ow Battery \rightarrow \mathrm{D}o(\mathrm{M}ove Mail Room)$. This rule represents the policy mapping for *all* states where $\mathrm{M}ail Ready \wedge \neg \mathrm{L}ow Battery$ holds. We can view this as a form of *abstraction*: details involving *other* variables are irrelevant to the choice of action when $\mathrm{M}ail Ready \wedge \neg \mathrm{L}ow Battery$ is known.

To make use of this structure, we require techniques that discover these regularities without enumerating states. Decision-theoretic generalizations of goal regression for FOMDPs have been developed recently that do just this [11, 6, 20]. These methods exploit the structure in the MDP representation to determine which variables are relevant to prediction of expected value and action choice, and which can be ignored. Furthermore, these determinations are conditional (e.g., $\mathrm{M}ail Ready$ may be relevant when $\neg \mathrm{L}ow Battery$ holds, but may be irrelevant when $\mathrm{L}ow Battery$ is true). These methods essentially organize and "cluster" the classic dynamic programming computations using the structure laid bare by the MDP representation. For instance, in [11] we use decision trees to represent policies and value functions, and exploit DBN and decision tree MDP representations to discover the appropriate decision tree structure. These methods often render large MDPs tractable because they obviate the need for state space enumeration.

These techniques have been extended to other representations such as algebraic decision diagrams [26] (which have proved to offer very large impact on computational savings). A general view of this approach in terms of automaton minimization is proposed in [14]. Furthermore, these representations provide tremendous leverage for approximation [19, 10, 16]. I will briefly review some of these developments in the talk.

Similar ideas have been applied to POMDPs. Though value functions for POMDPs map belief states to expected value, rather than system states, Sondik [45] has shown that these continuous functions are piecewise linear and convex: as a result they can be represented using a finite collection of *state-based* value functions. These state-based value functions can be computed by dynamic programming. In [12], we apply abstraction algorithms similar to those described above to discover suitable decision-tree structure for the state-based value functions that make up the POMDP, belief state-based, value function. Hansen and Feng [24] extend these ideas to more sophisticated POMDP algorithms and use an ADD representation, providing some encouraging results.

POMDPs offer an additional complication: for the approaches described above, an agent must maintain a belief state as the system evolves. After each action and observation of the system, this distribution over system states is updated and the optimal action for the new belief state is determined. This is, of course, a computationally intractable process in general since the number of system states is itself unmanageable. Furthermore, the process is online rather than offline, so computational intractability is a much more pressing concern. DBN representations are designed for precisely this reason, exploiting independencies among variables in order to more compactly maintain distributions as they evolve over time. Unfortunately, as shown convincingly by Boyen and Koller [13], exact belief state monitoring is intractable even given very concise DBN representations. However, they have shown how partially-observable processes can be monitored approximately. The choice of approximation method can be informed by the DBN representation of the system and error bounds on the difference between the true and approximate belief state can be constructed quite easily. In fact, we can view their approximation scheme as a form of abstraction.

Though the Boyen and Koller scheme was not designed for POMDPs, it can certainly be applied to POMDP belief state monitoring. The method for generating error bounds is not directly applicable unfortunately: it does not account for error in decision quality induced by error in belief state. Recent work by McAllester and Singh [34] has related belief state error to decision quality error in POMDPs. In the talk, I briefly describe work done jointly with Pascal Poupart on using DBN representations to make construct such bounds and to select an appropriate "abstraction" for the belief state monitoring process.

4 Decomposition

Another important way in which DBN problem representations can be exploited computationally is through problem decomposition. In many instances, MDPs have reward function devised of several additive, independent components (in the sense of multi-attribute utility theory [31]). Often the individual components of the reward function can be influenced only by certain actions or certain system variables. For each individual objective, we can often use the DBN representation of an MDP to (often rather easily) construct a smaller MDP comprised of only those variables relevant to the achievement of that objective [7]. In other cases, the decomposition may be given to us directly. These sub-MDPs will be considerably smaller than the original MDP (exponentially smaller in the number of irrelevant variables), and can be solved using standard techniques to produce a policy (and value function) that dictates how best to achieve that individual objective. Given the policies and value functions for these sub-MDPs, the question remains: how does one produce a globally-optimal (or near-optimal) policy using the component value functions for guidance?

The question of merging policies has been addressed in [7, 36, 44]. Essentially, the component value functions can be used as heuristics to guide the search for a global policy.

A related viewpoint is adopted in [32], where an additive structure is *imposed* on a value function. Given a set of basis functions (generally defined over some subset of the system variables), value determination for a given policy can be constrained to have the form of a weighted sum of the basis functions. The factored nature of the state space can be thus exploited in constructing basis functions. It seems clear that DBN representations of MDPs could be used to construct reasonable basis functions automatically, though this idea hasn't been pursued.

5 The Future

The use of structured representation to bring about the effective solution of both FOMDPs and POMDPs shows great promise. With FOMDPs, very large problems involving up to one billion states have been solved exactly using structured representations and solution techniques such as those described above. While POMDPs have proven to be more difficult to handle efficiently, structured approaches are beginning to offer some encouragement. There is of course much that remains to be done.

An important set of tasks involves the integration of structured representations and solution methods with other methods for solving MDPs. For example, while sampling has proven to be an effective means of belief state monitoring [28] and shows hope for solving POMDPs [46], DBN representations of dynamics and structured representation of value functions can be used to make sampling far more effective by focusing attention on (or diverting attention from) those areas of state space where variance in estimates can have a greater (or lesser) impact on decision quality. Many of these techniques (e.g., decomposition and abstraction) can be integrated with one another rather easily, since they tend to focus on complementary forms of structure.

Many other types of MDP structure can be captured or discovered more easily using AI-style problem representations as well. Reachability analysis can be made much more effective using DBN problem representations [8], and can aid in the solution of MDPs (much like the reachability analysis implemented by GraphPlan [5] accelerates goal regression). Large action spaces can sometimes be represented compactly using such representations [15], as well.

One of the most glaring deficiencies of these representations is the inability to deal with relational concepts and quantification, things taken for granted in classical AI knowledge representation. This isn't to say that logical representations of probabilistic concepts are unknown. Poole's independent choice logic [40] (to take one example) offers a means of representing stochastic action using relations and variables. Extensions of Bayesian networks offer similar possibilities [22]. An extension of the abstraction and decomposition ideas discussed in the talk to first-order representations of MDPs will provide a major step toward making MDPs a standard, practical tool for AI applications. Similarly, the extension of these ideas to continuous or hybrid domains will expand the range of practical applicability of MDPs considerably.

The activity at the MDP crossroads, where AI and OR meet yet again, is increasing. AI style representational methods and computational techniques are leading the way in taking FOMDPs and POMDPs from being simply a nice, conceptual, mathematical model of sequential decision making to becoming a practical technology for stochastic decision problems. This is indeed fortunate since the conceptual model is the right one for so many problems within AI, and outside.

References

[1] A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1–2):81–138, 1995.

[2] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, 1957.

[3] Donald A. Berry and Bert Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall, London, 1985.

[4] Dimitri P. Bertsekas and John. N. Tsitsiklis. *Neuro-dynamic Programming*. Athena, Belmont, MA, 1996.

[5] Avrim L. Blum and Merrick L. Furst. Fast planning through graph analysis. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1636–1642, Montreal, 1995.

[6] Craig Boutilier. Correlated action effects in decision theoretic regression. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 30–37, Providence, RI, 1997.

[7] Craig Boutilier, Ronen I. Brafman, and Christopher Geib. Prioritized goal decomposition of Markov decision processes: Toward a synthesis of classical and decision theoretic planning. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 1156–1162, Nagoya, 1997.

[8] Craig Boutilier, Ronen I. Brafman, and Christopher Geib. Structured reachability analysis for Markov decision processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 24–32, Madison, WI, 1998.

[9] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[10] Craig Boutilier and Richard Dearden. Approximating value trees in structured dynamic programming. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 54–62, Bari, Italy, 1996.

[11] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1104–1111, Montreal, 1995.

[12] Craig Boutilier and David Poole. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1168–1175, Portland, OR, 1996.

[13] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 33–42, Madison, WI, 1998.

[14] Thomas Dean and Robert Givan. Model minimization in Markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 106–111, Providence, 1997.

[15] Thomas Dean, Robert Givan, and Kee-Eung Kim. Solving planning problems with large state and action spaces. In *Fourth International Conference on Artificial Intelligence Planning Systems*, pages 102–110, Pittsburgh, PA, 1998.

[16] Thomas Dean, Robert Givan, and Sonia Leach. Model reduction techniques for computing approximately optimal solutions for Markov decision processes. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 124–131, Providence, RI, 1997.

[17] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[18] Thomas Dean and Shieu-Hong Lin. Decomposition techniques for planning in stochastic domains. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1121–1127, Montreal, 1995.

[19] Richard Dearden and Craig Boutilier. Abstraction and approximate decision theoretic planning. *Artificial Intelligence*, 89:219–283, 1997.

[20] Thomas G. Dietterich and Nicholas S. Flann. Explanation-based learning and reinforcement learning: A unified approach. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 176–184, Lake Tahoe, 1995.

[21] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[22] Nir Freidman, Daphne Koller, and Avi Pfeffer. Structured representation of complex stochastic systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 157–164, Madison, 1998.

[23] Hector Geffner and Blai Bonet. High-level planning and control with incomplete information using POMDPs. In *Proceedings Fall AAAI Symposium on Cognitive Robotics*, Orlando, FL, 1998.

[24] Eric A. Hansen and Zhengzhu Feng. Dynamic programming for pomdps using a factored state representation. In *Proceedings of the Fifth International Conference on AI Planning Systems*, Breckenridge, CO, 2000. to appear.

[25] Milos Hauskrecht, Nicolas Meuleau, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Hierarchical solution of Markov decision processes using macro-actions. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 220–229, Madison, WI, 1998.

[26] Jesse Hoey, Robert St-Aubin, Alan Hu, and Craig Boutilier. SPUDD: Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 279–288, Stockholm, 1999.

[27] Ronald A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, 1960.

[28] Michael Isard and Andrew Blake. CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–18, 1998.

[29] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[30] Michael Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1324–1331, Stockholm, 1999.

[31] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, New York, 1976.

[32] Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured mdps. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1332–1339, Stockholm, 1999.

[33] David McAllester and David Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 634–639, Anaheim, 1991.

[34] David McAllester and Satinder Singh. Approximate planning for factored POMDPs using belief state simplification. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 409–416, Stockholm, 1999.

[35] John McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[36] Nicolas Meuleau, Milos Hauskrecht, Kee-Eung Kim, Leonid Peshkin, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Solving very large weakly coupled Markov decision processes. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 165–172, Madison, WI, 1998.

[37] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, 1991.

[38] Ronald Parr. Flexible decomposition algorithms for weakly coupled Markov decision processes. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 422–430, Madison, WI, 1998.

[39] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.

[40] David Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94(1–2):7–56, 1997.

[41] Doina Precup, Richard S. Sutton, and Satinder Singh. Theoretical results on reinforcement learning with temporally abstract behaviors. In *Proceedings of the Tenth European Conference on Machine Learning*, pages 382–393, Chemnitz, Germany, 1998.

[42] Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation (Papers in Honor of John McCarthy)*, pages 359–380. Academic Press, San Diego, 1991.

[43] Ross D. Shachter. Evaluating influence diagrams. *Operations Research*, 33(6):871–882, 1986.

[44] Satinder P. Singh and David Cohn. How to dynamically merge Markov decision processes. In *Advances in Neural Information Processing Systems 10*, pages 1057–1063. MIT Press, Cambridge, 1998.

[45] Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

[46] Sebastian Thrun. Monte Carlo POMDPs. In *Proceedings of Conference on Neural Information Processing Systems*, 1999. to appear.

[47] Richard Waldinger. Achieving several goals simultaneously. In E. Elcock and D. Mitchie, editors, *Machine Intelligence 8: Machine Representations of Knowledge*, pages 94–136. Ellis Horwood, Chichester, England, 1977.