AI Magazine Volume 17 Number 4 (1996) (© AAAI)

The National Science Foundation Workshop on Reinforcement Learning

Sridhar Mahadevan and Leslie Pack Kaelbling

■ Reinforcement learning has become one of the most actively studied learning frameworks in the area of intelligent autonomous agents. This article describes the results of a three-day meeting of leading researchers in this area that was sponsored by the National Science Foundation. Because reinforcement learning is an interdisciplinary topic, the workshop brought together researchers from a variety of fields, including machine learning, neural networks, AI, robotics, and operations research. Thirty leading researchers from the United States, Canada, Europe, and Japan, representing from many different universities, government, and industrial research laboratories participated in the workshop. The goals of the meeting were to (1) understand limitations of current reinforcement-learning systems and define promising directions for further research; (2) clarify the relationships between reinforcement learning and existing work in engineering fields, such as operations research; and (3) identify potential industrial applications of reinforcement learning.

In recent years, a unifying viewpoint based on embedded autonomous agents has shaped much work in AI (Russell and Norvig 1994). Examples of such agents include robots operating in unstructured environments, softbots navigating the Internet, and even industrial controllers operating some complex machinery. Reinforcement learning has become one of the most actively studied learning frameworks in the area of embedded autonomous agents. Reinforcement learning has attracted researchers from an eclectic mix of fields, including machine learning, neural networks, robotics, AI, and engineering. In recognition of the growing importance of reinforcement learning, it seemed an opportune time to bring together leading researchers from these areas for a three-day meeting consisting of general and wide-ranging discussions. The National Science Foundation (NSF) sponsored the workshop with a generous grant to cover the travel and lodging costs of all participants. The participants sought to assess the state of the art of reinforcement learning today; outline promising directions for further work; clarify links between reinforcement learning and existing work in dynamic programming; and, finally, explore potential industrial applications of reinforcement learning.

What Is Reinforcement Learning?

Reinforcement learning is a general framework for describing learning problems in which an agent learns strategies for interacting with its environment (figure 1). The agent perceives something about the state of its environment and chooses what it thinks is an appropriate action. The world's state changes (not necessarily deterministically), and the agent receives a scalar reward, or reinforcement, indicating the utility of the new state for the agent. The agent's goal is to find, based on its experience with the environment, a strategy or an optimal policy for choosing actions that will yield as much reward as possible.

There are two major designs for a reinforcement-learning agent. In the *model-based approach*, the agent learns a model of the dynamics of the world and its rewards. Given the model, it tries to solve for the optimal control policy. In the *model-free approach*, the agent tries to learn the optimal control policy directly, without first constructing a world model. In either approach, the agent seeks to learn a policy that maximizes some cumulative measure of reinforcement received from the environment. The most well-known reinforcement-learning algorithms are based on a discounted framework where future rewards are reduced by some geometrically decreasing constant factor. The most well-known discounted algorithms include Q-learning, developed by Watkins (1989), and TD(λ), developed by Sutton (1988). These algorithms have been used to solve some large real-world sequential decision problems, including grand-master play in backgammon (Tesauro 1992), job-shop schedules for the space shuttle (Zhang and Dietterich 1995), a team of elevators in a multistory building (Crites and Barto 1996), and frequency assignments for cellular telephones (Bertsekas and Tsitsiklas 1996). As we discuss next, many of these problems can be formulated using a control-theory framework called *dynamic programming*.

Markov Decision Processes and Dynamic Programming

A key assumption underlying much research in reinforcement learning is that the agent-environment interaction can be viewed as a Markov decision process (MDP) (Puterman 1994). The MDP model implies that the current state of the environment perceived by the agent and the action selected by the agent together determine a fixed- (but unknown) probability distribution on the next state and immediate reward. This model is, thus, *memoryless*, in that the agent does not need to consider the history of previous states and actions in determining an optimal policy. However, there is growing attention in

that underlies much of the work in dynamic programming as well as reinforcement learning. He also described the standard dynamic programming algorithms, such as policy iteration and value iteration for computing optimal policies; variations on these algorithms, for example, modified policy iteration; and, finally, the relationship between discounted optimality and other optimality measures, such as maximizing the expected average reward.

One of the most influential models for reinforcement learning is the *actor-critic system* proposed by Barto, Sutton, and Anderson (1983). Here, the actor is responsible for executing a policy, and the critic learns to solve the credit-assignment problem of evaluating policies. In the second talk at the workshop, Dimitri Bertsekas (Massachusetts Institute of Technology [MIT]) discussed a class of actor-critic methods for approximate dynamic programming that has been used with considerable success for solving challenging large-scale problems. He showed that there is a fundamental structure, common to all these methods, that causes oscillations. In particular, he described a generically occurring phenomenon, called *chattering*, in which oscillation in policy space and convergence in parameter space (for example, the weights in a neural network stabilize) simultaneously occur. Furthermore, the limit to which the parameter sequence converges need not correspond to any of the policies of the problem. This result underlines the point that although reinforcement-learning techniques have had some good successes in practice, a great deal of theoretical work remains to understand what underlies their success.

Reinforcement Learning Andrew Barto (University of Massachusetts at Amherst) observed that the most dramatic reinforcement-learning successes to date have been achieved in completely offline applications in which experience was generated entirely using simulation models of the systems of interest. Examples include Tesauro's TD-GAMMON system (1992), the elevator-dispatching system developed in his group by Robert Crites (Crites and Barto 1996), and Zhang and Dietterich's (1995) National Aeronautics and Space Administration (NASA) job-shop–scheduling system. He pointed out that a great advantage of these methods is that although they require models, these models do not need to be explicit probability models of an MDP; simulation models, which are often much easier to obtain, suffice.

Richard Sutton (University of Massachusetts at Amherst) discussed some problems with Q-learning, one of the most popular model-free reinforcement-learning algorithms originally developed by Watkins (1989). Among the drawbacks noted in his talk were that Q-learning can be unstable even with linear-function approximators. Also, it can learn a policy that performs badly if the agent continues to explore because it is only optimal with no exploration. Finally, he noted that it does not work well with eligibility traces, such as used in TD(λ) (Sutton 1988). He observed that because it requires maximizing over actions to determine the utility of the next state, Q-learning could introduce systematic overestimation error (as noted by Thrun and Schwartz [1993]). He argued in favor of SARSA (Rummery and Niranjan 1994), a modified Q-learning algorithm that overcomes these problems.

Analysis of TD(λ) Satinder Singh (University of Colorado at Boulder) described his work with Peter Dayan (MIT) on how the bias and variance of the TD(λ) family of algorithms behaves with increasing experience. He described the effect of algorithm parameters such as λ and step size, and of problem parameters such as initial bias and cyclicity, on the behavior of learning curves.

Benjamin Van Roy (MIT) presented results on the TD algorithm as applied to approximating the cost-to-go function of a Markov chain using linear-function approximators. He described convergence results (with probability 1), a characterization of the limit of convergence, and a bound on the resulting approximation error. He also discussed the implications of two counterexamples with regard to the significance of online updating and linearly parameterized function approximators.

Undiscounted Reinforcement Learning Prasad Tadepalli (Oregon State University) argued that in contrast to the standard practice in reinforcement learning of maximizing the discounted total reward, in most real-world domains, the average reward received to a time step is a more natural metric. He introduced an average-reward reinforcement-learning method called H-learning (Tadepalli and Ok 1996a) and presented empirical results in several simple automated-guided vehicle scheduling domains. He also described a local linear-regression algorithm for approximating the value function. He discussed a Bayesian network approach to represent action models, where the topology of the network is part of the prior knowledge.

Sridhar Mahadevan (University of South Florida) presented a framework called *sensitive discount optimality*, the result of work by Blackwell (1962) and Veinott (1969), that offers an elegant way of linking the discounted and average-reward optimality criteria (Mahadevan 1996b). This framework is based on studying the properties of the expected cumulative discounted reward, as discounting tends to 1. He presented new model-free (Mahadevan 1996c) and model-based algorithms (Mahadevan 1996b), both derived from this framework, that not only optimize the expected average reward (*gain optimality*) but also maximize total reward among all gain-optimal policies (*bias optimality*).

Generalizing Markov Decision Processes Michael Littman (Brown University) described a generalized MDP model that unifies standard MDP models with alternating Markov games and information-state MDPs. The generalized MDP model applies to several different optimality criteria, including finite horizon, expected discounted sum, and risk-sensitive discounted reward. A key result here is that all the models subsumed by the generalized MDP model have an optimal-value function and policy and a general policy-iteration algorithm.

learning algorithms (both for prediction and control).

Hierarchical Models and Task Decomposition Singh (1994) summarized the work in reinforcement learning on hierarchical models and task decomposition. The basic idea is that faster learning can be achieved by decomposing the overall task into a collection of simpler subtasks. He discussed a mixture-model–based architecture for automatically decomposing sequential tasks.

Function Approximation Justin Boyan (CMU) described an algorithm for approximating the value function based on using efficient shortest-path algorithms from graph theory (Boyan and Moore 1996). He focused on the important subclass of acyclic tasks. His algorithm, called ROUT, can be used in large stochastic state spaces requiring function approximation. He showed significant improvements over TD(λ) in both efficiency and value-function approximation accuracy in several medium-sized domains.

Hidden State in Reinforcement Learning Although reinforcement learning has achieved notable successes in many application domains, it faces significant hurdles in domains where the state space is not completely observable. Ronald Parr (University of California at Berkeley) described some of the work he did in collaboration with Stuart Russell that is based on a particular type of Bayesian belief network called a *dynamic probabilistic network* (DPN). This network decomposes the representation of the state-transition model and the sensor model according to conditional independence relationships among the state and sensor variables. DPN models can be learned from observations, even in the partially observable case, using local gradient-descent techniques. Their aim is to demonstrate that the combination of these methods with techniques for approximate solution of partially observable MDPs should allow reinforcement learning to scale up to large, uncertain, partially observable decision problems (Russell and Parr 1995).

The most striking successes of reinforcement-learning techniques have been in their application to problems that are not traditionally viewed as learning problems.

Integrating Reinforcement Learning into AI

Several representational issues regarding integrating reinforcement learning into a general AI system were discussed at the workshop, including representing structured policies using Bayesian nets and planning using Bayesian nets.

Structured Policies Using Bayesian Nets Bayesian networks have been adopted widely in AI as a powerful tool for dealing with uncertainty. Craig Boutilier (1995) (University of British Columbia) illustrated how Bayesian networks could alleviate the problem of specifying and solving MDPs. He showed how networks reveal regularities and structure in the system dynamics and reward function that can be exploited computationally. He examined three different abstraction methods and described some ways of performing region-based dynamic programming in large, finite-state, and action problems.

Planning Using a Markov Decision Process Framework Steve Hanks (University of Washington) contrasted the view of work in decision-theoretic planning (Boutilier, Dean, and Hanks 1995; Draper, Hanks, and Weld 1994a, 1994b), where agents continually build suboptimal plans for achieving dynamic goals, with the work in reinforcement learning on computing optimal plans for a fixed goal. He argued that we should first try to understand the similarities and differences between the paradigmatic problems addressed by reinforcement-learning systems and those addressed by classical planners. This understanding would help integrate reinforcement learning into a broader problem-solving framework, where the value function might be partially provided as part of the task and, thus, change from task to task. Two interesting questions addressed in his talk were whether the reinforcement-learning approach could be applied equally well to higher-level planning and decision-making problems and, if not, whether there is a natural architectural interface between taskable problem-solving behavior and reactive or tactical behavior.

Representational Issues Thomas Dean (Brown University) pointed out that there is a tendency within the reinforcement-learning community for problems to be defined in terms of their solutions (algorithms) rather than the other way around. For example, he suggested that discounting has become a property of problems rather than a heuristic technique for generating policies with a particular sort of graded myopia. He suggested that Bayesian decision theory in general and graphic models in particular provide the languages and mathematics for framing decision problems involving uncertainty. He discussed in this talk how opportunities for exploiting structure are manifest in problem descriptions in which the state, value, and decision spaces are factored using variables, and the dependencies involving these variables were made explicit.

mator is well suited to the form of the value function, then powerful generalization can occur between similar states.

Reinforcement-learning methods have successfully been applied to elevator scheduling (Crites and Barto 1996), job-shop scheduling for NASA missions (Zhang and Dietterich 1995), backgammon (Tesauro 1992), and cellular telephone channel assignment (Bertsekas and Tsitsiklis 1996). There are a number of other ongoing applications, and case studies of systems of this kind will appear in Bertsekas and Tsitsiklis's new book (1996). There is considerable enthusiasm among members of the operations research community about these techniques, which enable the approximate solution of problems that were heretofore unaddressable.

One particularly interesting aspect of this development is that the reinforcement-learning techniques were developed as part of a basic research program whose focus was strategies that agents could use, online, to learn how to behave well in their particular environments. As it happens, these methods have extreme promise for solving large industrial problems, which were unanticipated during their development.

Impact of Reinforcement Learning on AI

Work on reinforcement learning is having a strong impact on other parts of AI, especially through the use of MDP models. Because most work in reinforcement learning addresses the problem of learning how to behave in sequential environments, there is a deep connection with work in AI planning. Because AI planning has begun to adopt models with a decision-theoretic orientation (Kushmerick, Hanks, and Weld 1995) and to be interested in partial policies rather than straight-line plans, there has been a convergence on MDPs as the basic model underlying all our work.

One of the biggest contributions that mainstream AI can make to work in reinforcement learning is in understanding how to use richer representations. Bayesian networks provide an ideal representation for stochastic state-transition and reward functions in complex domains; this representation has been used by Tadepalli (Tadepali and Ok 1996b), among others, for reinforcement-learning problems. In addition, there are now a number of good techniques for learning Bayesian networks from data, making this strategy plausible for acquiring world models. Although compact models are elegant, they will not really be useful until we can use them to solve MDP or reinforcement-learning problems more efficiently. Recent work by Boutilier (1995) and others seeks to exploit structure in the representation of an MDP to solve it more efficiently.

Although we know a good deal about the applications of reinforcement learning to control problems, it has recently been used successfully in a perception application. Bandera et al. (1996) used Q-learning to acquire an action-perception strategy to decide what parts of an object to foveate in an attempt to recognize the object's type. This result is encouraging, demonstrating that reinforcement-learning techniques have broad application potential.

Research Problems in Reinforcement Learning

Although there has been a great deal of progress both in the foundations and the application of reinforcement learning, even more open problems remain. We certainly cannot enumerate all of them (not least because there are surely many problems that have not even been discovered), but the list below highlights topics whose importance was noted at the workshop.

To solve problems with continuous or large finite-state spaces, it is crucial to approximate the value function during reinforcement learning or dynamic programming. A number of techniques often work well in practice; however, the theoretical properties of these methods are not yet well understood. In addition, there are some situations for which the existing methods are not suited; new algorithms must be developed.

Researchers have been surprised by the failures and successes of reinforcement learning. Problems that seem easy often turn out to be hard to solve, and vice versa. Although the size of the state space is relevant, it is by no means the determining factor. Other problem attributes, such as the shape of the value function and the degree to which reward is delayed, are thought to be important, but the relationship between these attributes and the overall difficulty for the current algorithms is not well understood.

The vast majority of reinforcement-learning work concentrates on the completely observable case, in which there is noise in the agent's actions but none in its observation of its environment. For many problems, this model is inadequate. Richer models, such as partially observable MDPs, capture a wider variety of problems but present an enormously more difficult optimization task. It has been shown that the exact solution of these problems is intractable, but there is a great need for work on approximate solution methods for these problems.

There has been a great deal of debate about the appropriate optimality measures to use in reinforcement learning. The discounted measure is often adopted for ease of computation rather than appropriateness to the problem. We are now in the position of having learning algorithms for average-reward measures, which leads us to the question of which measures are really appropriate for which kinds of problem. In addition, all this work uses *time-separable measures*, in which the objective is an additive function of the individual rewards on each time step. There might be some domains in which measures that depend on the whole trajectory are much more appropriate; further work is required in understanding how to optimize for such measures.

Reinforcement learning was originally developed as a model of behavior learning in animals and as a way of engineering behavior learning in artificial systems. We now have a fairly good understanding of how a single reinforcement-learning system works, but we have not really thought about the context in which it takes place. In a complex agent, we will have to deal with questions of com-

Montague, P.; Dayan P.; and Sejnowski, T. 1996. A Framework for Mesencephalic Dopamine Systems Based on Predictive Hebbian Learning. *Journal of Neuroscience* 16:1936–1947.

Montague, P.; Dayan P.; Person, C.; and Sejnowski, T. 1995. Bee Foraging in Uncertain Environments Using Predictive Hebbian Learning. *Nature* 377:725–728.

Puterman, M. 1994. *Markov Decision Processes: Discrete Dynamic Stochastic Programming.* New York: Wiley.

Rummery, G., and Niranjan, M. 1994. Online Q-Learning Using Connectionist Systems, Technical Report, CUED/F-INFENG/TR166, Cambridge University.

Rusell, S., and Norvig, P. 1994. *Artificial Intelligence: A Modern Approach*. New York: Prentice-Hall.

Russell, S., and Parr, R. 1995. Approximating Optimal Policies for Partially Observable Stochastic Domains. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Russell, S.; Binder, J.; Daphne, K.; and Kanazawa, K. 1995. Local Learning in Probabilistic Networks with Hidden Variables. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1146–1152. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Singh, S. 1994. Learning to Solve Markovian Decision Processes. Ph.D. thesis, Department of Computer Science, University of Massachusetts at Amherst.

Sutton, R. 1988. Learning to Predict by the Method of Temporal Differences. *Machine Learning* 3:9–44.

Sutton, R., ed. 1992. *Machine Learning* (Special Issue on Reinforcement Learning) 8(3–4).

Sutton, R., and Barto, A. 1997. Learning Values: An Introduction to Reinforcement Learning. Forthcoming.

Tadepalli, P., and Ok, D. 1996a. Auto-Exploratory Average-Reward Reinforcement Learning. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 881–887. Menlo Park, Calif.: American Association for Artificial Intelligence.

Tadepalli, P., and Ok, D. 1996b. Scaling Up Average Reward Reinforcement Learning by Approximating the Domain Models and the Value Function. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 471–479. San Francisco, Calif.: Morgan Kaufmann.

Takahashi, Y.; Asada, M.; and Hosoda, K. 1996. Reasonable Performance in Less Learning Time by Real Robots Based on Incremental State-Space Segmentation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS96). Washington, D.C.: IEEE Computer Society.

Tesauro, G. 1992. Practical Issues in Temporal Difference Learning. In *Reinforcement Learning*, ed. R. Sutton. Norwell, Mass.: Kluwer Academic.

Thrun, S., and Schwartz, A. 1995. Finding Structure in Reinforcement Learning. In *Neural Information Processing Systems (NIPS) 7: Proceedings of the 1994 Conference.* Cambridge, Mass.: MIT Press.

Thrun, S., and Schwartz, A. 1993. Issues in Using Function Approximation for Reinforcement Learning. In *Proceedings of the Fourth Connectionist Models Summer School.* Hillsdale, N.J.: Lawrence Erlbaum.

Uchibe, E.; Asada, M.; and Hosoda, K. 1996. Behavior Coordination for a Mobile Robot Using Modular Reinforcement Learning. In Proceedings of the 1996 International Conference on Intelligent Robots and Systems (IROS). Washington, D.C.: IEEE Computer Society.

Veinott, A. 1969. Discrete Dynamic Programming with Sensitive Discount Optimality Criteria. *Annals of Mathematical Statistics* 40(5): 1635–1660.

Watkins, C. 1989. Learning from Delayed Rewards. Ph.D. thesis, King's College.

Zhang, W., and Dietterich, T. 1995. A Reinforcement Learning Approach to Job-Shop Scheduling. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1114–1120. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Sridhar Mahadevan received his Ph.D. in computer science from Rutgers University in May 1990. He was employed as a research scientist at the IBM T. J. Watson Labs in Yorktown Heights, New York, until July 1993. He is currently an assistant professor in the Department of Computer Science and Engineering at the University of South Florida. His research interests are in the areas of reinforcement learning, machine learning, AI, expert systems, and robotics. He has published numerous articles in these areas. He has served on the program committees of several premier conferences, including the American Association for Artificial Intelligence Conference, the International Joint Conference on Artificial Intelligence, Intelligent Robots and Systems and International Machine Learning Conference. His current research in reinforcement learning is supported in part by a National Science Foundation (NSF) CAREER Award. He can be contacted at mahadeva @csee.usf.edu.

Leslie Pack Kaelbling is associate professor of computer science at Brown University. She previously held positions at the Artificial Intelligence Center of SRI International and Teleos Research. She received an A.B. in philosophy in 1983 and a Ph.D. in computer science in 1990, both from Stanford University. Kaelbling has done substantial research on programming paradigms and languages for embedded systems, mobile robot design and implementation, and reinforcement-learning algorithms. Her current research directions include the integration of learning modules into systems programmed by humans, algorithms for learning and navigating using hierarchical domain representations, and methods for learning perceptual strategies. In 1994, she was selected as a National Science Foundation presidential faculty fellow, and in 1996, she was elected to the executive council of the American Association for Artificial Intelligence.