Optimality Criteria in Reinforcement Learning

Sridhar Mahadevan*
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
mahadeva@csee.usf.edu

Abstract

Embedded autonomous agents, such as robots or softbots, are faced with solving *sequential decision problems*. Reinforcement learning (RL) is a particular approach to sequential decision problems, in which the agent-environment interaction is modeled as a Markov decision process (MDP). RL agents solve sequential decision problems by learning *optimal* policies for choosing actions. Thus, at the core of RL is the definition of what it means for a policy to be "optimal". In this paper, we discuss a variety of optimality metrics from the dynamic programming literature, and examine their suitability for RL. We also discuss the challenges in devising RL algorithms for the various metrics.

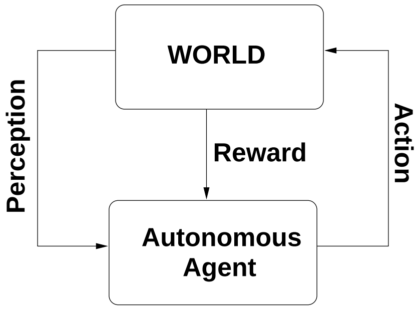

Figure 1: A sequential decision problem faced by an autonomous agent: how to choose actions that maximize rewards over the long term?

Sequential Decision Tasks

In recent years, a unifying viewpoint based on embedded autonomous agents has shaped much work in artificial intelligence (Kaelbling 1993; Rusell & Norvig 1994). Examples of such agents range from robots operating in unstructured environments, softbots navigating the Internet, or even industrial controllers operating some complex machinery. Such agents are faced with a difficult *sequential decision problem*, as illustrated in Figure 1. At every time instant, the agent observes the world to be in some state (perhaps, imperfectly), and must choose an action that (stochastically) alters the state of the environment. The goal of the agent is to behave *optimally*, that is, it needs to choose actions using a policy (either learned or programmed by its designer). The environment typically supplies the agent with some feedback regarding its performance, such as a reward or cost function. In this paper, we will be primarily interested in *infinite-horizon* decision tasks, where the agent continues to operate forever. The special case of tasks that have terminating goal states, such as a game or a robot finding a battery charger, can be easily modeled by assuming such states to be *absorbing*: the agent stays in that state forever.

Reinforcement learning (RL) is one general approach for formulating sequential decision problems, where the interaction between the agent and the environment is modeled as a Markov decision process (MDP) (Puterman 1994). RL has become one of the most actively studied learning frameworks for autonomous agents (Mahadevan & Kaelbling 1996; Kaelbing 1996; Sutton 1992). RL has proven to be particularly effective at solving large scale MDP's that were previously completely outside the scope of dynamic programming algorithms. Examples of practical successes of RL include a grandmaster backgammon player (Tesauro 1992), job-shop scheduling (Zhang & Dietterich 1995), elevator scheduling (Crites & Barto 1996), and cellular channel assignment (Bertsekas & Tsitsiklis 1996).

An RL agent solves sequential decision problems by learning an *optimal* policy mapping states to actions. Thus, the definition of optimality is at the very heart of RL. In this paper, we discuss a variety of different optimality criteria that have been previously studied in DP. We explore the relationships between these cri-

*Paper to be presented at the AAAI Fall Symposium on Learning Complex Behaviors in Adaptive Intelligent Systems, Nov. 9 - 11th, MIT, Boston, 1996.

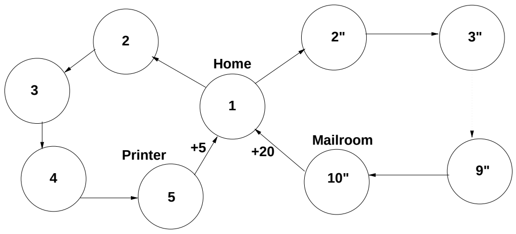

Figure 2: An example MDP with 2 possible policies, where the optimal policy depends on the metric used. The only action choice is in "home" state, where the robot can decide either to visit the printer or pick up mail. Picking up mail is a Blackwell-optimal policy (and consequently also bias and gain-optimal). Visiting the printer is discount-optimal if the discount factor $\gamma < 0.75$.

teria, as well as discuss the challenges in designing RL algorithms for them.

Markov Decision Processes

We begin by briefly reviewing the Markov Decision Process (MDP) framework underlying RL (Bertsekas 1995; Puterman 1994). An MDP consists of a (finite or infinite) set of states S, and a (finite or infinite) set of actions A for moving between states. In this paper we will assume that S and A are finite. We will denote the set of possible actions in a state x by $A(x)$. Associated with each action a is a state transition matrix $P(a)$, where $P_{xy}(a)$ represents the probability of moving from state x to y under action a. There is also a reward or payoff function $r : S \times A \rightarrow \mathcal{R}$, where $r(x, a)$ is the *expected* reward for doing action a in state x. Figure 2 illustrates a simple MDP.

A *stationary deterministic policy* is a mapping $\pi : S \rightarrow A$ from states to actions. In this paper we restrict our discussion to algorithms for such policies, since we are guaranteed that such policies exist for the optimality criteria being discussed. However, it is important to observe that once we relax some of the above assumptions, and consider problems with infinite states, or partially observable MDP's (Russell & Parr 1995; Littman, Cassandra, & Kaelbling 1995; Singh, Jaakkola, & Jordan 1994), the optimal policy may no longer be stationary. Any policy induces a state transition matrix P^π, where $P^\pi_{xy} = P_{xy}(\pi(x))$. Thus, any policy yields a Markov chain (S, P^π).

Any policy π generates a stream of rewards $R^\pi(t)$, which form a sequence of random real-valued variables. To differentiate various policies requires us to define some type of *optimality metric* on the reward stream. A simple example of such a metric is to compute the expected total sum of rewards (this and several other alternatives are defined more precisely below). The result of applying the optimality metric to a particular policy usually depends on the initial state from which the agent follows the policy. Thus, an optimality metric applied to a policy π results in a *value function* $V^\pi : S \rightarrow \mathcal{R}$ mapping states to real numbers.

Policies can be compared using the value functions resulting from following each policy. In particular, policy π_1 is preferable to π_2 if, over all states s, the value function for the first policy is larger than the value function for the second policy. Consequently, the optimal policy π^* is one that results in the optimal value function V^*, which is as good or better than all other (policy-induced) value functions over the whole state space.

A Variety of Optimality Metrics

We now describe a wide range of alternative optimality metrics, and briefly discuss their strengths and limitations. A more detailed discussion of these metrics can be found in the DP literature (e.g. see (Puterman 1994)). We address the computational issues for these metrics in the next section.

Total Reward Optimality The simplest (and most obvious) optimality metric is to maximize total reward. More precisely, the *expected total reward* of a policy π is

$$V^\pi(s) = \lim_{N \to \infty} E\left(\sum_{t=1}^{N} R^\pi_t(s)\right) \qquad (1)$$

where $R^\pi_t(s)$ is a random variable representing the reward received at time t, if the starting state is s, and actions are chosen using policy π. $E(.)$ denotes expected value. An optimal total reward policy π^* is one that maximizes this value function, over all states and actions.

The main drawback of the total reward optimality metric is that unless strong conditions are imposed, the total reward can easily be unbounded (e.g. as in Figure 2). Consequently, in the DP literature, two special cases of this metric have been studied: *positive* and *negative* DP models. Both models assume that the limit in Equation 1 exists; one condition that guarantees this limit is in MDP's where there are absorbing states with zero reward self-transitions, and all policies end up being absorbed in one of these states. Boyan (Boyan 1996) cites many practical applications from combinatorial optimization which satisfy this condition.

In negative DP, rewards are essentially viewed as "costs", and are always non-positive, i.e. $r(s, a) \leq 0$ for all actions a and states s. The interpretation

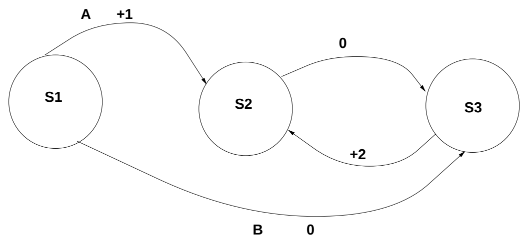

Figure 3: A simple MDP for which no policy is overtaking optimal. The difference in total reward earned by the two policies oscillates around 0.

here is that agent desires to minimize expected costs, since maximization of negative rewards is equivalent to minimization of expected costs. The aim is to get to a zero-cost absorbing state as quickly as possible. Conversely, in positive DP, rewards are always non-negative, i.e. $r(s, a) \geq 0$. The aim here is to delay termination as much as possible.

A classic application of positive and negative models is in the class of problems called *optimal stopping*: at every step, the agent can either choose to continue the process at a certain cost (or reward), or stop the process and earn a reward. Examples of optimal stopping problems include asset selling, sequential hypothesis testing, and gambling.

Overtaking Optimality Instead of using total reward, a more refined measure can be defined based on the asymptotic growth of the value function. Such a metric would not require the total reward to be bounded, and could still be useful in distinguishing among different policies. This metric is called *overtaking optimality* in the DP literature. More precisely, a policy π^* is termed *overtaking optimal* if

$$\lim_{N \to \infty} \left(V_N^{\pi^*}(s) - V_N^{\pi}(s) \right) \geq 0 \qquad (2)$$

over all policies π, where $V_N^{\pi^*}(s)$ represents the expected cumulative reward earned following π^* for N steps, starting from state s.

A key problem with this metric is that it is too stringent. Figure 3 illustrates a simple MDP for which no policy is overtaking optimal. The problem here is that the total reward of the neither policy starting from state $S1$ dominates the other, since action A in state $S1$ would generate the cumulative reward stream $1, 1, 3, 3, 5, 5, \ldots$, whereas the reward stream $0, 2, 2, 4, 4, \ldots$ would be generated by selecting action B in $S1$.

Average-Overtaking Optimality The problem with overtaking optimality in Figure 3 arises because the cumulative reward of the two policies oscillate around 0. This can be avoided by defining a slightly less stringent metric, called *average-overtaking* optimality, which damps out such periodic oscillations by taking the Cesaro average:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} \left(V_N^{\pi^*}(s) - V_N^{\pi}(s) \right) \geq 0 \qquad (3)$$

Interestingly, average-overtaking optimality is completely equivalent to *bias-optimality*, defined below, and so we postpone further discussion of this metric until then.

Discounted Optimality: Most of the previous work in RL has used this metric, where future rewards are geometrically attenuated by some discount factor $\gamma < 1$. In the discounted framework, a policy π is measured by a value function representing the cumulative discounted sum of rewards earned following that policy from any state s:

$$V_\gamma^\pi(s) = \lim_{N \to \infty} E \left(\sum_{t=0}^{N-1} \gamma^t R_t^\pi(s) \right),$$

where $\gamma < 1$ is the discount factor, and $R_t^\pi(s)$ is the reward received at time t starting from state s under policy π. An *optimal discounted policy* π^* maximizes the above value function over all states x and policies π, i.e. $V_\gamma^{\pi^*}(x) \geq V_\gamma^\pi(x)$.

The discounted framework is attractive because it is mathematically convenient to analyze. Discounting gives rise to a nice property called *contraction* (Bertsekas 1995), whereby every step of a DP algorithm (such as value iteration) reduces the distance between the current value function and the optimal value function. RL algorithms have been developed for the discounted framework, such as TD(λ) (Sutton 1988) and Q-learning (Watkins 1989), that have been demonstrated to work on large scale DP problems.

However, there are a number of problems with discounting. The convergence of the algorithms typically slows down for values of γ close to 1 (Mahadevan 1996b; Schwartz 1993). Also, the optimal policy often varies as the discount factor is changed. For example, in Figure 2, for $\gamma < 0.75$ the optimal policy is to service the printer, whereas for higher γ, the optimal policy is to pick up mail. For some more realistic examples from combinatorial optimization where choice of discount factor is crucial, see (Zhang & Dietterich 1995; Boyan 1996). Finally, for many tasks, such as a robot avoiding obstacles (Mahadevan & Connell 1992), an AGV transporting parts (Tadepalli & Ok 1994), or for process-oriented planning (Boutilier & Puterman 1995), the average reward seems a more natural optimality metric, and using discounted algorithms for the average reward metric has been shown to be unsafe (Mahadevan 1996b; Tadepalli & Ok 1996).

Gain Optimality: A very natural measure of optimality is for an agent to maximize the expected payoff per step. This metric, called *gain-optimality*, was first introduced in Howard's classic Ph.D. thesis (Howard 1960). The average reward $\rho^\pi(x)$ associated with a particular policy π at a state x is defined as

$$\rho^\pi(x) = \lim_{N\to\infty} \frac{E\left(\sum_{t=0}^{N-1} R_t^\pi(x)\right)}{N}, \quad \forall x \in S \qquad (4)$$

where $R_t^\pi(x)$ is the reward received at time t starting from state x, and actions are chosen using policy π. A gain-optimal policy π^* is one that maximizes the average reward over all states, that is, $\rho^{\pi^*}(x) \geq \rho^\pi(x)$ over all policies π and states x.

A detailed discussion of the average reward framework is given in (Mahadevan 1996b), which describes algorithms from DP, RL, adaptive control, and learning automata.

Bias Optimality: A key limitation of gain optimality is that it cannot distinguish between policies which have the same average reward, but which incur very different "startup" costs over the initial transient states. For example, in inventory control, an optimal policy is an (s, S) policy, whereby if the stock on hand falls below s units, the action taken is to restock the inventory to S units. Here, all states above S units are transient, and consequently, all policies that get rid of excess stock above S units (including selling or destroying) will be gain-optimal! More generally, transient states occur in goal-based tasks, such as finding a battery charger in a room, as well as in tasks with no absorbing goals, such as inventory control. A more selective criterion called *bias optimality* is needed to address the unselectivity of gain optimality. To define bias optimality, we need to evaluate policies using a different value function, namely the *average adjusted* sum of rewards earned following a policy,

$$V^\pi(s) = \lim_{N\to\infty} E\left(\sum_{t=0}^{N-1} (R_t^\pi(s) - \rho^\pi)\right), \quad \forall s \in S,$$

where ρ^π is the average reward associated with policy π. [1] The term $V^\pi(s)$ is usually referred to as the *bias* value.

A policy π^* is termed *bias-optimal* if it is gain-optimal, and it also maximizes bias values, that is $V^{\pi^*}(x) \geq V^\pi(x)$ over all $x \in S$ and policies π. The notion of bias-optimality was introduced by Blackwell (Blackwell 1962), who also pioneered the study of average reward MDP as the limiting case of the discounted framework. It can be shown that a stationary bias-optimal policy is guaranteed to exist for any finite MDP (Puterman 1994).

Blackwell Optimality Gain and bias optimality are instances of a more general optimality criterion called sensitive discount optimality. Sensitive discount optimality (Blackwell 1962; Veinott 1969) is based on studying the cumulative discounted sum of rewards, as the discount factor γ approaches 1. Since the cumulative discounted sum of rewards may be unbounded for $\gamma = 1$, the key idea here is to look for policies that remain optimal as $\gamma \to 1$. Blackwell (Blackwell 1962) showed that when $\gamma \to 1$, the cumulative discounted sum of rewards received by following any policy π, i.e. V_γ^π, can be expanded using a Laurent series expansion to obtain a sequence of terms, the first of which represents the *average reward* ρ^π of the policy π, the second of which represents the *average adjusted* sum of rewards V^π, and so on.

Blackwell also introduced a more refined notion of optimality called *Blackwell optimality*. A policy π^* is Blackwell optimal if there exists a $0 \leq \gamma^* < 1$ such that

$$V_\gamma^{\pi^*}(s) \geq V_\gamma^\pi(s) \quad \forall \gamma \geq \gamma^*, \gamma < 1, \;\; \forall \pi, \;\; \forall x \in S$$

Intuitively, a Blackwell optimal policy π^* is one that remains optimal for all values of the discount factor $\geq \gamma^*$, but less than 1. For example, in the MDP illustrated in Figure 2, picking up mail is Blackwell-optimal since it is discount-optimal for all $\gamma > 0.75$. Blackwell proved that there always exists a stationary deterministic policy that satisfies this metric for finite state and action spaces.

Sensitive Discount Optimality Veinott (Veinott 1969) showed that Blackwell optimality is actually the limiting case of a whole series of progressively more refined optimality criteria. He defined a sensitive-discount optimality criterion called *n-discount-optimality*:

Definition 1 *A policy* π^* *is* n-discount-optimal*, for* $n = -1, 0, 1, ...$, *if for each state* s*, and over all policies*

[1] This limit assumes that all policies are aperiodic. For periodic policies, we need to use the Cesaro limit $V^\pi(s) = \lim_{N\to\infty} \frac{\sum_{k=0}^{N-1} E\left(\sum_{t=0}^{k}(R_t^\pi(s)-\rho^\pi)\right)}{N}$.

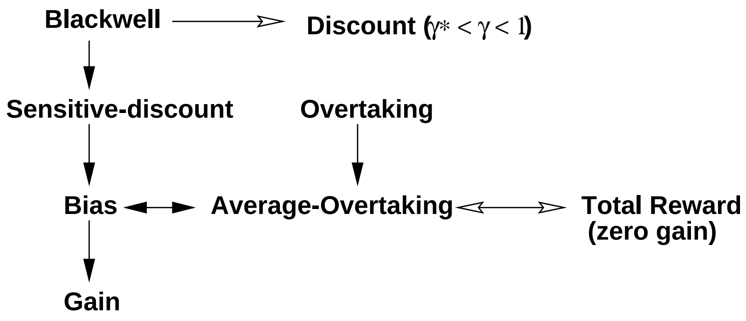

Figure 4: This figure illustrates the relationships between the various optimality metrics. Solid arrows represent unconditional implication. Outline arrows represent conditional implication.

π,

$$\lim_{\gamma \to 1} (1-\gamma)^{-n} \left(V_\gamma^{\pi^*}(s) - V_\gamma^{\pi}(s) \right) \geq 0.$$

We now consider some special cases of this general optimality metric to illustrate how it nicely generalizes the optimality metrics defined above.

Theorem 1 *(Veinott) π^* is a* -1-discount-optimal *policy if and only if π^* is a gain-optimal policy.*

Theorem 2 *(Veinott) π^* is a* 0-discount-optimal *policy if and only if π^* is a bias-optimal policy.*

Theorem 3 *(Veinott) π^* is a* n-discount-optimal *policy $\forall n \geq -1$ if and only if π^* is a Blackwell-optimal policy.*

These theorems provide the link between n-discount-optimality and the other optimality criteria defined above. An intuitive interpretation of n-discount-optimality when rewards correspond to monetary earnings is as follows. -1-discount-optimality or gain-optimality corresponds to maximizing the *rate of earnings*. On the other hand, 0-discount-optimality corresponds to maximizing *transient earnings*, before reaching recurrent states. Finally, 1-discount-optimality corresponds to maximizing *temporary earnings*, and so on.

Relationships between Various Metrics

Figure 4 illustrates the relationships between the various optimality metrics described above. The most refined optimality metric, which exists for any finite MDP, is Blackwell optimality. If a policy is Blackwell optimal, then it is n-discount-optimal, $\forall n \geq -1$, and hence the latter metric is less refined than the former. Bias and gain optimality are special cases of sensitive-discount optimality. Note that average-overtaking optimality is completely equivalent to bias optimality. Interestingly, it can be shown that if a policy is overtaking optimal, it is also average-overtaking optimal. The relationship of discount optimality (for fixed γ) with the other metrics is subtle: a Blackwell optimal policy will be discount optimal, but only for all values of $\gamma > \gamma^*$. Finally, if all policies result in zero average reward (gain), then total reward optimality is equivalent to bias optimality and average-overtaking optimality.

Computational Issues

We now turn to discuss the computational problem of designing an RL algorithm for the various metrics discussed so far. The discounted framework has clearly received the maximum attention in the RL literature. There is a well developed theory of model-free algorithms, like Q-learning (Tsitsiklis 1994), as well as several practical successes (mentioned earlier). These algorithms can be used with $\gamma = 1$, for MDP's with special properties (such as a zero cost absorbing goal state), and thus they can be used to compute total reward optimal policies. However, as noted in Boyan (Boyan 1996), in many combinatorial optimization problems, using one of these algorithms is likely to produce less than desirable results. More specialized algorithms may produce better performance.

Several RL algorithms have been developed for gain optimality, including R-learning (Schwartz 1993), H-learning (Tadepalli & Ok 1996), and a variant of Q-learning (Bertsekas 1995). The theory of average-case RL is still very under-developed, although promising new results have begun to appear (Bertsekas 1995). Also, no major empirical successes have yet been reported, but such results will undoubtedly appear as these algorithms are deployed on a wider scale.

However, these average reward algorithms cannot be used to select among multiple gain-optimal policies. This situation occurs in any MDP with transient states, including several important practical domains such as inventory control, queueing, maintenance, and scheduling. Recently, two new RL algorithms for computing bias-optimal algorithms have been developed (BOH, BOR) (Mahadevan 1996a; 1996c). The computational problem of determining bias-optimal policies appears more intricate, because a pair of nested optimality equations have to be solved. A general strategy for solving such nested equations is to use interleaved successive approximation, whereby both equations are solved together. Much remains to be done here in terms of empirically studying the performance of these algorithms on large scale MDP's.

Finally, no RL algorithms have yet been developed for computing Blackwell optimal policies, or even sensitive-discount optimal policies in general. There are provably convergent DP algorithms for computing both these metrics, but they are computationally intricate. Designing an efficient RL algorithm for these metrics appears to pose a significant challenge. Table 1 summarizes these observations by mentioning RL algorithms for some of the optimality metrics discussed in this paper.

Metric	RL Algorithms
Discounted	Q, TD
Total Reward	Q,TD
Gain	H, R
Bias (Avg. Overtaking)	BOH, BOR
Blackwell	?
Sensitive-discount	?

Table 1: This table lists some of the optimality metrics discussed in this paper, and identifies those for which RL algorithms have been developed.

Acknowledgements

I thank Justin Boyan for his comments on a draft of this paper. This research is supported in part by an NSF CAREER Award Grant No. IRI-9501852.

References

Bertsekas, D., and Tsitsiklis, J. 1996. *Neuro-Dynamic Programming.* Athena Scientific.

Bertsekas, D. 1995. *Dynamic Programming and Optimal Control.* Athena Scientific.

Blackwell, D. 1962. Discrete dynamic programming. *Annals of Mathematical Statistics* 33:719–726.

Boutilier, C., and Puterman, M. 1995. Process-oriented planning and average-reward optimality. In *Proceedings of the Fourteenth JCAI*, 1096–1103. Morgan Kaufmann.

Boyan, J. 1996. Learning evaluation functions. CMU CS Thesis Proposal.

Crites, B., and Barto, A. 1996. Improving elevator performance using reinforcement learning. In *Neural Information Processing Systems (NIPS) 8.*

Howard, R. 1960. *Dynamic Programming and Markov Processes.* MIT Press.

Kaelbing, L., ed. 1996. *Reinforcement Learning.* Kluwer Academic Press. Special Issue of Machine Learning Journal, Vol 22.

Kaelbling, L. 1993. *Learning in Embedded Systems.* MIT Press.

Littman, M.; Cassandra, T.; and Kaelbling, L. 1995. Learning policies for partially-observable environments: scaling up. In *Proceedings of the 12th International Conference on Machine Learning*, 362–370. Morgan Kaufmann.

Mahadevan, S., and Connell, J. 1992. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence* 55:311–365. Appeared originally as IBM TR RC16359, Dec 1990.

Mahadevan, S., and Kaelbling, L. 1996. The NSF workshop on reinforcement learning: Summary and observations. *AI Magazine.* To Appear in Winter Issue.

Mahadevan, S. 1996a. An average-reward reinforcement learning algorithm for computing bias-optimal policies. In *Proceedings of the Thirteenth AAAI.* Morgan Kaufmann. To Appear.

Mahadevan, S. 1996b. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning* 22:159–196.

Mahadevan, S. 1996c. Sensitive-discount optimality: Unifying average-reward and discounted reinforcement learning. In *Proceedings of the 13th International Conference on Machine Learning.* Morgan Kaufmann. To Appear.

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

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

Russell, S., and Parr, R. 1995. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the Fourteenth IJCAI.*

Schwartz, A. 1993. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the Tenth International Conference on Machine Learning*, 298–305. Morgan Kaufmann.

Singh, S.; Jaakkola, T.; and Jordan, M. 1994. Learning without state estimation in partially observable markov decision processes. In *Proceedings of the Eleventh International Conference on Machine Learning*, 284–292. Morgan Kaufmann.

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

Sutton, R., ed. 1992. *Reinforcement Learning.* Kluwer Academic Press. Special Issue of Machine Learning Journal Vol 8, Nos 3-4, May 1992.

Tadepalli, P., and Ok, D. 1994. H learning: A reinforcement learning method to optimize undiscounted average reward. Technical Report 94-30-01, Oregon State Univ.

Tadepalli, P., and Ok, D. 1996. Auto-exploratory average-reward reinforcement learning. In *Proceedings of the Thirteenth AAAI*, 881–887. AAAI/MIT Press.

Tesauro, G. 1992. Practical issues in temporal difference learning. In Sutton, R., ed., *Reinforcement Learning*. Kluwer Academic Publishers.

Tsitsiklis, J. 1994. Asynchronous stochastic approximation and Q-learning. *Machine Learning* 16:185–202.

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. Dissertation, King's College, Cambridge, England.

Zhang, W., and Dietterich, T. 1995. A reinforcement learning approach to job-shop scheduling. In *Proceedings of the Fourteenth IJCAI*.