Stochastic Optimization of Controlled Partially Observable Markov Decision Processes

Peter L. Bartlett
Research School of Info. Sciences and Eng.
Australian National University
ACT 0200, Australia
Peter.Bartlett@anu.edu.au

Jonathan Baxter
Whizbang! Labs, East
4616 Henry Street Pittsburgh
PA 15213, USA
Jonathan.Baxter@anu.edu.au

Abstract

We introduce an on-line algorithm for finding local maxima of the average reward in a Partially Observable Markov Decision Process (POMDP) controlled by a parameterized policy. Optimization is over the parameters of the policy.

The algorithm's chief advantages are that it requires only a single sample path of the POMDP, it uses only one free parameter $\beta \in [0, 1)$, which has a natural interpretation in terms of a bias-variance trade-off, and it requires no knowledge of the underlying state. In addition, the algorithm can be applied to infinite state, control and observation spaces.

We prove almost-sure convergence of our algorithm, and show how the correct setting of β is related to the mixing time of the Markov chain induced by the POMDP.

1 Introduction

Dynamic Programming is the method of choice for solving problems of decision making under uncertainty [5]. However, the application of Dynamic Programming becomes problematic in large or infinite state-spaces; in situations where the system dynamics are unknown; or when the state is only partially observed. In such cases one looks for approximate techniques that rely on simulation, rather than an explicit model, and parametric representations of either the value-function or the policy, rather than exact representations.

Simulation-based methods that rely on a parametric form of the value function tend to go by the name "Reinforcement Learning," and have been extensively studied in the Machine Learning literature [6, 22]. This approach has yielded some remarkable empirical successes in a number of different domains, including learning to play checkers [19], backgammon [23, 24], and chess [4], job-shop scheduling [27] and dynamic channel allocation [20].

Despite this success, most algorithms for training approximate value functions suffer from the same flaw: the performance of the greedy policy derived from the approximate value-function is not guaranteed to improve on each iteration, and in fact can be worse than the old policy by an amount equal to the *maximum* approximation error over all states. This can happen even when the parametric class contains a value function whose corresponding greedy policy is optimal. We illustrate this with an example in Section 2.

An alternative approach that circumvents this problem—the approach we pursue here—is to consider a class of *policies* parameterized by $\theta \in \mathbb{R}^K$, compute the gradient of the average reward with respect to θ, and then improve the policy by adjusting the parameters in the gradient direction. Unfortunately, for large-scale problems or problems where the system dynamics is unknown, the gradient will not be computable in closed form. Thus the challenging aspect of this approach is to find an algorithm for estimating the gradient via *simulation*.

This approach has a long history. Score function methods for estimating performance gradients in i.i.d. processes were first proposed in the sixties [1, 18] and were extended to regenerative processes (including MDPs) in [10, 11, 12, 16, 17], and independently for *episodic* POMDPs in [26], which introduced the REINFORCE algorithm. These algorithms are applicable when there is an identified recurrent state i^*, and the algorithm returns a gradient estimate each time i^* is entered. Formulae for the performance gradient of an MDP that rely on the existence of a recurrent state have also been given in [7, 8, 9], and for POMDPs in [21]. Williams' algorithm was generalized to the infinite-horizon setting in [13] and to more general reward structures in [15].

Policy-gradient algorithms for which convergence results have been proved all rely on the existence of an identifiable recurrent state. Although the assumptions we make in this paper about the POMDP ensure that every state is recurrent, we would expect that as the size of the state space increases, there will be a corre-

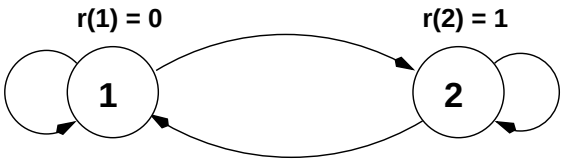

Figure 1: Two-state Markov Decsision Process

sponding increase in the expected time between visits to the identified recurrent state. Furthermore, the time between visits depends on the parameters, and states that are frequently visited for the initial value of the parameters may become very rare as performance improves. In addition, in an arbitrary POMDP it may be difficult to estimate the underlying states, and therefore to determine when the gradient estimate should be updated.

In view of these considerations, the main contribution of this paper is an on-line algorithm for optimizing the performance of parameterized policies for general POMDPs that does not rely on the existence of a single recurrent state. Instead, the algorithm follows an *approximate* gradient direction where the accuracy of approximation is controlled by a parameter $\beta \in [0,1)$. We show that the approximation is close to the true gradient provided $1/(1-\beta)$ is close to the *mixing time* of the Markov chain induced by the POMDP. However, β cannot be set arbitrarily close to 1 because the variance of the algorithm also depends on $1/(1-\beta)$. We prove almost-sure convergence of our algorithm.

2 An approximate value-function example

We illustrate the problems associated with approximate value-function methods in a two-state Markov decision process (Figure 1). Suppose there are two controls u_1, u_2 with corresponding transition probability matrices $P(u_1), P(u_2)$. We assume the actions have the same effect in each state, namely, if u_1 is chosen then the system makes a transition to state 1 with probability 1/3 and to state 2 with probability 2/3, regardless of the current state, while if u_2 is chosen then the transition probabilities are reversed. Since state 2 has a reward of 1, while state 1 has a reward of 0, the optimal policy is to always select action u_1. Under this policy the stationary distribution is $[\pi_1, \pi_2] = [1/3, 2/3]$, while the infinite-horizon discounted value of state i is

$$J_\alpha(i) = \mathbf{E}\left(\sum_{t=0}^{\infty} \alpha^t r(X_t) \middle| X_0 = i \right),$$

where $\alpha \in [0,1)$ is the discount factor, so that $J_\alpha(1) = \frac{2\alpha}{3(1-\alpha)}$, $J_\alpha(2) = 1 + \frac{2\alpha}{3(1-\alpha)}$.

Now, suppose we are trying to learn an approximate value function $\tilde{J}$ for this MDP, *i.e.* , $\tilde{J}(i) = w\phi(i)$ for each state $i = 1, 2$ and some scalar feature ϕ (ϕ must have dimensionality 1 to ensure that $\tilde{J}$ really is *approximate*). For $\tilde{J}$ to generate an optimal greedy policy, it must value state 2 above state 1. For the purposes of this illustration we choose $\phi(1) = 2, \phi(2) = 1$, so that for $\tilde{J}(2) > \tilde{J}(1)$, we require $w < 0$.

Temporal Difference learning (or TD(λ)) is one of the most popular techniques for training approximate value functions [22]. It has been shown that for linear functions, TD(1) converges to a weight w^* minimizing the expected squared loss under the stationary distribution [25]:

$$w^* = \text{argmin}_w \sum_{i=1}^{2} \pi_i \left[w\phi(i) - J_\alpha(i) \right]^2. \qquad (1)$$

Thus, if the optimal policy is being observed, we can substitute the previous expressions for π_1, π_2, ϕ and J_α and solve for w^*. Setting $\alpha = \frac{1}{2}$ yields $w^* = 7/9$, the wrong sign (the wrong sign results for any value of $\alpha \in [0,1)$. So we have a situation where the optimal policy is implementable as a greedy policy based on an approximate value function in the class (just choose any $w < 0$), yet TD(1) observing the optimal policy will converge to a value function whose corresponding greedy policy implements the suboptimal policy.

3 Parameterized POMDPs and average reward

For ease of exposition we consider finite POMDPs. Specifically, assume that there are n states $\mathcal{S} = \{1, \ldots, n\}$, N controls $\mathcal{U} = \{1, \ldots, N\}$ and M observations $\mathcal{Y} = \{1, \ldots, M\}$. For each state $i \in \mathcal{S}$ there is a corresponding reward $r(i)$ (control-dependent rewards are an easy extension of the present results). Each $u \in \mathcal{U}$ determines a stochastic matrix $P(u) = [p_{ij}(u)]$ where $p_{ij}(u)$ is the probability of making a transition from state i to state j give control u. For each state $i \in \mathcal{S}$, an observation $y \in \mathcal{Y}$ is generated independently according to a probability distribution $\nu(i)$ over observations in $\mathcal{Y}$. We denote the probability of observation y by $\nu_y(i)$. A *randomized policy* is simply a function μ mapping observations $y \in \mathcal{Y}$ into probability distributions over the controls $\mathcal{U}$. That is, for each observation y, $\mu(y)$ is a distribution over the controls in $\mathcal{U}$. Denote the probability under μ of control u given observation y by $\mu_u(y)$. Note that by concatenating observations we can make μ depend on observation histories, not simply the current observation.

To each randomized policy $\mu(\cdot)$ and observation distribution $\nu(\cdot)$, there corresponds a Markov chain in which state transitions are generated by first selecting an observation y in state i according to the distribution $\nu(i)$, then selecting a control u according to the distribu-

tion $\mu(y)$, and then generating a transition to state j according to the probability $p_{ij}(u)$. To parameterize these chains we parameterize the policies, so that μ now becomes a function $\mu(\theta, y)$ of a set of parameters $\theta \in \mathbb{R}^K$ as well as the observation y. The Markov chain corresponding to θ has state transition matrix $P(\theta) = [p_{ij}(\theta)]$ given by

$$p_{ij}(\theta) = \mathbf{E}_{y\sim\nu(i)}\mathbf{E}_{u\sim\mu(\theta,y)}p_{ij}(u). \tag{2}$$

We make the following assumptions on $P(\theta)$, the rewards $r(i)$, and the policies $\mu(\cdot, \theta)$:

Assumption 1 *Each $P(\theta)$ has a unique stationary distribution $\pi(\theta) := [\pi(\theta, 1), \ldots, \pi(\theta, n)]'$ satisfying the* balance equations

$$\pi'(\theta)P(\theta) = \pi'(\theta) \tag{3}$$

(throughout π' denotes the transpose of π).

Assumption 2 *The magnitudes of the rewards, $|r(i)|$, are uniformly bounded by $R < \infty$ for all states i.*

Assumption 3 *The derivatives,*

$$\frac{\partial\mu_u(\theta, y)}{\partial\theta_k}$$

exist for all $u \in \mathcal{U}$, $y \in \mathcal{Y}$ and $\theta \in \mathbb{R}^K$. The ratios

$$\left[\frac{\left|\frac{\partial\mu_u(\theta,y)}{\partial\theta_k}\right|}{\mu_u(\theta, y)}\right]_{y=1\ldots M;u=1\ldots N;k=1\ldots K}$$

are uniformly bounded by $B < \infty$ for all $\theta \in \mathbb{R}^K$.

Our goal is to find a $\theta \in \mathbb{R}^K$ maximizing the *long-term average reward*:

$$\eta(\theta) := \lim_{T\to\infty}\frac{1}{T}\mathbf{E}_\theta\left[\sum_{t=1}^{T} r(i_t)\right].$$

where $\mathbf{E}_\theta$ denotes the expectation over all sequences $i_0, i_1, \ldots$, with transitions generated according to $P(\theta)$. Under our assumptions, $\eta(\theta)$ is independent of the starting state i_0 and is equal to:

$$\begin{aligned}\eta(\theta) &= \sum_{i=1}^{n}\pi(\theta, i)r(i)\\ &= \pi'(\theta)r,\end{aligned} \tag{4}$$

where $r = [r(1), \ldots, r(n)]'$ (see, for example, [5]).

Under our assumptions, the gradient of $\eta(\theta)$, $\nabla\eta(\theta)$ exists and is given by

$$\nabla\eta(\theta) = \pi'(\theta)\nabla P(\theta)\left[I - P + e\pi'(\theta)\right]^{-1}r, \tag{5}$$

where $e\pi'(\theta)$ is the square matrix with the stationary distribution $\pi'(\theta)$ in every row. If we could compute (5), then standard gradient ascent methods could be used to find parameters at a local maximum of $\nabla\eta$. Unfortunately, although useful as an existence proof, (5) is of no use in the situations of interest here: namely unknown system dynamics or large state-spaces (the matrix inversion rules out the latter).

4 Stochastic gradient ascent on $\eta(\theta)$

The approach taken to optimization of $\eta(\theta)$ in this paper is *stochastic gradient ascent*. Algorithm 1 introduces OLPOMDP, an algorithm for finding a local maximum of $\eta(\theta)$ from a single sample path of the POMDP.

Algorithm 1 The OLPOMDP algorithm.

1: **Given:**

- Parameterized class of randomized policies $\{\mu(\theta, \cdot) : \theta \in \mathbb{R}^K\}$ satisfying Assumption 3.
- POMDP, which, when controlled by the randomized policies $\mu(\theta, \cdot)$, corresponds to a parameterized class of Markov chains satisfying Assumption 1.
- $\beta \in [0, 1)$, $\lambda > 0$.
- Arbitrary (unknown) starting state i_0.
- Observation sequence $y_0, y_1, \ldots$ generated by the POMDP with controls $u_0, u_1, \ldots$ generated randomly according to $\mu(\theta, y_t)$.
- Reward sequence $r(i_0), r(i_1), \ldots$ satisfying Assumption 2, where $i_0, i_1, \ldots$ is the (hidden) sequence of states of the Markov decision process.
- Step-sizes $\gamma_1, \gamma_2, \cdots > 0$ satisfying $\sum\gamma_t = \infty$ and $\sum\gamma_t^2 < \infty$.

2: Set $z_0 = 0$, $z_0 \in \mathbb{R}^K$.
3: **for** each observation y_t, control u_t, and subsequent reward $r(i_{t+1})$ **do**
4: $\quad z_{t+1} = \beta z_t + \dfrac{\nabla\mu_{u_t}(\theta, y_t)}{\mu_{u_t}(\theta, y_t)}$
5: $\quad \theta_{t+1} = \theta_t + \gamma_t\left(r(i_{t+1})z_{t+1} - \lambda\theta_t\right)$
6: **end for**

Observe that OLPOMDP does not need access to the underlying state and does not make use of recurrent states. To gain some intuition about the parameter updates made by the algorithm, ignore the $\lambda\theta_t$ term and consider the average of the other update term over a suitably long sample path. We will denote this aver-

age by $\nabla_\beta \eta(\theta)$ (the motivation behind the notation will become apparent shortly).

$$\nabla_\beta \eta(\theta) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r(i_t) z_t. \tag{6}$$

Strictly speaking, we should include an expectation over all paths in the right-hand side of (6). However, under our standing assumptions, the limit converges almost surely to the expectation.

Our first result is that $\nabla_\beta \eta(\theta)$ converges to $\nabla \eta(\theta)$, in the limit as β approaches 1 (see [3] for a proof). Thus, provided β is sufficiently close to 1, the average of the updates to the parameters θ in OLPOMDP will be close to the true gradient direction $\nabla \eta(\theta)$. In fact, we can go further and show that the approximation error between $\nabla_\beta \eta(\theta)$ and $\nabla \eta(\theta)$ depends on the relationship between β and the *mixing time* of the underlying Markov chain, where this is defined as follows:

Definition 4 *For any two probability distributions p, q over states, let $\|p - q\|_{TV}$ denote their* total variation *separation:*

$$\|p - q\|_{TV} = \sum_{i=1}^{n} |p_i - q_i|. \tag{7}$$

For any transition probability matrix P, let $p_i^{(t)}$ denote the i-th row of P^t. Define $d(t)$ by,

$$d(t) = \max_{i,j} \|p_i^{(t)} - p_j^{(t)}\|_{TV}. \tag{8}$$

Finally, define the mixing time *of P by*

$$\tau = \min\{t\colon d(t) \le 1/e\}. \tag{9}$$

For any transition probability matrix P with unique stationary distribution π, the Perron-Frobenius theorem states that $P^t \to e\pi'$, where $e\pi'$ is the square matrix with the stationary distribution in each row. Thus $d(t) \to 0$ and the mixing time τ is well defined. It can also be shown that $d(t)$ is a decreasing function of t.

Let $\tau(\theta)$ be the mixing time of $P(\theta)$ (defined in (2)). The following theorem relates the approximation error between $\nabla_\beta \eta(\theta)$ and $\nabla \eta(\theta)$ to $\tau(\theta)$. See [2] for a proof.

Theorem 5 *Under Assumptions 1, 2 and 3, there exists a constant $C = C(B, R, n)$ such that for all θ,*

$$\|\beta \nabla_\beta \eta(\theta) - \nabla \eta(\theta)\| \le C \tau(\theta) (1 - \beta). \tag{10}$$

Thus, if $1/(1-\beta)$ is large compared to the mixing time $\tau(\theta)$, $\nabla_\beta \eta(\theta)$ will be close to $\nabla \eta(\theta)$, and so the long-term average of the updates to the parameters in OLPOMDP will be close to the true gradient direction. In [2] we also showed that the *variance* of the estimates $\sum_{t=1}^{T} r(i_t) z_t$ produced after a finite number of steps T, scales as $1/(T(1-\beta)^2)$. Hence β has a natural interpretation in terms of a *bias/variance* trade-off: Setting β small in OLPOMDP ensures low variance in the long-term average of the parameter updates, but the expected direction of those updates may be a very biased estimate of the gradient. Conversely, setting β close to 1 ensures a low bias but a higher variance.

5 Convergence of Algorithm 1

Theorem 6 *As $t \to \infty$, $\theta_t \to L$ a.s., where L is the set of stable equilibrium points of the differential equation*

$$\frac{d\theta}{dt} = \nabla_\beta \eta(\theta) - \lambda\theta.$$

The proof uses the following result (see [14, Chapter 8] for a result of this form).

Lemma 7 *Consider updates of the form*

$$\theta_{t+1} = \theta_t + \gamma_t d(\theta_t, Z_t(\theta_t)), \tag{11}$$

where $\{Z_t(\theta)\}$ is an auxiliary *(homogeneous) ergodic Markov chain taking values in $\mathbb{R}^p$. For $z, z' \in \mathbb{R}^p$ and $\theta \in \mathbb{R}^k$, let $P(z, z'|\theta)$ denote the probability density of a transition from z to z'. Suppose that $\{Z_t(\theta)\}$, $P(z, z'|\theta)$, $d(\theta, z)$ and the step sizes γ_t satisfy the following conditions:*

1. *$\gamma_t \ge 0$, $\sum_{t=0}^{\infty} \gamma_t = \infty$, and $\sum_{t=0}^{\infty} \gamma_t^2 < \infty$.*
2. *$P(z, z'|\theta)$ is weakly continuous in (θ, z); that is, for all bounded and continuous $f\colon \mathbb{R}^p \to \mathbb{R}$,*

$$\int f(z') P(z, z'|\theta)\, dz'$$

   is continuous in (θ, z).
3. *$d(\theta, z)$ is continuous on $\mathbb{R}^k \times \mathbb{R}^p$.*
4. *$\{Z_t(\theta)\}$ is* positive recurrent, *which means for all $\theta \in \mathbb{R}^k$, $P(z, z'|\theta)$ has a unique stationary distribution $\pi(\theta, z)$.*
5. *$\bar{d}(\theta)$ is a continuous function of θ, where*

$$\bar{d}(\theta) := \int d(\theta, z) \pi(\theta, z)\, dz.$$

6. *For all initial distributions Z_0,*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} d(\theta, Z_t) = \bar{d}(\theta) \quad (w.p.1).$$

7. *θ_t is uniformly bounded for all t with probability 1.*

Now let $L \subset \mathbb{R}^k$ denote the stable equilibrium points of the ordinary differential equation,

$$\frac{d\theta(t)}{dt} = \bar{d}(\theta). \tag{12}$$

Then

$$\theta_t \to L \quad as \quad t \to \infty \quad (w.p.1). \tag{13}$$

Proof: *(of Theorem 6)* We apply Lemma 7 with

$$Z_t(\theta_t) = (y_t, u_t, i_{t+1}, z_{t+1}, r(i_{t+1}))$$
$$d(\theta_t, Z(\theta_t)) = r(i_{t+1}) z_{t+1} - \lambda\theta_t.$$

Clearly, $Z_t(\theta)$ is a Markov chain with a weakly continuous transition probability density, and d is continuous. Since the Markov chain (y_t, u_t, i_{t+1}) is ergodic, $Z_t(\theta)$ has a unique stationary distribution. The expectation $\bar{d}(\theta)$ of d under this stationary distribution is $\nabla_\beta \eta - \lambda\theta$, which is continuous, and the time average of d approaches $\bar{d}$ almost surely. The result follows from Lemma 7. ∎

6 Extensions to infinite state, observation, and control spaces

With no modification Algorithm 1 can be applied immediately to POMDPs with countably or uncountably infinite states $\mathcal{S}$ and observations $\mathcal{Y}$, and countable controls $\mathcal{U}$. In addition, with the appropriate interpretation of $\nabla\mu/\mu$, it can be applied to uncountable $\mathcal{U}$. Specifically, if $\mathcal{U}$ is a subset of $\mathbb{R}^N$ then $\mu(y, \theta)$ will be a probability *density* function on $\mathcal{U}$ with $\mu_u(y, \theta)$ the density at u.

7 Conclusion

We have presented a general algorithm (OLPOMDP) for finding local maxima of the average reward in POMDPs controlled by parameterized policies. The algorithm applies to arbitrary state, control and observation spaces. We proved almost sure convergence of the algorithm. The algorithm does not rely on recurrent states, but instead takes a parameter $\beta \in [0, 1)$, which is determined by the *mixing time* of the Markov chain induced by the POMDP.

References

[1] V. M. Aleksandrov, V. I. Sysoyev, and V. V. Shemeneva. Stochastic optimaization. *Engineering Cybernetics*, 5:11–16, 1968.

[2] P. L. Bartlett and J. Baxter. Estimation and approximation bounds for gradient-based reinforcement learning. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 133–141, 2000.

[3] J. Baxter and P. L. Bartlett. Infinite-Horizon Gradient-Based Policy Search. Technical report, Research School of Information Sciences and Engineering, Australian National University, July 1999.

[4] J. Baxter, A. Tridgell, and L. Weaver. Learning to Play Chess Using Temporal-Differences. *Machine Learning*, 40(3):243–264, 1999.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol II.* Athena Scientific, 1995.

[6] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

[7] X.-R. Cao and H.-F. Chen. Perturbation Realization, Potentials, and Sensitivity Analysis of Markov Processes. *IEEE Transactions on Automatic Control*, 42:1382–1393, 1997.

[8] X.-R. Cao and Y.-W. Wan. Algorithms for Sensitivity Analysis of Markov Chains Through Potentials and Perturbation Realization. *IEEE Transactions on Control Systems Technology*, 6:482–492, 1998.

[9] M. C. Fu and J. Hu. Smooth Perturbation Derivative Estimation for Markov Chains. *Operations Research Letters*, 15:241–251, 1994.

[10] P. W. Glynn. Stochastic approximation for Monte-Carlo optimization. In *Proceedings of the 1986 Winter Simulation Conference*, pages 356–365, 1986.

[11] P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33:75–84, 1990.

[12] P. W. Glynn and P. L'Ecuyer. Likelihood ratio gradient estimation for regenerative stochastic recursions. *Advances in Applied Probability, 27, 4 (1995)*, 27:1019–1053, 1995.

[13] H. Kimura, K. Miyazaki, and S. Kobayashi. Reinforcement learning in POMDPs with function approximation. In D. H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 152–160, 1997.

[14] H. J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications.* Springer, New York, 1997.

[15] P. Marbach and J. N. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes. Technical report, MIT, 1998.

[16] M. I. Reiman and A. Weiss. Sensitivity analysis via likelihood ratios. In *Proceedings of the 1986 Winter Simulation Conference*, 1986.

[17] M. I. Reiman and A. Weiss. Sensitivity analysis for simulations via likelihood ratios. *Operations Research*, 37, 1989.

[18] R. Y. Rubinstein. *Some Problems in Monte Carlo Optimization*. PhD thesis, 1969.

[19] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3:210–229, 1959.

[20] S. Singh and D. Bertsekas. Reinforcement learning for dynamic channel allocation in cellular telephone systems. In *Advances in Neural Information Processing Systems: Proceedings of the 1996 Conference*, pages 974–980. MIT Press, 1997.

[21] S. P. Singh, T. Jaakkola, and M. I. Jordan. Learning Without State-Estimation in Partially Observable Markovian Decision Processes. In *Proceedings of the Eleventh International Conference on Machine Learning*, 1994.

[22] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge MA, 1998. ISBN 0-262-19398-1.

[23] G. Tesauro. Practical Issues in Temporal Difference Learning. *Machine Learning*, 8:257–278, 1992.

[24] G. Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6:215–219, 1994.

[25] J. N. Tsitsikilis and B. Van-Roy. An Analysis of Temporal Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[26] R. J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 1992.

[27] W. Zhang and T. Dietterich. A reinforcement learning approach to job-shop scheduling. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1114–1120. Morgan Kaufmann, 1995.