Simulation-Based Optimization of Markov Reward Processes[1]

Peter Marbach

Center for Communications Sustems Research
University of Cambridge
10 Downing Street
Cambridge, CB2 3DS, UK
e-mail: marbach@ccsr.cam.ac.uk

John N. Tsitsiklis

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
e-mail: jnt@mit.edu

Abstract: We propose a simulation-based algorithm for optimizing the average reward in a Markov Reward Process that depends on a set of parameters. As a special case, the method applies to Markov Decision Processes where optimization takes place within a parametrized set of policies. The algorithm involves the simulation of a single sample path, and can be implemented on-line. A convergence result (with probability 1) is provided.

[1]This research was supported by contracts with Siemens AG, Munich, Germany, and Alcatel Bell, Belgium; and by contract DMI-9625489 with the National Science Foundation.

1 Introduction

Markov Decision Processes and the associated dynamic programming (DP) methodology [Ber95a, Put94] provide a general framework for posing and analyzing problems of sequential decision making under uncertainty. DP methods rely on a suitably defined value function that has to be computed for every state in the state space. However, many interesting problems involve very large state spaces ("curse of dimensionality"). In addition, DP assumes the availability of an exact model, in the form of transition probabilities. In many practical situations, such a model is not available and one must resort to simulation or experimentation with an actual system. For all of these reasons, dynamic programming in its pure form, may be inapplicable.

The efforts to overcome the aforementioned difficulties involve two main ideas:

1. The use of simulation to estimate quantities of interest, thus avoiding model-based computations.

2. The use of parametric representations to overcome the curse of dimensionality.

Parametric representations, and the associated algorithms, can be broadly classified into three main categories.

(a) *Parametrized value functions:* Instead of associating a value $V(i)$ with each state i, one uses a parametric form $\tilde{V}(i, r)$, where r is a vector of tunable parameters (weights), and $\tilde{V}$ is a so-called approximation architecture. For example, $\tilde{V}(i, r)$ could be the output of a multilayer perceptron with weights r, when the input is i. Other representations are possible, e.g., involving polynomials, linear combinations of feature vectors, state aggregation, etc. When the main ideas from DP are combined with such parametric representations, one obtains methods that go under the names of "reinforcement learning" or "neuro-dynamic programming"; see [BT96, SB98] for textbook expositions, as well as the references therein. A key characteristic is that policy optimization is carried out in an indirect fashion: one tries to obtain a good approximation of the optimal value function of dynamic programming, and uses it to construct policies that are close to optimal. Such methods are reasonably well, though not fully, understood and there have been some notable practical successes (see [BT96, SB98] for an overview), including the world-class backgammon player by Tesauro [Tes92].

(b) *Parametrized policies:* In an alternative approach, which is the one considered in this paper, the tuning of a parametrized value function is bypassed. Instead, one considers a class of policies described in terms of a parameter vector θ. Simulation is employed to estimate the gradient of the performance metric with respect to θ, and the policy is improved by updating θ in a gradient direction. In some cases, the required gradient can be estimated using IPA (infinitesimal perturbation analysis); see, e.g., [HC91, Gla91, CR94] and the references therein. For general Markov processes, and in the absence of special structure, IPA is inapplicable, but gradient estimation is still possible using "likelihood-ratio" methods [Gly86, Gly87, GG92, LEc90, GI89].

(c) *Actor-critic methods:* A third approach, which is a combination of the first two, includes parametrizations of the policy (actor) and of the value function (critic)

[BSA83]. While such methods seem particularly promising, theoretical understanding has been limited to the impractical case of lookup representations (one parameter per state) [KB98].

This paper concentrates on methods based on policy parametrization and (approximate) gradient improvement, in the spirit of item (b) above. While we are primarily interested in the case of Markov Decision Processes, almost everything applies to the more general case of Markov Reward Processes that depend on a parameter vector θ, and we proceed within this broader context.

We start with a formula for the gradient of the performance metric that has been presented in different forms and for various contexts in [Gly87, CC97, FH94, JSJ95, TH95, CW98]. We then suggest a method for estimating the terms that appear in that formula. This leads to a simulation-based method that updates the parameter vector θ at every regeneration time, in an approximate gradient direction. Furthermore, we show how to construct an on-line method that updates the parameter vector at each time step. The resulting method has some conceptual similarities with those described in [CR94] (that reference assumes, however, the availability of an IPA estimator, with certain guaranteed properties that are absent in our context) and in [JSJ95] (which, however, does not contain convergence results).

The method that we propose only keeps in memory and updates $2K + 1$ numbers, where K is the dimension of θ. Other than θ itself, this includes a vector similar to the "eligibility trace" in Sutton's temporal difference methods, and (as in [JSJ95]) an estimate $\tilde{\lambda}$ of the average reward under the current value of θ. If that estimate was accurate, our method would be a standard stochastic gradient algorithm. However, as θ keeps changing, $\tilde{\lambda}$ is generally a biased estimate of the true average reward, and the mathematical structure of our method is more complex than that of stochastic gradient algorithms. For reasons that will become clearer later, standard approaches (e.g., martingale arguments or the ODE approach) do not seem to suffice for establishing convergence, and a more elaborate proof is necessary.

Our gradient estimator can also be derived or interpreted in terms of likelihood ratios [Gly87, GG92]. It takes the same form as the one presented in p. 371 of [Gly87], but it is used differently. The development in [Gly87] leads to a consistent estimator of the gradient, assuming that a very large number of regenerative cycles are estimated, while keeping the policy parameter θ at a fixed value. Presumably, θ would be then updated after such a long simulation. In contrast, our method updates θ much more frequently and retains the desired convergence properties, despite the fact that any single cycle results in a biased gradient estimate.

An alternative simulation-based stochastic gradient method, again based on a likelihood ratio formula, has been provided in [Gly86], and uses the simulation of *two* regenerative cycles to construct an unbiased estimate of the gradient. We note some of the differences with the latter work. First, the methods in [Gly86] involve a larger number of auxiliary quantities that are propagated in the course of a regenerative cycle. Second, our method admits a modification (see Sections 4-5) that can make it applicable even if the time until the next regeneration is excessive (in which case, likelihood ratio-based methods suffer from excessive variance). Third, our estimate $\tilde{\lambda}$ of the average reward is obtained as a (weighted) average of all past rewards (not just over the last regenerative cycle). In contrast, an approach such as the one in [Gly86] would construct an independent estimate of $\tilde{\lambda}$ during each regenerative cycle, which should result in higher variance. Finally, our method brings forth and makes crucial use of the value (differential reward) function of dynamic programming. This is important because it paves the way for actor-critic methods in which the variance associated with the estimates of the differential rewards is potentially reduced by means of "learning" (value function approximation). Indeed, subsequent to the first writing of this paper, this latter approach has been pursued in [KT99, SMS99].

In summary, the main contributions of this paper are as follows.

1. We introduce a new algorithm for updating the parameters of a Markov Reward Process, on the basis of a single sample path. The parameter updates can take place either during visits to a certain recurrent state, or at every time step. We also specialize the method to Markov Decision Processes with parametrically represented policies. In this case, the method does not require the transition probabilities to be known.

2. We establish that the gradient (with respect to the parameter vector) of the performance metric converges to zero, with probability 1, which is the strongest possible result for gradient-related stochastic approximation algorithms.

3. The method admits approximate variants with reduced variance, such as the one described in Section 5, or various types of actor-critic methods.

The remainder of this paper is organized as follows. In Section 2, we introduce our framework and assumptions, and state some background results, including a formula for the gradient of the performance metric. In Section 3, we present an algorithm that performs updates during visits to a certain recurrent state, present our main convergence result, and provide a heuristic argument. Sections 4 and 5 deal with variants of the algorithm that perform updates at every time step. In Section 6, we specialize our methods to the case of Markov Decision Processes that are optimized within a possibly restricted set of parametrically represented randomized policies. We present some numerical results in Section 7, and conclude in Section 8. The lengthy proof of our main results is developed in the appendices.

2 Markov Reward Processes Depending on a Parameter

In this section, we present our general framework, make a few assumptions, and state some basic results that will be needed later.

We consider a discrete-time, finite-state Markov chain $\{i_n\}$ with state space $S = \{1, \ldots, N\}$, whose transition probabilities depend on a parameter vector $\theta \in \Re^K$, and are denoted by

$$p_{ij}(\theta) = P(i_n = j \mid i_{n-1} = i,\ \theta).$$

Whenever the state is equal to i, we receive a one-stage reward, that also depends on θ, and is denoted by $g_i(\theta)$.

For every $\theta \in \Re^K$, let $P(\theta)$ be the stochastic matrix with entries $p_{ij}(\theta)$. Let $\mathcal{P} = \{P(\theta) \mid \theta \in \Re^K\}$ be the set of all such matrices, and let $\overline{\mathcal{P}}$ be its closure. Note that every element of $\overline{\mathcal{P}}$ is also a stochastic matrix and, therefore, defines a Markov chain on the same state space. We make the following assumptions.

Assumption 1 *The Markov chain corresponding to every $P \in \overline{\mathcal{P}}$ is aperiodic. Furthermore, there exists a state i^* which is recurrent for every such Markov chain.*

We will often refer to the times that the state i^* is visited as *regeneration times.*

Assumption 2 *For every $i, j \in S$, the functions $p_{ij}(\theta)$ and $g_i(\theta)$ are bounded, twice differentiable, and have bounded first and second derivatives.*

The performance metric that we use to compare different policies is the average reward criterion $\lambda(\theta)$, defined by

$$\lambda(\theta) = \lim_{t\to\infty} \frac{1}{t} E_\theta \left[\sum_{k=0}^{t} g_{i_k}(\theta) \right].$$

Here, i_k is the state at time k, and the notation $E_\theta[\cdot]$ indicates that the expectation is taken with respect to the distribution of the Markov chain with transition probabilities $p_{ij}(\theta)$. Under Assumption 1, the average reward $\lambda(\theta)$ is well defined for every θ, and does not depend on the initial state. Furthermore, the balance equations

$$\sum_{i=1}^{N} \pi_i(\theta) p_{ij}(\theta) = \pi_j(\theta), \qquad j = 1, \ldots, N-1, \tag{1}$$

$$\sum_{i=1}^{N} \pi_i(\theta) = 1, \tag{2}$$

have a unique solution $\pi(\theta) = (\pi_1(\theta), \ldots, \pi_N(\theta))$, with $\pi_i(\theta)$ being the steady state probability of state i under that particular value of θ, and the average reward is equal to

$$\lambda(\theta) = \sum_{i=1}^{N} \pi_i(\theta) g_i(\theta). \tag{3}$$

We observe that the balance equations (1)-(2) are of the form

$$A(\theta)\pi(\theta) = a,$$

where a is a fixed vector and $A(\theta)$ is an $N \times N$ matrix. (Throughout the paper, all vectors are treated as column vectors.) Using the fact that $A(\theta)$ depends smoothly on θ, we have the following result.

Lemma 1 *Let Assumptions 1 and 2 hold. Then, $\pi(\theta)$ and $\lambda(\theta)$ are twice differentiable, and have bounded first and second derivatives.*

Proof: The balance equations are of the form $A(\theta)\pi(\theta) = a$, where the entries of $A(\theta)$ have bounded second derivatives (Assumption 2). Since the balance equations have a unique solution, the matrix $A(\theta)$ is always invertible, and Cramer's rule yields

$$\pi(\theta) = \frac{C(\theta)}{\det(A(\theta))}, \tag{4}$$

where $C(\theta)$ is a vector whose entries are polynomial functions of the entries of $A(\theta)$. Using Assumption 2, $C(\theta)$ and $\det(A(\theta))$ are twice differentiable and have bounded first and second derivatives.

More generally, suppose that $P \in \overline{\mathcal{P}}$, i.e., P is the limit of the stochastic matrices $P(\theta_k)$ along some sequence θ_k. The corresponding balance equations are again of the form $A(P)\pi = a$, where $A(P)$ is a matrix depending on P. Under Assumption 1, these balance equations have again a unique solution, which implies that $|\det(A(P))|$ is strictly positive. Note that $|\det(A(P))|$ is a continuous function of P, and P lies in the set $\overline{\mathcal{P}}$, which is closed and bounded. It follows that $|\det(A(P))|$ is bounded below by a positive constant c. Since every $P(\theta)$ belongs to $\overline{\mathcal{P}}$, it follows that $|\det(A(\theta))| \geq c > 0$, for every θ. This fact, together with Eq. (4) implies that $\pi(\theta)$ is twice differentiable and has bounded first and second derivatives. The same property holds true for $\lambda(\theta)$, as can be seen by differentiating twice the formula (3). $\square$

2.1 The Gradient of $\lambda(\theta)$

For any $\theta \in \Re^K$ and $i \in S$, we define the differential reward $v_i(\theta)$ of state i by

$$v_i(\theta) = E_\theta\left[\sum_{k=0}^{T-1} (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = i\right], \tag{5}$$

where i_k is the state at time k, and $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. With this definition, it is well known that $v_{i^*}(\theta) = 0$ and that the vector $v(\theta) = (v_1(\theta), \ldots, v_N(\theta))$ is a solution to the Poisson equation

$$g(\theta) = v + \lambda(\theta)e - P(\theta)v$$

where $g(\theta) = (g_1(\theta), \ldots, g_N(\theta)$ and e is equal to the all-one vector $(1, \ldots, 1)$.

The following proposition gives an expression for the gradient of the average reward $\lambda(\theta)$, with respect to θ. A related expression (in a somewhat different context) was given in [JSJ95], and a proof can be found in [CC97]. (The latter reference does not consider the case where $g_i(\theta)$ depends on θ, but the extension is immediate.) Given the importance of this result, and because existing proofs are somewhat involved, we provide a concise self-contained proof, for the benefit of the reader.

Proposition 1 *Let Assumptions 1 and 2 hold. Then,*

$$\nabla\lambda(\theta) = \sum_{i \in S} \pi_i(\theta)\left(\nabla g_i(\theta) + \sum_{j \in S} \nabla p_{ij}(\theta) v_j(\theta)\right).$$

Proof: We carry out the proof using vector notation, and using the superscript T to denote vector transposition. All gradients are taken with respect to θ, but to unclutter notation, the dependence on θ is suppressed.

We start with the Poisson equation $g = v + \lambda e - Pv$ and left-multiply both sides with $\nabla\pi^T$, to obtain

$$(\nabla\pi^T)g = (\nabla\pi^T)v + \lambda\nabla\pi^T e - (\nabla\pi^T)(Pv). \tag{6}$$

Note that $\pi^T e = 1$, which yields $\nabla \pi^T e = 0$. Using the balance equation $\pi^T P = \pi^T$, we obtain

$$\nabla \pi^T = \nabla(\pi^T P) = (\nabla \pi^T)P + \pi^T(\nabla P).$$

We right-multiply both sides by v, and use the resulting relation to rewrite the right-hand side of Eq. (6), leading to

$$(\nabla \pi^T)g = \pi^T(\nabla P)v.$$

Thus,

$$\nabla \lambda = \nabla(\pi^T g) = \pi^T(\nabla g) + (\nabla \pi^T)g = \pi^T(\nabla g) + \pi^T(\nabla P)v,$$

which is the desired result. □

Equation (3) for $\lambda(\theta)$ suggests that $\nabla \lambda(\theta)$ could involve terms of the form $\nabla \pi_i(\theta)$, but the expression given by Proposition 1 involves no such terms. This property is very helpful in producing simulation-based estimates of $\nabla \lambda(\theta)$.

2.2 An Idealized Gradient Algorithm

Given that our goal is to maximize the average reward $\lambda(\theta)$, it is natural to consider gradient-type methods. If the gradient of $\lambda(\theta)$ could be computed exactly, we would consider a gradient algorithm of the form

$$\theta_{k+1} = \theta_k + \gamma_k \nabla \lambda(\theta_k).$$

Based on the fact that $\lambda(\theta)$ has bounded second derivatives, and under suitable conditions on the stepsizes γ_k, it would follow that $\lim_{k\to\infty} \nabla \lambda(\theta_k) = 0$ and that $\lambda(\theta_k)$ converges [Ber95b].

Alternatively, if we could use simulation to produce an unbiased estimate h_k of $\nabla \lambda(\theta_k)$, we could then employ the stochastic gradient iteration

$$\theta_{k+1} = \theta_k + \gamma_k h_k.$$

The convergence of such a method can be established if we use a diminishing stepsize sequence and make suitable assumptions on the estimation errors. While one can construct unbiased estimates of the gradient [Gly86], it does not appear possible to use them in an algorithm which updates the parameter vector θ at every time step – which is a desirable property, as discussed in Section 3.4. This difficulty is bypassed by the method developed in the following.

3 The Simulation–Based Method

In this section, we develop a simulation-based algorithm in which the gradient $\nabla \lambda(\theta)$ is replaced with a biased estimate, obtained by simulating a single sample path. We will eventually show that the bias asymptotically vanishes, which will then lead to a convergence result. For technical reasons, we make the following assumption on the transition probabilities $p_{ij}(\theta)$.

Assumption 3 *For every i and j, there exists a bounded function $L_{ij}(\theta)$ such that*

$$\nabla p_{ij}(\theta) = p_{ij}(\theta) L_{ij}(\theta), \qquad \forall\ \theta.$$

Note that when $p_{ij}(\theta) > 0$, we have

$$L_{ij}(\theta) = \frac{\nabla p_{ij}(\theta)}{p_{ij}(\theta)},$$

which can be interpreted as a likelihood ratio derivative term [LEc90]. Assumption 3 holds automatically if there exists a positive scalar ϵ, such that for every $i, j \in S$, we have

$$\text{either} \quad p_{ij}(\theta) = 0, \quad \forall\, \theta, \qquad\qquad \text{or} \quad p_{ij}(\theta) \geq \epsilon, \quad \forall\, \theta.$$

3.1 Estimation of $\nabla\lambda(\theta)$

Throughout this subsection, we assume that the parameter vector θ is fixed to some value. Let $\{i_n\}$ be a sample path of the corresponding Markov chain, possibly obtained through simulation. Let t_m be the time of the mth visit at the recurrent state i^*. We refer to the sequence $i_{t_m}, i_{t_m+1}, \ldots, i_{t_{m+1}}$ as the mth *regenerative cycle*, and we define its *length* T_m by

$$T_m = t_{m+1} - t_m.$$

For a fixed θ, the random variables T_m are independent identically distributed, and have a (common) finite mean, denoted by $E_\theta[T]$.

Our first step is to rewrite the formula for $\nabla\lambda(\theta)$ in the form

$$\nabla\lambda(\theta) = \sum_{i\in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} p_{ij}(\theta) L_{ij}(\theta) v_j(\theta) \right).$$

Estimating the term $\pi_i(\theta)\nabla g_i(\theta)$ through simulation is straightforward, assuming that we are able to compute $\nabla g_i(\theta)$ for any given i and θ. The other term can be viewed as the expectation of $v_j(\theta)L_{ij}(\theta)$, with respect to the steady-state probability $\pi_i(\theta)p_{ij}(\theta)$ of transitions from i to j. Furthermore, the definition (5) of $v_j(\theta)$, suggests that if $t_m < n \leq t_{m+1} - 1$, and $i_n = j$, we can use

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} \left(g_{i_k}(\theta) - \tilde{\lambda} \right), \tag{7}$$

to estimate $v_j(\theta)$, where $\tilde{\lambda}$ is some estimate of $\lambda(\theta)$. Note that $v_{i^*}(\theta) = 0$ and does not need to be estimated. For this reason, we let

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = 0, \qquad \text{if } n = t_m.$$

By accumulating the above described estimates over a regenerative cycle, we are finally led to an estimate of the direction of $\nabla\lambda(\theta)$ given by

$$F_m(\theta, \tilde{\lambda}) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{v}_{i_n}(\theta, \tilde{\lambda}) L_{i_{n-1} i_n}(\theta) + \nabla g_{i_n}(\theta) \right). \tag{8}$$

The random variables $F_m(\theta, \tilde{\lambda})$ are independent and identically distributed for different values of m, because the transitions during distinct regenerative cycles are independent.

We define $f(\theta, \tilde{\lambda})$ to be the expected value of $F_m(\theta, \tilde{\lambda})$, namely,

$$f(\theta, \tilde{\lambda}) = E_\theta[F_m(\theta, \tilde{\lambda})]. \tag{9}$$

The following proposition confirms that the expectation of $F_m(\theta, \tilde{\lambda})$ is aligned with $\nabla\lambda(\theta)$, to the extent that $\tilde{\lambda}$ is close to $\lambda(\theta)$.

Proposition 2 *We have*

$$f(\theta, \tilde{\lambda}) = E_\theta[T]\nabla\lambda(\theta) + G(\theta)(\lambda(\theta) - \tilde{\lambda}),$$

where

$$G(\theta) = E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} (t_{m+1} - n) L_{i_{n-1}i_n}(\theta)\right]. \tag{10}$$

Proof: Note that for $n = t_m + 1, \ldots, t_{m+1} - 1$, we have

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(\theta) - \lambda(\theta)) + (t_{m+1} - n)(\lambda(\theta) - \tilde{\lambda}).$$

Therefore,

$$F_m(\theta, \tilde{\lambda}) = \sum_{n=t_m+1}^{t_{m+1}-1} a_n L_{i_{n-1}i_n}(\theta) + \sum_{n=t_m+1}^{t_{m+1}-1} (t_{m+1} - n)(\lambda(\theta) - \tilde{\lambda}) L_{i_{n-1}i_n}(\theta) + \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta),$$

where

$$a_n = \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(\theta) - \lambda(\theta)). \tag{11}$$

We consider separately the expectations of the three sums above. Using the definition of $G(\theta)$, the expectation of the second sum is equal to $G(\theta)(\lambda(\theta) - \tilde{\lambda})$. We then consider the third sum. As is well known, the expected sum of rewards over a regenerative cycle is equal to the steady-state expected reward times the expected length of the regenerative cycle. Therefore, the expectation of the third sum is

$$E_\theta\left[\sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta)\right] = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \nabla g_i(\theta). \tag{12}$$

We now focus on the expectation of the first sum. For $n = t_m + 1, \ldots, t_{m+1} - 1$, let

$$\Delta_n = (a_n - v_{i_n}(\theta)) L_{i_{n-1}i_n}(\theta).$$

Let $\mathcal{F}_n = \{i_0, \ldots, i_n\}$ stand for the history of the process up to time n. By comparing the definition (11) of a_n with the definition (5) of $v_{i_n}(\theta)$, we obtain

$$E_\theta[a_n \mid \mathcal{F}_n] = v_{i_n}(\theta). \tag{13}$$

It follows that $E_\theta[\Delta_n \mid \mathcal{F}_n] = 0$.

Let $\chi_n = 1$ if $n < t_{m+1}$, and $\chi_n = 0$, otherwise. For any $n > t_m$, we have

$$E_\theta[\chi_n \Delta_n \mid \mathcal{F}_{t_m}] = E_\theta[E_\theta[\chi_n \Delta_n \mid \mathcal{F}_n] \mid \mathcal{F}_{t_m}] = E_\theta[\chi_n E_\theta[\Delta_n \mid \mathcal{F}_n] \mid \mathcal{F}_{t_m}] = 0.$$

We then have

$$\begin{aligned} E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} \Delta_n \mid \mathcal{F}_{t_m}\right] &= E_\theta\left[\sum_{n=t_m+1}^{\infty} \chi_n \Delta_n \mid \mathcal{F}_{t_m}\right] \\ &= \sum_{n=t_m+1}^{\infty} E_\theta[\chi_n \Delta_n \mid \mathcal{F}_{t_m}] \\ &= 0. \end{aligned}$$

(The interchange of the summation and the expectation can be justified by appealing to the dominated convergence theorem.)

We therefore have

$$E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} a_n L_{i_{n-1}i_n}(\theta)\right] = E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} v_{i_n}(\theta) L_{i_{n-1}i_n}(\theta)\right].$$

The right-hand side can be viewed as the total reward over a regenerative cycle of a Markov reward process, where the reward associated with a transition from i to j is $v_j(\theta)L_{ij}(\theta)$. Recalling that any particular transition has steady-state probability $\pi_i(\theta)p_{ij}(\theta)$ of being from i to j, we obtain

$$E_\theta\left[\sum_{n=t_m+1}^{t_{m+1}-1} a_n L_{i_{n-1}i_n}(\theta)\right] = E_\theta[T] \sum_{i\in S}\sum_{j\in S} \pi_i(\theta) p_{ij}(\theta) L_{ij}(\theta) v_j(\theta). \tag{14}$$

By combining Eqs. (12) and (14), and comparing with the formula for $\nabla\lambda(\theta)$, we see that the desired result has been proved. $\square$

3.2 An Algorithm that Updates at Visits to the Recurrent State

We now use the approximate gradient direction provided by Proposition 2, and propose a simulation-based algorithm that performs updates at visits to the recurrent state i^*. We use the variable m to index the times when the recurrent state i^* is visited, and the corresponding updates. The form of the algorithm is the following. At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m and an average reward estimate $\tilde{\lambda}_m$. We then simulate the process according to the transition probabilities $p_{ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited, and update according to

$$\theta_{m+1} = \theta_m + \gamma_m F_m(\theta_m, \tilde{\lambda}_m), \tag{15}$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m), \tag{16}$$

where γ_m is a positive stepsize sequence (cf. Assumption 4) and $\eta > 0$ allows to scale the stepsize for updating $\tilde{\lambda}$ by a positive constant. To see the rationale behind Eq. (16), note that the expected update direction for $\tilde{\lambda}$ is

$$E_\theta\left[\sum_{n=t_m}^{t_{m+1}-1}(g_{i_n}(\theta)-\tilde{\lambda})\right] = E_\theta[T](\lambda(\theta)-\tilde{\lambda}), \tag{17}$$

which moves $\tilde{\lambda}$ closer to $\lambda(\theta)$.

Assumption 4 *The stepsizes γ_m are nonnegative and satisfy*

$$\sum_{m=1}^{\infty}\gamma_m = \infty, \qquad \sum_{m=1}^{\infty}\gamma_m^2 < \infty.$$

Assumption 4 is satisfied, for example, if we let $\gamma_m = 1/m$. It can be shown that if θ is held fixed, but $\tilde{\lambda}$ keeps being updated according to Eq. (16), then $\tilde{\lambda}$ converges to $\lambda(\theta)$. However, if θ is also updated according to Eq. (15), then the estimate $\tilde{\lambda}_m$ can "lag behind" $\lambda(\theta_m)$. As a consequence, the expected update direction for θ will not be aligned with the gradient $\nabla\lambda(\theta)$.

An alternative approach that we do not pursue is to use different stepsizes for updating $\tilde{\lambda}$ and θ. If the stepsize used to update θ is, in the limit, much smaller than the stepsize used to update $\tilde{\lambda}$, the algorithm exhibits a two-time scale behavior of the form studied in [Bor97]. In the limit, $\tilde{\lambda}_m$ is an increasingly accurate estimate of $\lambda(\theta_m)$, and the algorithm is effectively a stochastic gradient algorithm. However, such a method would make slower progress, as far as θ is concerned. Our convergence results indicate that this alternative approach is not necessary.

We can now state our main result, which is proved in Appendix A.

Proposition 3 *Let Assumptions 1-4 hold, and let $\{\theta_m\}$ be the sequence of parameter vectors generated by the above described algorithm. Then, $\lambda(\theta_m)$ converges and*

$$\lim_{m\to\infty}\nabla\lambda(\theta_m) = 0,$$

with probability 1.

3.3 A Heuristic Argument

In this subsection, we approximate the algorithm by a suitable ODE (as in [Lju77]), and establish the convergence properties of the ODE. While this argument does not constitute a proof, it illustrates the rationale behind our convergence result.

We replace the update directions by their expectations under the current value of θ. The resulting deterministic update equations take the form

$$\begin{aligned}
\theta_{m+1}^d &= \theta_m^d + \gamma_m f(\theta_m^d, \tilde{\lambda}_m^d), \\
\tilde{\lambda}_{m+1}^d &= \tilde{\lambda}_m^d + \eta\gamma_m E_{\theta_m^d}[T](\lambda(\theta_m^d) - \tilde{\lambda}_m^d),
\end{aligned}$$

where $f(\theta,\tilde{\lambda})$ is given by Proposition 2, and where θ_m^d and $\tilde{\lambda}_m^d$ are the deterministic counterparts of θ_m and $\tilde{\lambda}_m$, respectively. With an asymptotically vanishing stepsize, and after rescaling time, this deterministic iteration behaves similar to the following system of differential equations:

$$\dot{\theta}_t = \nabla\lambda(\theta_t) + \frac{G(\theta_t)}{E_{\theta_t}[T]}(\lambda(\theta_t) - \tilde{\lambda}_t), \tag{18}$$

$$\dot{\tilde{\lambda}}_t = \eta\Big(\lambda(\theta_t) - \tilde{\lambda}_t\Big). \tag{19}$$

Note that $\tilde{\lambda}_t$ and $\lambda(\theta_t)$ are bounded functions since the one-stage reward $g_i(\theta)$ is finite-valued and, therefore, bounded. We will now argue that $\tilde{\lambda}_t$ converges.

We first consider the case where the initial conditions satisfy $\tilde{\lambda}_0 \leq \lambda(\theta_0)$. We then claim that

$$\tilde{\lambda}_t \leq \lambda(\theta_t), \qquad \forall\, t > 0. \tag{20}$$

Indeed, suppose that at some time t_0 we have $\tilde{\lambda}_{t_0} = \lambda(\theta_{t_0})$. If $\nabla\lambda(\theta_{t_0}) = 0$, then we are at an equilibrium point of the differential equations, and we have $\tilde{\lambda}_t = \lambda(\theta_t)$ for all subsequent times. Otherwise, if $\nabla\lambda(\theta_{t_0}) \neq 0$, then $\dot{\theta}_{t_0} = \nabla\lambda(\theta_{t_0})$, and $\dot{\lambda}(\theta_{t_0}) > 0$. At the same time, we have $\dot{\tilde{\lambda}}_{t_0} = 0$, and this implies that $\tilde{\lambda}_t < \lambda(\theta_t)$ for t slightly larger than t_0. The validity of the claim (20) follows. As long as $\tilde{\lambda}_t \leq \lambda(\theta_t)$, $\tilde{\lambda}_t$ is nondecreasing and since it is bounded, it must converge.

Suppose now that the initial conditions satisfy $\tilde{\lambda}_0 > \lambda(\theta_0)$. As long as this condition remains true, $\tilde{\lambda}_t$ is nonincreasing. There are two possibilities. If this condition remains true for all times, then $\tilde{\lambda}_t$ converges. If not, then there exists a time t_0 such that $\tilde{\lambda}_{t_0} = \lambda(\theta_{t_0})$, which takes us back to the previously considered case.

Having concluded that $\tilde{\lambda}_t$ converges, we can use Eq. (19) to argue that $\lambda(\theta_t)$ must also converge to the same limit. Then, in the limit, θ_t evolves according to $\dot{\theta}_t = \nabla\lambda(\theta_t)$, from which it follows that $\nabla\lambda(\theta_t)$ must go to zero.

We now comment on the nature of a rigorous proof. There are two common approaches for proving the convergence of stochastic approximation methods. One method relies on the existence of a suitable Lyapunov function and a martingale argument. In our context, $\lambda(\theta)$ could play such a role. However, as long as $\tilde{\lambda}_m \neq \lambda(\theta_m)$, our method cannot be expressed as a stochastic gradient algorithm and this approach does not go through. (Furthermore, it is unclear whether another Lyapunov function would do.) The second proof method, the so-called ODE approach, shows that the trajectories followed by the algorithm converge to the trajectories of a corresponding deterministic ODE, e.g., the ODE given by Eqs. (18)-(19). This line of analysis generally requires the iterates to be bounded functions of time. In our case, such a boundedness property is not guaranteed to hold. For example, if θ stands for the weights of a neural network, it is certainly possible that certain "neurons" asymptotically saturate, and the corresponding weights drift to infinity. We therefore need a line of argument specially tailored to our particular algorithm. In rough terms, it proceeds along the same lines as the above provided deterministic analysis, except that we must also ensure that the stochastic terms are not significant.

3.4 Implementation Issues

For systems involving a large state space, as is the case in many applications, the interval between visits to the state i^* can be large. Consequently,

(a) the parameter vector θ gets updated only infrequently;

(b) the estimate $F_m(\theta)$ can have a large variance.

In the following, we will address these two issues and propose two modified versions: one which updates θ at every time step, and one which reduces the variance of the updates.

4 An Algorithm that Updates at Every Time Step

In this section, we develop an algorithm which updates the parameter vector θ at every time step. We start by indicating an economical way of computing the update direction $F_m(\theta, \tilde{\lambda})$. This will allow us to break $F_m(\theta)$ into a sum of incremental updates carried out at each time step.

Taking into account that $\tilde{v}_{i_{t_m}}(\theta, \tilde{\lambda}) = 0$, Eq. (8) becomes

$$\begin{aligned} F_m(\theta, \tilde{\lambda}) &= \sum_{n=t_m+1}^{t_{m+1}-1} \tilde{v}_{i_n}(\theta, \tilde{\lambda}) L_{i_{n-1}i_n}(\theta) + \sum_{n=t_m}^{t_{m+1}-1} \nabla g_{i_n}(\theta) \\ &= \sum_{n=t_m+1}^{t_{m+1}-1} \left(\nabla g_{i_n}(\theta) + L_{i_{n-1}i_n}(\theta) \sum_{k=n}^{t_{m+1}-1} (g_{i_k}(\theta) - \tilde{\lambda}) \right) + \nabla g_{i^*}(\theta) \\ &= \sum_{k=t_m+1}^{t_{m+1}-1} \left(\nabla g_{i_k}(\theta) + (g_{i_k}(\theta) - \tilde{\lambda}) \sum_{n=t_m+1}^{k} L_{i_{n-1}i_n}(\theta) \right) + \nabla g_{i^*}(\theta) \\ &= \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m+1}-1} \Big(\nabla g_{i_k}(\theta) + (g_{i_k}(\theta) - \tilde{\lambda}) z_k \Big), \end{aligned}$$

where

$$z_k = \sum_{n=t_m+1}^{k} L_{i_{n-1}i_n}(\theta) = \sum_{n=t_m+1}^{k} \frac{\nabla p_{i_{n-1}i_n}(\theta)}{p_{i_{n-1}i_n}(\theta)}, \qquad k = t_m+1, \ldots, t_{m+1}-1,$$

is a vector (of the same dimension as θ) that becomes available at time k. It can be updated recursively, with

$$z_{t_m} = 0, \tag{21}$$

and

$$z_{k+1} = z_k + L_{i_k i_{k+1}}(\theta), \qquad k = t_m, \ldots, t_{m+1}-2. \tag{22}$$

We note that z_k is the likelihood ratio derivative that commonly arises in likelihood ratio gradient estimation [Gly87, GG92].

The preceding formulas suggest the following algorithm which updates θ at every time step. At a typical time k, the state is i_k, and the values of θ_k, z_k, and $\tilde{\lambda}_k$ are available from the previous iteration. We update θ and $\tilde{\lambda}$ according to

$$\begin{aligned} \theta_{k+1} &= \theta_k + \gamma_k \Big(\nabla g_{i_k}(\theta_k) + (g_{i_k}(\theta_k) - \tilde{\lambda}_k) z_k \Big), \\ \tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta \gamma_k (g_{i_k}(\theta_k) - \tilde{\lambda}_k). \end{aligned}$$

We then simulate a transition to the next state i_{k+1} according to the transition probabilities $p_{ij}(\theta_{k+1})$, and finally update z by letting

$$z_{k+1} = \begin{cases} 0, & \text{if } i_{k+1} = i^*, \\ z_k + L_{i_k i_{k+1}}(\theta_k), & \text{otherwise.} \end{cases}$$

In order to implement the algorithm, on the basis of the above equations, we only need to maintain in memory $2K+1$ scalars, namely $\tilde{\lambda}$, and the vectors θ, z.

To prove convergence of this version of the algorithm, we have to strengthen Assumption 1 of Section 2. Assumption 1 states that for every fixed parameter θ, we will eventually reach the state i^*. Here, we need to make sure that this will remain so, even if θ keeps changing; see [Mar98] for further discussion of this assumption.

Assumption 5 *There exist a state $i^* \in S$ and a positive integer N_0, such that, for every state $i \in S$ and every collection $\{P_1, \ldots, P_{N_0}\}$ of N_0 matrices in the set $\overline{\mathcal{P}}$, we have*

$$\sum_{n=1}^{N_0} [\Pi_{l=1}^{n} P_l]_{ii^*} > 0.$$

We also impose an additional condition on the stepsizes.

Assumption 6 *The stepsizes γ_k are nonincreasing. Furthermore, there exists a positive integer p and a positive scalar A such that*

$$\sum_{k=n}^{n+t} (\gamma_n - \gamma_k) \leq A t^p \gamma_n^2, \qquad \forall\, n, t > 0.$$

Assumption 6 is satisfied, for example, if we let $\gamma_k = 1/k$. With this choice, and if we initialize $\tilde{\lambda}$ to zero, it is easily verified that $\tilde{\lambda}_k$ is equal to the average reward obtained in the first k transitions.

We have the following convergence result, which is proved in Appendix B.

Proposition 4 *Let Assumptions 1-6 hold, and let $\{\theta_k\}$ be the sequence of parameter vectors generated by the above described algorithm. Then, $\lambda(\theta_k)$ converges and*

$$\lim_{k \to \infty} \nabla \lambda(\theta_k) = 0,$$

with probability 1.

The algorithm of this section is similar to the algorithm of the preceding one, except that θ is continually updated in the course of a regenerative cycle. Because of the diminishing stepsize, these incremental updates are asymptotically negligible and the difference between the two algorithms is inconsequential. Given that the algorithm of the preceding section converges, Proposition 4 is hardly surprising. The technique in our convergence proof use is similar to the one in [CR94]. However, mapped into the context of parameterized Markov reward processes, [CR94] assumes that the transition probabilities $p_{ij}(\theta)$ are independent of θ (the one-stage rewards $g_i(\theta)$ can still depend on θ). The situation here is more general and a separate proof is needed.

5 An Algorithm that may Reduce the Variance

When the length of a regeneration cycle is large, the vector z_k will also become large before it is reset to zero, resulting in high variance for the updates. (This is a generic difficulty associated with likelihood ratio methods.) For this reason, it may be preferable to introduce a forgetting factor $\alpha \in (0,1)$ and update z_k according to

$$z_{k+1} = \begin{cases} 0, & \text{if } i_{k+1} = i^*, \\ \alpha z_k + L_{i_k i_{k+1}}(\theta_k), & \text{otherwise.} \end{cases}$$

This modification, which resembles the algorithm introduced in [JSJ95], can reduce the variance of a typical update, but introduces a new bias in the update direction. Given that gradient-type methods are fairly robust with respect to small biases, this modification may result in improved practical performance; see the numerical results in Section 7.

Similar to [JSJ95], this modified algorithm can be justified if we approximate the differential reward with

$$v_i(\theta) \approx E_\theta \left[\sum_{k=0}^{T} \alpha^k (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = i \right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ (which is increasingly accurate as $\alpha \uparrow 1$), use the estimate

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{T} \alpha^k \left(g_{i_k}(\theta) - \tilde{\lambda} \right),$$

instead of Eq. (7), and then argue similar to Section 3. The analysis of this algorithm is carried out in [Mar98] and, with less detail, in [MT99].

6 Markov Decision Processes

In this section, we indicate how to apply our methodology to Markov decision processes. An important feature, which is evident from the formulas provided at the end of this section, is that the algorithm is "model-free:" as long as the process can be simulated or is available for observation, explicit knowledge of the transition probabilities $p_{ij}(\theta)$ is not needed.

We consider a Markov Decision Processes [Ber95a, Put94] with finite state space $S = \{1, \ldots, N\}$ and finite action space $U = \{1, \ldots, M\}$. At any state i, the choice of a control action $u \in U$ determines the probability $p_{ij}(u)$ that the next state is j. The immediate reward at each time step is of the form $g_i(u)$, where i and u is the current state and action, respectively.

A (*randomized*) *policy* is defined as a mapping

$$\mu : S \mapsto [0,1]^M,$$

with components $\mu_u(i)$ such that

$$\sum_{u \in U} \mu_u(i) = 1, \qquad \forall\, i \in S.$$

Under a policy μ, and whenever the state is equal to i, action u is chosen with probability $\mu_u(i)$, independent of everything else. If for every state i there exists a single u for which $\mu_u(i)$ is positive (and, therefore, unity), we say that we have a *pure* policy.

For problems involving very large state spaces, it is impossible to even describe an arbitrary pure policy μ, since this requires a listing of the actions corresponding to each state. This leads us to consider policies described in terms of a parameter vector $\theta = (\theta_1, \ldots, \theta_K)$, whose dimension K is tractably small. We are interested in a method that performs small incremental updates of the parameter θ. A method of this type can work only if the policy has a smooth dependence on θ, and this is the main reason why we choose to work with randomized policies.

We allow θ to be an arbitrary element of $\Re^K$. With every $\theta \in \Re^K$, we associate a randomized policy $\mu(\theta)$, which at any given state i chooses action u with probability $\mu_u(i,\theta)$. Naturally, we require that every $\mu_u(i,\theta)$ be nonnegative and that $\sum_{u\in U} \mu_u(i,\theta) = 1$. Note that the resulting transition probabilities are given by

$$p_{ij}(\theta) = \sum_{u\in U} \mu_u(i,\theta) p_{ij}(u), \tag{23}$$

and the expected reward per stage is given by

$$g_i(\theta) = \sum_{u\in U} \mu_u(i,\theta) g_i(u).$$

The objective is to maximize the average reward under policy $\mu(\theta)$, which is denoted by $\lambda(\theta)$. This is a special case of the framework of Section 2. We now discuss the various assumptions introduced in earlier sections.

In order to satisfy Assumption 1, it suffices to assume that there exists a state i^* which is recurrent under every pure policy, a property which is satisfied in many interesting problems. In order to satisfy Assumption 2, it suffices to assume that the policy has a smooth dependence on θ; in particular, that $\mu_u(i,\theta)$ is twice differentiable (in θ) and has bounded first and second derivatives. Finally, Assumption 3 is implied by the following condition.

Assumption 7 *For every i and u, there exists a bounded function $L_u(i,\theta)$ such that*

$$\nabla \mu_u(i,\theta) = \mu_u(i,\theta) L_u(i,\theta), \qquad \forall\ \theta.$$

This assumption can be satisfied in a number of ways:

(a) Consider a smoothly parametrized function $r_u(i,\theta)$ that maps state-action pairs (i,u) to real numbers, and suppose that

$$\mu_u(i,\theta) = \frac{\exp(r_u(i,\theta))}{\sum_v \exp(r_v(i,\theta))}.$$

Assumption 7 is satisfied once we assume that $r_u(i,\theta)$ has bounded first and second derivatives. This particular form is common in the neural network literature: the $r_u(i,\theta)$ are the outputs of a neural network with input (i,u) and internal weights θ, and an action u is selected by a randomized "soft maximum."

(b) We may artificially restrict to policies for which there exists some $\epsilon > 0$ such that

$$\mu_u(i,\theta) \geq \epsilon, \qquad \forall\, i, u, \theta.$$

Such policies introduce a minimal amount of "exploration," and ensure that every action will be tried infinitely often. This can be beneficial because the available experience with simulation-based methods for Markov decision processes indicates that performance can substantially degrade in the absence of exploration: a method may stall within a poor set of policies for the simple reason that the actions corresponding to better policies have not been sufficiently explored.

Since $\sum_{u\in U} \mu_u(i,\theta) = 1$ for every θ, we have $\sum_{u\in U} \nabla\mu_u(i,\theta) = 0$, and

$$\nabla g_i(\theta) = \sum_{u\in U} \nabla\mu_u(i,\theta)\big(g_i(u) - \lambda(\theta)\big).$$

Furthermore,

$$\sum_{j\in S} \nabla p_{ij}(\theta) v_j(\theta) = \sum_{j\in S}\sum_{u\in U} \nabla\mu_u(i,\theta) p_{ij}(u) v_j(\theta).$$

Using these relations in the formula for $\nabla\lambda(\theta)$ provided by Proposition 1, and after some rearranging, we obtain

$$\nabla\lambda(\theta) = \sum_{i\in S}\sum_{u\in U} \pi_i(\theta)\mu_u(i,\theta) q_{i,u}(\theta) \frac{\nabla\mu_u(i,\theta)}{\mu_u(i,\theta)},$$

where

$$\begin{aligned} q_{i,u}(\theta) &= \big(g_i(u) - \lambda(\theta)\big) + \sum_{j\in S} p_{ij}(u) v_j(\theta) \\ &= E_\theta\left[\sum_{k=0}^{T-1} \big(g_{i_k}(u_k) - \lambda(\theta)\big) \mid i_0 = i, u_0 = u\right], \end{aligned}$$

and where i_k and u_k is the state and control at time k. Thus, $q_{i,u}(\theta)$ is the differential reward if control action u is first applied in state i, and policy $\mu(\theta)$ is followed thereafter. It is the same as Watkins' Q-factor [Wat89], suitably modified for the average reward case.

From here on, we can proceed as in Section 3 and obtain an algorithm that updates θ at the times t_m that state i^* is visited. The form of the algorithm is

$$\begin{aligned} \theta_{m+1} &= \theta_m + \gamma_m F_m(\theta_m, \tilde\lambda_m), \\ \tilde\lambda_{m+1} &= \tilde\lambda_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} \big(g_{i_n}(u_n) - \tilde\lambda_m\big), \end{aligned}$$

where

$$F_m(\theta_m, \tilde\lambda_m) = \sum_{n=t_m}^{t_{m+1}-1} \tilde q_{i_n,u_n} \frac{\nabla\mu_{u_n}(i_n,\theta_m)}{\mu_{u_n}(i_n,\theta_m)},$$

and

$$\tilde q_{i_n,u_n} = \sum_{k=n}^{t_{m+1}-1} \big(g_{i_k}(u_k) - \tilde\lambda_m\big).$$

Similar to Section 4, an on-line version of the algorithm is also possible. The convergence results of Sections 3 and 4 remain valid, with only notation changes in the proof.

7 Experimental Results for an Admission Control Problem

In this section, we describe some numerical experiments with a call admission control problem. This problem arises when a service provider with limited resources (bandwidth) has to accept or reject incoming calls of several types, while taking into account current congestion. The objective is to maximize long-term average revenue. More details on the experiments reported here can be found in [Mar98].

7.1 Problem Formulation

Consider a communication link with a total bandwidth of B units, which supports a finite set $\{1, 2, \ldots, M\}$ of different service types. Each service type is characterized by its bandwidth requirement $b(m)$, its call arrival rate $\alpha(m)$, and its average holding time $1/\beta(m)$, where we assume that the calls (customers) arrive according to independent Poisson processes, and that the holding times are exponentially (and independently) distributed. When a new customer requests a connection, we can decide to reject, or, if enough bandwidth is available, to accept the customer. Once accepted, a customer of class m seizes $b(m)$ units of bandwidth for the duration of the call. Whenever a call of service type m gets accepted, we receive an immediate reward of $c(m)$ units. The reward $c(m)$ can be interpreted as the price customers of service type m are paying for using $b(m)$ units of bandwidth of the link for the duration of the call. The goal of the link provider is to exercise call admission control in a way that maximizes the long term revenue.

Using uniformization, the problem is easily transformed into a discrete-time Markov decision process. The state can be taken to be of the form $i = (s(1), \ldots, s(M), \omega)$, where $s(m)$ denotes the number of active calls of type m, and ω indicates the type of event that triggers the next transition (a departure or arrival of a call, together with the type of the call). If ω indicates an arrival of a call of class m and if there is enough free bandwidth to accommodate it, there are two available decisions, namely, u_a (accept) or u_r (reject).

We consider randomized policies of the following form. If there is an arrival of a call of class m, we accept it with probability

$$\mu_{u_a}(i, \theta) = \frac{1}{1 + \exp(s \cdot b - \theta(m))}.$$

Here, $s \cdot b = \sum_m s(m)b(m)$ is the currently occupied bandwidth and $\theta(m)$, the mth component of θ, is a policy parameter. Note that

$$\mu_{u_a}(i, \theta) \geq 0.5 \quad \text{if and only if} \quad s \cdot b \leq \theta(m).$$

Thus, $\theta(m)$ can be interpreted as a "fuzzy" threshold on system occupancy, which determines whether type m calls are to be admitted or rejected.

In our experiments, we consider a link with a total bandwidth of 10 units, and three different call types. The detailed parameters are given in Table 1. The number of link configuarions (i.e., possible choices of s that do not violate the link capacity constraint) turns out to be 286.

Any state (s, ω) in which $s = (0, \ldots, 0)$ and ω corresponds to an arrival of a new call, is recurrent under any policy, and can therefore play the role of i^*. Even for moderately loaded systems, the time between consecutive visits to such a state can be extremely large, in which case we expect our methods to be very slow.

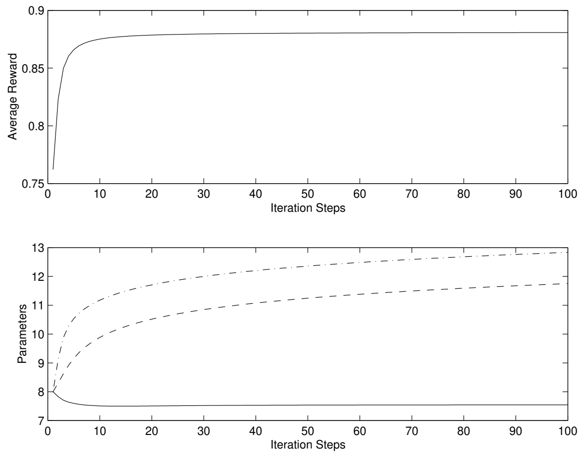

Figure 1: Parameter vectors and average rewards (computed exactly) of the corresponding admission control policies, obtained by the idealized gradient algorithm. The solid, dashed, and dash-dot line correspond to the threshold values θ_1, θ_2, and θ_3, associated with service types 1, 2, and 3, respectively.

7.2 Results

Optimal Policy

Since the state space is relatively small, an optimal policy can be obtained using standard dynamic programming methods [Ber95a]. The optimal average reward is equal to 0.8868. (Of course, the optimal average reward within the restricted class of randomized policies that we have introduced earlier will have to be less than that.) Under an optimal policy, customers of type 2 and 3 are accepted whenever there is available bandwidth. Customers of type 1 are accepted only if the currently used bandwidth does not exceed 7.

Idealized Gradient Algorithm

For such a small example, we can numerically calculate $\nabla\lambda(\theta)$, for any given θ, which allows us to implement the idealized algorithm

$$\theta_{k+1} = \theta_k + \gamma_k \nabla\lambda(\theta_k)$$

of Section 2.2. The evolution of this algorithm, starting with $\theta_0 = (8, 8, 8)$, is shown in Figure 1. After 100 iterations, we have $\theta_{100} = (7.5459, 11.7511, 12.8339)$, and the corresponding average reward is equal to 0.8808, which is very close to optimal. The probabilities of accepting a new call are given in Figure 2.

Simulation-Based Algorithm that Updates at Every Time Step

We implemented a streamlined version of the algorithm given Section 4, where we reset the vector z_k not only at visits to the recurrent state i^*, but at visits to all states $i = (s, \omega)$ for which $s = (0, ..., 0)$. A justification of this modification, which does not change the mean

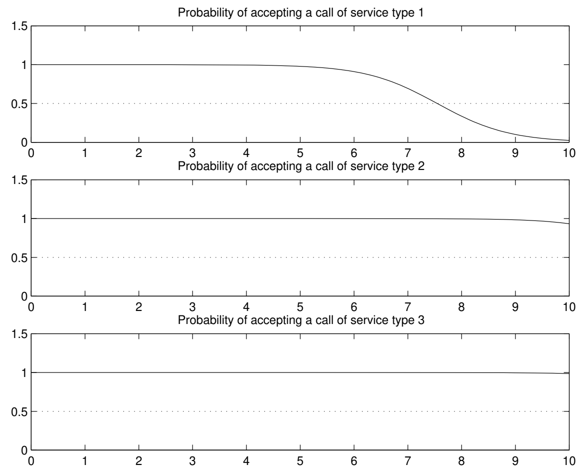

Figure 2: Probabilities of accepting a new call, as a function of the already occupied bandwidth, under the control policy associated with the parameter vector $(7.5459, 11.7511, 12.8339)$ obtained by the idealized gradient algorithm.

direction of the update, is given in [Mar98]. We started with the same initial parameter $\theta = (8, 8, 8)$, and the initial estimate of the average reward $\tilde{\lambda}_0$ was set to 0.78. The scaling factor in the update equation was chosen to be $\eta = 0.1$. The corresponding trajectories of the parameter vectors and average reward are given in Figure 3. We have the following observations:

1. The algorithm makes rapid progress in the beginning, improving the average reward from 0.78 to 0.87 within the first $1 \cdot 10^6$ iteration steps.

2. After $1 \cdot 10^6$ iterations, the algorithm makes only slow progress obtaining after $8 \cdot 10^6$ iterations the parameter vector

$$\theta_{8 \cdot 10^6} = (7.3540, 10.6850, 11.7713)$$

   which corresponds to an admission control policy with an average reward of 0.8789. This average reward still slightly below the average reward of 0.8808 obtained by the idealized gradient algorithm.

3. The fluctuations in the estimate of the average reward remain small and the performance of the control policies never deteriorates.

This behavior is not unlike the idealized algorithm (see Figure 1), where the average reward improves rapidly in the beginning, but only slowly in the later iterations.

The probabilities of accepting a new call under the control policy obtained with the simulation-based algorithm are given in Figure 4.

Modified Simulation-Based Algorithm

We finally consider the modified algorithm of Section 5, using a forgetting factor of $\alpha = 0.99$. As expected, it makes much faster progress; see Figure 5.

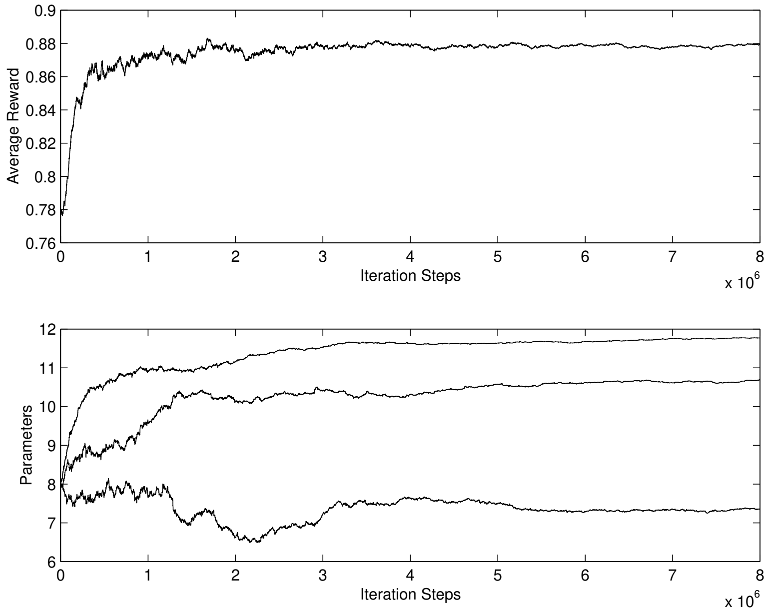

Figure 3: Parameter vectors, and estimates of the average reward, obtained by the simulation-based algorithm. The scaling factor for the iteration steps is 10^6.

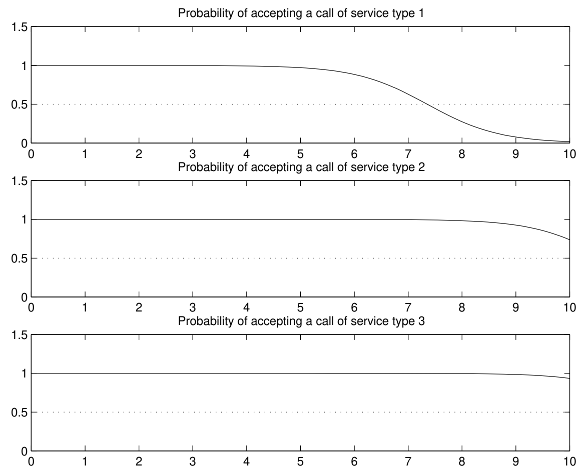

Figure 4: Probabilities of accepting a new call, given as a function of the used bandwidth on the link, under the control policy associated with the parameter vector $(7.3540, 10.6850, 11.7713)$ obtained by the simulation-based algorithm.

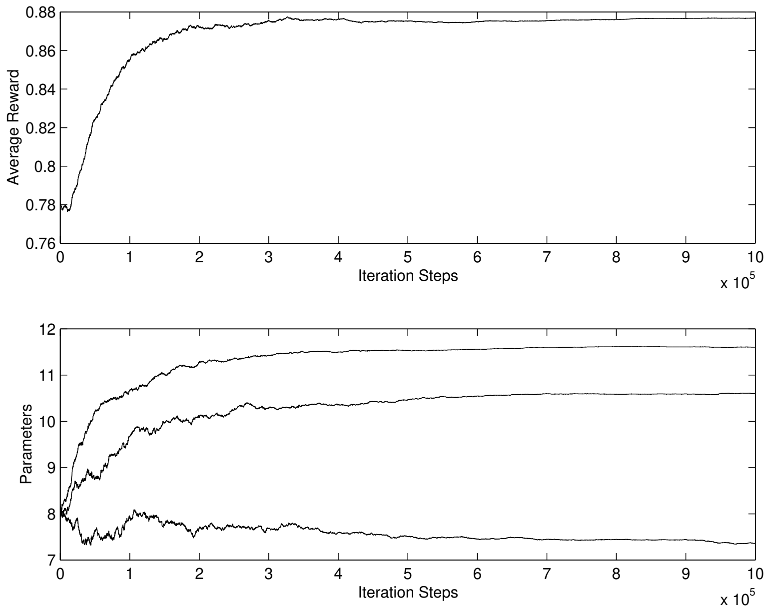

Figure 5: Parameter vectors, and estimates of the average reward, obtained by modified simulation-based algorithm using a discount factor $\alpha = 0.99$. The scaling factor for the iteration steps is 10^5.

After 10^6 iterations, we obtain a parameter vector of $\theta = (7.3553, 10.6034, 11.6073)$ and an average reward of 0.8785, which is essentially the same as for the unmodified algorithm after $8 \cdot 10^6$ iterations. Thus, the use of a forgetting factor speeds up convergence by an order of magnitude, while introducing a negligible bias.

8 Conclusions

We have presented a simulation-based method for optimizing a Markov Reward Process whose transition probabilities depend on a parameter vector θ, or a Markov Decision Process in which we restrict to a parametric set of randomized policies. The method involves simulation of a single sample path. Updates can be carried out either when the recurrent state i^* is visited, or at every time step. We have also proposed a modified, possibly more practical method, and have provided some encouraging numerical results.

Regarding further research, there is a need for more computational experiments in order to delineate the class of practical problems for which this methodology is useful. In particular, further analysis and experimentation is needed for the modified on-line algorithm of Section 5. In addition, the possibility of combining such methods with "learning" (function approximation) of the differential reward function needs to be explored. On the technical side, it may be possible to extend the results to the case of an infinite state space, and to relate the speed of convergence to the mixing time of the underlying Markov chains.

Acknowledgments

We are grateful to Oliver Mihatsch for suggesting the form of Assumption 3, and the referees for many suggestions for improving the paper.

References

[Ber95a] D. P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific, Belmont, MA, 1995.

[Ber95b] D. P. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, MA, 1995.

[Bor97] V.S. Borkar, "Stochastic Approximation with Two Time Scales," Systems and Control Letters, Vol. 29, pp. 291-294, 1997.

[BSA83] A. Barto, R. Sutton, and C. Anderson, "Neuron-Like Elements that Can Solve Difficult Learning Control Problems," IEEE Trans. on Systems, Man and Cybernetics, Vol. 13, pp. 835-846, 1983.

[BT96] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, MA, 1996.

[BT97] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient Convergence in Gradient Methods," Lab. for Info. and Decision Systems Report LIDS-P-2301, Massachusetts Institute of Technology, Cambridge, MA, 1997.

[CC97] X. R. Cao and H. F. Chen, "Perturbation Realization, Potentials, and Sensitivity Analysis of Markov Processes," IEEE Transactions on Automatic Control, Vol. 42, pp. 1382-1393, 1997.

[CR94] E. K. P. Chong and P. J. Ramadage, "Stochastic Optimization of Regenerative Systems Using Infinitesimal Perturbation Analysis," IEEE Trans. on Automatic Control, Vol. 39, pp. 1400-1410, 1994.

[CW98] X. R. Cao and Y. W. Wan, "Algorithms for Sensitivity Analysis of Markov Systems through Potentials and Perturbation Realization," IEEE Trans. on Control Systems Technology, Vol. 6, pp. 482-494, 1998.

[Del96] B. Delyon, "General Results on the Convergence of Stochastic Algorithms," IEEE Trans. on Automatic Control, Vol. 41, pp. 1245-1255, 1996.

[FH94] M. C. Fu and J. Hu, "Smoothed Perturbation Analysis Derivative Estimation for Markov Chains," Operations Research Letters, Vol. 15, pp. 241-251, 1994.

[Gla91] P. Glasserman, Gradient Estimation Via Perturbation Analysis, Kluwer Academic, Boston, 1991.

[GG92] P. Glasserman and P. W. Glynn, "Gradient Estimation for Regenerative Processes," Proceedings of the 1992 Winter Simulation Conference, pp. 280-288, 1992.

[Gly86] P. W. Glynn, "Stochastic Approximation for Monte Carlo Optimization," Proceedings of the 1986 Winter Simulation Conference, pp. 285-289, 1986.

[Gly87] P. W. Glynn, "Likelihood Ratio Gradient Estimation: an Overview," Proceedings of the 1987 Winter Simulation Conference, pp. 366-375, 1987.

[GI89] P. W. Glynn and D. L. Iglehart, "Importance Sampling for Stochastic Simulation," Management Science, Vol. 35, No. 11, pp. 1367-1392, 1989.

[HC91] Y. C. Ho and X. R. Cao, Perturbation Analysis of Discrete Event Systems, Kluwer Academic Publisher, Boston, MA, 1991.

[JSJ95] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems," Advances in Neural Information Processing Systems, Vol. 7, pp. 345-352, Morgan Kaufman, San Francisco, CA, 1995.

[KB98] V. R. Konda and V. S. Borkar, "Actor-Critic Like Learning Algorithms for Markov Decision Processes," submitted.

[KT99] V. R. Konda and J. N. Tsitsiklis, "Actor-Critic Algorithms," to appear in the Proceedings of the 1999 Neural Information Processing Systems Conference.

[LEc90] P. L'Ecuyer, "A Unified View of the IPA, SF, and LR Gradient Estimation Techniques," Management Science, Vol. 36, No. 11, pp. 1364-1383, 1990.

[Lju77] L. Ljung, "Analysis of Recursive Stochastic Algorithms," IEEE Trans. on Automatic Control, Vol. 22, pp. 551-575, 1977.

[Mar98] P. Marbach, "Simulation-Based Optimization of Markov Decision Processes," doctoral thesis, Dept. of EECS, MIT, Cambridge, MA, 1998.

[MT99] P. Marbach and J. N. Tsitsiklis, "Simulation-Based Optimization of Markov Reward Processes: Implementation Issues," in Proceedings of the 1999 IEEE Conference on Decision and Control, Phoenix, AZ, December 1999.

[Put94] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, New York, NY, 1994.

[SB98] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

[SMS99] R. S. Sutton, D. McAllester, S. Singh and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," to appear in the Proceedings of the 1999 Neural Information Processing Systems Conference.

[Tes92] G. J. Tesauro, "Practical Issues in Temporal Difference Learning," Machine Learning, Vol. 8, pp. 257-277, 1992.

[TH95] V. Tresp and R. Hofmann, "Missing and Noisy Data in Nonlinear Time-Series Prediction," in Neural Networks for Signal Processing, S. F. Girosi, J. Mahoul, E. Manolakos and E. Wilson, eds., IEEE Signal Processing Society, New York, New York, 1995, pp. 1-10.

[Wat89] C. Watkins, "Learning from Delayed Rewards," Ph.D. Thesis, Cambridge University, UK, 1989.

A Proof of Proposition 3

In this appendix, we prove convergence of the algorithm

$$\begin{aligned}\theta_{m+1} &= \theta_m + \gamma_m F_m(\theta_m, \tilde{\lambda}_m),\\ \tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m),\end{aligned}$$

where

$$F_m(\theta_m, \tilde{\lambda}_m) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{v}_{i_n}(\theta_m, \tilde{\lambda}_m) L_{i_{n-1}i_n}(\theta_m) + \nabla g_{i_n}(\theta_m)\right),$$

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} \left(g_{i_k}(\theta) - \tilde{\lambda}\right), \qquad n = t_m + 1, \ldots, t_{m+1} - 1,$$

and

$$\tilde{v}_{i_{t_m}}(\theta, \tilde{\lambda}) = 0.$$

For notational convenience, we define the augmented parameter vector $r_m = (\theta_m, \tilde{\lambda}_m)$, and write the update equations in the form

$$r_{m+1} = r_m + \gamma_m H_m(r_m),$$

where

$$H_m(r_m) = \begin{bmatrix} F_m(\theta_m, \tilde{\lambda}_m) \\ \eta \displaystyle\sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m) \end{bmatrix}. \tag{24}$$

Let

$$\mathcal{F}_m = \{\theta_0, \tilde{\lambda}_0, i_0, i_1, \ldots, i_{t_m}\}$$

stand for the history of the algorithm up to and including time t_m. Using Proposition 2 and Eq. (17), we have

$$E[H_m(r_m) \mid \mathcal{F}_m] = h(r_m),$$

where

$$h(r) = \begin{bmatrix} E_\theta[T]\nabla\lambda(\theta) + G(\theta)(\lambda(\theta) - \tilde{\lambda}) \\ \eta E_\theta[T](\lambda(\theta) - \tilde{\lambda}) \end{bmatrix}.$$

We then rewrite the algorithm in the form

$$r_{m+1} = r_m + \gamma_m h(r_m) + \varepsilon_m, \tag{25}$$

where

$$\varepsilon_m = \gamma_m (H_m(r_m) - h(r_m))$$

and note that

$$E[\varepsilon_m \mid \mathcal{F}_m] = 0.$$

The proof rests on the fact that ε_m is "small," in a sense to be made precise, which will then allow us to mimic the heuristic argument of Section 3.3.

A.1 Preliminaries

In this subsection, we establish a few useful bounds and characterize the behavior of ε_m.

Lemma 2

(a) There exist constants C and $\rho < 1$ such that

$$P_\theta(T = k) \le C\rho^k, \qquad \forall\ k,\ \theta,$$

where the subscript θ indicates that we are considering the distribution of the length of the regeneration cycle $T_m = t_{m+1} - t_m$ under a particular choice of θ. In particular, $E_\theta[T]$ and $E_\theta[T^2]$ are bounded functions of θ.

(b) The function $G(\theta)$ is well defined and bounded.

(c) The sequence $\tilde{\lambda}_m$ is bounded, with probability 1.

(d) The sequence $h(r_m)$ is bounded, with probability 1.

Proof:

(a) For any transition probability matrix $P \in \overline{\mathcal{P}}$, and because of Assumption 1, the probability of reaching i^* in N steps is bounded below by some positive $\epsilon(P)$, for every initial state. Furthermore, $\epsilon(P)$ can be taken to be a continuous function of P. Using the compactness of $\overline{\mathcal{P}}$, we have $\epsilon^* = \min_{P \in \overline{\mathcal{P}}} \epsilon(P) > 0$, and the result follows with $\rho = (\epsilon^*)^{1/N}$.

(b) Note that

$$E_\theta \left[\sum_{n=t_m+1}^{t_{m+1}-1} \left\| (t_{m+1} - n) L_{i_{n-1} i_n}(\theta) \right\| \right] \le C E_\theta[T^2],$$

where C is a bound on $\|L_{ij}(\theta)\|$ (cf. Assumption 3). The right-hand side is bounded by the result of part (a). It follows that the expectation defining $G(\theta)$ exists and is a bounded function of θ.

(c) Using Assumption 4 and part (a) of this lemma, we obtain

$$E \left[\sum_{m=1}^{\infty} \gamma_m^2 (t_{m+1} - t_m)^2 \right] < \infty,$$

which implies that $\gamma_m(t_{m+1} - t_m)$ converges to zero, with probability 1. Note that

$$\tilde{\lambda}_{m+1} \le (1 - \gamma_m(t_{m+1} - t_m))\tilde{\lambda}_m + \gamma_m(t_{m+1} - t_m)C,$$

where C is an upper bound on $g_i(\theta)$. For large enough m, we have $\gamma_m(t_{m+1} - t_m) \le 1$, and $\tilde{\lambda}_{m+1} \le \max\{\tilde{\lambda}_m, C\}$, from which it follows that the sequence $\tilde{\lambda}_m$ is bounded above. By a similar argument, the sequence $\tilde{\lambda}_m$ is also bounded below.

(d) Consider the formula that defines $h(r)$. Parts (a) and (b) show that $E_{\theta_m}[T]$ and $G(\theta_m)$ are bounded. Also, $\lambda(\theta_m)$ is bounded since the $g_i(\theta)$ are bounded (Assumption 2). Furthermore, $\nabla\lambda(\theta_m)$ is bounded, by Lemma 1. Using also part (c) of this lemma, the result follows. $\square$

Lemma 3 *There exists a constant C (which is random but finite with probability 1) such that*

$$E[\|\varepsilon_m\|^2 \mid \mathcal{F}_m] \le C\gamma_m^2, \qquad \forall\, m,$$

and the series $\sum_m \varepsilon_m$ converges with probability 1.

Proof: Recall that $g_{i_m}(\theta_m)$ and $\tilde{\lambda}_m$ are bounded with probability 1 (Assumption 2 and Lemma 2(c)). Thus, for $n = t_m, \ldots, t_{m+1}-1$, we have $|\tilde{v}_{i_n}(\theta, \tilde{\lambda})| \le C(t_{m+1}-t_m)$, for some constant C. Using this bound in the definition of $F_m(\theta_m, \tilde{\lambda}_m)$, we see that for almost all sample paths, we have

$$\|F_m(\theta_m, \tilde{\lambda}_m)\| \le C(t_{m+1} - t_m)^2,$$

for some new constant C. Using Lemma 2(a), the conditional variance of $F_m(\theta_m, \tilde{\lambda}_m)$, given $\mathcal{F}_m$, is bounded. Similar arguments also apply to the last component of $H_m(r_m)$. Since $\varepsilon_m = \gamma_m(H_m(r_m) - E[H_m(r_m) \mid \mathcal{F}_m])$, the first statement follows.

Fix a positive integer c and consider the sequence

$$w_n^c = \sum_{m=1}^{\min\{M(c),n\}} \varepsilon_m,$$

where $M(c)$ is the first time m such that $E[\|\varepsilon_m\|^2 \mid \mathcal{F}_m] > c\gamma_m^2$. The sequence w_n^c is a martingale with bounded second moment, and therefore converges with probability 1. This is true for every positive integer c. For (almost) every sample path, there exists some c such that $M(c) = \infty$. After discarding a countable union of sets of measure zero (for each c, the set of sample paths for which w_n^c does not converge), it follows that for (almost) every sample path, $\sum_m \varepsilon_m$ converges. □

We observe the following consequences of Lemma 3. First, ε_m converges to zero with probability 1. Since γ_m also converges to zero and the sequence $h(r_m)$ is bounded, we conclude that

$$\lim_{m\to\infty}(\theta_{m+1}-\theta_m) = 0, \qquad \lim_{m\to\infty}(\lambda(\theta_{m+1})-\lambda(\theta_m)) = 0, \qquad \lim_{m\to\infty}(\tilde{\lambda}_{m+1}-\tilde{\lambda}_m) = 0,$$

with probability 1.

A.2 Convergence of $\tilde{\lambda}_m$ and $\lambda(\theta_m)$

In this subsection, we prove that $\tilde{\lambda}_m$ and $\lambda(\theta_m)$ converge to a common limit. The flow of the proof is similar to the heuristic argument of Section 3.3.

We will be using a few different Lyapunov functions to analyze the behavior of the algorithm in different "regions." The lemma below involves a generic Lyapunov function ϕ and characterizes the changes in $\phi(r)$ caused by the updates

$$r_{m+1} = r_m + \gamma_m h(r_m) + \varepsilon_m.$$

Let $\mathcal{D}_c = \{(\theta, \tilde{\lambda}) \in \Re^{K+1} \mid |\tilde{\lambda}| \le c\}$. We are interested in Lyapunov functions ϕ that are twice differentiable and for which ϕ, $\nabla\phi$, and $\nabla^2\phi$ are bounded on $\mathcal{D}_c$ for every c. Let Φ be the set of all such Lyapunov functions. For any $\phi \in \Phi$, we define

$$\varepsilon_m(\phi) = \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m) \cdot h(r_m),$$

where for any two vectors a, b, we use $a \cdot b$ to denote their inner product.

Lemma 4 *If $\phi \in \Phi$, then the series $\sum_m \varepsilon_m(\phi)$ converges with probability 1.*

Proof: Consider a sample path of the random sequence $\{r_m\}$. Using part (c) of Lemma 2, and after discarding a set of zero probability, there exists some c such that $r_m \in \mathcal{D}_c$ for all m. We use the Taylor expansion of $\phi(r)$ at r_m, and obtain

$$\begin{aligned}
\varepsilon_m(\phi) &= \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m) \cdot h(r_m) \\
&\leq \nabla\phi(r_m) \cdot (r_{m+1} - r_m) + M\|r_{m+1} - r_m\|^2 - \gamma_m \nabla\phi(r_m) \cdot h(r_m) \\
&= \nabla\phi(r_m) \cdot \varepsilon_m + M\|r_{m+1} - r_m\|^2,
\end{aligned}$$

where M is a constant related to the bound on the second derivatives of $\phi(\cdot)$ on the set $\mathcal{D}_c$. A symmetric argument also yields

$$\nabla\phi(r_m) \cdot \varepsilon_m - M\|r_{m+1} - r_m\|^2 \leq \varepsilon_m(\phi).$$

Using the boundedness of $\nabla\phi$ on the set $\mathcal{D}_c$, the same martingale argument as in the proof of Lemma 3 shows that the series $\sum_m \nabla\phi(r_m) \cdot \varepsilon_m$ converges with probability 1. Note that $\|r_{m+1} - r_m\| = \|\gamma_m h(r_m) + \varepsilon_m\|$, which yields

$$\|r_{m+1} - r_m\|^2 \leq 2\gamma_m^2 \|h(r_m)\|^2 + 2\|\varepsilon_m\|^2.$$

The sequence $h(r_m)$ is bounded (Lemma 2) and γ_m^2 is summable (Assumption 4). Furthermore, it is an easy consequence of Lemma 3 that ε_m is also square summable. We conclude that $\|r_{m+1} - r_m\|$ is square summable, and the result follows. □

¿From now on, we will concentrate on a single sample path for which the sequences ε_m and $\varepsilon_m(\phi)$ (for the Lyapunov functions to be considered) are summable. Accordingly, we will be omitting the "with probability 1" qualification.

The next lemma shows that if the error $\tilde{\lambda}_m - \lambda(\theta_m)$ in estimating the average reward is positive but small, then it tends to decrease. The proof uses $\tilde{\lambda} - \lambda(\theta)$ as a Lyapunov function.

Lemma 5 *Let L be such that $\|G(\theta)\| \leq L$ for all θ, and let*

$$\phi(r) = \phi(\theta, \tilde{\lambda}) = \tilde{\lambda} - \lambda(\theta).$$

We have $\phi \in \Phi$. Furthermore, if $0 \leq \tilde{\lambda} - \lambda(\theta) \leq \eta/L^2$, then

$$\nabla\phi(r) \cdot h(r) \leq 0.$$

Proof: The fact that $\phi \in \Phi$ is a consequence of Lemma 1. We now have

$$\nabla\phi(r) \cdot h(r) = -\eta(\tilde{\lambda} - \lambda(\theta))E_\theta[T] - \|\nabla\lambda(\theta)\|^2 E_\theta[T] + (\tilde{\lambda} - \lambda(\theta))\nabla\lambda(\theta) \cdot G(\theta).$$

Using the inequality $|a \cdot b| \leq \|a\|^2 + \|b\|^2$, to bound the last term, and the fact $E_\theta[T] \geq 1$, we obtain

$$\nabla\phi(r) \cdot h(r) \leq -\eta(\tilde{\lambda} - \lambda(\theta)) + L^2(\tilde{\lambda} - \lambda(\theta))^2,$$

which is nonpositive as long as $0 \leq \tilde{\lambda} - \lambda(\theta) \leq \eta/L^2$. □

In the next two lemmas, we establish that if $|\tilde{\lambda}_m - \lambda(\theta_m)|$ remains small during a certain time interval, then $\tilde{\lambda}_m$ cannot decrease by much. We first introduce a Lyapunov function that captures the behavior of the algorithm when $\tilde{\lambda} \approx \lambda(\theta)$.

Lemma 6 *As in Lemma 5, let* L *be such that* $\|G(\theta)\| \leq L$*. Let also*

$$\phi(r) = \phi(\theta, \tilde{\lambda}) = \lambda(\theta) - (L^2/\eta)(\lambda(\theta) - \tilde{\lambda})^2.$$

We have $\phi \in \Phi$*. Furthermore, if* $|\lambda(\theta) - \tilde{\lambda}| \leq \eta/4L^2$*, then*

$$\nabla\phi(r) \cdot h(r) \geq 0.$$

Proof: The fact that $\phi \in \Phi$ is a consequence of Lemma 1. We have

$$\nabla_\theta \phi(\theta, \tilde{\lambda}) = \Big(1 - (2L^2/\eta)(\lambda(\theta) - \tilde{\lambda})\Big)\nabla\lambda(\theta),$$

and

$$\nabla_{\tilde{\lambda}} \phi(\theta, \tilde{\lambda}) = (2L^2/\eta)(\lambda(\theta) - \tilde{\lambda}).$$

Therefore, assuming that $|\lambda(\theta) - \tilde{\lambda}| \leq \eta/4L^2$, and using the Schwartz inequality, we obtain

$$\begin{aligned}
\nabla\phi(r) \cdot h(r) &= \Big(1 - (2L^2/\eta)(\lambda(\theta) - \tilde{\lambda})\Big)\Big(\|\nabla\lambda(\theta)\|^2 E_\theta[T] + (\lambda(\theta) - \tilde{\lambda})G(\theta) \cdot \nabla\lambda(\theta)\Big) \\
&\quad + 2L^2(\lambda(\theta) - \tilde{\lambda})^2 E_\theta[T] \\
&\geq \frac{1}{2}\|\nabla\lambda(\theta)\|^2 - \frac{3}{2}|\lambda(\theta) - \tilde{\lambda}|L\|\nabla\lambda(\theta)\| + 2L^2(\lambda(\theta) - \tilde{\lambda})^2 \\
&\geq 0.
\end{aligned}$$

□

Lemma 7 *Consider the same function* ϕ *as in Lemma 6, and the same constant* L*. Let* α *be some positive scalar smaller than* $\eta/4L^2$*. Suppose that for some integers* n *and* n'*, with* $n' > n$*, we have*

$$|\lambda(\theta_n) - \tilde{\lambda}_n| \leq \alpha, \qquad |\lambda(\theta_{n'}) - \tilde{\lambda}_{n'}| \leq \alpha,$$

and

$$|\lambda(\theta_m) - \tilde{\lambda}_m| \leq \frac{\eta}{4L^2}, \qquad m = n+1, \ldots, n'-1.$$

Then,

$$\tilde{\lambda}_{n'} \geq \tilde{\lambda}_n - 2\alpha\Big((L^2\alpha/\eta) + 1\Big) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi).$$

Proof: Using Lemma 6, we have

$$\nabla\phi(r_m) \cdot h(r_m) \geq 0, \qquad m = n, \ldots, n'-1.$$

Therefore, for $m = n, \ldots, n'-1$, we have

$$\begin{aligned}
\phi(r_{m+1}) &= \phi(r_m) + \gamma_m \nabla\phi(r_m) \cdot h(r_m) + \varepsilon_m(\phi) \\
&\geq \phi(r_m) + \varepsilon_m(\phi),
\end{aligned}$$

and

$$\phi(r_{n'}) \geq \phi(r_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi). \tag{26}$$

Note that $|\phi(r_n) - \tilde{\lambda}_n| \leq (L^2\alpha^2/\eta) + \alpha$, and $|\phi(r_{n'}) - \tilde{\lambda}_{n'}| \leq (L^2\alpha^2/\eta) + \alpha$. Using these inequalitites in Eq. (26), we obtain the desired result. □

Lemma 8 *We have* $\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0$.

Proof: Suppose that the result is not true, and we will derive a contradiction. Since $\lambda(\theta_{m+1}) - \lambda(\theta_m)$ and $\tilde{\lambda}_{m+1} - \tilde{\lambda}_m$ converge to zero, there exists a scalar $\epsilon > 0$ and an integer n, such that either $\lambda(\theta_m) - \tilde{\lambda}_m > \epsilon$, or $\lambda(\theta_m) - \tilde{\lambda}_m < -\epsilon$, for all $m > n$. Without loss of generality, let us consider the first possibility.

Recall that the update equation for $\tilde{\lambda}$ is of the form

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) + \delta_m,$$

where δ_m is the last component of the vector ε_m, which is summable by Lemma 3. Given that $\lambda(\theta_m) - \tilde{\lambda}_m$ stays above ϵ, the sequence $\eta\gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m)$ sums to infinity. As δ_m is summable, we conclude that $\tilde{\lambda}_m$ converges to infinity, which contradicts the fact that it is bounded. □

The next lemma shows that the condition $\lambda(\theta_m) \geq \tilde{\lambda}_m$ is satisfied, in the limit.

Lemma 9 *We have* $\liminf_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) \geq 0$.

Proof: Suppose the contrary. Then, there exists some $\epsilon > 0$ such that the inequality

$$\tilde{\lambda}_m - \lambda(\theta_m) > \epsilon$$

holds infinitely often. Let $\beta = \min\{\epsilon, \eta/L^2\}$, where L is the constant of Lemma 5. Using Lemma 8, we conclude that $\tilde{\lambda}_m - \lambda(\theta_m)$ crosses infinitely often from a value smaller than $\beta/3$ to a value larger than $2\beta/3$. In particular, there exist infinitely many pairs n, n', with $n' > n$, such that

$$0 < \tilde{\lambda}_n - \lambda(\theta_n) < \frac{1}{3}\beta, \qquad \tilde{\lambda}_{n'} - \lambda(\theta_{n'}) > \frac{2}{3}\beta,$$

and

$$\frac{1}{3}\beta \leq \tilde{\lambda}_m - \lambda(\theta_m) \leq \frac{2}{3}\beta, \qquad m = n+1, \ldots, n'-1.$$

We use the Lyapunov function

$$\phi(r) = \phi(\theta, \tilde{\lambda}) = \tilde{\lambda} - \lambda(\theta),$$

and note that

$$\phi(r_{n'}) \geq \phi(r_n) + \frac{\beta}{3}. \tag{27}$$

For $m = n, \ldots, n'-1$, we have $0 < \tilde{\lambda} - \lambda(\theta) < \beta \leq \eta/L^2$. Lemma 5 applies and shows that $\nabla\phi(r_m) \cdot h(r_m) \leq 0$. Therefore,

$$\phi(r_{n'}) = \phi(r_n) + \sum_{m=n}^{n'-1} \Big(\gamma_m \nabla\phi(r_m) \cdot h(r_m) + \varepsilon_m(\phi)\Big) \leq \phi(r_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi).$$

By Lemma 4, $\sum_m \varepsilon_m(\phi)$ converges, which implies that $\sum_{m=n}^{n'-1} \varepsilon_m(\phi)$ becomes arbitrarily small. This contradicts Eq. (27) and completes the proof. □

We now continue with the central step in the proof, which consists of showing that $\lim_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) = 0$. Using Lemma 9, it suffices to show that we cannot have

$\limsup_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) > 0$. The main idea is the following. Whenever $\lambda(\theta_m)$ becomes significantly larger than $\tilde{\lambda}_m$, then $\tilde{\lambda}_m$ is bound to increase significantly. On the other hand, by Lemma 7, whenever $\lambda(\theta_m)$ is approximately equal to $\tilde{\lambda}_m$, then $\tilde{\lambda}_m$ cannot decrease by much. Since $\tilde{\lambda}_m$ is bounded, this will imply that $\lambda(\theta_m)$ can become significantly larger than $\tilde{\lambda}_m$ only a finite number of times.

Lemma 10 *We have* $\lim_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) = 0$.

Proof: We will assume the contrary and derive a contradiction. By Lemma 9, we have $\liminf_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) \geq 0$. So if the desired result is not true, we must have $\limsup_{m\to\infty}(\lambda(\theta_m) - \tilde{\lambda}_m) > 0$, which we will assume to be the case. In particular, there is some $A > 0$ such that $\lambda(\theta_m) - \tilde{\lambda}_m > A$, infinitely often. Without loss of generality, we assume that $A \leq \eta/4L^2$, where L is the constant of Lemmas 5 and 6. Let $\alpha > 0$ be some small constant (with $\alpha < A/2$), to be specified later. Using Lemma 9, we have $\lambda(\theta_m) - \tilde{\lambda}_m > -\alpha$ for all large enough m. In addition, by Lemma 8, the condition $|\lambda(\theta_m) - \tilde{\lambda}_m| \leq \alpha$ holds infinitely often. Thus, the algorithm can be broken down into a sequence of cycles, where in the beginning and at the end of each cycle we have $|\lambda(\theta_m) - \tilde{\lambda}_m| \leq \alpha$, while the condition $\lambda(\theta_m) - \tilde{\lambda}_m > A$ holds at some intermediate time in the cycle.

We describe the stages of such a cycle more precisely. A typical cycle starts at some time N with $|\lambda(\theta_N) - \tilde{\lambda}_N| \leq \alpha$. Let n'' be the first time after time N that $\lambda(\theta_{n''}) - \tilde{\lambda}_{n''} > A$. Let n' be the last time before n'' such that $\lambda(\theta_{n'}) - \tilde{\lambda}_{n'} < A/2$. Let also n be the last time before n' such that $\lambda(\theta_n) - \tilde{\lambda}_n < \alpha$. Finally, let n''' be the first time after n'' such that $|\lambda(\theta_{n'''}) - \tilde{\lambda}_{n'''}| < \alpha$. The time n''' is the end of the cycle and marks the beginning of a new cycle.

Recall that the changes in θ_m and $\tilde{\lambda}_m$ converge to zero. For this reason, by taking N to be large enough, we can assume that $\lambda(\theta_n) - \tilde{\lambda}_n \geq 0$. To summarize our construction, we have $N < n < n' < n'' < n'''$, and

$$
\begin{aligned}
&|\lambda(\theta_N) - \tilde{\lambda}_N| < \alpha, \\
&0 \leq \lambda(\theta_n) - \tilde{\lambda}_n < \alpha, \\
&|\lambda(\theta_m) - \tilde{\lambda}_m| \leq A, \qquad m = N, \ldots, n'' - 1, \\
&\lambda(\theta_{n'}) - \tilde{\lambda}_{n'} < \frac{A}{2}, \\
&\lambda(\theta_{n''}) - \tilde{\lambda}_{n''} > A \\
&\alpha \leq \lambda(\theta_m) - \tilde{\lambda}_m \leq A, \qquad m = n+1, \ldots, n'' - 1, \\
&\frac{A}{2} \leq \lambda(\theta_m) - \tilde{\lambda}_m \leq A, \qquad m = n'+1, \ldots, n'' - 1, \\
&\alpha \leq \lambda(\theta_m) - \tilde{\lambda}_m, \qquad m = n'', \ldots, n''' - 1.
\end{aligned}
$$

Our argument will use the Lyapunov functions

$$\phi(r) = \phi(\theta, \tilde{\lambda}) = \lambda(\theta) - (L^2/\eta)\left(\lambda(\theta) - \tilde{\lambda}\right)^2,$$

where L is as in Lemma 5 and 6, and

$$\psi(r) = \psi(\theta, \tilde{\lambda}) = \tilde{\lambda} - \lambda(\theta).$$

We have

$$\varepsilon_m(\phi) = \phi(r_{m+1}) - \phi(r_m) - \gamma_m \nabla\phi(r_m)\cdot h(r_m),$$

and we define $\varepsilon_m(\psi)$ by a similar formula. By Lemma 4, the series $\sum_m \varepsilon_m(\phi)$ and $\sum_m \varepsilon_m(\psi)$ converge. Also, let

$$\delta_m = \tilde{\lambda}_{m+1} - \tilde{\lambda}_m - \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m).$$

We observe that δ_m is the last component of ε_m and therefore, the series $\sum_m \delta_m$ converges and $\lim_{m\to\infty}\delta_m = 0$. Finally, let C be a constant such that $|\nabla\psi(r_m)\cdot h(r_m)| \le C$, for all m, which exists because $\psi\in\Phi$ and because the sequences $h(r_m)$ and $\tilde{\lambda}_m$ are bounded.

Using the above observations, we see that if the beginning time N of a cycle is chosen large enough, then for any k, k' such that $N\le k\le k'$, we have

$$\begin{aligned}
\gamma_k C &\le \frac{A}{32},\\
\left|\sum_{m=k}^{k'}\varepsilon_m(\phi)\right| &\le \frac{A^2}{96C},\\
\left|\sum_{m=k}^{k'}\varepsilon_m(\psi)\right| &\le \frac{A}{32},\\
\left|\sum_{m=k}^{k'}\delta_m\right| &\le \eta\frac{A^2}{8C}.
\end{aligned}$$

Finally, we assume that α has been chosen small enough so that

$$2(\alpha + (L^2\alpha^2/\eta)) \le \eta\frac{A^2}{96C}.$$

Using the fact that $\lambda(\theta_{n'+1}) - \tilde{\lambda}_{n'+1} \ge A/2$, we have

$$\lambda(\theta_{n'}) - \tilde{\lambda}_{n'} = \lambda(\theta_{n'+1}) - \tilde{\lambda}_{n'+1} + \gamma_{n'}\nabla\psi(r_{n'})\cdot h(r_{n'}) + \varepsilon_{n'}(\psi) \ge \frac{A}{2} - \frac{A}{16}.$$

Furthermore, we have

$$\begin{aligned}
\frac{A}{2} &\le \Big((\lambda(\theta_{n''}) - \tilde{\lambda}_{n''}) - (\lambda(\theta_{n'}) - \tilde{\lambda}_{n'})\Big)\\
&= -\psi(r_{n''}) + \psi(r_{n'})\\
&= -\sum_{m=n'}^{n''-1}\gamma_m\nabla\psi(r_m)\cdot h(r_m) - \sum_{m=n'}^{n''-1}\varepsilon_m(\psi)\\
&\le \sum_{m=n'}^{n''-1}\gamma_m C + \frac{A}{32},
\end{aligned}$$

which implies that

$$\sum_{m=n'}^{n''-1}\gamma_m \ge \frac{A}{2C} - \frac{A}{32C}.$$

Then,

$$\begin{aligned}
\tilde{\lambda}_{n'''} &= \tilde{\lambda}_n + \sum_{m=n}^{n'''-1} \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) + \sum_{m=n}^{n'''-1} \delta_m \\
&\geq \tilde{\lambda}_n + \sum_{m=n'}^{n''-1} \eta\gamma_m(\lambda(\theta_m) - \tilde{\lambda}_m) + \sum_{m=n}^{n'''-1} \delta_m \\
&\geq \tilde{\lambda}_n + \eta\left(\frac{A}{2C} - \frac{A}{32C}\right)\left(\frac{A}{2} - \frac{A}{16}\right) - \eta\frac{A^2}{8C} \\
&\geq \tilde{\lambda}_n + \eta\frac{A^2}{24C}.
\end{aligned}$$

We have shown so far that $\tilde{\lambda}_m$ has a substantial increase between time n and n'''. We now show that $\tilde{\lambda}_m$ can only have a small decrease in the time between N and n. Indeed, by Lemma 7, we have

$$\tilde{\lambda}_n \geq \tilde{\lambda}_N - 2(\alpha + L^2\alpha^2) + \sum_{m=N}^{n-1} \varepsilon_m(\phi).$$

By combining these two properties, we obtain

$$\begin{aligned}
\tilde{\lambda}_{n'''} &\geq \tilde{\lambda}_N - 2(\alpha + L^2\alpha^2) - \eta\frac{A^2}{96C} + \eta\frac{A^2}{24C} \\
&\geq \tilde{\lambda}_N + \eta\frac{A^2}{48C}.
\end{aligned}$$

We have shown that $\tilde{\lambda}_m$ increases by a positive amount during each cycle. Since $\tilde{\lambda}_m$ is bounded above, this proves that there can only be a finite number of cycles, and a contradiction has been obtained. □

Lemma 11 *The sequences $\tilde{\lambda}_m$ and $\lambda(\theta_m)$ converge.*

Proof: Consider the function $\phi(r) = \lambda(\theta) - (L^2/\eta)(\lambda(\theta) - \tilde{\lambda})^2$, and the same constant L as in Lemma 6. Let α be a scalar such that $0 < \alpha \leq \eta/(4L^2)$. By the preceding lemma and by Lemma 4, there exists some N such that if $N \leq n \leq n'$, we have

$$|\lambda(\theta_n) - \tilde{\lambda}_n| \leq \alpha,$$

and

$$\left|\sum_{m=n}^{n'-1} \varepsilon_m(\phi)\right| \leq \alpha.$$

Using Lemma 6,

$$\phi(\theta_{n'}) \geq \phi(\theta_n) + \sum_{m=n}^{n'-1} \varepsilon_m(\phi) \geq \phi(\theta_n) - \alpha, \qquad N \leq n \leq n',$$

or

$$\lambda(\theta_{n'}) - (L^2/\eta)(\lambda(\theta_{n'}) - \tilde{\lambda}_{n'})^2 \geq \lambda(\theta_n) - (L^2/\eta)(\lambda(\theta_n) - \tilde{\lambda}_n)^2 - \alpha,$$

which implies

$$\lambda(\theta_{n'}) \geq \lambda(\theta_n) - (L^2\alpha^2/\eta) - \alpha, \qquad N \leq n \leq n'.$$

Therefore,

$$\liminf_{n'\to\infty} \lambda(\theta_{n'}) \geq \lambda(\theta_n) - (L^2\alpha^2/\eta) - \alpha, \qquad N \leq n,$$

and this implies that

$$\liminf_{m\to\infty} \lambda(\theta_m) \geq \limsup_{m\to\infty} \lambda(\theta_m) - (L^2\alpha^2/\eta) - \alpha.$$

Since α can be chosen arbitrarily small, we have $\liminf_{m\to\infty} \lambda(\theta_m) \geq \limsup_{m\to\infty} \lambda(\theta_m)$, and since the sequence $\lambda(\theta_m)$ is bounded, we conclude that it converges. Using also Lemma 10, it follows that $\tilde{\lambda}_m$ converges as well. □

A.3 Convergence of $\nabla\lambda(\theta_m)$

In the preceding subsection, we have shown that $\lambda(\theta_m)$ and $\tilde{\lambda}_m$ converge to a common limit. It now remains to show that $\nabla\lambda(\theta_m)$ converges to zero.

Since $\lambda(\theta_{t_m}) - \tilde{\lambda}_{t_m}$ converges to zero, the algorithm is of the form

$$\theta_{m+1} = \theta_m + \gamma_m E_{\theta_m}[T](\nabla\lambda(\theta_m) + e_m) + \epsilon_m,$$

where e_m converges to zero and ϵ_m is a summable sequence. This is a gradient method with errors, similar to the methods considered in [Del96] and [BT97]. However, [Del96] assumes the boundedness of the sequence of iterates, and the results of [BT97] do not include the term e_m. Thus, while the situation is very similar to that considered in these references, a separate proof is needed.

We will first show that $\liminf_{m\to\infty} \|\nabla\lambda(\theta_m)\| = 0$. Suppose the contrary. Then, there exists some $\epsilon > 0$ and some N such that $\|\nabla\lambda(\theta_m)\| > \epsilon$ for all $m > N$. In addition, by taking N large enough, we can also assume that $\|e_m\| \leq \epsilon/2$. Then, it is easily checked that

$$\nabla\lambda(\theta_m) \cdot (\nabla\lambda(\theta_m) + e_m) \geq \frac{\epsilon^2}{2}.$$

Let $\phi(r) = \lambda(\theta)$. Note that $\phi \in \Phi$. We have

$$\begin{aligned} \lambda(\theta_{m+1}) &= \lambda(\theta_m) + \gamma_m E_{\theta_m}[T]\nabla\lambda(\theta_m) \cdot (\nabla\lambda(\theta_m) + e_m) + \varepsilon_m(\phi) \\ &\geq \lambda(\theta_m) + \gamma_m \frac{\epsilon^2}{2} + \varepsilon_m(\phi). \end{aligned} \tag{28}$$

Since $\varepsilon_m(\phi)$ is summable (Lemma 4), but $\sum_m \gamma_m = \infty$, we conclude that $\lambda(\theta_m)$ converges to infinity, which is a contradiction.

Next we show that $\limsup_{m\to\infty} \|\nabla\lambda(\theta_m)\| = 0$. Suppose the contrary. Then, there exists some $\epsilon > 0$ such that $\|\nabla\lambda(\theta_n)\| > \epsilon$ for infinitely many indices n. For any such n, let n' be the first subsequent time that $\|\nabla\lambda(\theta_{n'})\| < \epsilon/2$. Then,

$$\begin{aligned}
\frac{\epsilon}{2} &\leq \|\nabla\lambda(\theta_n)\| - \|\nabla\lambda(\theta_{n'})\| \\
&\leq \|\nabla\lambda(\theta_n) - \nabla\lambda(\theta_{n'})\| \\
&\leq C\|r_n - r_{n'}\| \\
&= C\left\|\sum_{m=n}^{n'-1} \gamma_m h(r_m) + \sum_{m=n}^{n'-1} \varepsilon_m\right\| \\
&\leq C\sum_{m=n}^{n'-1} \gamma_m \|h(r_m)\| + C\left\|\sum_{m=n}^{n'-1} \varepsilon_m\right\|,
\end{aligned}$$

for some constant C, as $\nabla^2\lambda(\theta)$ is bounded (Lemma 1). Recall that $\|h(r_m)\|$ is bounded by some constant B. Furthermore, when n is large enough, the summability of the sequence ε_m yields $C\|\sum_{m=n}^{n'-1}\varepsilon_m\| \leq \epsilon/4$. This implies that $\sum_{m=n}^{n'-1}\gamma_m \geq \epsilon/4CB$. By an argument very similar to the one that led to Eq. (28), it is easily shown that there exists some $\beta > 0$ such that

$$\lambda(\theta_{n'}) \geq \lambda(\theta_n) + \beta,$$

which contradicts the convergence of the sequence $\lambda(\theta_m)$. □

B Proof of Proposition 4

In this section, we prove the convergence of the on-line method introduced in Section 4, which is described by

$$\begin{aligned}
\theta_{k+1} &= \theta_k + \gamma_k\Big(\nabla g_{i_k}(\theta_k) + (g_{i_k}(\theta_k) - \tilde{\lambda}_k)z_k\Big), \\
\tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \eta\gamma_k(g_{i_k}(\theta_k) - \tilde{\lambda}_k), \\
z_{k+1} &= \begin{cases} 0, & \text{if } i_{k+1} = i^* \\ z_k + \dfrac{\nabla p_{i_k i_{k+1}}(\theta_k)}{p_{i_k i_{k+1}}(\theta_k)}, & \text{otherwise.} \end{cases}
\end{aligned}$$

The proof has many common elements with the proof of Proposition 3. For this reason, we will only discuss the differences in the two proofs. In addition, whenever routine arguments are used, we will only provide an outline.

As in Appendix A, we let $r_k = (\theta_k, \tilde{\lambda}_k)$. Note, however, the different meaning of the index k which is now advanced at each time step, whereas in Appendix A it was advanced whenever the state i^* was visited. We also define an augmented state $x_k = (i_k, z_k)$.

We rewrite the update equations as

$$r_{k+1} = r_k + \gamma_k R(x_k, r_k),$$

where

$$R(x_k, r_k) = \begin{bmatrix} \nabla g_{i_k}(\theta_k) + (g_{i_k}(\theta_k) - \tilde{\lambda}_k)z_k \\ \eta\Big(g_{i_k}(\theta_k) - \tilde{\lambda}_k\Big) \end{bmatrix}. \tag{29}$$

Consider the sequence of states $(i_0, i_1, \ldots)$ visited during the execution of the algorithm. As in Section 3, we let t_m be the mth time that the recurrent state i^* is visited. Also, as in Appendix A, we let

$$\mathcal{F}_m = \{\theta_0, \tilde{\lambda}_0, i_0, \ldots, i_{t_m}\}$$

stand for the history of the algorithm up to and including time t_m.

The parameter θ_k keeps changing between visits to state i^*, which is a situation somewhat different than that considered in Lemma 2(a). Nevertheless, using Assumption 5, a similar argument applies and shows that for any positive integer s, there exists a constant D_s such that

$$E\left[(t_{m+1} - t_m)^s \mid \mathcal{F}_m\right] \leq D_s. \tag{30}$$

We have

$$\begin{aligned} r_{t_{m+1}} &= r_{t_m} + \sum_{k=t_m}^{t_{m+1}-1} \gamma_k R(x_k, r_k) \\ &= r_{t_m} + \tilde{\gamma}_m \hat{h}(r_{t_m}) + \varepsilon_m, \end{aligned} \tag{31}$$

where $\tilde{\gamma}_m$ and ε_m are given by

$$\tilde{\gamma}_m = \sum_{k=t_m}^{t_{m+1}-1} \gamma_k, \tag{32}$$

$$\varepsilon_m = \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \left(R(x_k, r_k) - \hat{h}(r_{t_m}) \right),$$

and $\hat{h}$ is a scaled version of the function h in Appendix A, namely,

$$\hat{h}(r) = \frac{h(r)}{E_\theta[T]} = \begin{bmatrix} \nabla\lambda(\theta) + \frac{G(\theta)}{E_\theta[T]}(\lambda(\theta) - \tilde{\lambda}) \\ \eta\left(\lambda(\theta) - \tilde{\lambda}\right) \end{bmatrix}. \tag{33}$$

We note the following property of the various stepsize parameters.

Lemma 12

(a) For any positive integer s, we have

$$E\left[\sum_{m=1}^{\infty} \gamma_{t_m}^2 (t_{m+1} - t_m)^s\right] < \infty.$$

(b) We have

$$\sum_{m=1}^{\infty} \tilde{\gamma}_m = \infty, \qquad \sum_{m=1}^{\infty} \tilde{\gamma}_m^2 < \infty,$$

with probability 1.

Proof: (a) ¿From Eq. (30), and because γ_{t_m} is $\mathcal{F}_m$-measurable, we have

$$E[\gamma_{t_m}^2 (t_{m+1} - t_m)^s] = E\left[\gamma_{t_m}^2 E[(t_{m+1} - t_m)^s \mid \mathcal{F}_m]\right] \leq E[\gamma_{t_m}^2] D_s.$$

Hence,

$$\sum_{m=1}^{\infty} E[\gamma_{t_m}^2 (t_{m+1} - t_m)^s] \leq D_s \sum_{k=1}^{\infty} \gamma_k^2 < \infty,$$

and the result follows.

(b) By Assumption 4, we have

$$\sum_{m=1}^{\infty} \tilde{\gamma}_m = \sum_{k=1}^{\infty} \gamma_k = \infty.$$

Furthermore, since the sequence γ_k is nonincreasing (Assumption 5), we have

$$\tilde{\gamma}_m^2 \leq \gamma_{t_m}^2 (t_{m+1} - t_m)^2.$$

Using part (a) of the lemma, we obtain that $\sum_{m=1}^{\infty} \tilde{\gamma}_m^2$ has finite expectation and is therefore finite with probability 1. □

Without loss of generality, we assume that $\eta\gamma_k \leq 1$ for all k. Then, the update equation for $\tilde{\lambda}_k$ implies that $|\tilde{\lambda}_k| \leq \max\{|\tilde{\lambda}_0|, C\}$, where C is a bound on $|g_i(\theta)|$. Thus, $|\tilde{\lambda}_k|$ is bounded by a deterministic constant, which implies that the magnitude of $\hat{h}(r_k)$ is also bounded by a deterministic constant.

We now observe that Eq. (31) is of the same form as Eq. (25) that was studied in the preceding appendix, except that we now have r_{t_m} in place of r_m, $\tilde{\gamma}_m$ in place of γ_m, and $\hat{h}(r_{t_m})$ in place of $h(r_m)$. By Lemma 12(b), the new stepsizes satisfy the same conditions as those imposed by Assumption 4 on the stepsizes γ_m of Appendix A. Also, in the next subsection, we show that the series $\sum_m \varepsilon_m$ converges. Once these properties are established, the arguments in Appendix A remain valid and show that $\lambda(\theta_{t_m})$ converges, and that $\nabla\lambda(\theta_{t_m})$ converges to zero. Furthermore, we will see in the next subsection that the total change of θ_k between consecutive visits to i^* converges to zero. This implies that $\lambda(\theta_k)$ converges and that $\nabla\lambda(\theta_k)$ converges to zero, and Proposition 4 is established.

B.1 Summability of ε_k and Convergence of the Changes in θ_k

This subsection is devoted to the proof that the series $\sum_m \varepsilon_m$ converges, and that the changes of θ_k between visits to i^* converge to zero.

We introduce some more notation. The evolution of the augmented state $x_k = (i_k, z_k)$ is affected by the fact that θ_k changes at each time step. Given a time t_m at which i^* is visited, we define a "frozen" augmented state $x_k^F = (i_k^F, z_k^F)$ which evolves the same way as x_k except that θ_k is held fixed at θ_{t_m} until the next visit at i^*. More precisely, we let $x_{t_m}^F = x_{t_m}$. Then, for $k \geq t_m + 1$, we let i_k^F evolve as a time-homogeneous Markov chain with transition probabilities $p_{ij}(\theta_{t_m})$. We also let $t_{m+1}^F = \min\{k > t_m \mid i_k^F = i^*\}$ be the first time after t_m that i_k^F is equal to i^*, and

$$z_{k+1}^F = z_k^F + L_{i_k^F i_{k+1}^F}(\theta_{t_m}).$$

We start by breaking down ε_m as follows:

$$\begin{aligned}
\varepsilon_m &= \sum_{k=t_m}^{t_{m+1}-1} \gamma_k \left(R(x_k, r_k) - \hat{h}(r_{t_m})\right) \\
&= \varepsilon_m^{(1)} + \varepsilon_m^{(2)} + \varepsilon_m^{(3)} + \varepsilon_m^{(4)} + \varepsilon_m^{(5)},
\end{aligned}$$

where

$$
\begin{aligned}
\varepsilon_m^{(1)} &= \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k)\, \hat{h}(r_{t_m}), \\
\varepsilon_m^{(2)} &= \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}^F-1} \left[R(x_k^F, r_{t_m}) - \hat{h}(r_{t_m}) \right], \\
\varepsilon_m^{(3)} &= \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} \left[R(x_k, r_{t_m}) - \hat{h}(r_{t_m}) \right] \\
&\quad -\gamma_{t_m} \sum_{k=t_m}^{t_{m+1}^F-1} \left[R(x_k^F, r_{t_m}) - \hat{h}(r_{t_m}) \right], \\
\varepsilon_m^{(4)} &= \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} \left[R(x_k, r_k) - R(x_k, r_{t_m}) \right], \\
\varepsilon_m^{(5)} &= \sum_{k=t_m}^{t_{m+1}-1} (\gamma_k - \gamma_{t_m})\, R(x_k, r_k).
\end{aligned}
$$

We will show that each one of the series $\sum_m \varepsilon_m^{(n)}$, $n = 1, \ldots, 5$, converges with probability 1.

We make the following observations. The ratio $L_{i_k i_{k+1}}(\theta_k)$ is bounded because of Assumption 3. This implies that between the times t_m and t_{m+1} that i^* is visited, the magnitude of z_k is bounded by $C(t_{m+1}-t_m)$ for some constant C. Similarly, the magnitude of z_k^F is bounded by $C(t_{m+1}^F - t_m)$. Using the boundedness of $\tilde{\lambda}_k$ and $\hat{h}(r_k)$, together with the update equations for θ_k and $\tilde{\lambda}_k$, we conclude that there exists a (deterministic) constant C, such that for every m, we have

$$
\begin{aligned}
\|R(x_k, r_k)\| &\leq C(t_{m+1} - t_m), && k \in \{t_m, \ldots, t_{m+1} - 1\}, && (34) \\
\|R(x_k^F, r_k)\| &\leq C(t_{m+1}^F - t_m), && k \in \{t_m, \ldots, t_{m+1}^F - 1\}, && (35) \\
\|r_k - r_{t_m}\| &\leq C\gamma_{t_m}(t_{m+1} - t_m)^2, && k \in \{t_m, \ldots, t_{m+1} - 1\}, && (36) \\
\|R(x_k, r_{t_m}) - R(x_k, r_k)\| &\leq C\gamma_{t_m}(t_{m+1} - t_m)^3, && k \in \{t_m, \ldots, t_{m+1} - 1\}. && (37)
\end{aligned}
$$

Lemma 13 *The series $\sum_m \varepsilon_m^{(1)}$ converges with probability 1.*

Proof: Let B be a bound on $\|\hat{h}(r_k)\|$. Then, using Assumption 5, we have

$$
\|\varepsilon_m^{(1)}\| \leq B \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k) \leq BA\gamma_{t_m}^2 (t_{m+1} - t_m)^p.
$$

Then, Lemma 12(a), implies that $\sum_m \|\varepsilon_m^{(1)}\|$ has finite expectation, and is therefore finite with probability 1. □

Lemma 14 *The series $\sum_m \varepsilon_m^{(2)}$ converges with probability 1.*

Proof: When the parameters θ and $\tilde{\lambda}$ are frozen to their values at time t_m, the total update $\sum_{k=t_m}^{t_{m+1}^F-1} R(x_k^F, r_{t_m})$ coincides with the update $H_m(r_m)$ of the algorithm studied in Appendix A. Using the discussion in the beginning of that appendix, we have $E[H_m(r_m) \mid \mathcal{F}_m] = h(r_{t_m})$. Furthermore, observe that

$$E\left[\sum_{k=t_m}^{t_{m+1}^F-1} \hat{h}(r_{t_m}) \mid \mathcal{F}_m\right] = \hat{h}(r_{t_m})E_{\theta_{t_m}}[T] = h(r_{t_m}).$$

Thus, $E[\varepsilon_m^{(2)} \mid \mathcal{F}_m] = 0$. Furthermore, using Eq. (34), we have

$$E[\|\varepsilon_m^{(2)}\|^2 \mid \mathcal{F}_m] \leq C\gamma_{t_m}^2(t_{m+1} - t_m)^4.$$

Using Lemma 12(a), we obtain

$$E\left[\sum_{m=1}^{\infty} \|\varepsilon_m^{(2)}\|^2\right] < \infty.$$

Thus, $\sum_m \varepsilon_m^{(2)}$ is martingale with bounded variance and, therefore, converges. □

Lemma 15 *The series $\sum_m \varepsilon_m^{(3)}$ converges with probability 1.*

Proof: The proof is based on a coupling argument. For $k = t_m, \ldots, t_{m+1} - 1$, the two processes x_k and x_k^F can be defined on the same probability space as follows. Suppose that i_k and i_k^F are both equal to some particular state i. We partition the unit interval into N subintervals, each of length $p_{ij}(\theta_k)$, $j = 1, \ldots, N$. The next state i_{k+1} is obtained by generating a uniform random variable U and selecting the state j associated with the particular subinterval into which U belongs. The same random variable U is used to select i_{k+1}^F, except that we now have a partition into subintervals of length $p_{ij}(\theta_k^F)$. The probability that U causes i_{k+1} and i_{k+1}^F to be different is bounded by $N \max_{i,j} |p_{ij}(\theta_k) - p_{ij}(\theta_k^F)|$. Using the assumption that the transition probabilities depend smoothly on θ, as well as Eq. (36), we obtain

$$P(i_{k+1}^F \neq i_{k+1} \mid i_k^F = i_k) \leq B\|\theta_k - \theta_k^F\| \leq B\|r_k - r_{t_m}\| \leq BC\gamma_{t_m}(t_{m+1} - t_m)^2, \qquad (38)$$

for some constants B and C.

We define $\mathcal{E}_m$ to be the event

$$\mathcal{E}_m = \{x_k^F \neq x_k \text{ for some } k = t_m, \ldots, t_{m+1}\}.$$

Using Eq. (38), we obtain

$$P(\mathcal{E}_m \mid t_m, t_{m+1}) \leq BC \sum_{k=t_m}^{t_{m+1}-1} \gamma_{t_m}(t_{m+1} - t_m)^2 = BC\gamma_{t_m}(t_{m+1} - t_m)^3.$$

Note that if the event $\mathcal{E}_m$ does not occur, then $\varepsilon_m^{(3)} = 0$. Thus,

$$E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}] = P(\mathcal{E}_m \mid t_m, t_{m+1})E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}, \mathcal{E}_m].$$

Since $\hat{h}(r_k)$ is bounded, and using also the bounds (34)-(35), we have

$$\|\varepsilon_m^{(3)}\| \leq \gamma_{t_m} C((t_{m+1} - t_m)^2 + (t_{m+1}^F - t_m)^2),$$

for some new constant C. We conclude that

$$E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}, \mathcal{E}_m] \leq \gamma_{t_m} C\Big((t_{m+1} - t_m)^2 + E[(t_{m+1}^F - t_m)^2 \mid t_m, t_{m+1}, \mathcal{E}_m]\Big).$$

Now, it is easily verified that

$$\begin{aligned} E[(t_{m+1}^F - t_m)^2 \mid t_m, t_{m+1}, \mathcal{E}_m]) &\leq 2E[(t_{m+1}^F - t_{m+1})^2 \mid t_m, t_{m+1}, \mathcal{E}_m] + 2(t_{m+1} - t_m)^2 \\ &\leq C(t_{m+1} - t_m)^2, \end{aligned}$$

for some new constant C. By combining these inequalities, we obtain

$$E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}, \mathcal{E}_m] \leq C\gamma_{t_m}(t_{m+1} - t_m)^2,$$

and

$$E[\|\varepsilon_m^{(3)}\| \mid t_m, t_{m+1}] \leq BC\gamma_{t_m}^2(t_{m+1} - t_m)^5,$$

for some different constant C. Using Lemma 12(a), $\sum_m \|\varepsilon_m^{(3)}\|$ has finite expectation, and is therefore finite with probability 1. □

Lemma 16 *The series $\sum_m \varepsilon_m^{(4)}$ converges with probability 1.*

Proof: Using Eq. (37), we have

$$\|\varepsilon_m^{(4)}\| \leq \gamma_{t_m} \sum_{k=t_m}^{t_{m+1}-1} C\gamma_{t_m}(t_{m+1} - t_m)^3 = C\gamma_{t_m}^2(t_{m+1} - t_m)^4.$$

Using Lemma 12(a), $\sum_m \|\varepsilon_m^{(4)}\|$ has finite expectation, and is therefore finite with probability 1. □

Lemma 17 *The series $\sum_m \varepsilon_m^{(5)}$ converges with probability 1.*

Proof: Using Assumption 5 and the bound (34) on $\|R(x_k, r_k)\|$, we have

$$\|\varepsilon_m^{(5)}\| \leq C(t_{m+1} - t_m) \sum_{k=t_m}^{t_{m+1}-1} (\gamma_{t_m} - \gamma_k) \leq AC\gamma_{t_m}^2(t_{m+1} - t_m)^{p+1}.$$

Using Lemma 12(a), $\sum_m \|\varepsilon_m^{(5)}\|$ has finite expectation, and is therefore finite with probability 1. □

We close by establishing the statement mentioned at the end of the preceding subsection, namely, that the changes in r_k (and, therefore, the changes in θ_k as well) between visits to the recurrent state i^* tend to zero as time goes to infinity. Indeed, Eq. (34) establishes a bound on $\|r_k - r_{t_m}\|$ for $k = t_m, \ldots, t_{m+1} - 1$, which converges to zero because of Lemma 12(a).