Approximate Gradient Methods in Policy-Space Optimization of Markov Reward Processes [1]

Peter Marbach

Department of Computer Science
University of Toronto
Toronto, ON, M5S 3H4
e-mail: marbach@cs.toronto.edu

John N. Tsitsiklis

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

Abstract

We consider a discrete time, finite state Markov reward process that depends on a set of parameters. We start with a brief review of (stochastic) gradient descent methods that tune the parameters in order to optimize the average reward, using a single (possibly simulated) sample path of the process of interest. The resulting algorithms can be implemented online, and have the property that the gradient of the average reward converges to zero with probability 1. On the other hand, the updates can have a high variance, resulting in slow convergence. We address this issue and propose two approaches to reduce the variance. These approaches rely on approximate gradient formulas, which introduce an additional bias into the update direction. We derive bounds for the resulting bias terms and characterize the asymptotic behavior of the resulting algorithms. For one of the approaches considered, the magnitude of the bias term exhibits an interesting dependence on the time it takes for the rewards to reach steady-state. We also apply the methodology to Markov reward processes with a reward-free termination state, and an expected total reward criterion. We use a call admission control problem to illustrate the performance of the proposed algorithms.

Keywords: Markov reward processes, simulation-based optimization, policy-space optimization.

[1] This research was supported by contracts with Siemens AG, Munich, Germany, and Alcatel Bell, Belgium; and by contract ACI-9873339 with the National Science Foundation. A preliminary version of this paper was presented at the 38th IEEE Conference on Decision and Control, in December 1999 [MT99].

1 Introduction

We consider discrete time, finite state Markov reward processes in which the transition probabilities and one-stage rewards depend on a parameter vector $\theta \in \Re^K$. We propose simulation-based algorithms for tuning the parameter θ to optimize either the average reward, or the expected reward-to-go. Compared with earlier work [MT01], these algorithms have a smaller variance and therefore tend to perform better in practice. Most of the paper focuses on methods for optimizing the average reward; the resulting methodology is readily applied to Markov processes with a reward-free termination state where the optimizer wants to maximize the expected reward-to-go. Here, we only outline the results for the latter case and refer to [Mar98] for a detailed discussion.

In earlier work [MT01], we proposed a method for tuning the parameter θ to optimize the average reward, denoted by $\lambda(\theta)$. The method relies on simulations to produce an estimate of the gradient of the average reward. It can be implemented online, and has the property that the gradient of the average reward converges to zero with probability 1 (which is the strongest possible result for gradient-related stochastic approximation algorithms). In addition, the method can be applied to average reward Markov decision processes (with finite state and action spaces) in which one restricts to a parametric class of randomized control policies that depend on a parameter vector θ. In this setting, the method does not require the transition probabilities and one-stage rewards to be explicitly known, but only assumes that a sample path and its associated reward sequence can be observed.

A drawback of the algorithms proposed in [MT01] is that the updates may have a high variance, which can result in slow convergence. This is because they essentially employ a renewal period (interval between visits to a certain recurrent state) to produce an estimate of the gradient. If the length of a typical renewal period is large (as tends to be the case when the state space is large), then the variance of the corresponding estimate will also be large. In this paper, we address this issue and propose two approaches to reduce the variance: one which estimates the gradient based on trajectories which tend to be shorter than a renewal period, and another which employs a discount factor. However, the resulting algorithms introduce an additional bias into the update direction. As a result, we cannot guarantee the convergence of $\nabla\lambda(\theta)$ to zero. We will nevertheless establish a result of the form

$$\liminf_{m\to\infty} \|\nabla\lambda(\theta_m)\| \le D,$$

where the constant D is an upper bound on the magnitude of the bias. Thus, if the bound D is small, then the gradient $\nabla\lambda(\theta_m)$ is small infinitely often. We interpret the bias bound D in terms of qualitative properties of the underlying process. In particular, for the case where a discount factor α is employed, we show that D is small, as long as the effect of the current state on the expected reward n steps later falls at a rate faster than α^n. (A sufficient – but not necessary – condition for this to happen is that the underlying Markov chain reaches steady-state at a rate faster than α^n.) As gradient-type methods tend to be robust with respect to small biases, the algorithms we propose are expected to perform better in practice. We provide a numerical case study to illustrate this point.

We provide some brief comments on the related literature and we refer to [MT01] for a more detailed comparison. The starting point for the methods we consider is a certain formula for the gradient of $\lambda(\theta)$, which has been presented in various forms and for various contexts in [Cao00, CC97, CW98, FH94, Gly87, JSJ95, TH95, W92]. The idea of using simulation to estimate the gradient of a performance metric with respect to a parameter vector is in the spirit of infinitesimal perturbation analysis (IPA), specialized to Markov reward processes [CR94, CC97, CW98, FH94, FH97], and has also attracted much attention in the more recent reinforcement learning literature

[W92, JSJ95, BB99]. Finally, the introduction of a discount factor, mostly with the purpose of limiting the variance of the gradient estimates, appears in [JSJ95, KMK97], as well as in the more recent references [BB99, Cao00]. The results reported in this paper have also been presented in [Mar98] and [MT99].

The rest of the paper is structured as follows. In Sections 2 and 3, we provide a brief summary of the framework and results of [MT01]. In Section 3.2, we propose two approaches to reduce the variance in the update, which we study in more detail in Sections 4 and 5, respectively, where we also state our main results. In Section 6, we outline how the methodology can be applied to Markov reward processes with a reward-free terminal state, and an expected total reward criterion. In Section 7, we briefly mention how the algorithms of the previous sections can be applied to Markov decision processes. Finally, in Section 8, we provide numerical results from a case study involving an admission control problem.

2 Formulation

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 $\theta \in \Re^K$. We denote the one-step transition probabilities by $P_{ij}(\theta)$, $i, j \in S$, and the n-step transition probabilities by $P_{ij}^n(\theta)$, i.e.,

$$P_{ij}(\theta) = P(i_1 = j \mid i_0 = i, \theta), \quad \text{and } P_{ij}^n(\theta) = P(i_n = j \mid i_0 = i, \theta), \quad n = 1, 2, \ldots ,$$

where i_n stands for the state of the chain at time n. 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 (the set of all limit points of $\mathcal{P}$). 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 (Recurrence) *The Markov chain corresponding to every $P \in \overline{\mathcal{P}}$ is aperiodic. Furthermore, there exists a state $i^* \in S$ which is recurrent for every such Markov chain.*

Assumption 2 (Regularity) *For all states $i, j \in S$, the transition probability $P_{ij}(\theta)$, and the one-state reward $g_i(\theta)$, are bounded, twice differentiable, and have bounded first and second derivatives. Furthermore, we have*

$$\nabla P_{ij}(\theta) = P_{ij}(\theta) L_{ij}(\theta), \quad \theta \in \Re^K$$

for some bounded function $L_{ij}(\cdot)$.

Assumption 1 allows us to use the recurrent state i^* as a reference state and to employ results of renewal theory (see for example [Gal95]) for our analysis. Assumption 2 (Regularity) ensures that the transition probabilities $P_{ij}(\theta)$ and the one-stage reward $g_i(\theta)$ depend smoothly on θ, and that the quotient $\nabla P_{ij}(\theta)/P_{ij}(\theta) = L_{ij}(\theta)$ is well behaved.

Under Assumption 1, the balance equations

$$\pi'(\theta) P(\theta) = \pi'(\theta)$$

have a unique solution for every $\theta \in \Re^K$, where $\pi'(\theta)$ is the row vector $\big(\pi_1(\theta), ..., \pi_N(\theta)\big)$, and $\pi_i(\theta)$ is the steady state probability of state i in the Markov chain with transition probabilities $P_{ij}(\theta)$.

As a performance metric associated with the parameter θ, we use the average reward criterion

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

Here, i_k is the state visited 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 (Recurrence), the average reward $\lambda(\theta)$ is well defined for every θ, and does not depend on the initial state.

We define the the differential reward $v_i(\theta)$ of state $i \in S$, and the mean recurrence time $E_\theta[T]$ by

$$\begin{aligned} v_i(\theta) &= E_\theta \left[\sum_{k=0}^{T-1} \big(g_{i_k}(\theta) - \lambda(\theta)\big) \mid i_0 = i \right], \\ E_\theta[T] &= E_\theta[T \mid i_0 = i^*], \end{aligned}$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that the recurrent state i^* is visited. We have $v_{i^*}(\theta) = 0$. The following lemma, established in [MT01], states that $\lambda(\theta)$, $E_\theta[T]$ and $v_i(\theta)$, $i \in S$, depend smoothly on θ.

Lemma 1 *Let Assumption 1 (Recurrence) and Assumption 2 (Regularity) hold. Then, $\lambda(\theta)$, $E_\theta[T]$ and $v_i(\theta)$, $i \in S$, are (as functions of θ) bounded, twice differentiable, and have bounded first and second derivatives. Furthermore, for every integer $s > 0$, there exists a constant D_s, such that for all $\theta \in \Re^K$, we have*

$$E_\theta[T^s] = E_\theta[T^s \mid i_0 = i^*] \le D_s,$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited.

3 Background

To maximize the average reward $\lambda(\theta)$, we will use a gradient-type method of the form

$$\theta := \theta + \gamma F(\theta),$$

where $F(\theta)$ is a simulation-based estimate of $\nabla\lambda(\theta)$, and γ is a positive step size. In order to construct such an estimate $F(\theta)$, we start with the gradient formula

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

(see [CC97], [Gly87], or [MT01], for a derivation) which we rewrite as

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

where $L_{ij}(\theta)$ is as in Assumption 2.

Let the parameter vector θ be fixed to some value, and let $\{i_n\}$ be a sample path of the corresponding Markov chain, possibly obtained through simulation. Furthermore, let t_m be the time of the mth visit at the recurrent state i^*, i.e. $i_{t_m} = i^*$ for $m = 1, 2, \ldots$. Consider the estimate of $\nabla\lambda(\theta)$ given by

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

where

$$\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{t_{m+1}-1} \left(g_{i_k}(\theta) - \tilde{\lambda} \right), \quad t_m < n < t_{m+1}, \tag{2}$$

is an estimate of the differential reward $v_{i_n}(\theta)$, and $\tilde{\lambda}$ is some estimate of $\lambda(\theta)$. Noting that $v_{i^*}(\theta) = 0$, we let

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

Assumption 1 (Recurrence) allows us to employ renewal theory (see, for example, [Gal95]) to obtain the following result, which states that the expectation of $F_m(\theta, \tilde{\lambda})$ is aligned with $\nabla\lambda(\theta)$ to the extent that $\tilde{\lambda}$ is close to $\lambda(\theta)$ (see [MT01]).

Proposition 1 *We have*

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

where

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

and $E_\theta[T]$ is the mean recurrence time.

3.1 An Algorithm that Updates at Visits to the Recurrent State

Using the estimate of the gradient $\nabla\lambda(\theta)$ given above, we obtain an algorithm which updates the parameter vector θ at visits to the recurrent state i^*. At the same time, the estimate $\tilde{\lambda}$ of the average reward gets updated to drive the bias term $G(\theta)(\lambda(\theta) - \tilde{\lambda})$ to zero.

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

$$\theta_{m+1} = \theta_m + \gamma_m F_m(\theta_m, \tilde{\lambda}_m), \tag{3}$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m), \tag{4}$$

where η is a positive scalar and γ_m is a step size sequence which satisfies the following assumption.

Assumption 3 (Step Size) *The step sizes γ_m are nonnegative and satisfy*

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

This assumption is satisfied, for example, if we let $\gamma_m = 1/m$. We then have the following convergence result [MT01].

Proposition 2 *Let Assumption 1 (Recurrence), Assumption 2 (Regularity), and Assumption 3 (Step Size) 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.

3.2 Variance reduction methods

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, the parameter vector θ gets updated only infrequently, and the estimate $F_m(\theta, \tilde{\lambda})$ can have a large variance. In [MT01], we extended the method of Section 3.1, so that the parameter vector gets updated at every time step. We will give a summary of the corresponding algorithm in the next subsection. In addition, we will consider two ways of reducing the variance in the updates, by replacing the estimate $\tilde{v}_{i_n}(\theta)$ of the differential reward $v_{i_n}(\theta)$ [cf. Eq. (2)] by alternative estimates.

In the first approach, we truncate the length of the sample path for estimating the differential reward by replacing the (generally large) time until we reach the recurrent state i^*, by the (generally smaller) first time that a *set of states* S^*, containing i^*, is reached. Given a simulated trajectory $(i_0, i_1, \ldots)$ under the parameter θ, this leads us to estimate $v_{i_n}(\theta)$ by

$$\tilde{v}_{S^*,i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{\hat{T}-1} \Big(g_{i_k}(\theta) - \tilde{\lambda} \Big), \tag{5}$$

where

$$\hat{T} = \min\{k > n \mid i_k \in S^*\}$$

is the first future time that a state in the set S^* is visited.

In the second approach, we introduce a discount factor $\alpha \in (0,1)$ and form the estimate

$$\tilde{v}_{\alpha,i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{T-1} \alpha^k \Big(g_{i_k}(\theta) - \tilde{\lambda} \Big), \tag{6}$$

where $T = \min\{k > n \mid i_k = i^*\}$ is the first future time the state i^* is visited.

3.3 An Algorithm that Updates at Every Time Step

By rearranging terms (see [MT01]), we can rewrite the estimate $F_m(\theta, \tilde{\lambda})$ in the following form

$$F_m(\theta, \tilde{\lambda}) \quad = \quad \nabla g_{i^*}(\theta) + \sum_{k=t_m+1}^{t_{m+1}-1} \Big(\nabla g_{i_k}(\theta) + (g_{i_k}(\theta) - \tilde{\lambda}) z_k \Big),$$

where

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

is a vector (of the same dimension as θ) that becomes available at time k.

Using this expression, we obtain the following algorithm which updates the parameter vector at each 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

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

where η is a positive scalar, and γ_k is a step size parameter. We then simulate a transition to the next state i_{k+1} according to the transition probabilities $P_{ij}(\theta_{k+1})$, and update z by letting

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

From a theoretical point of view, the algorithm of this subsection and Subsection 3.1 differ only by certain small terms that are of second order in the stepsize. This is because θ_k moves by $O(\gamma)$ between successive visits to i^*. Under a minor additional assumption on the step size (which again holds for $\gamma_k = 1/k$) and on the recurrence property of the state i^*, it can be shown that such $O(\gamma^2)$ modifications do not affect the asymptotic behavior and that this algorithm converges, i.e., $\lambda(\theta_k)$ converges and

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

with probability 1 (we refer to [Mar98, MT01] for a detailed proof). On the other hand, there are clear practical advantages when $E_\theta[T]$ is very large.

In the remainder of the paper, we incorporate the variance reducing estimates of $v_{i_n}(\theta)$ of Section 3.2 into the algorithms of Sections 3.1 and 3.3, and study the resulting biases and convergence properties.

4 Truncating the Sample Path to Reduce the Variance

In this section, we use Eq. (5) to produce an estimate of the gradient $\nabla\lambda(\theta)$, and then proceed to analyze the resulting gradient-like algorithms for tuning θ.

4.1 An Estimate of the Gradient $\nabla\lambda(\theta)$

Let the parameter $\theta \in \Re^K$ be fixed to some value and let $(i_1, i_2, \ldots)$ be a simulated trajectory of the Markov chain with transition probabilities $P_{ij}(\theta)$. Let t_m be the time of the mth visit to the recurrent state i^*. We fix a set $S^* \subset S$ containing i^*. Let $\kappa(m)$ be the number of times a state in the set S^* is visited in the interval $k = t_m + 1, \ldots, t_{m+1} - 1$, let $t_{m,n}$ be the time of the nth visit to such a state, and let $t_{m,0}$ and $t_{m,\kappa(m)+1}$ be equal to t_m and t_{m+1}, respectively. Using these definitions, we consider the estimate $F_{S^*,m}(\theta,\tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$ given by

$$F_{S^*,m}(\theta,\tilde{\lambda}) = \sum_{k=t_m}^{t_{m+1}-1} \Big(\tilde{v}_{S^*,i_k}(\theta,\tilde{\lambda})L_{i_{k-1}i_k}(\theta) + \nabla g_{i_k}(\theta)\Big), \tag{7}$$

where, for $t_{m,n} \le k < t_{m,n+1}$, $n = 0, \ldots, \kappa(m)$, we set

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

For $k = t_m$, we let $\tilde{v}_{i_k}(\theta, \tilde{\lambda}) = 0$.

We define

$$f_{S^*}(\theta, \tilde{\lambda}) = E_\theta[F_{S^*,m}(\theta, \tilde{\lambda})],$$

and we have the following result. The proof parallels the proof of Proposition 1 given in [MT01], and is omitted.

Proposition 3 *We have*

$$f_{S^*}(\theta, \tilde{\lambda}) = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_{S^*,j}(\theta) \right) + G_{S^*}(\theta)(\lambda(\theta) - \tilde{\lambda}),$$

where

$$G_{S^*}(\theta) \quad = \quad E_\theta \left[\sum_{n=t_m+1}^{t_{m,1}-1} (t_{m,1} - n) L_{i_{n-1}j_n}(\theta) + \sum_{k=1}^{\kappa(m)} \sum_{n=t_{m,k}}^{t_{m,k+1}-1} (t_{m,k+1} - n) L_{i_{n-1}j_n}(\theta) \right],$$

and

$$\begin{aligned} v_{S^*,j}(\theta) &= E_\theta \left[\sum_{k=0}^{\hat{T}-1} (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = j \right], \qquad j \in S \backslash \{i^*\}, \\ v_{S^*,i^*}(\theta) &= 0, \end{aligned}$$

with $\hat{T} = \min\{k > 0 \mid i_k \in S^*\}$ being the first future time that the set S^* is visited.

Note that the expression for $f_{S^*}(\theta, \tilde{\lambda})$ in Proposition 3 is of the same form as the expectation of the original estimate $F_m(\theta, \tilde{\lambda})$ given in Proposition 1, except that the bias term $G(\theta)(\tilde{\lambda} - \lambda(\theta))$ in Proposition 1 is replaced by $G_{S^*}(\theta)(\tilde{\lambda} - \lambda(\theta))$, and the exact value of the differential reward $v_j(\theta)$ is replaced by the approximation $v_{S^*,j}(\theta)$. Replacing $G(\theta)$ by $G_{S^*}(\theta)$ is inconsequential to the behavior of the algorithm. Replacing $v_j(\theta)$ with $v_{S^*,j}(\theta)$ introduces an additional bias $E_\theta[T]\sigma_{S^*}(\theta)$, where

$$\sigma_{S^*}(\theta) \quad = \quad \sum_{i \in S} \pi_i(\theta) \left(\sum_{j \in S} \nabla P_{ij}(\theta)(v_{S^*,j}(\theta) - v_j(\theta)) \right).$$

4.2 A Bound on the Bias $\sigma_{S^*}(\theta)$

In this subsection, we derive an upper bound on the magnitude of $\sigma_{S^*}(\theta)$. Clearly, $\sigma_{S^*}(\theta)$ will be small if the difference $v_{S^*,i}(\theta) - v_i(\theta)$ is small for every θ and every i. However, a weaker condition is possible. In particular, using the fact that

$$\sum_{j \in S} \nabla P_{ij}(\theta) = 0, \quad \text{for all } i \in S \text{ and all } \theta \in \Re^K,$$

we see that the bias $\sigma_{S^*}(\theta)$ will be zero, if for any i, and all j such that $\nabla P_{ij}(\theta) \neq 0$, $v_{S^*,j}(\theta) - v_j(\theta)$ takes a constant (possibly nonzero) value, that could also depend on i. This leads us to the following definitions. Let

$$S_i = \left\{ j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K \right\},$$

and

$$\hat{v}_{S^*,i}(\theta) = v_{S^*,i}(\theta) - v_i(\theta).$$

Also, let $\bar{N}$ be given by

$$\bar{N} = \max\{|S_i| \mid i \in S\},$$

where $|S_i|$ is the number of states in the set S_i. Thus, $\bar{N}$ bounds the number of possible transitions from any state $i \in S$ whose probability depends on θ. We have the following result.

Proposition 4 *Let Assumption 1 (Recurrence) and Assumption 2 (Regularity) hold. Furthermore, let ϵ be such that, for all states $i \in S$, we have*

$$\left|\hat{v}_{S^*,j}(\theta) - \hat{v}_{S^*,j'}(\theta)\right| \leq \epsilon, \qquad \text{if } j, j' \in S_i.$$

Then

$$\|\sigma_{S^*}(\theta)\| \leq \bar{N} C \epsilon,$$

where C is a bound on $\|\nabla P_{ij}(\theta)\|$.

Note that by Lemma 1 and by Assumption 2 (Regularity) the bound C on $\|\nabla P_{ij}(\theta)\|$ is finite.

Proof: For every state $i \in S$, let $j_r(i)$ be a state in

$$S_i = \left\{ j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K \right\}$$

which we use as a reference state. By assumption, for all states $i \in S$ and for all states $j \in S_i$, we have

$$\left|\hat{v}_{S^*,j}(\theta) - \hat{v}_{S^*,j_r(i)}(\theta)\right| \leq \epsilon.$$

Using the fact

$$\sum_{j \in S} \nabla P_{ij}(\theta) = 0,$$

it follows that

$$\begin{aligned}
\|\sigma_{S^*}(\theta)\| &= \left\| \sum_{i \in S} \pi_i(\theta) \sum_{j \in S_i} \nabla P_{ij}(\theta) \hat{v}_{S^*,j}(\theta) \right\| \\
&= \left\| \sum_{i \in S} \pi_i(\theta) \sum_{j \in S_i} \nabla P_{ij}(\theta) \Big(\hat{v}_{S^*,j}(\theta) - \hat{v}_{S^*,j_r(i)}(\theta) \Big) \right\| \\
&\leq \bar{N} C \epsilon,
\end{aligned}$$

which completes the proof. □

Proposition 4 suggests that in order to keep the bias $\sigma_{S^*}(\theta)$ small one should choose S^* such that, for all states $i \in S$ and for all states $j, j' \in S_i$, the difference $\left|\hat{v}_{S^*,j}(\theta) - \hat{v}_{S^*,j'}(\theta)\right|$ is small. The following example illustrates this result.

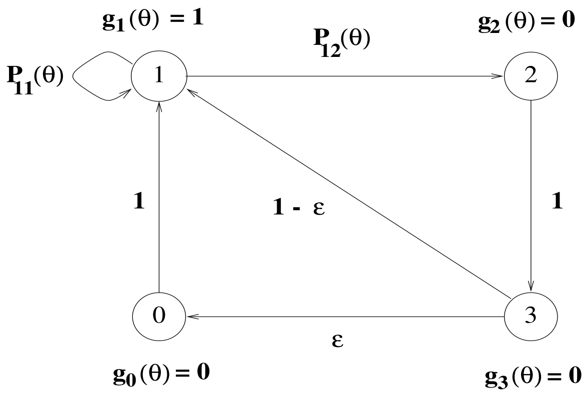

Figure 1: Structure of the Markov reward process in Example 1.

Example 1 Let ϵ be a scalar in $(0, 1]$, and consider the Markov reward process on the state space $S = \{0, 1, 2, 3\}$, with transition probabilities

$$\begin{aligned} P_{01}(\theta) &= P_{23}(\theta) = 1, \\ P_{11}(\theta) &= \frac{1}{2}\left[\frac{\exp(\theta)}{1+\exp(\theta)}\right], \\ P_{12}(\theta) &= 1 - P_{11}(\theta), \\ P_{30}(\theta) &= \epsilon, \qquad P_{31}(\theta) = 1 - \epsilon, \end{aligned}$$

and one-stage rewards

$$g_1(\theta) = 1, \quad \text{and} \quad g_0(\theta) = g_2(\theta) = g_3(\theta) = 0.$$

The structure of this Markov reward process is given in Figure 1. Note that Assumption 1 (Recurrence) and Assumption 2 (Regularity) are satisfied, with state 0 serving as the recurrent state i^*. Define the set S^* to be $\{0, 3\}$ and consider the estimates $\tilde{v}_{S^*,1}(\theta, \tilde{\lambda})$ and $\tilde{v}_1(\theta, \tilde{\lambda})$ of the differential reward of state 1. Note that $\tilde{v}_{S^*,1}(\theta, \tilde{\lambda})$ has the same distribution as the random variable X_θ described by

$$P\Big(X_\theta = n(1-\tilde{\lambda}) - \tilde{\lambda}\Big) = \Big(1 - P_{12}(\theta)\Big)^{n-1} P_{12}(\theta), \quad n = 1, 2, \ldots,$$

and that the estimate $\tilde{v}_1(\theta, \tilde{\lambda})$ has the same distribution as the random variable

$$Y_\theta = \sum_{n=1}^{N} \Big(X_{\theta,n} - \tilde{\lambda}\Big),$$

where $(X_{\theta,n})$ is a sequence of IID random variables with the distribution of X_θ, and N is a random variable, independent of $(X_{\theta,n})$, with distribution

$$P(N = n) = (1-\epsilon)^{n-1}\epsilon, \quad n = 1, 2, \ldots \quad .$$

Using $\tilde{v}_{S^*,1}(\theta,\tilde{\lambda})$, instead of $\tilde{v}_1(\theta,\tilde{\lambda})$, then reduces the variance by the factor

$$\frac{\mathrm{Var}\left(\tilde{v}_1(\theta,\tilde{\lambda})\right)}{\mathrm{Var}\left(\tilde{v}_{S^*,1}(\theta,\tilde{\lambda})\right)} = \frac{E[N]\mathrm{Var}(X_\theta) + E[(X_\theta - \tilde{\lambda})^2]\mathrm{Var}(N)}{\mathrm{Var}(X_\theta)} \geq E[N] = \frac{1}{\epsilon},$$

which becomes large when ϵ becomes small.

Furthermore, note that the set $S_1 = \{j \in S \mid \nabla P_{1j}(\theta) \neq 0 \text{ for some } \theta \in \Re\} = \{1,2\}$, $S_0 = S_2 = S_3 = \emptyset$, and

$$\hat{v}_{S^*,1}(\theta) = \hat{v}_{S^*,2}(\theta) = v_3(\theta).$$

Therefore, by Proposition 4, using $\tilde{v}_{S^*,i}(\theta,\tilde{\lambda})$ can significantly reduce the variance in the estimate of the differential reward $v_1(\theta)$ without introducing a bias into the estimate of the gradient $\nabla\lambda(\theta)$.

An important special case of Proposition 4 arises when the set S^* has the following property:

$$|v_i(\theta) - v_{i^*}(\theta)| \leq \frac{\epsilon}{2}, \qquad \text{for all } \theta \in \Re^K \text{ and all } i \in S^*; \tag{8}$$

since $v_{i^*}(\theta) = 0$, this is the same as requiring $|v_i(\theta)| \leq \epsilon/2$, for all $i \in S^*$ and all θ. It is then easily verified that

$$|\hat{v}_{S^*,j}(\theta)| \leq \epsilon/2,$$

for all j, and the assumption in Proposition 4 is satisfied. In practice, condition (8) can be satisfied by picking S^* to be small enough so that $v_i(\theta)$ does not vary much within the set S^*, but should also be large enough so that the set S^* is typically entered much earlier than the state i^* is visited (see Section 8.1.1 for an application).

4.3 An Algorithm that Updates at Visits to the Recurrent State

We will now use the estimate $F_{S^*,m}(\theta,\tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$ to formulate an algorithm which updates the parameter vector θ at visits to the recurrent state i^*. Again, we use the variable m to index the times when the state i^* is visited and the corresponding updates. At the time t_m, 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

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

where η is a positive scalar, and γ_k is a step size parameter. We have the following result.

Proposition 5 *Let Assumption 1 (Recurrence), Assumption 2 (Regularity), and Assumption 3 (Step Size) hold, and let D be such that, for all $\theta \in \Re^K$, we have*

$$\|\sigma_{S^*}(\theta)\| \le D,$$

where $\sigma_{S^*}(\theta)$ is as in Proposition 4. Furthermore, let $\{\theta_m\}$ be the sequence of parameter vectors generated by the above described algorithm. Then,

$$\liminf_{m\to\infty} \|\nabla\lambda(\theta_m)\| \le D,$$

with probability 1.

We defer the proof of this proposition to Appendix A.

Proposition 5 establishes that if the bias $\|\sigma_{S^*}(\theta)\|$ is small, then the gradient $\nabla\lambda(\theta_m)$ is small at infinitely many visits to the recurrent state i^*.

4.4 An Algorithm that Updates at Every Time Step

Similar to Section 3.2, we can break down the total update $F_{S^*,m}(\theta_m, \tilde{\lambda}_m)$ in the algorithm in Section 4.3 into a sum of incremental updates carried out at each time step, and derive the following algorithm which updates the parameter vector at each time step. At a typical time k, the state is i_k, and the values of θ_k, $\tilde{\lambda}_k$ and z_k, are available from the previous iteration. We update θ according to

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

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

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

Similar to Subsection 3.3, it can be shown that the conclusions of Proposition 5 remain valid for this algorithm as well, under some minor additional assumptions (see [Mar98]).

5 Using a Discount Factor to Reduce the Variance

Given a simulated trajectory $(i_0, i_1, ...)$ under the parameter θ, we produce in this section an estimate of the gradient $\nabla\lambda(\theta)$ by using another expression for estimating the differential reward of state i_n, namely

$$\tilde{v}_{\alpha,i_n}(\theta, \tilde{\lambda}) = \sum_{k=n}^{T-1} \alpha^k \Big(g_{i_k}(\theta) - \tilde{\lambda}\Big), \tag{9}$$

where $T = \min\{k > n \mid i_k = i^*\}$ is the first future time that the state i^* is visited and $\alpha \in (0,1)$ is a discount factor.

5.1 An Estimate of the Gradient $\nabla\lambda(\theta)$

Let the parameter $\theta \in \Re^K$ be fixed to some value and let $(i_1, i_2, \ldots)$ be a simulated trajectory of the Markov chain with transition probabilities $P_{ij}(\theta)$. Furthermore, let t_m be the time of the mth visit at the recurrent state i^* and consider the following estimate $F_{\alpha,m}(\theta, \tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$,

$$F_{\alpha,m}(\theta, \tilde{\lambda}) = \sum_{n=t_m}^{t_{m+1}-1} \left(\tilde{v}_{\alpha,i_n}(\theta, \tilde{\lambda}) L_{i_{n-1}i_n}(\theta) + \nabla g_{i_n}(\theta) \right), \tag{10}$$

where, for $t_m \leq n \leq t_{m+1} - 1$, we set

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

We have the following result for $f_\alpha(\theta, \tilde{\lambda})$, which we define to be the expected value of $F_{\alpha,m}(\theta, \tilde{\lambda})$, namely,

$$f_\alpha(\theta, \tilde{\lambda}) = E_\theta[F_{\alpha,m}(\theta, \tilde{\lambda})].$$

Proposition 6 *We have*

$$f_\alpha(\theta, \tilde{\lambda}) = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) v_{\alpha,j}(\theta) \right) + G_\alpha(\theta)(\lambda(\theta) - \tilde{\lambda}),$$

where

$$G_\alpha(\theta) = E_\theta \left[\sum_{n=t_m}^{t_{m+1}-1} \sum_{k=n}^{t_{m+1}-1} \alpha^{k-n} L_{i_{n-1}i_n}(\theta) \right],$$

and

$$v_{\alpha,j}(\theta) = E_\theta \left[\sum_{k=0}^{T-1} \alpha^k (g_{i_k}(\theta) - \lambda(\theta)) \mid i_0 = j \right], \qquad j \in S,$$

with $T = \min\{k > 0 \mid i_k = i^*\}$ being the first future time that the state i^* is visited.

As in Section 4, the expression for $f_\alpha(\theta, \tilde{\lambda})$ in Proposition 6 is of the same form as the expectation of the original estimate $F_m(\theta, \tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$, except that the bias term $G(\theta)(\tilde{\lambda} - \lambda(\theta))$ is replaced by $G_\alpha(\theta)(\tilde{\lambda} - \lambda(\theta))$, and the exact value of the differential reward $v_j(\theta)$ is replaced by the approximation $v_{\alpha,j}(\theta)$.

5.2 A Bound on the Bias $\sigma_\alpha(\theta)$

In this subsection, we analyze the bias

$$E_\theta[T]\sigma_\alpha(\theta) = E_\theta[T] \sum_{i \in S} \pi_i(\theta) \left(\sum_{j \in S} \nabla P_{ij}(\theta) \left(v_{\alpha,j}(\theta) - v_j(\theta) \right) \right),$$

which is due to replacing $v_j(\theta)$ with $v_{\alpha,j}(\theta)$, and derive a bound for the magnitude of $\sigma_\alpha(\theta)$.

To do that, we consider the "mixing behavior" of the Markov reward process, i.e. we define scalars A and β, with $0 \leq \beta < 1$, such that, for all states $i \in S$ and all integers $n \geq 0$, we have

$$\Big| \sum_{j \in S} \Big(P_{ij}^n(\theta) - \pi_j(\theta) \Big) g_j(\theta) \Big| \leq A\beta^n.$$

Such constants are guaranteed to exist under Assumption 1 (Recurrence). In particular, β can be taken to be an upper bound on the second largest of the magnitudes of the eigenvalues of the stochastic matrices $P(\theta)$. This setting becomes interesting when β is small relative to α, which corresponds to "fast mixing". Let us emphasize, however, that the value of β may turn out to be small even if the Markov chain takes a long time to reach equilibrium. All that is required is that the expected reward $\sum_{j \in S} P_{ij}^n(\theta) g_j(\theta)$, n steps into the future, can be well approximated by the average reward $\sum_{j \in S} \pi_j(\theta) g_j(\theta) = \lambda(\theta)$. In other words, the value of β is not determined by how long it takes for the chain to reach steady-state, but rather by how long it takes for the rewards to reach steady-state.

Proposition 7 *Let Assumption 1 (Recurrence) and Assumption 2 (Regularity) hold. Furthermore, let the constants β, $0 \leq \beta < 1$, and A, be such that for all $\theta \in \Re^K$, for all $i \in S$, and all integers $n \geq 0$, we have*

$$\Big| \sum_{j \in S} \Big(P_{ij}^n(\theta) - \pi_j(\theta) \Big) g_j(\theta) \Big| \leq A\beta^n.$$

We then have

$$\|\sigma_\alpha(\theta)\| \quad \leq \quad \frac{AC\bar{N}}{1-\alpha\beta} \left(\frac{\beta(1-\alpha)}{1-\beta} + \sum_{i \in S} \pi_i(\theta) E_\theta \Big[\alpha^T \mid i_0 = i \Big] \right),$$

where $T = \min\{k > 0 \mid i_k = i^*\}$, C is a bound on $\|\nabla P_{ij}(\theta)\|$, and $\bar{N}$ is the same bound as in Proposition 4.

Proof: We introduce some additional notation. For any $\alpha \in [0,1]$, we let $v_{\alpha,i}^\infty(\theta)$, $i \in S$, be given by

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

The above infinite infinite sum is well-defined and finite even for the special case where $\alpha = 1$, because under our assumptions, the summands converge exponentially fast to zero. Furthermore, the resulting differential reward $v_{1,i}^\infty(\theta)$ turns out to be the same as the earlier defined $v_i(\theta)$, modulo an additive constant. That is, there exists a constant c such that

$$v_i(\theta) = v_{1,i}^\infty(\theta) + c, \qquad \text{for all } i. \tag{11}$$

We then have

$$\begin{aligned} \sigma_\alpha(\theta) \quad &= \quad \sum_{i \in S} \pi_i(\theta) \left(\sum_{j \in S} \nabla P_{ij}(\theta) \Big(v_{\alpha,j}(\theta) - v_{\alpha,j}^\infty(\theta) \Big) \right) \\ &\quad + \sum_{i \in S} \pi_i(\theta) \left(\sum_{j \in S} \nabla P_{ij}(\theta) \Big(v_{\alpha,j}^\infty(\theta) - v_{1,j}^\infty(\theta) \Big) \right) \end{aligned}$$

$$+\sum_{i\in S}\pi_i(\theta)\left(\sum_{j\in S}\nabla P_{ij}(\theta)\Big(v^{\infty}_{1,j}(\theta)-v_j(\theta)\Big)\right). \qquad (12)$$

Using Eq. (11) and the property $\sum_j \nabla P_{ij}(\theta)=0$, the third sum above vanishes.

Let us consider the second sum. Using also the property $\lambda(\theta)=\sum_{j\in S}\pi_j(\theta)g_j(\theta)$, we obtain

$$\begin{aligned}
|v^{\infty}_{\alpha,i}(\theta)-v^{\infty}_{1,i}(\theta)| &\leq \sum_{k=0}^{\infty}|\alpha^k-1|\left|\sum_{j\in S}P^k_{ij}(\theta)\Big(g_j(\theta)-\lambda(\theta)\Big)\right| \\
&= \sum_{k=0}^{\infty}(1-\alpha^k)\left|\sum_{j\in S}\Big(P^k_{ij}(\theta)-\pi_j(\theta)\Big)g_j(\theta)\right| \\
&\leq A\sum_{k=0}^{\infty}(1-\alpha^k)\beta^k \\
&= A\frac{\beta}{1-\beta}\cdot\frac{1-\alpha}{1-\alpha\beta}
\end{aligned}$$

and

$$\left\|\sum_{i\in S}\pi_i(\theta)\left(\sum_{j\in S}\nabla P_{ij}(\theta)\Big(v^{\infty}_{\alpha,j}(\theta)-v^{\infty}_{1,j}(\theta)\Big)\right)\right\|\leq\frac{AC\bar{N}}{1-\alpha\beta}\cdot\frac{\beta(1-\alpha)}{1-\beta}.$$

We finally provide a bound for the first sum. By Assumption 2 (Regularity) and Lemma 1, $|g_{i_k}(\theta)-\lambda(\theta)|$ is bounded and we have, for $\alpha\in(0,1)$,

$$\begin{aligned}
v^{\infty}_{\alpha,i}(\theta) &= \sum_{k=0}^{\infty}\alpha^k E_\theta\left[(g_{i_k}(\theta)-\lambda(\theta))\mid i_0=i\right] \\
&= E_\theta\left[\sum_{k=0}^{\infty}\alpha^k(g_{i_k}(\theta)-\lambda(\theta))\mid i_0=i\right].
\end{aligned}$$

It follows that

$$\begin{aligned}
|v_{\alpha,i}(\theta)-v^{\infty}_{\alpha,i}(\theta)| &= \left|E_\theta\left[\sum_{k=T}^{\infty}\alpha^k\Big(g_{i_k}(\theta)-\lambda(\theta)\Big)\mid i_0=i\right]\right| \\
&= \left|\sum_{t=1}^{\infty}E_\theta\left[\sum_{k=T}^{\infty}\alpha^k\Big(g_{i_k}(\theta)-\lambda(\theta)\Big)\mid i_0=i,T=t\right]P_\theta(T=t\mid i_0=i)\right| \\
&= \left|\sum_{t=1}^{\infty}\alpha^t E_\theta\left[\sum_{k=0}^{\infty}\alpha^k\Big(g_{i_k}(\theta)-\lambda(\theta)\Big)\mid i_0=i^*\right]P_\theta(T=t\mid i_0=i)\right| \\
&\leq E_\theta[\alpha^T\mid i_0=i]\sum_{k=0}^{\infty}\alpha^k\left|\sum_{j\in S}P^k_{i^*j}\Big(g_j(\theta)-\lambda(\theta)\Big)\right| \\
&= E_\theta[\alpha^T\mid i_0=i]\sum_{k=0}^{\infty}\alpha^k\left|\sum_{j\in S}\Big(P^k_{i^*j}-\pi_j(\theta)\Big)g_j(\theta)\right|
\end{aligned}$$

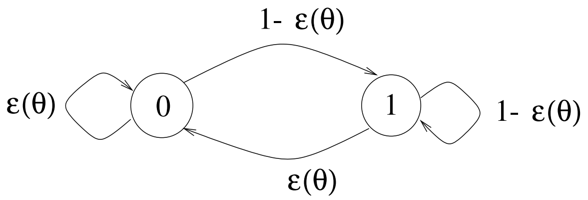

Figure 2: Structure of the Markov reward process in Example 2.

$$
\begin{aligned}
&\leq \quad E_\theta[\alpha^T \mid i_0 = i] \cdot A \sum_{k=0}^{\infty} \alpha^k \beta^k \\
&= \quad E_\theta[\alpha^T \mid i_0 = i] \frac{A}{1-\alpha\beta},
\end{aligned}
$$

and we obtain

$$
\left\| \sum_{i\in S} \pi_i(\theta) \left(\sum_{j\in S} \nabla P_{ij}(\theta) \Big(v_{\alpha,j}(\theta) - v^{\infty}_{\alpha,j}(\theta) \Big) \right) \right\| \quad \leq \quad \frac{AC\bar{N}}{1-\alpha\beta} \cdot \sum_{i\in S} \pi_i(\theta) E_\theta[\alpha^T \mid i_0 = i].
$$

The result then follows. □

Proposition 7 indicates that the bias will be small as long as β is moderate (not too close to 1, which corresponds to fast mixing), the discount factor is chosen large enough so that $1-\alpha$ is significantly smaller than $1-\beta$, and $\sum_{i\in S} \pi_i(\theta) E_\theta[\alpha^T \mid i_0 = i]$ is also small. Note that the latter term will be small if the time T to reach i^* starting from a random state [drawn according to the steady-state distribution $\pi(\theta)$], is large; this is the typical case in large scale problems. The following example serves as an illustration.

Example 2 We consider the Markov chain given in Fig. 2, where we set

$$
\epsilon(\theta) = \frac{\exp(\theta)}{1+\exp(\theta)}.
$$

Note that $\|\nabla P_{ij}(\theta)\| < \frac{1}{4}$, for all $\theta \in \Re$. The steady state probabilities are

$$
\pi_0(\theta) = \epsilon(\theta) \quad \text{and} \quad \pi_1(\theta) = 1 - \epsilon(\theta),
$$

which implies that

$$
\sum_{j\in S} \Big(P_{ij}(\theta) - \pi_j(\theta) \Big) = 0, \qquad \text{for } i = 0, 1,
$$

and the process reaches steady state in a single step. Therefore, we can set the constants β, and A, of Proposition 7 equal to 0, and 1, respectively. Choosing the state $i = 0$ as the recurrent state i^*, we obtain

$$
E_\theta[\alpha^T \mid i_0 = 0] = E_\theta[\alpha^T \mid i_0 = 1] = \frac{\epsilon(\theta)\alpha}{1-\alpha(1-\epsilon(\theta))},
$$

and the bound on the bias becomes

$$\|\sigma_\alpha(\theta)\| \leq \frac{1}{2} \cdot \frac{\epsilon(\theta)\alpha}{1 - \alpha(1 - \epsilon(\theta))}.$$

Thus, the bias can be made arbitrarily small by letting α approach 0.

5.3 An Algorithm that Updates at Visits to the Recurrent State

Using the estimate $F_{\alpha,m}(\theta, \tilde{\lambda})$ of the gradient $\nabla\lambda(\theta)$, we can formulate an algorithm which updates the parameter vector θ at visits to the state i^* according to

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

where η is a positive scalar, and where γ_m is a step size parameter for which Assumption 3 (Step Size) holds. This situation is identical to the one in Section 4.3 and Proposition 5 remains valid, except that D must now stand for the bound on $\sigma_\alpha(\theta)$.

With this algorithm, the time between updates, and the resulting variance, would still be large. This problem is alleviated by the variant that we introduce next.

5.4 An Algorithm that Updates at Every Time Step

We now consider a variant of the algorithm of the preceding subsection but in which an update is carried out at each step. At a typical time k, the state is i_k, and the values of θ_k and z_k, are available from the previous iteration. We update θ according to

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

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

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

Similar to Subsection 3.3 and 4.3, under some minor additional assumptions (see [Mar98]), the algorithm of this and the preceding subsection exhibit the same asymptotic behavior.

5.5 A Modified Estimate

Instead of using the expression given by Eq. (9), we could estimate the differential reward of state i by

$$\tilde{v}_{\alpha,i}^{\infty}(\theta, \tilde{\lambda}) = \sum_{k=0}^{\infty} \alpha^k \Big(g_{i_k}(\theta) - \tilde{\lambda} \Big).$$

Using this new estimate, one obtains an algorithm that updates at each time step and which is identical to the one in the preceding subsection, except that the state i^* does not play a special role, and the vector z_k is not reset at visits to the recurrent state i^*.

In [BB99], it is shown that, as long as θ is unchanged, the estimate $\tilde{v}^{\infty}_{\alpha,i}(\theta,\tilde{\lambda})$ can be used to produce an estimate of the gradient $\nabla\lambda(\theta)$ for which the bound on the bias is proportional to $(1-\alpha)/(1-\beta)$. The same conclusion is obtained with our approach: with this new estimate, the first sum in Eq. (12) disappears, and the term $\sum_{i\in S}\pi_i(\theta)E_\theta\left[\alpha^T \mid i_0 = i\right]$ is eliminated from the bias bound of Proposition 7. In this respect the variant considered in this subsection has a somewhat better bias bound. However, from a practical point of view, the two algorithms are essentially the same. If the visits to i^* are very rare (as is typical in large problems), the term $\sum_{i\in S}\pi_i(\theta)E_\theta\left[\alpha^T \mid i_0 = i\right]$ in the bias bound is very small. This reflects the fact that the term $\tilde{v}^{\infty}_{\alpha,i}(\theta,\tilde{\lambda}) - \tilde{v}_{\alpha,i}(\theta,\tilde{\lambda})$ which causes the difference between the two algorithms is very small with high probability.

From a mathematical point of view, the convergence analysis of the modified algorithm discussed here is actually much more involved, because the updates during a "renewal cycle" are affected by discounted terms originating in previous renewal cycles. This introduces certain dependencies and martingale-based tools are harder to apply. We feel that the difference between the two algorithms is not significant enough to warrant a long separate proof.

6 Optimizing the Weighted Reward-to-Go

In this section, we outline how the methodology of the previous sections can be applied to Markov reward processes with a reward-free termination state i^* that is eventually reached from every initial state $i \in S$. We make the following assumption.

Assumption 4 (Termination) *There exists a state $i^* \in S$, such that, for every parameter vector $\theta \in \Re^K$, we have*

$$g_{i^*}(\theta) = 0 \qquad \textit{and} \qquad P_{i^*i^*}(\theta) = 1,$$

and, for every state $i \in S$ and every transition matrix $P \in \overline{\mathcal{P}}$, we have

$$P^N_{ii^*} > 0,$$

where N is the number of states in the state space S.

When Assumption 4 (Termination) holds, the reward-to-go $J_i(\theta)$ of state i is defined as

$$J_i(\theta) = E_\theta\left[\sum_{k=0}^{\infty} g_{i_k}(\theta) \mid i_0 = i\right] = E_\theta\left[\sum_{k=0}^{T-1} g_{i_k}(\theta) \mid i_0 = i\right], \tag{13}$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. Note that $J_{i^*}(\theta) = 0$.

In this setting, we associate with each possible parameter vector θ, a weighted reward performance measure $\chi(\theta)$, defined by

$$\chi(\theta) = \sum_{i\in S}\bar{\pi}_i J_i(\theta), \tag{14}$$

where $\bar{\pi} = (\bar{\pi}_1, ..., \bar{\pi}_N)$ is a given probability distribution on the state space S. This performance measure corresponds to a situation where a decision maker wants to maximize the expected reward-to-go, given that the initial state of the system is equal to i with probability $\bar{\pi}_i$.

We would like to point out that optimizing the weighted reward-to-go is not equivalent to finding a control policy which optimizes the reward-to-go simultaneously for all states (which is the goal of dynamic programming). We chose the weighted reward-to-go as an objective function because there might not exist a parameter vector $\theta^* \in \Re^K$ that maximizes the reward-to-go simultaneously for all states. However, if there exists a parameter vector $\theta^* \in \Re^K$, such that for all vectors $\theta \in \Re^K$ and for all states $i \in S$ we have

$$J_i(\theta^*) \geq J_i(\theta),$$

then θ^* maximizes the weighted average reward-to-go $\chi(\theta)$ for every probability distribution $\bar{\pi}$ over the state space S. In this case, maximizing the weighted reward-to-go is equivalent to finding a parameter θ^* which maximizes the reward-to-go for all states.

6.1 The Gradient of $\chi(\theta)$

In this section we derive an expression for the the gradient of the weighted reward-to-go $\chi(\theta)$ with respect to θ, and propose a gradient algorithm for tuning θ so as to improve $\chi(\theta)$.

We start out by defining a new Markov reward process on the state space S with transition probabilities

$$P_{\bar{\pi},ij} = \begin{cases} P_{ij}(\theta), & i \neq i^*, \\ \bar{\pi}_j, & i = i^*, \end{cases}$$

and one-stage rewards

$$g_{\bar{\pi},i}(\theta) = g_i(\theta).$$

Note that this new Markov reward process differs from the original one only in the transition probabilities from the termination state i^* to other states $j \in S$. These transition probabilities are now equal to $\bar{\pi}_j$. This means that whenever the termination state i^* is reached, then the new process moves to state j (and restarts the original process with j as the initial state) with probability $\bar{\pi}_j$.

In the following, we use the notation $P_{\bar{\pi},\theta}(\cdot)$, to denote the probability distribution induced by the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$. Accordingly, we use $E_{\bar{\pi},\theta}[\cdot]$ to indicate that the expectation are taken with respect to the probability distribution of the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$.

For the Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$, let $\pi_{\bar{\pi},i}(\theta)$ be the steady state probability distribution of being in state $i \in S$, let $\lambda_{\bar{\pi}}(\theta) = \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) g_{\bar{\pi},i}(\theta)$ be the average reward, and let $E_{\bar{\pi},\theta}[T]$ be the mean recurrence time, i.e., we have

$$E_{\bar{\pi},\theta}[T] = E_{\bar{\pi},\theta}\left[T \mid i_0 = i^*\right],$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time that state i^* is visited. We then obtain the the following proposition which gives an expression for the gradient of the weighted reward-to-go $\chi(\theta)$ with respect to θ (for a proof see [Mar98]).

Proposition 8 *Let Assumption 2 (Regularity) and Assumption 4 (Termination) hold. Then,*

$$\nabla \chi(\theta) = E_{\bar{\pi},\theta}[T] \sum_{i \in S} \pi_{\bar{\pi},i}(\theta) \left(\nabla g_i(\theta) + \sum_{j \in S} \nabla P_{ij}(\theta) J_j(\theta) \right).$$

6.2 Estimation of $\nabla\chi(\theta)$

Similar to Section 3, we rewrite the formula for $\nabla\chi(\theta)$ given by Proposition 8 in the form

$$\nabla\chi(\theta) = E_{\bar{\pi},\theta}[T]\sum_{i\in S}\pi_{\bar{\pi},i}(\theta)\left(\nabla g_i(\theta) + \sum_{j\in S} P_{ij}(\theta)L_{ij}(\theta)J_j(\theta)\right),$$

where $L_{ij}(\theta)$ is as in Assumption 2.

Let the parameter vector θ be fixed to some value, and let $\bar{\pi}$ be a given probability distribution on the state space S. Furthermore, let $(i_1, i_2, ...)$ be a sample path of the corresponding Markov chain with transition probabilities $P_{\bar{\pi},ij}(\theta)$ and let t_m be the time of the mth visit at the termination state i^*. We refer to the sequence $i_{t_m}, i_{t_m+1}, \ldots, i_{t_{m+1}} - 1$ as the mth renewal cycle. Consider then the estimate of $\nabla\chi(\theta)$ given by

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

where

$$\tilde{J}_{i_n}(\theta) = \sum_{k=n}^{t_{m+1}-1} g_{i_k}(\theta), \qquad t_m < n \le t_{m+1} - 1,$$

is an estimate of the reward-to-go $J_{i_n}(\theta)$. Noting that $J_{i^*}(\theta) = 0$, we let

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

Note that the random variables $F_m(\theta)$ are independent and identically distributed for different values of m, because the transitions during distinct renewal cycles are independent. We define $f_{\bar{\pi}}(\theta)$ to be the expected value of $F_m(\theta)$, namely,

$$f_{\bar{\pi}}(\theta) = E_{\bar{\pi},\theta}[F_m(\theta)].$$

The following proposition confirms that the expectation of $F_m(\theta)$ is an unbiased estimate of $\nabla\chi(\theta)$ (for a proof see [Mar98]).

Proposition 9 *We have*

$$f_{\bar{\pi}}(\theta) = \nabla\chi(\theta).$$

6.3 An Algorithm that Updates at Visits to the Termination State

We now use the estimate of the gradient direction provided by Proposition 9 to propose a simulation-based algorithm that performs updates at visits to the state i^*. We use the variable m to index the times when the recurrent state i^* is visited, and the corresponding updates.

At the time t_m that state i^* is visited for the mth time, we have available a current vector θ_m. We then simulate the process according to the transition probabilities $P_{\bar{\pi},ij}(\theta_m)$ until the next time t_{m+1} that i^* is visited and update according to

$$\begin{aligned}\theta_{m+1} &= \theta_m + \gamma_m F_m(\theta_m)\\ &= \theta_m + \gamma_m \sum_{n=t_m}^{t_{m+1}-1}\left(\tilde{J}_{i_n}(\theta)L_{i_{n-1}i_n}(\theta) + \nabla g_{i_n}(\theta)\right)\end{aligned}$$

where γ_m is a positive step size parameter. We can rewrite this iteration as as

$$\theta_{m+1} = \theta_m + \gamma_m \nabla \chi(\theta_m) + \varepsilon_m,$$

where

$$\varepsilon_m = \gamma_m (F_m(\theta_m) - \nabla \chi(\theta_m)).$$

Proposition 9 the implies that we have

$$E_\theta[\varepsilon_m] = E_\theta[\gamma_m (F_m(\theta_m) - \nabla \chi(\theta_m))] = 0.$$

Therefore, the algorithm we propose here can be interpreted as a gradient algorithm with a stochastic error term ε_m that is a zero mean random vector. It is then not surprising that we have the following convergence result (for a proof see [Mar98]).

Proposition 10 *Let Assumption 2 (Regularity) , Assumption 4 (Termination) , and Assumption 3 (Step Size) hold, and let $\{\theta_m\}$ be the sequence of parameter vectors generated by the above described algorithm. Then, with probability 1, $\chi(\theta_m)$ converges and*

$$\lim_{m \to \infty} \nabla \chi(\theta_m) = 0.$$

An algorithm which updates the parameter vector at each time step, as described in Section 3.3, can also be derived based on the above update rule. Variance reducing modifications, in the spirit of Sections 4 and 5, are straightforward. Detailed descriptions of the resulting methods can be found in [Mar98] and are omitted from this paper.

7 Markov Decision Processes

As shown in [MT01], the algorithms of the previous sections can be applied to Markov decision processes that are defined on a finite state space $S = \{1, \ldots, N\}$ and a finite action space $U = \{1, \ldots, L\}$. At any state i, the choice of a control action $u \in U$ determines the transition probabilities $P_{ij}(u)$, and the one-stage rewards $g_i(u)$. We consider a parametrized family of randomized policies that associate with each parameter vector $\theta \in \Re^K$ the probability $\mu_u(i, \theta)$ that control action u is applied at state i. The corresponding transition probabilities are given by

$$P_{ij}(\theta) = \sum_{u \in U} \mu_u(i, \theta) P_{ij}(u), \tag{15}$$

and the expected reward per stage is given by

$$g_i(\theta) = \sum_{u \in U} \mu_u(i, \theta) g_i(u). \tag{16}$$

Our original algorithms, as described in Sections 3.1 and 3.3 were shown in [MT98] to have natural counterparts for the case of Markov decision processes. Variance reducing modifications, in the spirit of Sections 4 and 5, are straightforward. Detailed descriptions of the resulting methods can be found in [Mar98] and are omitted from this paper. A case study to illustrate the methodology is presented, however, in the next section. We would like to point out that the resulting algorithms do not require the transition probabilities $P_{ij}(u)$ and one-stage rewards $g_i(u)$ to be known, but only assume that the optimizer has access to an observation of a sample path and the associated rewards.

Table 1: Call Types.

CALL TYPE m	1	2	3
BANDWIDTH DEMAND $b(m)$	1	1	1
ARRIVAL RATE $\alpha(m)$	1.8	1.6	1.4
AVERAGE HOLDING TIME $1/\beta(m)$	1/0.6	1/0.5	1/0.4
IMMEDIATE REWARD $c(m)$	1	2	4

8 Numerical Results

As a case study, we use an admission control problem. More details on the experiments reported here can be found in [Mar98].

Consider a provider of a communication link with total bandwidth of B units, that supports a finite set $\{1, 2, \ldots, M\}$ of different call types. When a customer requests a new connection for a call, the provider can decide to reject, or, if enough bandwidth is available, to maybe accept the call. Once accepted, a call of class m seizes $b(m)$ units of bandwidth. Whenever a call of class m gets accepted, the provider receives an immediate reward of $c(m)$ units, which is the price the customer pays 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.

Assuming that class m calls arrive according to independent Poisson processes (with rate $\alpha(m)$), and that the holding times of class m calls are exponentially (and independently) distributed (with mean $1/\beta(m)$), the problem can be formulated as a discrete-time Markov decision process (see [Mar98]), where the state i is of the form $i = (s(1), \ldots, s(M), \omega)$. Here $s(m)$, $m = 1, ..., M$, 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) [2]

We define a randomized policy as a function of $\theta = (\theta(1), ..., \theta(M)) \in \Re^M$, where M is the number of different service types. The provider accepts a new call of class m with probability

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

where $s \cdot b = \sum_m s(m)b(m)$ is the currently occupied bandwidth. Note that

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

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

8.1 Experiments

We consider a link with a total bandwidth of $B = 10$ units, which supports three different call types (see Table 1). The number of link configurations (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, can serve as the recurrent state i^*.

[2]The event ω needs to be included into the state i, because the decision (accept or reject) and the associated reward depend explicitly on the type of the arriving call.

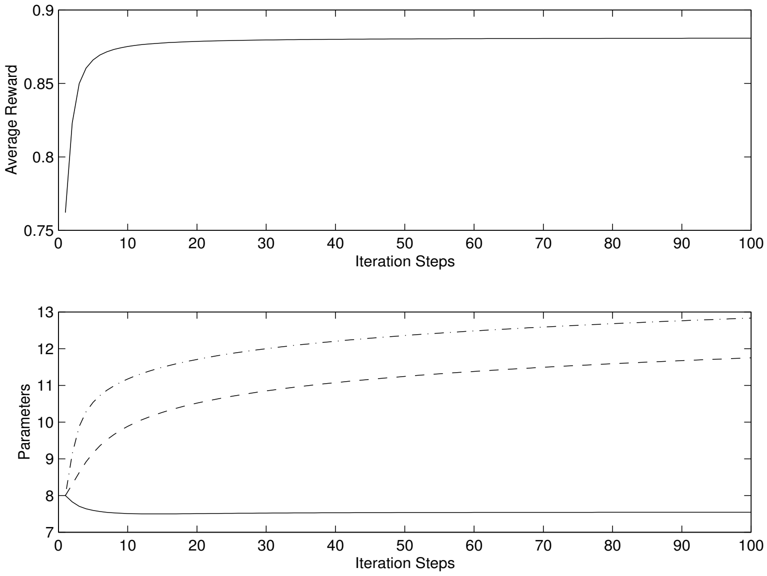

Figure 3: Parameter vectors θ_k and the average rewards $\lambda(\theta_k)$ (computed exactly) of the idealized gradient algorithm. The solid, dashed, and dash-dot line correspond to $\theta_k(1)$, $\theta_k(2)$, and $\theta_k(3)$, respectively. After 100 iterations, the parameter vector θ_{100} is equal to (7.5459,11.7511,12.8339) which corresponds to an average reward of 8.6318.

For this case, we can compute an optimal call admission control policy using methods of dynamic programming [Ber95a]. The policy accepts customers of service type 1 if the currently used bandwidth does not exceed the threshold value of 7 units, while customers of service type 2 and 3 get always accepted (if enough bandwidth is available). The corresponding optimal average reward is equal to 8.6902. In [MT01], we implemented for this problem an idealized gradient algorithm (where we used the exact value of $\nabla\lambda(\theta)$ to update the parameter vector) and the simulation-based algorithm of Section 3.3. As a reference, we give in Figure 3, and 4, the trajectories of the parameter vector and (estimates of the) average reward for the idealized algorithm, and the algorithm of Section 3.3, respectively[3]. Note that the simulation-based algorithm of Section 3.3 makes fast progress in the beginning, improving the average reward from 7.64 to 8.53 within $1 \cdot 10^6$ iteration steps. After $8 \cdot 10^6$ iterations, the average reward is 8.6064, which is still slightly below 8.6318, the average reward of the idealized gradient algorithm.

In the next two subsections, we apply the modified algorithms of Section 4.4 and 5.4 to this problem. Our result illustrate that these algorithms are robust with respect to a small bias term in the update direction and converge faster than the original simulation-based algorithm of Section 3.3.

[3]In [MT01], the average reward is given for the sampled discrete-time problem (as defined in [Mar98]) for a sampling rate equal to 9.8; the average reward for the continuous-time system (used in this paper) is then obtained by multiplying the average reward of the discrete-time system with the sampling rate 9.8.

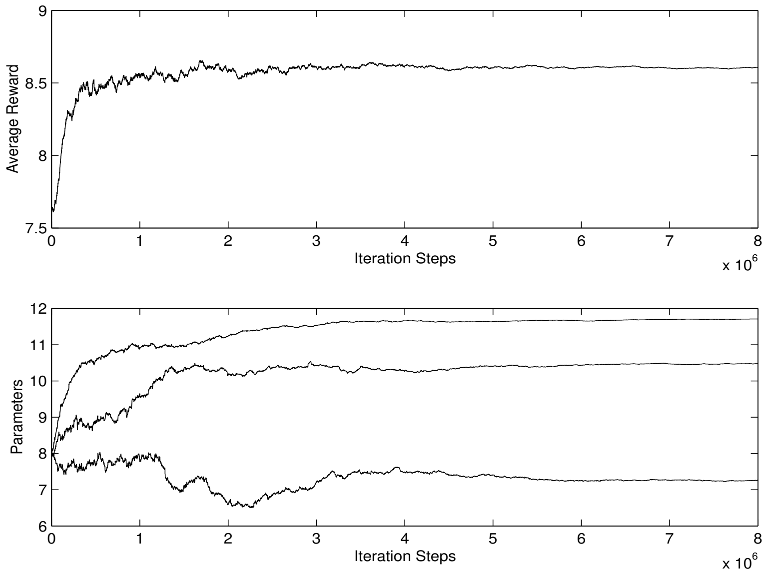

Figure 4: Parameters $\theta_k(1)$, $\theta_k(2)$, and $\theta_k(3)$, and estimates of the average reward $\tilde{\lambda}_k$, obtained by the algorithm which updates at each iteration step.

8.1.1 Modified Algorithm Using Truncated Sample Paths

Recall that any state $i = (s, \omega)$ in which $s = (0, \ldots, 0)$, and ω corresponds to an arrival of a new call, can serve as the recurrent state i^*. This leads us to consider a set S^* of the form

$$S^* = \left\{ i = (s, \omega) \in S \mid \sum_{m=1}^{3} s(m)b(m) \leq B_0 \right\}, \qquad B_0 > 0,$$

for the algorithm of Section 4.4 which uses truncated sample paths to reduce the variance. One would expect that the bias introduced by using the set S^* is small when B_0 is small. Figure 5 gives the trajectories of the parameter vector for the deterministic version of this algorithm (where we replaced the random estimate $F_{S^*,m}(\theta)$ used to update θ by its mean) for several values of B_0. Comparing Figure 5 with Figure 3 shows that the algorithm is robust in the presence of a small bias, and for B_0 equal to 5 and 7, the effect of the additional bias is negligible.

Using $B_0 = 7$, we implement the algorithm of Section 4.4 (see Figure 6). As expected, it makes much faster progress than the algorithm of Section 3.3. After 150,000 iterations steps the average reward is roughly equal to 8.53, and after $1 \cdot 10^6$ iterations the average reward is 8.6117 (which is even slightly higher than the one obtained with the original algorithm of Section 3.3).

8.1.2 Modified Algorithm Using a Discount Factor

Next, we consider the algorithm of Section 5.4 which uses a discount factor α, $0 < \alpha < 1$, to reduce the variance. First, we consider its deterministic version where we use the mean of the estimates $F_{\alpha,m}(\theta)$ to update the parameter vector θ. The resulting iterations for values of the discount factor α equal to 0.5, 0.9, 0.99, and 0.999, are given in Figure 7. We observe that the deterministic version of the algorithm is robust with respect to a small bias, and for α equal to 0.99 and 0.999, the effect

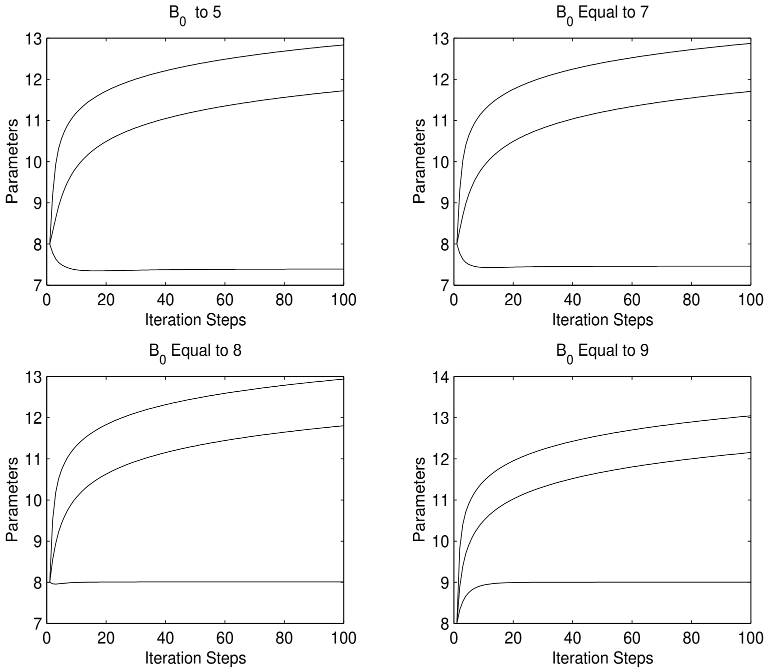

Figure 5: Parameters $\theta_k(1)$, $\theta_k(2)$, and $\theta_k(3)$, of the deterministic version of the algorithm which uses truncated sample paths.

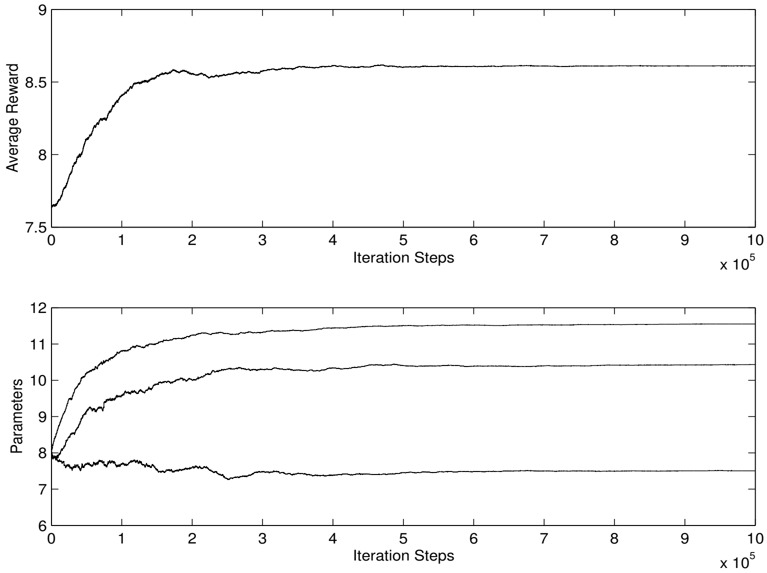

Figure 6: Parameters $\theta_k(1)$, $\theta_k(2)$, and $\theta_k(3)$, and estimates of the average reward $\tilde{\lambda}_k$, of the algorithm which uses truncated sample paths. The value for B_0 is set equal to 7.

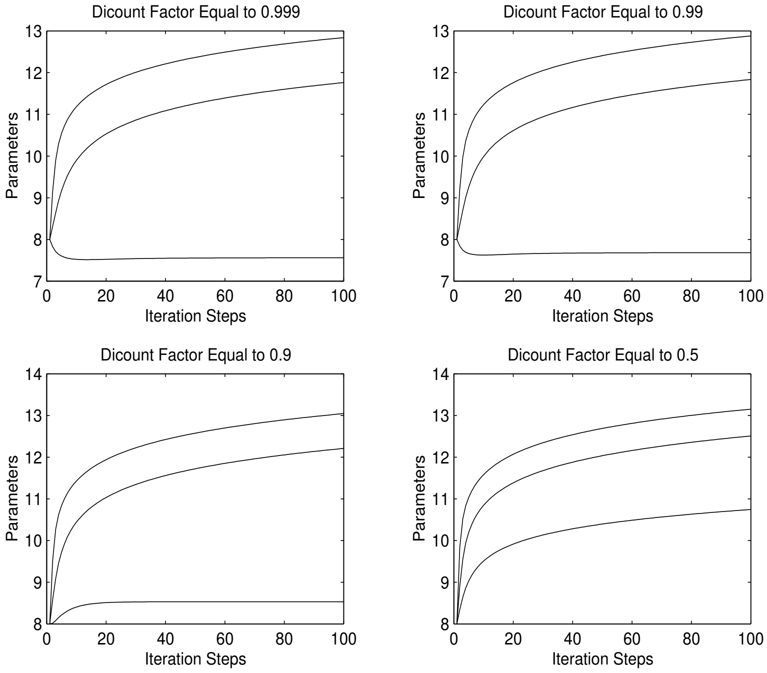

Figure 7: Parameters $\theta_k(1)$, $\theta_k(2)$ of the deterministic version of the algorithm which uses a discount factor.

of the additional bias is negligible. In the simulation-based version (implemented with $\alpha = 0.99$), the average reward is equal to 8.6128 after $1 \cdot 10^6$ iterations (see Figure 8).

9 Conclusions

We have proposed two approaches to reduce the variance of the updates in a simulation-based method for optimizing Markov reward processes that depend on a parameter vector. The resulting algorithms introduce an additional bias into the update direction, for which certain bounds were derived. In addition, we carried out a convergence analysis and showed that when the bias is small, then the algorithms will infinitely often lead to policies at which the gradient of average reward is small, and can therefore be expected to be close to a local optimum. The numerical results for an admission control problem are encouraging: compared with the original algorithm, the modified algorithms obtain essentially the same average reward, but converge much faster.

References

[BB99] J. Baxter and P. L. Bartlett. Direct Gradient-Based Reinforcement Learning: I. Gradient Estimation Algorithms. Unpublished manuscript, November 1999.

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

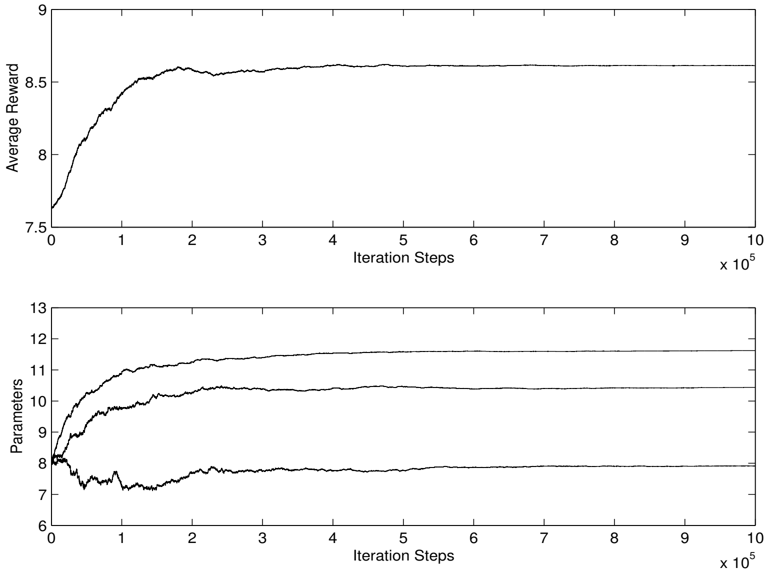

Figure 8: Parameters $\theta_k(1)$, $\theta_k(2)$, and $\theta_k(3)$, and estimates of the average reward $\tilde{\lambda}_k$, obtained by modified simulation-based algorithm using a discount factor $\alpha = 0.99$.

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

[Cao00] X. R. Cao. A Unified Approach to Markov Decision Problems and Performance Sensitivity Analysis. *Automatica*, 36:771-774, 2000.

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

[CR94] E. K. P. Chong and P. J. Ramadage. Stochastic Optimization of Regenerative Systems Using Infinitesimal Perturbation Analysis. *IEEE Trans. on Automatic Control*, 39:1400–1410, 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*, 6:482–494, 1998.

[FH94] M. C. Fu and J.-Q. Hu. Smoothed Perturbation Analysis Derivative Estimation for Markov Chains. *Operations Research Letters*, 15:241–251, 1994.

[FH97] M. Fu and J.-Q. Hu. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications.* Kluwer Academic Publisher, Boston, MA, 1997.

[Gal95] R. G. Gallager. *Discrete Stochastic Processes.* Kluwer Academic Publishers, Boston/Dordrech/London, 1995.

[Gly86] P. W. Glynn. Stochastic Approximation for Monte Carlo Optimization. *Proceedings of the 1986 Winter Simulation Conference*, pages 285–289, 1986.

[Gly87] P. W. Glynn. Likelihood Ratio Gradient Estimation: An Overview. *Proceedings of the 1987 Winter Simulation Conference*, pages 366–375, 1987.

[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.

[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*, 7:835–846, 1995.

[KMK97] H. Kimura, K. Miyazaki, and S. Kobayashi. Reinforcement Learning in POMDPs with Function Approximation. In D. H. Fisher, editor, *Proceedings of the 14th International Conference on Machine Learning*, pp. 152-160, 1997.

[Mar98] P. Marbach. Simulation-based Optimization of Markov Decision Processes. *PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, MA*, 1998.

[MT01] P. Marbach and J. N. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes. *IEEE Transactions on Automatic Control,* Vol. 46, No. 2, pp. 191-209, 2001.

[MT99] P. Marbach and J. N. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes: Implementation Issues. *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, pp. 1769-1774, December 1999.

[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.

[W92] R. J. Williams. Simple Statistical Gradient–Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, Vol. 8, pp. 229–256, 1992.

A Proof of Proposition 5

In this section, we analyze the algorithm proposed in Section 4 (the same analysis applies verbatim to the algorithm of Section 5). We will take the same approach as in [MT01] (and accordingly omit those parts that are identical to the proof in [MT01]). In particular, we will use a few different Lyapunov functions to analyze the algorithm in different regions.

Before we start with the proof of Proposition 5, we introduce some additional notation and definitions. Recall the update equations

$$\theta_{m+1} = \theta_m + \gamma_m F_{S^*,m}(\theta_m, \tilde{\lambda}_m), \quad (17)$$

$$\tilde{\lambda}_{m+1} = \tilde{\lambda}_m + \eta\gamma_m \sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m), \quad (18)$$

where the estimate $F_{S^*,m}(\theta_m, \tilde{\lambda}_m)$ of the gradient $\nabla\lambda(\theta_m)$ is defined by Eq. (7) in Section 4. We rewrite (17) and (18) as

$$\begin{aligned}\theta_{m+1} &= \theta_m + \gamma_m \Big(E_{\theta_m}[T]\nabla\lambda(\theta_m) + E_{\theta_m}[T]\sigma_{S^*}(\theta_m) + G_{S^*}(\theta_m)(\lambda(\theta_m) - \tilde{\lambda}_m)\Big) + \varepsilon_{\theta,m},\\ \tilde{\lambda}_{m+1} &= \tilde{\lambda}_m + \eta\gamma_m E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) + \varepsilon_{\lambda,m},\end{aligned}$$

where $G_{S^*}(\theta)$ and $\sigma_{S^*}(\theta)$ are defined in Propositions 3 and 4 in Section 4, and

$$\begin{aligned}
\varepsilon_{\theta,m} &= \gamma_m \left(F_{S^*,m}(\theta_m, \tilde{\lambda}_m) - E_{\theta_m}\left[F_{S^*,m}(\theta_m, \tilde{\lambda}_m)\right]\right), \\
\varepsilon_{\lambda,m} &= \eta\gamma_m \left(\sum_{n=t_m}^{t_{m+1}-1} (g_{i_n}(\theta_m) - \tilde{\lambda}_m) - E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) \right).
\end{aligned}$$

Let us establish some properties of $G_{S^*}(\theta)$, $\varepsilon_{\theta,m}$, and $\varepsilon_{\lambda,m}$. The following lemma states that $G_{S^*}(\theta)$ is a bounded function of θ.

Lemma 2 *Let Assumption 1 (Recurrence) and Assumption 2 (Regularity) hold, then there exists a constant L such that, for all $\theta \in \Re^K$, we have*

$$\|G_{S^*}(\theta)\| \leq L.$$

Proof: Recall the definition of $G_{S^*}(\theta)$, namely

$$G_{S^*}(\theta) = E_\theta \left[\sum_{n=t_m+1}^{t_{m,1}-1} (t_{m,k+1} - n) L_{i_{n-1}j_n}(\theta) + \sum_{k=1}^{\kappa(m)} \sum_{n=t_{m,k}}^{t_{m,k+1}-1} (t_{m,k+1} - n) L_{i_{n-1}j_n}(\theta) \right].$$

Let C be a bound on $\left\| L_{ij}(\theta) \right\|$, which exists by Assumption 2 (Regularity). Then we obtain

$$\|G_{S^*}(\theta)\| \quad \leq \quad C E_\theta \left[(t_{m+1} - t_m)^2 \right].$$

By Lemma 1, the expectation $E_\theta\left[(t_{m+1} - t_m)^2\right]$ is bounded, and the result follows. □

Similar to [MT01], we define the augmented parameter vector $r_m = (\theta_m, \tilde{\lambda}_m)$, and write the update equations in the form

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

where

$$\begin{aligned}
h(r_m) &= = \begin{bmatrix} E_{\theta_m}[T]\nabla\lambda(\theta_m) + G_{S^*}(\theta_m)(\lambda(\theta_m) - \tilde{\lambda}_m) \\ \eta E_{\theta_m}[T](\lambda(\theta_m) - \tilde{\lambda}_m) \end{bmatrix}, \\
\rho(r_m) &= \begin{bmatrix} E_{\theta_m}[T]\sigma_{S^*}(\theta_m) \\ 0 \end{bmatrix}, \\
\varepsilon_m &= \begin{bmatrix} \varepsilon_{\theta,m} \\ \varepsilon_{\lambda,m} \end{bmatrix}.
\end{aligned}$$

Also, we define the set $\mathcal{D}_c = \{(\theta, \tilde{\lambda}) \in \Re^{K+1} \mid |\tilde{\lambda}| \leq c\}$, and the set Φ, which contains all functions $\phi : \Re^{K+1} \mapsto \Re$ that are twice differentiable and which have the property that, for every $c \geq 0$, $\nabla\phi$ and $\nabla^2\phi$ are bounded on $\mathcal{D}_c$. For $\phi \in \Phi$, let $\varepsilon_m(\phi)$ be given by

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

By slightly adapting the martingale argument given in [MT01], we obtain the following lemma.

Lemma 3 *For every function $\phi \in \Phi$, the series $\sum_m \varepsilon_m(\phi)$ converges with probability 1.*

We now proceed with the main body of the proof of Proposition 5. We will concentrate on a single sample path for which the sequence $\varepsilon_m(\phi)$ (for the Lyapunov functions to be considered) is summable. Accordingly, we will be omitting the "with probability 1" qualification.

We show in the next lemma that when $\|\nabla\lambda(\theta_m)\|$ is nonzero, and the two quantities $\|\sigma_{S^*}(\theta_m)\|$ and $|\lambda(\theta_m) - \tilde{\lambda}_m|$ are small enough, then the difference $\lambda(\theta_m) - \tilde{\lambda}_m$ increases. Remember that there exists a constant L such that, for all $\theta \in \Re^K$,

$$\|G_{S^*}(\theta)\| \leq L.$$

Lemma 4 *Let L be such that $\|G_{S^*}(\theta)\| \leq L$, for all $\theta \in \Re^K$. For $\kappa \geq 0$, let*

$$B(\theta,\kappa) = \frac{E_\theta[T]\|\nabla\lambda(\theta)\|\Big(\|\nabla\lambda(\theta)\| - \|\sigma_{S^*}(\theta)\|\Big) - \kappa}{\eta E_\theta[T] + \|\nabla\lambda(\theta)\|L}$$

and let

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

We have $\phi \in \Phi$. Furthermore, if $B(\theta,\kappa) > 0$ and $|\tilde{\lambda} - \lambda(\theta)| \leq B(\theta,\kappa)$, then

$$\nabla\phi(r)\cdot\Big(h(r) + \rho(r)\Big) \geq \kappa.$$

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

$$\begin{aligned}
\nabla\phi(r)\cdot\big(h(r)+\rho(r)\big) &= \begin{pmatrix} \nabla\lambda(\theta) \\ -1 \end{pmatrix} \cdot \begin{pmatrix} E_\theta[T]\nabla\lambda(\theta) + G_{S^*}(\theta)(\lambda(\theta)-\tilde{\lambda}) + E_\theta[T]\sigma_{S^*}(\theta) \\ \eta E_\theta[T](\lambda(\theta)-\tilde{\lambda}) \end{pmatrix} \\
&= -\eta E_\theta[T]\Big(\lambda(\theta)-\tilde{\lambda}\Big) + E_\theta[T]\|\nabla\lambda(\theta)\|^2 \\
&\quad +\Big(\lambda(\theta)-\tilde{\lambda}\Big)\nabla\lambda(\theta)\cdot G_{S^*}(\theta) + E_\theta[T]\nabla\lambda(\theta)\cdot\sigma_{S^*}(\theta) \\
&\geq -\eta E_\theta[T]|\lambda(\theta)-\tilde{\lambda}| + E_\theta[T]\|\nabla\lambda(\theta)\|^2 \\
&\quad -L|\lambda(\theta)-\tilde{\lambda}|\|\nabla\lambda(\theta)\| - E_\theta[T]\|\nabla\lambda(\theta)\|\|\sigma_{S^*}(\theta)\| \\
&= -|\lambda(\theta)-\tilde{\lambda}|\Big(\eta E_\theta[T] + L\|\nabla\lambda(\theta)\|\Big) \\
&\quad +E_\theta[T]\|\nabla\lambda(\theta)\|\Big(\|\nabla\lambda(\theta)\| - \|\sigma_{S^*}(\theta)\|\Big).
\end{aligned}$$

Note that when $B(\theta,\kappa) > 0$ and $|\tilde{\lambda} - \lambda(\theta)| \leq B(\theta,\kappa)$, then we have

$$\nabla\phi(r)\cdot\Big(h(r)+\rho(r)\Big) \geq \kappa.$$

□

By the same argument as in [MT01], we obtain the following lemma.

Lemma 5 *We have* $\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0$.

We are now ready to prove Proposition 5.

Proof of Proposition 5: We assume that Proposition 5 is not true and proceed in two steps as follows.

(1) We show that when the proposition is not true, then there exists a constant $\beta > 0$ (which we define below), such that
$$\limsup_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| \geq \beta. \tag{19}$$

(2) Using the result of Step (1), we derive a contradiction.

We introduce some notation. When Proposition 5 is not true, then there exists a constant $\epsilon > 0$, and an integer M, such that
$$\|\nabla\lambda(\theta_m)\| > D + \epsilon, \qquad \text{for all } m \geq M.$$
Let B be a bound on $\|\nabla\lambda(\theta)\|$, and let $T_{\min}$ and $T_{\max}$ be such that, for all $\theta \in \Re^K$, we have
$$1 \leq T_{\min} \leq E_\theta[T] \leq T_{\max}.$$
Such constants exist by Lemma 1. Furthermore, let $D' = D + \epsilon$, and let
$$\beta = T_{min} D' \epsilon / (\eta T_{max} + LB),$$
where L is the bound on $\|G_{S^*}(\theta)\|$ used in the statement of the proposition.

Step (1): Suppose that the proposition is not true and, furthermore, that the condition given by Eq.(19) does not hold. Then there exists an integer M_0, and a scalar $\kappa > 0$, such that, for all $m > M_0$,
$$|\lambda(\theta_m) - \tilde{\lambda}_m| \leq \beta - \frac{\kappa}{\eta T_{\max} + BL} \leq \frac{T_{min} D'\epsilon - \kappa}{\eta T_{\max} + BL} \leq \frac{E_{\theta_m}[T]\|\nabla\lambda(\theta_m)\|\Big(\|\nabla\lambda(\theta_m)\| - \|\sigma_{S^*}(\theta_m)\|\Big) - \kappa}{\eta E_{\theta_m}[T] + \|\nabla\lambda(\theta_m)\| L}.$$
Therefore, Lemma 4 applies and we obtain, for $m > M_0$, that
$$\begin{aligned} \phi(r_{m+1}) &= \phi(r_m) + \gamma_m \nabla\phi(r_m) \cdot \Big(h(r_m) + \rho(r_m)\Big) + \varepsilon_m(\phi) \\ &\geq \phi(r_m) + \gamma_m \kappa + \varepsilon_m(\phi). \end{aligned}$$
As $\lim_{m\to\infty} \varepsilon_m(\phi) = 0$ and $\sum_{m=1}^{\infty} \gamma_m = \infty$, it follows that $\phi(r_m) = \lambda(\theta_m) - \tilde{\lambda}_m$ diverges. This is a contradiction to Lemma 5, which states that
$$\liminf_{m\to\infty} |\lambda(\theta_m) - \tilde{\lambda}_m| = 0.$$

Step (2): Suppose that Proposition 5 is not true. Using the result of Step (1), together with Lemma 5, it follows that there are infinitely many pairs n, n', with $n' > n$, such that
$$\phi(r_{n'}) - \phi(r_n) = \Big(\lambda(\theta_{n'}) - \tilde{\lambda}_{n'}\Big) - \Big(\lambda(\theta_n) - \tilde{\lambda}_n\Big) < -\frac{1}{2}\beta,$$
and, for $m = n, \ldots, n' - 1$,
$$\begin{aligned} \|\nabla\lambda(\theta_m)\| &> D', \\ |\lambda(\theta_m) - \tilde{\lambda}_m| &< \beta. \end{aligned}$$

This implies that, for $m = n, \ldots, n' - 1$, we have

$$\|\nabla\lambda(\theta_m)\| - \|\sigma_{S^*}(\theta_m)\| \geq \epsilon > 0$$

and

$$\begin{aligned} |\lambda(\theta_m) - \tilde{\lambda}_m| &< \beta \\ &= \frac{T_{min} D' \epsilon}{\eta T_{max} + BL} \\ &\leq \frac{E_{\theta_m}[T] \|\nabla\lambda(\theta_m)\| \Big(\|\nabla\lambda(\theta_m)\| - \|\sigma_{S^*}(\theta_m)\| \Big)}{\eta E_{\theta_m}[T] + \|\nabla\lambda(\theta_m)\| L}. \end{aligned}$$

Therefore, Lemma 4 applies, and we obtain, for $m = n, \ldots, n' - 1$, that

$$\nabla\phi(r_m) \cdot \big(h(r_m) + \rho(r_m)\big) \geq 0.$$

Combining these results, we have

$$\begin{aligned} -\frac{1}{2}\beta &> \phi(r_{n'}) - \phi(r_n) \\ &= \sum_{m=n}^{n'-1} \Big(\gamma_m \nabla\phi(r_m) \cdot \big(h(r_m) + \rho(r_m)\big) + \varepsilon_m(\phi) \Big) \geq \sum_{m=n}^{n'-1} \varepsilon_m(\phi). \end{aligned} \tag{20}$$

By Lemma 3, the series $\sum_m \varepsilon_m(\phi)$ converges and the term $\left\| \sum_{m=n}^{n'-1} \varepsilon_m(\phi) \right\|$ becomes arbitrarily small. This leads to a contradiction in Eq. (20) and completes the proof. □