Simulation-Based Optimization of Markov Reward Processes: Implementation Issues[1]

Peter Marbach[2] John N. Tsitsiklis[3]

Abstract

We consider discrete time, finite state space Markov reward processes which depend on a set of parameters. Previously, we proposed a simulation-based methodology to tune the parameters to optimize the average reward. The resulting algorithms converge with probability 1, but may have a high variance. Here we propose two approaches to reduce the variance, which however introduce a new bias into the update direction. We report numerical results which indicate that the resulting algorithms are robust with respect to a small bias.

1 Introduction

In [MT98] we considered finite state Markov reward processes for which the transition probabilities and one-stage rewards depend on a parameter vector $\theta \in \Re^K$, and proposed a simulation-based methodology which uses gradient estimates for tuning the parameter θ to optimize the average reward. The resulting algorithms can be implemented online and have 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). A drawback of these algorithms is that the updates may have a high variance, which can lead to slow convergence. This is due to the fact they essentially employ a renewal period to produce an estimate of the gradient. If the length of a typical renewal period is large (as it tends to be the case for systems involving a large state space) then the variance in the corresponding estimate can become quite high. 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 one which employs a discount factor. The resulting algorithms reduce the variance of a typical update, but introduce an additional bias into the update direction. As gradient-type methods tend to be robust with respect to small biases, the resulting algorithm may have an improved practical performance. A numerical study that we carried out suggests that this is indeed the case. For a comparison of our approach with other simulation-based optimization methods such as the likelihood-ratio method [Gly86, Gly87], the infinitesimal perturbation analysis (IPA) [CR94, CC97, CW98, FH94, FH97], and neuro-dynamic programming/reinforcement learning in [JSJ95], we refer to [MT98].

2 Formulation

In this section we give a summary of the general framework, which is slightly more general than the one in [MT98] (see [Mar98] for a detailed discussion). 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 $P_{ij}^n(\theta)$. 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 (Recurrence) *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.*

Assumption 2 (Regularity) *For all states $i, j \in S$ and for all $P \in \overline{\mathcal{P}}$, 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*

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

where the function $L_{ij}(\theta)$ is bounded.

[1]This research was supported by contracts with Siemens AG, Munich, Germany, and Alcatel Bell, Belgium; and by contracts DMI-9625489 and ACI-9873339 with the National Science Foundation

[2]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; current affiliation: Center for Communication Systems Research, Cambridge University, UK; email: p.marbach@ccsr.cam.ac.uk

[3]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; email: jnt@mit.edu

Assumption 1 (Recurrence) states that under every transition matrix $P \in \overline{\mathcal{P}}$ we will eventually revisit the state i^*. This property allows us to employ results of renewal theory (see for example [Gal95]) for our analysis. Assumption 2 (Regularity) assures that the transition probabilities $P_{ij}(\theta)$ and the one-stage reward $g_i(\theta)$ depend "smoothly" on θ, and that the quotient $\nabla P_{ij}(\theta)/P_{ij}(\theta)$ is "well behaved". As a performance metric 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

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

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

3 A Gradient-Based Algorithm for Updating θ

Given that our goal is to maximize the average reward $\lambda(\theta)$, it is natural to consider gradient-type methods. Using the following expression for the gradient $\nabla\lambda(\theta)$ of the average reward with respect to θ (see [MT98] for a derivation),

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

we could consider an algorithm of the following form

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

Alternatively, if we could use simulation to produce an unbiased estimate $F_k(\theta)$ of $\nabla\lambda(\theta_k)$ and employ the approximate gradient iteration

$$\theta_{k+1} = \theta_k + \gamma_k F_k(\theta)$$

to tune θ. Unbiased estimates of the gradient exist [Gly86], but it is not clear how to use them in an algorithm which has the properties we will require in the following [1]. This difficulty is bypassed by the method developed in the following. We will again employ an estimate of the gradient to update the parameter vector θ which however is biased. Convergence of the resulting simulation based method can be established by showing that the bias (asymptotically) vanishes.

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

We rewrite the expression for the gradient $\nabla\lambda(\theta)$ as follows (see [MT98],

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

where $L_{ij}(\theta) = \nabla P_{ij}(\theta)/P_{ij}(\theta$. This expression suggests the following way to estimate the gradient $\nabla\lambda(\theta)$. 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^*. 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 for $i_n = j$

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

is an estimate of the differential reward $v_j(\theta)$ and $\tilde{\lambda}$ is some estimate of $\lambda(\theta)$. For $n = t_m$ we set $\tilde{v}_{i_n}(\theta, \tilde{\lambda}) = 0$. 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 extend that $\tilde{\lambda}$ is close the $\lambda(\theta)$ (see [MT98, Mar98]).

Proposition 1 *We have*

$$\begin{aligned} E_\theta \left[F_m(\theta, \tilde{\lambda}) \right] &= E_\theta[T] \nabla\lambda(\theta) + G(\theta)(\lambda(\theta) - \tilde{\lambda}) \\ &= E_\theta \sum_{i\in S} \pi_i(\theta) \left(\nabla g_i(\theta) + \sum_{j\in S} \nabla P_{ij}(\theta) v_j(\theta) \right) + \ldots \\ &\quad + G(\theta)(\lambda(\theta) - \tilde{\lambda}) \end{aligned}$$

where

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

[1] [Gly86] develops an algorithm using unbiased estimates of the gradient which updates the parameter vector at visits to the recurrent state i^*. It is not known if this algorithm can be extended so that the parameter gets updated at every time step, a property which we will require based on the discussion in Section 4.

3.2 An Algorithm that Updates at Visits to the Recurrent State

Using $F_m(\theta, \tilde{\lambda})$ as an estimate of the gradient direction, we can formulate the following algorithm which updates θ at visits to the recurrent state i^*. 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

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

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

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 $\gamma_m = 1/m$.

Note that the update in (3) is a biased estimate of the gradient direction (see Proposition 1). By showing that the update (4) drives this bias (asymptotically) to zero, we obtain the following convergence result (see [MT98, Mar98]).

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, with probability 1, $\lambda(\theta_m)$ converges and*

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

4 Implementation Issues

For systems involving a large state space, the interval between visits to the state i^* can be large. This means that in the algorithm proposed above the parameter vector θ gets updated only infrequently and the estimate $F_m(\theta)$ can have a large variance. In [MT98], we have shown how the above algorithm can be extended so that the parameter vector gets updated at every time step. Here, we will in addition consider two approaches to reduce the variance in the updates, which are based on two alternative estimates of the differential reward.

4.1 Reducing the Variance

If the time until we reach the recurrent state i^* from state i is large, it may be desirable to choose a subset S^* of S containing i^* and to estimate $v_i(\theta)$ through

$$\tilde{v}_{S^*,i}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \left(g_{i_k}(\theta) - \tilde{\lambda}\right), \quad (5)$$

where $T = \min\{k > 0 \mid i_k \in S^*\}$ is the first future time a state in the set S^* is visited. Note that it takes fewer time steps to reach the set S^* than to revisit the recurrent state i^*; therefore $\tilde{v}_{S^*,i}(\theta, \tilde{\lambda})$ has typically a smaller variance than the estimate based on (2). As an alternative approach, we may use a factor α, $0 \le \alpha < 1$, to discount future rewards. This leads to the following estimate,

$$\tilde{v}_{\alpha,i}(\theta, \tilde{\lambda}) = \sum_{k=0}^{T-1} \alpha^k \left(g_{i_k}(\theta) - \tilde{\lambda}\right), \quad (6)$$

where $T = \min\{k > 0 \mid i_k = i^*\}$ is the first future time the state i^* is visited. In Sections 5 and 6 we will use (5) and (6) to derive modified estimates of the gradient $\nabla\lambda(\theta)$.

5 Using the Set S^* to Reduce the Variance

Let the parameter $\theta \in \Re^K$ be fixed to some value and let $(i_1, i_2, ...)$ be a trajectory of the corresponding Markov chain. Let t_m be the time of the mth time to the termination state i^*. Furthermore, given a set $S^* \subset S$ containing i^*, let $\kappa(m)$ be the number of times between t_m and t_{m+1} that a state in the set $S^*\backslash\{i^*\}$ is visited, let the $t_{m,n}$ be the time of the nth visit, 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

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

where, for $t_{m,n} \le k < t_{m,n+1}$, and $i_k \neq i^*$, we set

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

and, for $k = t_m$, we set $\tilde{v}_{i_k}(\theta, \tilde{\lambda}) = 0$.

One can show (see [Mar98]), that the expectation $E_\theta[F_{S^*,m}(\theta, \tilde{\lambda})]$ is of the same form as the expectation of the original estimate $F_m(\theta, \tilde{\lambda})$ in Proposition 1, however the exact value of the differential reward $v_j(\theta)$ is replaced by the approximation

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

where $T = \min\{k > 0 \mid i_k \in S^*\}$. This introduces a new bias term $\sigma_{S^*}(\theta)$ into the estimate of the gradient direction which is equal to

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

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

To derive an upper bound on $\|\sigma_{S^*}(\theta)\|$, we use some suitable assumptions on the bias in the estimate of the differential reward given by

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

The basic idea is the following. As $\sum_{j \in S} \nabla P_{ij}(\theta) = 0$, for all $i \in S$ and all $\theta \in \Re^K$, we only have to know the "relative magnitudes" of the differential rewards to compute $\nabla\lambda(\theta)$, i.e. we have that for all constant c

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

Therefore, if $\hat{v}_{S^*,i}(\theta) = \hat{v}_{S^*,i'}(\theta)$, for all $i, i' \in S$, then the bias term $\|\sigma_{S^*}(\theta)\|$ is equal to 0. In fact, the term $\|\sigma_{S^*}(\theta)\|$ vanishes under the weaker assumption that for all states $i \in S$ we have $\hat{v}_{S^*,j}(\theta) = \hat{v}_{S^*,j'}(\theta)$, if $j, j' \in S_i$, where $S_i = \{j \in S \mid \nabla P_{ij}(\theta) \neq 0 \text{ for some } \theta \in \Re^K\}$. Now assume that there exists a ϵ such that, for all states $i \in S$, we have

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

Then we have that under Assumption 1 (Recurrence) and Assumption 2 (Regularity)

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

where $T_{\max}$ is an upper-bound on $E_\theta[T]$, C is a bound on $\|\nabla P_{ij}(\theta)\|$, and $\bar{N} = \max\{N_i \mid i \in S\}$ with where N_i is the number of states in the set S_i. Therefore, 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. A proof for this result is given in [Mar98].

5.2 An Algorithm that Updates at Visit to the Recurrent State

Consider the following version of algorithm in Section 3 to update θ and $\tilde{\lambda}$ at visits to the recurrent state i^*,

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

where we use the estimate $F_{S^*,m}(\theta_m)$ in place of $F_m(\theta_m)$. Under Assumption 1 (Recurrence) and Assumption 2 (Regularity), we have that with probability 1

$$\liminf_{m \to \infty} \|\nabla\lambda(\theta_m)\| \leq \frac{D}{T_{\min}}.$$

Here D is a bound on the bias $\|\sigma_{S^*}(\theta)\|$ and $T_{\min} \geq 1$ is lower bound on $E_\theta[T]$ (see [Mar98]). This establishes that if the bias $\|\sigma_{S^*}(\theta)\|$ tends to be small, then the gradient $\nabla\lambda(\theta_m)$ is small at infinitely many visits to the recurrent state i^*. However, it does not say anything about the behavior of the sequence $\lambda(\theta_m)$ or how we might detect instances at which the average reward $\lambda(\theta_m)$ is high. This result can be strengthened in the following sense (see [Mar98]): if the upper-bound D on $\|\sigma_{S^*}(\theta)\|$ is small enough, then $\tilde{\lambda}_m$ overestimates the average reward $\lambda(\theta_m)$ at most by a little, i.e. there exists a bound $B(D)$ such that

$$0 \leq \limsup_{m \to \infty} (\tilde{\lambda}_m - \lambda(\theta_m)) \leq B(D).$$

This implies that $\tilde{\lambda}_m$ can be used to detect instances where the average reward $\lambda(\theta_m)$ is high. Similar to [MT98], one can derive a version of this algorithm which updates θ at each time step (see [Mar98]).

6 Using a Discount Factor to Reduce the Variance

Let the parameter $\theta \in \Re^K$ be fixed to some value and let $(i_1, i_2, ...)$ be a sample path of the corresponding Markov chain. Furthermore, given a discount factor $\alpha \in (0, 1)$, 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 $\nabla\lambda(\theta)$,

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

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

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

and, for $n = t_m$, we set $\tilde{v}_{\alpha,i_n}(\theta, \tilde{\lambda}) = 0$. As in the previous section, the expectation $E_\theta[F_{\alpha,m}(\theta, \tilde{\lambda})]$ is of the same form as the expectation of the original estimate $F_m(\theta, \tilde{\lambda})$, except that the exact value of the differential reward $v_j(\theta)$ is replaced by the approximation

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

where $T = \min\{k > 0 \mid i_k = i^*\}$.

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

To derive a bound on the resulting new bias $\sigma_\alpha(\theta)$ which is equal to

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

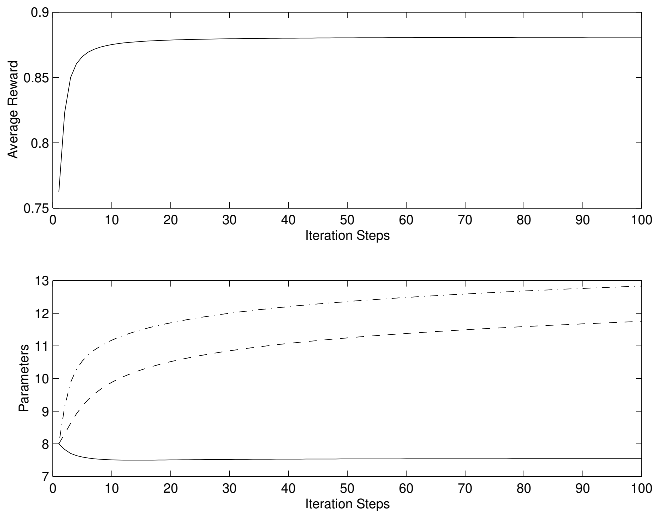

Figure 1: Trajectory of the exact average reward (top) and threshold parameters (bottom) of the idealized gradient algorithm.

we use the following assumption. There exist scalars β, $0 \leq \beta < 1$, and A such that for, all states $i \in S$ and all integers $n \geq 0$, we have

$$\Big|\sum_{j\in S}\Big(P^n_{ij}(\theta) - \pi_j(\theta)\Big) g_i(\theta)\Big| \leq A\beta^n. \tag{9}$$

We can think of β as a "mixing constant" for the Markov reward processes associated with $\overline{\mathcal{P}}$. It can be shown that under assumption Assumption 1 (Recurrence) such constants A and β exist (see [Mar98]). It then follows, that under Assumption 1 (Recurrence) and Assumption 2 (Regularity) ,

$$\|\sigma_\alpha(\theta)\| \quad \leq \quad A T_{\max} C \bar{N}\left(\frac{1}{1-\beta} - \frac{1}{1-\alpha\beta}\right),$$

where $T_{\max}$ is an upper-bound on $E_\theta[T]$, C is a bound on $\|\nabla P_{ij}(\theta)\|$, and $\bar{N}$ is the same as in Section 5. This result states that the bound on the bias $\sigma_\alpha(\theta)$ vanishes as $\beta \downarrow 0$ and as $\alpha \uparrow 1$. A proof is given in [Mar98]. Again, we can use the estimate $F_{\alpha,m}(\theta,\tilde{\lambda})$ to derive an algorithm which updates the parameter vector θ either at visits to the recurrent state i^* or at every time step (see [Mar98]). The results of the previous section also hold in this case.

7 Experimental Results

We applied the algorithms proposed above to the problem where a provider of a communication link has to accept and reject incoming calls of several types, while taking into account current congestion. Calls are charged according to the type they belong to and the goal of the provider is to maximize long-term average reward. We consider the following admission control policy described in terms a "fuzzy threshold" parameter $\theta(m)$ for each call type $m = 1, ..., M$. Given that the used link bandwidth is equal to B, a new call of type m is accepted with probability

$$\frac{1}{1+\exp(B-\theta(m))},$$

and is otherwise rejected. A detailed description of the case study can be found in [Mar98]. In the following, we provide a short summary of the main results for a case study involving 3 different service types.

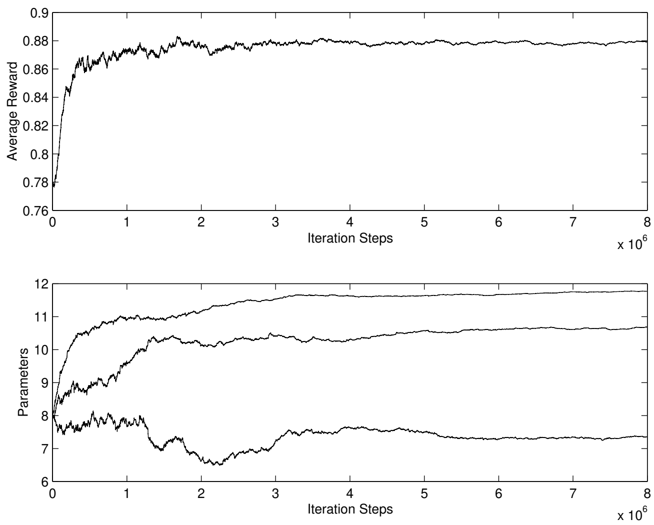

Figure 2: Trajectory of the average reward (top) and threshold parameters (bottom) of the simulation-based algorithm of Section 3 (the scaling factor for the iteration steps is 10^6).

7.1 Idealized Gradient Algorithm

For this case, we were able to compute the exact gradient $\nabla\lambda(\theta)$ and to implement the idealized gradient algorithm of Section 2. The corresponding trajectories are given in Figure 1. The resulting average reward is 0.8808 (the optimal average reward obtained by dynamic programming is 0.8868).

7.2 Simulation-Based Optimization

Figure 2, 3, and 4, shows the results for the versions of the simulation-based algorithms of Section 2, Section 5 (for the choice of S^* see [Mar98]), and Section 6 (with $\alpha = 0.99$), respectively, where the parameter θ gets updated at every time step. We make here the following observations.

1. All three algorithms make rapid progress in the beginning, improving the average reward from

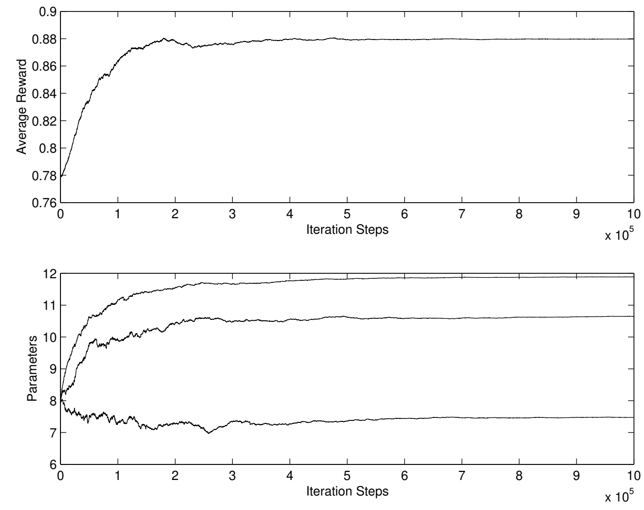

Figure 3: Trajectory of the average reward (top) and threshold parameters (bottom) of the simulation-based algorithm of Section 5 (the scaling factor for the iteration steps is 10^5).

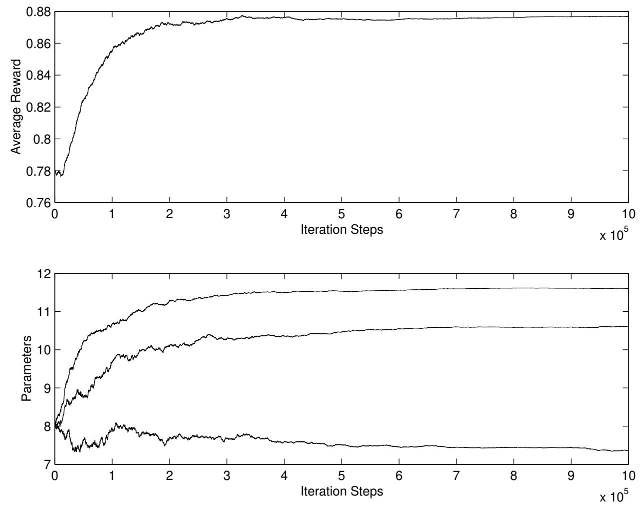

Figure 4: Trajectory of the average reward (top) and threshold parameters (bottom) of the simulation-based algorithm of Section 6 (the scaling factor for the iteration steps is 10^5).

0.78 to 0.87 within the first $1 \cdot 10^6$ iteration steps for the algorithm of Section 2 and within first $1 \cdot 10^5$ iteration steps for the algorithms of Section 5 and 6. After reaching an average reward of 0.87, the algorithms progresses only slow. This is not unlike the behavior of the idealized algorithm (see Figure 1).

2. The simulation-based algorithms of Section 2 attains an average reward of 0.8798 after $8 \cdot 10^6$ iteration steps, while the algorithms of Section 5 and 6 obtain an average reward of 0.8792, and 0.8785 respectively, after 10^6 iteration steps.

These results are encouraging, as the algorithms with reduced variance speed up the convergence by an order of magnitude, while introducing a negligible bias.

References

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

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

[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*, 7:835–846, 1995.

[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.

[MT98] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of markov reward processes. *Technical Report, LIDS-P 2411, Lab. for Info. and Decision System, Massachusetts Institute of Technology, MA*, 1998.