Discrete Event Dyn Syst (2010) 20:3–17
DOI 10.1007/s10626-009-0078-3

On-Line Policy Gradient Estimation with Multi-Step Sampling

Yan-Jie Li · Fang Cao · Xi-Ren Cao

Received: 22 May 2008 / Accepted: 14 July 2009 / Published online: 28 July 2009

Abstract In this note, we discuss the problem of the sample-path-based (on-line) performance gradient estimation for Markov systems. The existing on-line performance gradient estimation algorithms generally require a standard importance sampling assumption. When the assumption does not hold, these algorithms may lead to poor estimates for the gradients. We show that this assumption can be relaxed and propose algorithms with multi-step sampling for performance gradient estimates; these algorithms do not require the standard assumption. Simulation examples are given to illustrate the accuracy of the estimates.

Keywords Markov reward processes · Policy gradient · On-line estimation · Performance potentials

Y.-J. Li (✉) · F. Cao · X.-R. Cao
Department of Electronic and Computer Engineering,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
e-mail: whylyj@ustc.edu

Present Address:
Y.-J. Li
Division of Control and Mechatronics Engineering,
Harbin Institute of Technology Shenzhen
Graduate School, Shenzhen, China

Present Address:
F. Cao
School of Electronics and Information Engineering,
Beijing Jiaotong University, Beijing, China

1 Introduction

The policy gradient approach has recently received increasing attention in the optimization and reinforcement learning communities (Baxter and Bartlett 2001; Baxter et al. 2001; Greensmith et al. 2004; Cao 2005). It is closely related to perturbation analysis in the discrete event dynamic system theory (Cao and Chen 1997; Cao and Wan 1998). With the policy gradient estimates, performance optimization algorithms can be developed for Markov systems (Baxter et al. 2001; Marbach and Tsitsiklis 2001; Cao 2007). Compared with the value-function methods, the policy gradient approaches can avoid the problems associated with policy degradation (Baxter and Bartlett 2001). However, the existing on-line policy gradient estimation algorithms generally need a standard assumption in importance sampling (Marbach and Tsitsiklis 2001; Baxter and Bartlett 2001; Greensmith et al. 2004; Cao 2005). When the assumption does not hold, these on-line policy gradient estimation algorithms will lead to poor estimates.

In this note, we give examples to illustrate that the existing on-line policy gradient approaches cannot provide an accurate gradient estimate when the assumption does not hold. We then show that this assumption can be relaxed and propose a few new algorithms based on multi-step sampling; these algorithms do not require this assumption. All the algorithms can be implemented on sample paths and policy gradients can be estimated on line.

2 Problem formulation

Consider an ergodic (irreducible and aperiodic) discrete time Markov chain $\boldsymbol{X} = \{X_l, l = 0, 1, \ldots\}$ on a finite state space $\mathcal{S} = \{1, 2, \ldots, M\}$ with transition probability matrix $P = [p(j|i)] \in [0, 1]^{M\times M}$, where X_l denotes the system state at time l and $p(j|i)$ denotes the one-step transition probability from state $i \in \mathcal{S}$ to state $j \in \mathcal{S}$. Let $\pi = (\pi(1), \pi(2), \ldots, \pi(M))$ be the row vector representing its steady-state probabilities, then we have the following balance equations

$$\pi P = \pi, \pi e = 1, \tag{1}$$

where $e = (1, 1, \ldots, 1)^T$ is an M-dimensional column vector whose components all equal 1 and the superscript " T " denotes transpose. Let $f = (f(1), f(2), \ldots, f(M))^T$ be a column vector with $f(i)$ being the expected immediate reward at state i, $i = 1, 2, \ldots, M$. We consider the long-run average reward defined as

$$\eta = \sum_{i\in S} \pi(i) f(i) = \pi f = \lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} f(X_l), \qquad w.p.1.$$

For Markov chain $\boldsymbol{X}$, we have the following Poisson equation

$$(I - P)g + \eta e = f, \tag{2}$$

where I is the $M \times M$ identity matrix. Its solution $g = (g(1), g(2), \ldots, g(M))^T$ is called a *performance potential* and $g(i)$ is the potential at state i. (It is equivalent to the value function in dynamic programming, or the "differential" or "relative cost vector" (Bertsekas 1995), or the "bias" (Puterman 1994)). The solution to Eq. 2 can

be obtained only up to an additive constant, i.e., if g is a solution to Eq. 2, then so is $g + ce$, where c is any constant. It is well known (Cao and Chen 1997; Cao 2005, 2007) that both

$$g(i) = E\left\{\sum_{l=0}^{\infty}[f(X_l) - \eta]\Big| X_0 = i\right\}, \tag{3}$$

and

$$g(i) = E\left\{\sum_{l=0}^{L_i(i^*)-1}[f(X_l) - \eta]\Big| X_0 = i\right\}, i \neq i^*; \qquad g(i^*) = 0, \tag{4}$$

are the solutions to the Poisson equation (Eq. 2), where $L_i(i^*)$ denotes the first passage time from state i to a reference state i^* (which can be chosen arbitrarily) and "E" denotes the expectation.

If the transition probability matrix depends on a parameter $\theta \in \Theta$, that is, $P(\theta) = [p_\theta(j|i)]$, where Θ is the parameter set, and $P(\theta)$ is ergodic and differentiable for any $\theta \in \Theta$, (for simplicity, we assume f does not depend on θ), the performance gradient of $\eta(\theta)$ with respect to θ is (see e.g. Cao (2005, 2007))

$$\frac{d\eta(\theta)}{d\theta} = \pi(\theta)\frac{dP(\theta)}{d\theta}g(\theta), \tag{5}$$

where $\eta(\theta)$, $\pi(\theta)$ and $g(\theta)$ are the average reward, steady-state probability and performance potential corresponding to transition matrix $P(\theta)$, respectively. If $P(\theta)$ has a linear structure, i.e., $P(\theta) = P + \theta Q$, where $Q = [q(j|i)]$ is an $M \times M$ matrix with $Qe = 0$ (e.g., $Q = P' - P$, where P' be another irreducible and aperiodic transition probability matrix and $\theta \in [0, 1]$), then the performance gradient has the following simple structure

$$\frac{d\eta(\theta)}{d\theta} = \pi(\theta)Qg(\theta). \tag{6}$$

Because Q indicates the direction of the derivative in Eq. 6, we call it a *direction matrix*. We will develop gradient estimates based on Eq. 6; this does not lose any generality because we can simply replace Q with $\frac{dP(\theta)}{d\theta}$ in the algorithms to obtain estimates in the form of Eq. 5.

There are a number of sample-path-based policy gradient estimation algorithms in literature (Cao and Wan 1998; Marbach and Tsitsiklis 2001; Baxter and Bartlett 2001; Greensmith et al. 2004; Cao 2005). These algorithms generally use a standard technique in simulation called importance sampling. The basic principles can be summarized as follows (Cao 2005). For convenience, we consider the performance gradient at $\theta = 0$. From the performance gradient formula (6), we have

$$\begin{aligned}\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} &= \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}\pi(i)q(j|i)g(j) \\ &= \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}\pi(i)p(j|i)\frac{q(j|i)}{p(j|i)}g(j) \qquad (7) \\ &= E_\pi\left\{\frac{q(X_{l+1}|X_l)}{p(X_{l+1}|X_l)}g(X_{l+1})\right\}, \qquad (8)\end{aligned}$$

where E_π denotes the expectation with respect to the steady-state distribution π and the transition matrix P. Then, following the basic formula in Cao (2005), we have

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} = \lim_{L\to\infty}\frac{1}{L}\sum_{l=0}^{L-1}\frac{q(X_{l+1}|X_l)}{p(X_{l+1}|X_l)}g(X_{l+1}), \qquad w.p.1. \tag{9}$$

Since the ratio $\frac{q(j|i)}{p(j|i)}$ is used in Eq. 7, we need the following standard importance sampling assumption:

$$For\ any\ i,\ j\in\mathcal{S},\ if\ q(j|i)\neq 0,\ then\ p(j|i)>0. \tag{10}$$

This assumption limits the application of these gradient-estimation algorithms. When there exists some i, $j\in\mathcal{S}$ such that $q(j|i)\neq 0$ but $p(j|i)=0$, since the transition from state i to state j does not occur in the simulation, the estimation algorithms based on Eq. 9 will lead to a poor gradient estimate. This is clearly illustrated by the following example.

Example 1 Consider a Markov chain with transition probability matrix P and matrix Q defined as

$$P=\begin{bmatrix} 0.2 & 0 & 0.8 & 0\\ 0 & 0.5 & 0 & 0.5\\ 0.3 & 0 & 0 & 0.7\\ 0 & 0.6 & 0.4 & 0\end{bmatrix},\quad Q=\begin{bmatrix} -0.2 & 0.5 & -0.8 & 0.5\\ 0.7 & -0.5 & 0.3 & -0.5\\ -0.3 & 0.4 & 0.6 & -0.7\\ 0.5 & -0.6 & -0.4 & 0.5\end{bmatrix},$$

and $f=(1,2,3,4)^T$. We compute the steady-state probability π and the potential g by the balance equations (Eq. 1) and the Poisson equation (Eq. 2), respectively, and obtain $g=(0.8630, 2.1788, 3.0998, 3.7577)^T+c(1,1,1,1)^T$ and $\pi=(0.0718, 0.4019, 0.1914, 0.3349)$. Thus, from Eq. 6, the performance gradient at $\theta=0$ is $\frac{d\eta(\theta)}{d\theta}|_{\theta=0}=-0.6633$. However, since $q(j|i)=-p(j|i)$ for all i, $j\in\mathcal{S}$ with $p(j|i)>0$, all the terms $\frac{q(X_{l+1}|X_l)}{p(X_{l+1}|X_l)}$ in Eq. 9 are -1. Thus, the performance gradient obtained by Eq. 9 is $-\pi g$! Note that $-\pi g$ is different for different constant c. For instance, when we consider the potential (3), $c=-\eta=2.7894$, we have $-\pi g=0$. Moreover, when we consider the potential (4) with $i^*=1$, $c=-0.8630$, we have $-\pi g=-1.9264$. From this simple example, we find all the performance gradient estimation algorithms based on Eq. 9 cannot provide an accurate gradient estimate.

3 The performance gradient estimates with multi-step sampling

In this section, we show that the standard importance sampling assumption (10) can be relaxed and propose some policy gradient estimation algorithms that may treat the cases where assumption (10) does not hold.

When assumption (10) does not hold, there exists a pair of states i, $j\in\mathcal{S}$ such that $q(j|i)\neq 0$ and $p(j|i)=0$. To illustrate the idea, we first assume that for any pair of states i, $j\in\mathcal{S}$ such that $q(j|i)\neq 0$, there exists a state, denoted as $u_{i,j}$, such that

$p(u_{i,j}|i)\,p(j|u_{i,j}) > 0$. The performance gradient formula (6) can be written as follows (cf. Eqs. 7 and 8):

$$\begin{aligned}\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} &= \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}\pi(i)q(j|i)g(j)\\ &= \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}}\pi(i)\sum_{u\in\mathcal{S}}p(u|i)p(j|u)\frac{q(j|i)}{\sum_{u\in\mathcal{S}}p(u|i)p(j|u)}g(j)\\ &= E_\pi\left\{\frac{q(X_{l+2}|X_l)}{\sum_{u\in\mathcal{S}}p(u|X_l)p(X_{l+2}|u)}g(X_{l+2})\right\}.\end{aligned}$$

With this equation, we have

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} = \lim_{L\to\infty}\frac{1}{L}\sum_{l=0}^{L-1}\frac{q(X_{l+2}|X_l)}{\sum_{u\in\mathcal{S}}p(u|X_l)p(X_{l+2}|u)}g(X_{l+2}),\qquad w.p.1. \tag{11}$$

The performance gradient can be estimated according to Eq. 11, in which we do not sample X_l and X_{l+1}, but X_l and X_{l+2} at each time step. Since $p(u_{i,j}|i)\,p(j|u_{i,j}) > 0$, the probability that the system moves from state i to state j in two steps, denoted as $p^{(2)}(j|i) := \sum_{u\in\mathcal{S}}p(u|i)p(j|u)$, is always positive; i.e., the sum on the denominator of Eq. 11 is always positive when $q(X_{l+2}|X_l) \neq 0$.

In general, for some systems, there may not exist such a state $u_{i,j}$ for all $i, j \in \mathcal{S}$. Fortunately, for any ergodic transition matrix P, we have Cinlar (1975)

$$\lim_{n\to\infty} P^n = e\pi.$$

Thus, there must exist a K such that $p^{(K)}(j|i) > 0$ for all $i, j \in \mathcal{S}$, where $p^{(K)}(j|i)$ denotes the probability that the system moves to state j at the Kth step from state i. We only need to find a K such that $p^{(K)}(j|i) > 0$ for all i, j with $q(j|i) \neq 0$. Thus, K might be a small integer for a particular problem. We can evaluate $p^{(K)}(j|i)$ by iteration

$$x_0 = e_j,\quad x_{n+1} = Px_n,\quad n = 0, 1, \ldots, K-1,\quad p^{(K)}(j|i) = e_i * x_K, \tag{12}$$

where e_j (or e_i) denotes a column vector whose jth (or ith) component is 1 and others are zero. Because this is a series of matrix-vector multiplications, the worst case complexity of these computations is $O(KS^2)$. The matrix P is usually sparse. Using sparse matrix data structures and sparse multiplication algorithms, the practical complexity is $O(\rho KS)$ where $\rho \ll S$ and depends on the degree of sparsity. Generally matrix Q is also sparse, thus the total computational complexity to compute $p^{(K)}(j|i)$ for all i, j such that $q(j|i) \neq 0$ is acceptable.

With the above discussion, we rewrite the performance gradient formulas (7) and (8) in the following general form:

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} = \sum_{i\in\mathcal{S}}\pi(i)\sum_{j\in\mathcal{S}}p^{(K)}(j|i)\frac{q(j|i)}{p^{(K)}(j|i)}g(j) = E_\pi\left\{\frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)}g(X_{l+K})\right\}.$$

Thus, the performance gradient can be estimated according to

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} = \lim_{L\to\infty}\frac{1}{L}\sum_{l=0}^{L-1}\frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)}g(X_{l+K}),\qquad w.p.1. \tag{13}$$

Equation 13 is an extension of Eq. 9, in which we assume that

$$For\ any\ i,\ j \in \mathcal{S},\ if\ q(j|i) \neq 0,\ then\ p^{(K)}(j|i) > 0. \tag{14}$$

From the above discussion, such a K always exists. Therefore, Eq. 14 can be viewed as a specification on K rather than an assumption.

On the basis of Eq. 13, we may develop the policy gradient estimation algorithms. Following the same idea used in Cao (2005), we may use any sample-path-based estimate $\hat{g}(X_{l+K}, X_{l+K+1}, ...)$, with $E[\hat{g}(X_{l+K}, X_{l+K+1}, ...)|X_{l+K}] \approx g(X_{l+K})$, to replace the $g(X_{l+K})$ in Eq. 13. In this way, using different sample-path-based estimates of the potentials, we may obtain different gradient estimates, and these gradient estimates do not require the assumption (10) once a proper integer K is determined.

First, let us consider the discounted potential approximation. That is, let

$$g(X_{l+K}) \approx E\left\{\sum_{k=l+K}^{\infty} \alpha^{k-(l+K)}[f(X_k) - \eta]\Big| X_{l+K}\right\},\quad \alpha \in (0, 1),$$

which can approximate potential $g(X_{l+K})$ in Eq. 3 when α is close to 1. Thus, we may choose $\hat{g}(X_{l+K}, X_{l+K+1}, ...) = \sum_{k=l+K}^{\infty} \alpha^{k-(l+K)}[f(X_k) - \eta]$ as an estimate of potential $g(X_{l+K})$ and obtain

$$\begin{aligned}\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} &\approx \lim_{L\to\infty} \frac{1}{L}\sum_{l=0}^{L-1} \frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)} \sum_{k=l+K}^{\infty} \alpha^{k-(l+K)}[f(X_k) - \eta] \\ &= \lim_{L\to\infty} \frac{1}{L}\sum_{l=0}^{L-1} \frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)} \sum_{k=l+K}^{L+K-1} \alpha^{k-(l+K)}[f(X_k) - \eta].\end{aligned}$$

Interchanging the order of the two sums, we have

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} \approx \lim_{L\to\infty} \frac{1}{L}\sum_{l=0}^{L-1}[f(X_{l+K}) - \eta] \sum_{k=0}^{l} \alpha^{l-k} \frac{q(X_{k+K}|X_k)}{p^{(K)}(X_{k+K}|X_k)}. \tag{15}$$

With Eq. 15, we may develop the following Algorithm 1. This algorithm does not require the condition (10) for the policy gradient algorithms proposed in Baxter and Bartlett (2001) and Cao (2005). In the algorithm, Δ_l similarly converges to a biased estimate of the performance derivative as $l \to \infty$.

Algorithm 1 (With discounted potential approximation)

1. Given a state sequence $X_0, X_1, \ldots$ generated by transition probability matrix P and discount factor α;
2. Set $Z_0 = 0$, $\eta_0 = 0$, $\Delta_0 = 0$ and $l = 0$;
3. At time $l + K$, $l = 0, 1, 2, \ldots$, with state X_l and state X_{l+K}, do

$$Z_{l+1} = \alpha Z_l + \frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)}.$$

$$\eta_{l+1} = \eta_l + \frac{1}{l+1}[f(X_{l+K}) - \eta_l].$$

$$\Delta_{l+1} = \Delta_l + \frac{1}{l+1}\{[f(X_{l+K}) - \eta_{l+1}]Z_{l+1} - \Delta_l\}.$$

Next, let us use the perturbation realization factors in Cao and Chen (1997) to estimate the potentials. With this approach, the potentials are estimated based on Eq. 4. We first choose any state, denoted as i^*, as a reference state. For convenience, we set $X_0 = i^*$ and define $u_0 = 0$ and $u_{m+1} = \min\{n : n > u_m; X_n = i^*\}$ be the sequence of regenerative points. Then, for any time step k, there always exists an $m(k)$ such that $u_{m(k)} \le k < u_{m(k)+1}$. Set $g(i^*) = 0$. From Eq. 4, for any $X_k \neq i^*$, $u_{m(k)} < k < u_{m(k)+1}$, we have

$$g(X_k) = E\left\{ \sum_{l=k}^{u_{m(k)+1}-1} [f(X_l) - \eta] \Big| X_k \right\}.$$

Thus, we may use $\sum_{k=l+K}^{u_{m(l+K)+1}-1}[f(X_k) - \eta]$ to estimate potential $g(X_{l+K})$ in Eq. 13 when $X_{l+K} \neq i^*$; if $X_{l+K} = i^*$, $g(X_{l+K}) = 0$. Then, we obtain

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} = \lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} \left\{ \frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)} \sum_{k=l+K}^{u_{m(l+K)+1}-1} [f(X_k) - \eta][1 - I_{i^*}(X_{l+K})] \right\},$$

where $I_{i^*}(x)$ is an indicator function, i.e., $I_{i^*}(x) = 1$ if $x = i^*$, otherwise 0. Interchanging the order of the two sums, we have

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} = \lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} \left\{ [f(X_{l+K}) - \eta] \sum_{k=u_{m(l+K)}-K+1}^{l} \frac{q(X_{k+K}|X_k)}{p^{(K)}(X_{k+K}|X_k)} \right\}. \quad (16)$$

Denote $\sum_{k=u_{m(l+K)}-K+1}^{l} \frac{q(X_{k+K}|X_k)}{p^{(K)}(X_{k+K}|X_k)}$ by Z_{l+1}. We set $Z_{l+1} = 0$ if $X_{l+K} = i^*$ and let k begin from 0 if $u_{m(l+K)} - K + 1 < 0$ in Eq. 16.

With Eq. 16, we can develop the following policy gradient estimation Algorithm 2. This algorithm may be applied to the optimization schemes proposed in Marbach and Tsitsiklis (2001) to deal with the cases where assumption (10) does not hold. Δ_{l+1} in Eq. 18 calculates the average in Eq. 16, which converges to an unbiased estimate of the performance derivative as $l \to \infty$.

Algorithm 2 (With perturbation realization factors)

1. Given a state sequence $X_0, X_1, \ldots$ generated by transition probability matrix P and a reference state i^*;
2. Set $Z_0 = 0$, $\eta_0 = 0$, $\Delta_0 = 0$ and $l = 0$;
3. At time $l + K$, $l = 0, 1, 2, \ldots$, with state X_l and state X_{l+K}, do

$$Z_{l+1} = \begin{cases} Z_l + \dfrac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)}, & \text{if } X_{l+K} \neq i^*; \\ 0, & \text{if } X_{l+K} = i^*. \end{cases} \quad (17)$$

$$\eta_{l+1} = \eta_l + \frac{1}{l+1}[f(X_{l+K}) - \eta_l].$$

$$\Delta_{l+1} = \Delta_l + \frac{1}{l+1}\{[f(X_{l+K}) - \eta_{l+1}]Z_{l+1} - \Delta_l\}. \quad (18)$$

Finally, let us focus on the potential approximations obtained by truncation. That is, we use

$$g(X_{l+K}) \approx E\left\{\sum_{k=l+K}^{l+K+T}[f(X_k)-\eta]\Big|X_{l+K}\right\}.$$

When T is large, this gives a good approximation of potential $g(X_{l+K})$ in Eq. 3. T needs to be carefully chosen to balance the bias and variance of the estimate (Cao and Wan 1998). Thus, we may use $\sum_{k=l+K}^{l+K+T}[f(X_k)-\eta]$ to estimate $g(X_{l+K})$ in Eq. 13 and obtain

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} \approx \lim_{L\to\infty}\left\{\frac{1}{L}\sum_{l=0}^{L-1}\frac{q(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)}\sum_{k=l+K}^{l+K+T}[f(X_k)-\eta]\right\}.$$

Interchanging the order of the two sums, we have

$$\frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0} \approx \lim_{L\to\infty}\left\{\frac{1}{L}\sum_{l=0}^{L-1}[f(X_{l+K+T})-\eta]\sum_{k=l}^{l+T}\frac{q(X_{k+K}|X_k)}{p^{(K)}(X_{k+K}|X_k)}\right\}. \tag{19}$$

From Eq. 19, we have the following Algorithm 3, which improves the similar algorithm presented in Cao and Wan (1998).

Algorithm 3 (With potential approximation by truncation)

1. Given a state sequence $X_0, X_1, \ldots$ generated by transition probability matrix P and a truncation parameter T;
2. Set $Z_0 = \sum_{k=0}^{T}\frac{q(X_{k+K}|X_k)}{p^{(K)}(X_{k+K}|X_k)}$, $\eta_0 = 0$, $\Delta_0 = 0$ and $l = 1$;
3. At time $l+K+T$, $l = 1, 2, \ldots$, with states X_{l-1}, X_{l+T} and states X_{l+K-1}, X_{l+K+T}, do

$$Z_l = Z_{l-1} + \frac{q(X_{K+T+l}|X_{T+l})}{p^{(K)}(X_{K+T+l}|X_{T+l})} - \frac{q(X_{l+K-1}|X_{l-1})}{p^{(K)}(X_{l+K-1}|X_{l-1})}.$$

$$\eta_l = \eta_{l-1} + \frac{1}{l}[f(X_{K+T+l}) - \eta_{l-1}].$$

$$\Delta_l = \Delta_{l-1} + \frac{1}{l}\{[f(X_{K+T+l}) - \eta_l]Z_l - \Delta_{l-1}\}.$$

4 Further considerations

In this section, we show some further considerations that may utilize more information on the sample path and reduce the computation. These are achieved based on the following observation.

For any particular integer n, if there exist states u and $v \in \mathcal{S}$ such that $p^{(n)}(v|u) = 0$ but $q(v|u) \neq 0$, although $\lim_{L\to\infty}\frac{1}{L}\sum_{l=0}^{L-1}\frac{q(X_{l+n}|X_l)}{p^{(n)}(X_{l+n}|X_l)}g(X_{l+n})$ cannot estimate the gradient, it does estimate the sum of $\pi(i)q(j|i)g(j)$ for all $i, j \in \mathcal{S}$ with $p^{(n)}(j|i) > 0$. All the algorithms in Section 3 choose a K, and therefore the information contained

in n-step state transitions, with $0 < n < K$, are not utilized. Thus, we may use this information to get a more accurate estimate.

Let K be a positive integer such that $p^{(K)}(j|i) > 0$ for all $i, j \in \mathcal{S}$ with $q(j|i) \neq 0$. Define matrix $\tilde{Q}$ as follows,

$$\tilde{Q} = [\tilde{q}(j|i)], \quad \tilde{q}(j|i) = \frac{1}{K_{ij}} q(j|i),$$

where K_{ij} denotes the number of the positive numbers in $\{p(j|i), p^{(2)}(j|i), \ldots, p^{(K)}(j|i)\}$. For example, if $K=3$ and $p(j|i)=0$, $p^{(2)}(j|i)>0$, $p^{(3)}(j|i)>0$, then $K_{ij} = 2$.

From the definition of $\tilde{Q}$, $\lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} \frac{\tilde{q}(X_{l+K}|X_{l+K-n})}{p^{(n)}(X_{l+K}|X_{l+K-n})} g(X_{l+K})$ yields an estimate of the sum of $\frac{1}{K_{ij}} \pi(i) q(j|i) g(j)$ for all $i, j \in \mathcal{S}$ with $p^{(n)}(j|i) > 0$. Thus, we have

$$\lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} \left\{ \left[\frac{\tilde{q}(X_{l+K}|X_{l+K-1})}{p(X_{l+K}|X_{l+K-1})} + \cdots + \frac{\tilde{q}(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)} \right] g(X_{l+K}) \right\}$$

$$= \sum_{(i,j)\in S_1} \frac{1}{K_{ij}} \pi(i) q(j|i) g(j) + \cdots + \sum_{(i,j)\in S_K} \frac{1}{K_{ij}} \pi(i) q(j|i) g(j) \quad w.p.1 \tag{20}$$

$$= \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}} \pi(i) q(j|i) g(j) = \frac{d\eta(\theta)}{d\theta}\Big|_{\theta=0}, \qquad w.p.1, \tag{21}$$

where S_n denotes the set of state pairs (i, j) such that $p^{(n)}(j|i) > 0, n = 1, 2, \ldots, K$. Note that for any particular $i, j \in \mathcal{S}$, the term $\pi(i)q(j|i)g(j)$ appears in Eq. 20 K_{ij} times.

With Eq. 21, we can design the following Algorithm 4, which utilizes all the information contained in n-step transitions, $n = 1, 2, \ldots, K$, before each time $l + K$, $l = 0, 1, \ldots$. (Here, we only provide the algorithm with the discounted potential approximation; algorithms with perturbation realization factors and potential truncation approximation can be developed in a similar way.)

Algorithm 4 (With more information)

1. Given a state sequence $X_0, X_1, \ldots$ generated by transition probability matrix P and a discount factor α;
2. Set $Z_0 = 0$, $\eta_0 = 0$, $\Delta_0 = 0$ and $l = 0$;
3. At time $l + K$, $l = 0, 1, 2, \ldots$, with states $X_l, X_{l+1}, \ldots, X_{l+K}$, do

$$Z_{l+1} = \alpha Z_l + \left[\frac{\tilde{q}(X_{l+K}|X_{l+K-1})}{p(X_{l+K}|X_{l+K-1})} + \cdots + \frac{\tilde{q}(X_{l+K}|X_l)}{p^{(K)}(X_{l+K}|X_l)} \right].$$

$$\eta_{l+1} = \eta_l + \frac{1}{l+1}[f(X_{l+K}) - \eta_l].$$

$$\Delta_{l+1} = \Delta_l + \frac{1}{l+1}\{[f(X_{l+K}) - \eta_{l+1}]Z_{l+1} - \Delta_l\}.$$

In Algorithm 4, we need to compute $p^{(2)}(j|i), \ldots, p^{(K)}(j|i)$ for all $i, j \in \mathcal{S}$ with $q(j|i) \neq 0$ to determine K_{ij} and matrix $\tilde{Q}$. To reduce computation, we may define

the following matrixes to replace matrix $\tilde{Q}$:

$$Q_1 = Q, \qquad Q_n = [q_n(j|i)],$$

where

$$q_n(j|i) = \begin{cases} 0, & if\ p^{(m)}(j|i) > 0\ for\ some\ 1 \le m < n, \\ q(j|i), & otherwise, \end{cases} \qquad 1 < n \le K.$$

Let K^* be the minimal K such that $p^{(K)}(j|i) > 0$ for all $i, j \in \mathcal{S}$ with $q(j|i) \neq 0$, then it follows that $Q_n = \mathbf{0}, n > K^*$. From the definition of Q_n, we can find that $\lim_{L\to\infty} \frac{1}{L}\sum_{l=0}^{L-1} \frac{q_n(X_{l+K^*}|X_{l+K^*-n})}{p^{(n)}(X_{l+K^*}|X_{l+K^*-n})} g(X_{l+K^*})$ estimates the sum of $\pi(i)q(j|i)g(j)$ for all $i, j \in \mathcal{S}$ with $p^{(n)}(j|i) > 0$ and $p^{(m)}(j|i) = 0, 1 \le m < n \le K^*$. Thus, we may use

$$\lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} \left[\frac{q_1(X_{l+K^*}|X_{l+K^*-1})}{p(X_{l+K^*}|X_{l+K^*-1})} + \cdots + \frac{q_{K^*}(X_{l+K^*}|X_l)}{p^{(K^*)}(X_{l+K^*}|X_l)} \right] g(X_{l+K^*})$$

to estimate the gradient and develop the on-line policy gradient estimation algorithms. The advantage of this approach is that if $p^{(n)}(j|i) > 0$, we needn't compute the probabilities $p^{(n+1)}(j|i), p^{(n+2)}(j|i), \ldots, P^{(K^*)}(j|i)$ because $q_m(j|i) = 0, m > n$. Thus, we may use the well-known Dijkstra's algorithm (Cormen et al. 2001) to determine the smallest transition step number from state $i \in \mathcal{S}$ to state $j \in \mathcal{S}$ and only compute the transition probability with smallest transition step number for each state pair i, j such that $q(j|i) \neq 0$.

Finally, when K is large enough, we have $p^{(K)}(X_{l+K}|X_l) \approx \pi(X_{l+K})$ and Eq. 13 becomes

$$\left.\frac{d\eta(\theta)}{d\theta}\right|_{\theta=0} \approx \lim_{L\to\infty} \frac{1}{L} \sum_{l=0}^{L-1} \frac{q(X_{l+K}|X_l)}{\pi(X_{l+K})} g(X_{l+K}), \qquad w.p.1. \tag{22}$$

However, with Eq. 22, we need to know $\pi(i)$ for all states $i \in \mathcal{S}$.

5 Simulation results

In this section, the simulation results for the above algorithms are given. We first consider the simple Example 1 and then consider a practical example.

For Example 1, if $K \ge 3$, $p^{(K)}(j|i) > 0$ for all $i, j \in \mathcal{S}$. For comparison, we consider $K = 1, 2, 3, 4, 5, 10, 100$. Then $K = 1, 2$ correspond to the cases that assumption (14) does not hold. We set $\alpha = 0.9$ in Algorithm 1, set $i^* = 1$ in Algorithm 2 and set $T = 10$ in Algorithm 3. Running these algorithms 10 times, respectively, each with 100,000 transitions, the simulation results are listed in Table 1, Table 2 and Table 3, respectively, in which "Mean" denotes the average of the ten gradient estimates and "SD" denotes the standard deviation of these ten estimates.

From these results, we observe that when $K = 1, 2$, all the gradient estimates have a large bias. Note that the case $K = 1$ corresponds to the standard importance sampling case. Moreover, we may find that the unbiased estimate obtained by Algorithm 2 has a larger SD than the biased estimates obtained by Algorithms 1 and 3.

Table 1 Simulation results of Algorithm 1 with $\alpha = 0.9$

K	1	2	3	4	5	10	100	Theoretic
Mean	0.0031	0.0709	−0.6403	−0.6433	−0.6362	−0.6450	−0.6338	−0.6633
SD	0.0248	0.0107	0.0336	0.0418	0.0324	0.0315	0.0355	

Table 2 Simulation results of Algorithm 2 with $i^* = 1$

K	1	2	3	4	5	10	100	Theoretic
Mean	−1.9762	−0.0050	−0.6566	−0.6718	−0.6658	−0.6645	−0.6691	−0.6633
SD	0.0812	0.0407	0.0453	0.0468	0.0416	0.0443	0.0447	

Table 3 Simulation results of Algorithm 3 with $T = 10$

K	1	2	3	4	5	10	100	Theoretic
Mean	−0.0090	0.0681	−0.6749	−0.6549	−0.6473	−0.6746	−0.6460	−0.6633
SD	0.0246	0.0231	0.0395	0.0343	0.0296	0.0355	0.0215	

Table 4 Simulation results of Algorithm 4 with $\alpha = 0.9$

K	3	4	5	6	10	100	Theoretic
Mean	−0.6516	−0.6553	−0.6479	−0.6484	−0.6541	−0.6535	−0.6633
SD	0.0252	0.0149	0.0205	0.0259	0.0202	0.0151	

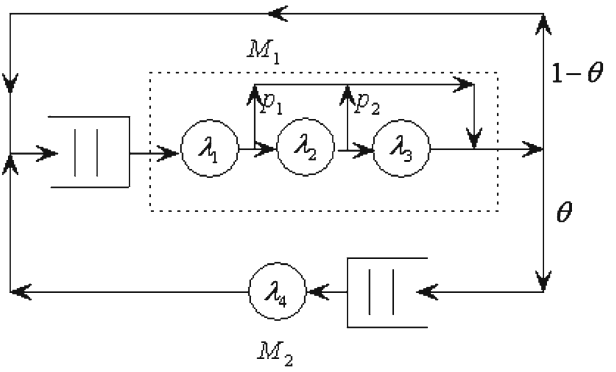

Fig. 1 Simulation results of existing algorithms

Table 5 Simulation results of existing algorithms

$K = 1$	$\alpha = 0.99$	$i^* = 1$	$T = 20$
Mean	0.0803	1.7310	0.0337
SD	0.2307	0.0796	0.0356

The discount factor α in Algorithm 1 and truncation parameter T in Algorithm 3 play important role in the tradeoff between bias and SD. A larger α or T will lead to a larger SD but a smaller α or T will lead to a larger bias. Generally, the choices of α and T are problem dependent. We also estimate the performance gradient in Example 1 by using Algorithm 4 with $\alpha = 0.9$ and samely run the algorithm 10 times. The simulation results are listed in Table 4. Compared with Algorithm 1, since this algorithm utilizes more information to estimate the gradient, it indeed provides more accurate estimates.

Example 2 Consider a manufacturing system (Cao 2007) consists of two machines, M_1 and M_2, and N pieces of works, as shown in Fig. 1. Each work piece undertakes three consecutive operations at M_1. The service times at these three operations are exponential distribution with rates λ_1, λ_2 and λ_3, respectively. M_2 has only one operation with an exponential distributed service time with rate λ_4. A work piece, after the completion of its service at each operation of M_1, will leave M_1 with probability $p_i, i = 1, 2, 3$, with $p_3 = 1$, or go to next operation with probability $1 - p_i$; if the work piece leaves M_1, it will go to M_2 with probability θ, where $\theta \in (0, 1]$ is a tuning parameter, or return M_1 with probability $1 - \theta$.

To optimize the parameter θ, we first want to obtain the derivative of system performance with respect to θ. For this example, the marginal case $\theta = 1$ is very important and the gradient information at $\theta = 1$ is our concern. We may use the embedded Markov chain to model this system and easily verify that $\theta = 1$ leads to the case that the standard importance sampling assumption does not hold. We consider $N = 3$ and assume the performance function is $f = [0, 1, 2, 3, 4, 5, 6, 7, 0, 0]^T$. The simulation results of the existing algorithms for the case that $\lambda_1 = \lambda_3 = \lambda_4 = 1, \lambda_2 = 2$, $p_1 = 0.2$ and $p_2 = 0.4$ are listed in Table 5, where we also run these algorithms 10 times. Compared with the theoretical value of gradient $\frac{d\eta}{d\theta}|_{\theta=1} = -0.5348$, the existing policy gradient estimates (K=1) are very poor and even with wrong direction. However, the algorithms with $K = 4$ provide good estimates as described in Table 6. Here, the computational complexity of iteration (12) is 22 multiplications and the total computation to obtain $p^{(K)}(j|i)$ is $4 * 18 * 22$, where "18" is the number of (i, j)'s such that $q(j|i) \neq 0$ and the sparse structures of P and $\frac{dP}{d\theta}$ have been used.

Table 6 Simulation results of algorithms with $K = 4$

$K = 4$	$\alpha = 0.99$	$i^* = 1$	$T = 20$	Algorithm 4	Theoretic
Mean	−0.5107	−0.5542	−0.5306	−0.5313	−0.5348
SD	0.0747	0.0559	0.0435	0.0273	

6 Conclusion

In this note, we proposed a few new sample-path-based (on-line) algorithms with multi-step sampling for estimating the performance gradients of Markov systems; these algorithms do not require the standard importance sampling assumption (10) used in the existing algorithms. We also discussed possible ways to improve the accuracy of the estimates and reduce the computation. These algorithms can be used in performance optimization in a wider class of systems.

One possible issue related to these algorithms is that the multi-step transition probabilities need to be computed. We hope that more efficient algorithms can be developed, where the estimation of the multi-step transition probabilities might be incorporated into the algorithms.

It is interesting to note that the two approaches, the policy gradient approach in reinforcement learning and perturbation analysis in discrete event dynamic systems, have different emphasis. Perturbation analysis focuses on "constructing", or deriving, formulas for performance gradients by studying the system dynamics to determine the effect of any fictitiously introduced perturbations on the system performance (Cao and Chen 1997; Cao and Wan 1998; Cao 2007). The policy gradient approach, on the other hand, focuses on developing efficient and practical algorithms to estimate the performance gradients by using the gradient formulas (Baxter and Bartlett 2001; Baxter et al. 2001; Greensmith et al. 2004; Cao 2005). A combination of both may lead to better perspectives and results.

Acknowledgement This work is supported by a grant from Hong Kong UGC.

References

Baxter J, Bartlett PL (2001) Infinite-horizon policy-gradient estimation. J Artif Intell Res 15: 319–350

Baxter J, Bartlett PL, Weaver L (2001) Experiments with infinite-horizon policy-gradient estimation. J Artif Intell Res 15:351–381

Bertsekas DP (1995) Dynamic programming and optimal control, vols I and II. Athena Scientific, Belmont

Cao XR (2005) A basic formula for online policy gradient algorithms. IEEE Trans Automat Contr 50(5):696–699

Cao XR (2007) Stochastic learning and optimization: a sensitivity-based approach. Springer, New York

Cao XR, Chen HF (1997) Perturbation realization, potentials and sensitivity analysis of Markov processes. IEEE Trans Automat Contr 42(10):1382–1393

Cao XR, Wan YW (1998) Algorithms for sensitivity analysis of Markov systems through potentials and perturbation realization. IEEE Trans Control Syst Technol 6(4):482–494

Cinlar E (1975) Introduction to stochastic processes. Prentice Hall, Englewood Cliffs

Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms, 2nd edn. MIT and McGraw-Hill, Cambridge

Greensmith E, Bartlett PL, Baxter J (2004) Variance reduction techniques for gradient estimates in reinforcement learning. J Mach Learn Res 5:1471–1530

Marbach P, Tsitsiklis JN (2001) Simulation-based optimization of Markov reward processes. IEEE Trans Automat Contr 46(2):191–209

Puterman ML (1994) Markov decision processes: discrete stochastic dynamic programming. Wiley, New York

Yan-Jie Li received the M.S. and Ph.D. degrees from University of Science and Technology of China, in 2006. He then worked as a research associate in Hong Kong University of Science and Technology until August 2008. Now he is an assistant professor in division of control and Mechatronics, Harbin Institute of Technology Shenzhen Graduate School. His research interests include the control and optimization of stochastic systems and reinforcement learning.

Fang Cao received the M.S. and Ph.D. degrees from Hong Kong University of Science and Technology, in 2008. Now, she is an assistant professor in the School of Electronics and Information Engineering, Beijing Jiaotong University. Her research interests include stochastic learning and optimization.

Xi-Ren Cao received the M.S. and Ph.D. degrees from Harvard University, in 1981 and 1984, respectively, where he was a research fellow from 1984 to 1986. He then worked as a consultant engineer/engineering manager at Digital Equipment Corporation, Massachusetts, U.S.A, until October 1993. Then he joined the Hong Kong University of Science and Technology (HKUST), where he is currently chair professor, director of the Research Center for Networking. He held visiting positions at Harvard University, University of Massachusetts at Amherst, AT&T Labs, University of Maryland at College Park, University of Notre Dame, Tsinghua University, University of Science and Technology of China, and other universities. Dr. Cao owns three patents in data- and telecommunications and published three books in the area of stochastic learning and optimization and discrete event dynamic systems. He received the Outstanding Transactions Paper Award from the IEEE Control System Society in 1987, the Outstanding Publication Award from the Institution of Management Science in 1990, and the Outstanding Service Award from IFAC in 2008. He was elected as a Fellow of IEEE in 1995, and as a Fellow of IFAC in 2008. He is Editor-in-Chief of Discrete Event Dynamic Systems: Theory and Applications, Associate Editor at Large of IEEE Transactions of Automatic Control, and he served as the Chairman of IEEE Fellow Evaluation Committee of IEEE Control System Society (2005-2007), and member on the Board of Governors of IEEE Control Systems Society. He is the chairman of IFAC Coordinating Committee on Systems and Signals (2006–2011) and on the Technical Board of IFAC, He is/was associate editor of a number of international journals and chairman of a few technical committees of international professional societies. His current research areas include discrete event dynamic systems, stochastic learning and optimization, performance analysis of communication systems, signal processing, and financial engineering. 2