

Two-timescale algorithms for simulation optimization of hidden Markov models

SHALABH BHATNAGAR¹, MICHAEL C. FU^{2,*}, STEVEN I. MARCUS³ and SHASHANK BHATNAGAR⁴

¹Institute for Systems Research, ²The Robert H. Smith School of Business and Institute for Systems Research
E-mail: mfu@rhsmith.umd.edu, and ³Department of Electrical and Computer Engineering and Institute for Systems Research,
University of Maryland, College Park, MD 20742, USA

⁴Department of Physics, University of Addis Ababa, P. O. Box 21926, Code 1000, Addis Ababa, Ethiopia

Received May 1999 and accepted June 2000

We propose two finite difference two-timescale Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms for simulation optimization of hidden Markov models. Stability and convergence of both the algorithms is proved. Numerical experiments on a queueing model with high-dimensional parameter vectors demonstrate orders of magnitude faster convergence using these algorithms over related $(N + 1)$ -Simulation finite difference analogues and another Two-Simulation finite difference algorithm that updates in cycles.

1. Introduction

A popular approach to simulation optimization of discrete event systems with continuous-valued parameters is based on stochastic approximation (Kushner and Clark, 1978; Pflug, 1996; Kushner and Yin, 1997; Andradóttir, 1998). Gradient descent stochastic approximation algorithms are typically used to perform function optimization in cases where the function to be optimized is difficult to compute analytically, see for instance, Fu (1994). Several stochastic approximation schemes that have been used for optimization of long run average performance measures suffer from the drawback that they require data aggregation (for averaging) over regeneration epochs (Fu, 1990; Chong and Ramadge, 1994). These epochs can be very infrequent (particularly for large systems/networks), making the scheme extremely slow in practice. In Chong and Ramadge (1993), an algorithm that updates the parameter after a fixed number of customers is presented and convergence proved. However, the algorithms in Fu (1990) and also Chong and Ramadge (1993, 1994) all require the availability of direct gradient estimates, and are all based on infinitesimal perturbation analysis (Ho and Cao, 1991). In L'Ecuyer *et al.* (1994) and L'Ecuyer and Glynn (1994), various stochastic approximation algorithms governed by finite difference estimates (as well as direct gradient estimates) were considered for opti-

mizing a steady-state performance measure with respect to a scalar parameter in a single-server queue.

In Bhatnagar and Borkar (1997, 1998), a more general setting for vector parameters and long run average performance measures is considered, in which the parameter is updated at deterministic instants that are obtained using two timescales: a faster timescale at which the system evolves, and a slower timescale at which the parameter is updated. Specifically, in Bhatnagar and Borkar (1997), the parameter is updated at deterministically increasing time instants that are in turn obtained using two timescales (see also Bartusek and Makowski (1994)). On the other hand, in Bhatnagar and Borkar (1998), the parameter is updated at every instant using coupled iterations that are governed by different timescales. However as with any other forward finite difference scheme (L'Ecuyer *et al.*, 1994; L'Ecuyer and Glynn, 1994), these schemes also require $(N + 1)$ parallel simulations for an N -vector parameter. A proposed alternative in Bhatnagar and Borkar (1997) uses only two parallel simulations at any instant by moving the algorithm in bigger loops or cycles but results in slow convergence.

Spall (1992) proposed a stochastic approximation technique that requires only two simulations for a parameter vector of any dimension and updates all parameter components at every instant. This technique came to be known as Simultaneous Perturbation Stochastic Approximation (SPSA), since it simultaneously perturbs the various parameter components randomly, most commonly by using independent and identically distributed (i.i.d.), symmetric Bernoulli random variables

*Corresponding author

in the two simulations, and uses the estimates thus obtained for updating the parameter. It has been applied in various contexts; for instance, see Fu and Hill (1997) for an application of SPSA to optimization of discrete event systems. Most of the work in discrete event systems, except Fu and Hill (1997), is on low-dimensional problems (Chong and Ramadge, 1993; L'Ecuyer *et al.*, 1994; L'Ecuyer and Glynn, 1994; Kushner and Vazquez-Abad, 1996; Bhatnagar and Borkar, 1997, 1998). In Spall (1992) and Kushner and Yin (1997), a general idea for high-dimensional problems is proposed.

Motivated by all of the above considerations, in this paper we develop SPSA variants (that we call SPSA-1 and SPSA-2) of the two-timescale algorithms of Bhatnagar and Borkar (1997, 1998) (respectively) for optimizing high-dimensional parameters in hidden Markov models (cf. Elliott *et al.* (1995)). The algorithm SPSA-1 uses only two parallel simulations and updates the parameter at increasing time instants as in Bhatnagar and Borkar (1997). The algorithm SPSA-2 also uses only two parallel simulations but has the added advantage (over the algorithm in Bhatnagar and Borkar (1998)) that it allows for data aggregation over a fixed number of instants in between two successive updates of the parameter for better performance. Moreover, the algorithm in Bhatnagar and Borkar (1998) was only for ordinary Markov processes and not hidden Markov models (which is a more general setting) considered here. We prove the convergence of both of these schemes and numerically demonstrate the algorithms on a feedback queueing network with high-dimensional parameters. These schemes are found to converge orders of magnitude faster than their $(N+1)$ -Simulation analogues in Bhatnagar and Borkar (1997, 1998), and also the Two-Simulation algorithm of Bhatnagar and Borkar (1997) that moves in cycles.

Hidden Markov models arise in many queueing and stochastic control applications. To illustrate a very simple instance of a hidden Markov model, consider an $M/G/1$ queue. Let $\{q_n\}$ represent the queue length process observed at customer arrival epochs. Similarly, let $\{r_n\}$ represent the sequence of residual service times of the customers in service at these epochs. Then the joint process $\{(q_n, r_n)\}$ is Markov. In most real life applications only the process $\{q_n\}$ is observed whereas $\{r_n\}$ is not. Thus in this example, $\{q_n\}$ represents a hidden Markov model.

In Bhatnagar *et al.* (1999a), the algorithms SPSA-1 and a special case of SPSA-2 were applied for the closed loop feedback control of Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks, by considering parameterized feedback policies. A finite state setting was considered there and as a result there was no problem with stability of the schemes. We develop these algorithms in this paper in the framework of hidden Markov models with an unbounded state space, and therefore stability issues are explicitly addressed. The convergence of SPSA-1 in the finite state setting of

Bhatnagar *et al.* (1999a) was proven in Bhatnagar *et al.* (1999b). The convergence analysis of SPSA-2 in any setting has not been shown earlier.

The rest of the paper is organized as follows: in the next section, we formulate the optimization problem, present the assumptions on the system and give a result on tightness of the stationary measures for the hidden Markov model. In Section 3, we present our SPSA algorithms (SPSA-1 and SPSA-2) and compare their performance with their corresponding analogues in Bhatnagar and Borkar (1997, 1998). In Section 4, we present the detailed convergence analyses for both algorithms. In order to save space, however, we skip the more straightforward steps in the analysis and refer the interested reader to Bhatnagar *et al.* (2000) for all the details. In Section 5, numerical results comparing the SPSA algorithms with those of Bhatnagar and Borkar (1997, 1998) for a simple queueing system are presented. Finally, Section 6 provides the concluding remarks.

2. The optimization problem

The process that we seek to optimize is an \mathcal{R}^d -valued parameterized (with parameter $\theta \in \mathcal{R}^N$) Hidden Markov Model (HMM) represented by $\{Y(j), j \geq 0\}$ and is given by the set of coupled iterations

$$X(j+1) = F(X(j), Y(j), \xi(j), \theta), \quad (1)$$

$$Y(j+1) = G(X(j), Y(j), \eta(j), \theta), \quad (2)$$

$j \geq 0$. Here the state process $\{X(j)\}$ is \mathcal{R}^q -valued and is unobserved or hidden. Further, $\{\xi(j)\}$, $\{\eta(j)\}$ are i.i.d. sequences in \mathcal{R}^l and \mathcal{R}^s , respectively, and are mutually independent. The maps $F: \mathcal{R}^q \times \mathcal{R}^d \times \mathcal{R}^l \times \mathcal{R}^N \rightarrow \mathcal{R}^q$ and $G: \mathcal{R}^q \times \mathcal{R}^d \times \mathcal{R}^s \times \mathcal{R}^N \rightarrow \mathcal{R}^d$ are measurable. Further, $\theta \triangleq (\theta_1, \dots, \theta_N)^T \in \mathcal{R}^N$ represents the parameter to be tuned in order to minimize a certain cost function $J(\theta)$ (defined below). We will assume that θ takes values in a set of the form $C \triangleq \prod_{i=1}^N [\theta_{i,\min}, \theta_{i,\max}]$. We assume that the joint process $\{(X(j), Y(j))\}$ is ergodic Markov for every fixed θ , and has stationary distribution $\mu_\theta(dx, dy)$. Let $\nu_\theta(dy)$ be the marginal (corresponding to $\{Y(j)\}$) of this stationary distribution. Also, let $p_\theta(x, y; dx', dy')$ represent the transition kernel of $\{(X(j), Y(j))\}$. Let $h: \mathcal{R}^d \rightarrow \mathcal{R}$ be a given bounded and continuous cost function. Our aim then is to find a θ in the set C that minimizes the average cost

$$J(\theta) \triangleq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=0}^{N-1} h(Y(j)).$$

Since $\{(X(j), Y(j))\}$ is ergodic Markov (for fixed θ), the above limit exists and

$$J(\theta) = \int h(y) \nu_\theta(dy).$$

Remark. As shown in Borkar (1993), very general classes of random processes have an HMM representation if one is permitted time nonhomogeneity namely, replace F and G in (1) and (2) above with $F_j, G_j, j \geq 0$. Also, $\{\xi(j)\}$ and $\{\eta(j)\}$ can be taken to be uniformly distributed on $[0, 1]$ without loss of generality.

We now proceed with the rest of the model. As mentioned earlier, the parameter θ in (1) and (2) has to be tuned in order to minimize $J(\theta)$. Let $\tilde{\theta}_n \in C$ represent the parameter value at instant n . Thus in particular, we will consider the following dynamics:

$$X(j+1) = F(X(j), Y(j), \xi(j), \tilde{\theta}_j), \quad (3)$$

$$Y(j+1) = G(X(j), Y(j), \eta(j), \tilde{\theta}_j), \quad (4)$$

$j \geq 0$. Let $\mathcal{G}_n \triangleq \sigma(X(j), Y(j), \xi(j), \eta(j), \tilde{\theta}_j, j \leq n), n \geq 0$, represent the σ -field generated by $\{X(j), Y(j), \xi(j), \eta(j), \tilde{\theta}_j, j \geq 0\}$ up to instant n . Note that since $\{\tilde{\theta}_j\}$ can be any arbitrarily defined parameter sequence, $\{(X(j), Y(j))\}$ as in (3) and (4) is in general not Markov. We thus force $\{(X(j), Y(j))\}$ to be Markov by assuming that any sequence $\{\tilde{\theta}_j, j \geq 0\}$ satisfies

$$\begin{aligned} P(X(n+1) \in A, Y(n+1) \in B | \mathcal{G}_n) \\ = p_{\tilde{\theta}_n}(X(n), Y(n); A, B), \end{aligned} \quad (5)$$

for any A, B , Borel in \mathcal{R}^d and \mathcal{R}^d respectively. We call any such $\{\tilde{\theta}_n\}$ satisfying (5), an *M-Sequence*.

Next, we list the following assumptions on our system. Assumptions (A1), (A2) and (A3) are needed to prove convergence of our first algorithm SPSA-1, while Assumptions (A1), (A2') and (A3) are required for convergence of algorithm SPSA-2. Both algorithms SPSA-1 and SPSA-2 are presented in Section 3.

Assumptions

- (A1) The average cost $J(\theta)$ is continuously differentiable.
- (A2) The map $(\theta, x, y) \rightarrow p_\theta(x, y; dx', dy')$ is continuous.
- (A2') For any $h \in C(\mathcal{R}^{q+d})$ vanishing at ∞ ,

$$\lim_{\|(x,y)\| \rightarrow \infty} \int p_\theta(x, y; dx', dy') h(x', y') = 0,$$

uniformly over $\theta \in C$.

- (A3) *Liapunov Stability Condition:* There exist nonnegative $V \in C(\mathcal{R}^{q+d})$, $K \subset \mathcal{R}^{q+d}$ compact and $\epsilon_0 > 0$ such that under any M-Sequence $\{\tilde{\theta}_n\}$,

1. $\lim_{\|(x,y)\| \rightarrow \infty} V(x, y) = \infty$,
2. $\sup_n E[V(X(n), Y(n))^2] < \infty$,
3. $E[V(X(n+1), Y(n+1)) | \mathcal{G}_n] \leq V(X(n), Y(n)) - \epsilon_0$, whenever $(X(n), Y(n)) \notin K, n \geq 0$.

In the above, $\|\cdot\|$ represents sup norm. Note that a sufficient condition for Assumption (A1) to be satisfied is that the parameterized stationary distribution $\mu_\theta(dx, dy)$ of the ergodic Markov process $\{(X(n), Y(n))\}$ be con-

tinuously differentiable in the parameter θ . In Vazquez-Abad and Kushner (1992), certain sufficiency conditions for showing the latter have been given. Assumptions (A2) and (A2') are technical conditions that are satisfied routinely by most systems, see Bhatnagar and Borkar (1997, 1998). Finally, Assumption (A3) is required to ensure that the system remains stable, and is a standard assumption.

Recall that a sequence $\{\hat{\mu}_n\}$ of probability measures on S is tight if for each $\epsilon > 0$, there exists a compact set $K_\epsilon \subset S$ such that $\hat{\mu}_n(K_\epsilon) > 1 - \epsilon$ for all n . The following lemma shows that the sets of all parameterized stationary distributions $\{\mu_\theta, \theta \in C\}$ and their marginals $\{v_\theta, \theta \in C\}$ are tight and crucially uses Assumption (A3) for its proof. This result is required later in the analysis.

Lemma 2.1. $\{\mu_\theta, \theta \in C\}$ (resp., $\{v_\theta, \theta \in C\}$) is compact and the map $\theta \rightarrow \mu_\theta$ (resp., v_θ) is continuous.

Proof. Follows in exactly the same manner as Lemma 2.1 of Bhatnagar and Borkar (1997). ■

3. Two-timescale SPSA algorithms

In this section, we present our two-timescale SPSA algorithms and compare their performance with corresponding two-timescale finite difference algorithms in Bhatnagar and Borkar (1997, 1998). In order to put things in proper perspective and to clearly bring out the advantages of our SPSA algorithms, we first begin with the $(N+1)$ -Simulation finite difference stochastic approximation algorithm of Bhatnagar and Borkar (1997) that we refer to as $(N+1)$ -Simulation FDSA-1 and its corresponding Two-Simulation alternative (proposed in that paper) referred here as Two-Simulation FDSA-1. We then present our first SPSA algorithm (SPSA-1). Later, we illustrate the $(N+1)$ -Simulation finite difference algorithm of Bhatnagar and Borkar (1998) that we refer to as $(N+1)$ -Simulation FDSA-2, followed by its generalized SPSA version (SPSA-2). Finally, we briefly compare all these algorithms and argue the reasons for the superior performance of SPSA-1 and SPSA-2 over the algorithms in Bhatnagar and Borkar (1997, 1998).

3.1. The algorithms

The algorithms presented here are called two-timescale algorithms since they are governed by two step-size sequences (or timescales) $\{a(n)\}$ and $\{b(n)\}$ defined below. Before proceeding, we define some notation. Let $\delta > 0$ be a fixed small constant. Let $\pi_i(x) \triangleq \min(\max(\theta_{i,\min}, x), \theta_{i,\max})$, $i = 1, \dots, N$, denote the point closest to $x \in \mathcal{R}$ in the interval $[\theta_{i,\min}, \theta_{i,\max}] \subset \mathcal{R}$, $i = 1, \dots, N$, and $\pi(\theta)$ be defined by the vector $\pi(\theta) \triangleq (\pi_1(\theta_1), \dots, \pi_N(\theta_N))^T$. Then $\pi(\theta)$ is a projection of $\theta \in \mathcal{R}^N$ onto the set C . Define sequences $\{a(n)\}$ and $\{b(n)\}$ as follows: $a(0) = b(0) = 1$,

$a(i) = i^{-1}$, $b(i) = i^{-\alpha}$, $i \geq 1$, and with $1/2 < \alpha < 1$. Then clearly,

$$\begin{aligned} \frac{a(n+1)}{a(n)}, \frac{b(n+1)}{b(n)} &\rightarrow 1, \quad \text{as } n \rightarrow \infty, \quad (6) \\ \sum_n a(n) &= \sum_n b(n) = \infty, \\ \sum_n a(n)^2, \sum_n b(n)^2 &< \infty, \quad a(n) = o(b(n)). \quad (7) \end{aligned}$$

Define $\{n_m, m \geq 0\}$ as follows: $n_0 = 1$ and $n_{m+1} = \min\{j > n_m \mid \sum_{i=n_m+1}^j a(i) \geq b(m)\}$, $m \geq 1$. Then $\{n_m\}$ represents a deterministically increasing sequence of points. In SPSSA-1 (as also $(N+1)$ -Simulation FDSA-1 and Two-Simulation FDSA-1), $\{n_m\}$ defines the parameter update instants of the algorithm.

Note that any Finite Difference Stochastic Approximation (FDSA) scheme ordinarily requires $(N+1)$ parallel simulations for an N -vector parameter. The two-timescale stochastic approximation algorithm in Bhatnagar and Borkar (1997) for an N -vector parameter is thus as follows:

$(N+1)$ -Simulation FDSA-1

The first simulation corresponds to $\{(X(j), Y(j))\}$ and is governed by $\{\theta_j\}$ that is in turn defined by $\theta_j = \theta(m)$, for $n_m \leq j < n_{m+1}$. The remaining N parallel simulations are represented by $\{(X_i(j), Y_i(j))\}$, $i = 1, \dots, N$, and are respectively governed by $\{\tilde{\theta}^i(j)\}$, $i = 1, \dots, N$, with $\tilde{\theta}^i(j) = \theta(m) + \delta e_i$, for $n_m \leq j < n_{m+1}$, and where e_i is the unit vector with 1 in the i th component. Now, for $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(m+1) &= \pi_i \left(\theta_i(m) + \sum_{j=n_m+1}^{n_{m+1}} a(j) \left(\frac{h(Y(j)) - h(Y_i(j))}{\delta} \right) \right). \quad (8) \end{aligned}$$

It is clear that one would require $N+1$ parallel simulations using this algorithm. An alternative scheme was proposed in Bhatnagar and Borkar (1997) that uses only two parallel simulations at any instant. This it achieves by moving the algorithm in 'cycles', in each of which only one component is updated. This scheme that we call Two-Simulation FDSA-1 is as follows:

Two-Simulation FDSA-1

The first simulation here corresponds to $\{(X(j), Y(j))\}$ and is governed by $\{\theta_j\}$ where the parameter θ_j is the N -vector $\theta_j = (\theta_{j,1}, \dots, \theta_{j,N})^T$ with $\theta_{j,i} = \theta_i(m)$, for $n_{Nm+i-1} \leq j < n_{Nm+i}$, $i = 1, \dots, N$, $m \geq 0$ (with N being the dimensionality of the parameter vector), where $\theta_{j,i}$ (resp. $\theta_i(m)$) is the i th component of θ_j (resp. $\theta(m)$). The second simulation is now represented as $\{(\tilde{X}(j), \tilde{Y}(j))\}$

and is governed by $\{\hat{\theta}_j\}$ defined by $\hat{\theta}_j = \theta(m) + \delta e_i$, for $n_{Nm+i-1} \leq j < n_{Nm+i}$, $i = 1, \dots, N$, $m \geq 0$. The algorithm is as follows: For $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(m+1) &= \pi_i \left(\theta_i(m) + \sum_{j=n_{Nm+i-1}+1}^{n_{Nm+i}} a(j) \left(\frac{h(Y(j)) - h(\tilde{Y}(j))}{\delta} \right) \right). \quad (9) \end{aligned}$$

Thus using this scheme, the whole parameter is updated once every n_{Nm} steps instead of the n_m steps required for one update using the $(N+1)$ -Simulation FDSA-1 version (8) of it. Next, we present our first randomized difference SPSSA algorithm (SPSSA-1).

Let for any $i \geq 0$, $\Delta(i) \triangleq (\Delta_{1,i}, \dots, \Delta_{N,i})^T$ be a vector of i.i.d. mean-zero random variables with each $\Delta_{j,i}$, $j = 1, \dots, N$, taking values in a compact set $E \subset \mathcal{R}^N$. We assume that these random variables satisfy Condition (B) below.

Condition (B). There exists a constant $\bar{K} < \infty$, such that for any $i \geq 0$, and $l \in \{1, \dots, N\}$, $E[(\Delta_{l,i})^{-2}] \leq \bar{K}$.

Further, we assume that $\{\Delta(i)\}$ is a mutually independent sequence with $\Delta(i)$ independent of $\sigma(\theta(l), l \leq i)$, the filtration (σ -field) generated by the sequence of parameter updates. Condition (B) is a standard condition in SPSSA algorithms. Minor variants of this condition are for instance available in Spall (1992). Note that distributions like Gaussian and Uniform are precluded while using Condition (B). An important consequence of $E[\Delta_{l,i}^{-2}] < \infty$ is that $P(\Delta_{l,i} = 0) = 0$.

We now proceed with our first algorithm, SPSSA-1, wherein we use only two parallel simulations and update all parameter components every n_m instants by perturbing all of these simultaneously along random directions in the two simulations.

SPSSA-1

Consider two parallel simulations $\{(X^k(j), Y^k(j))\}$, $k = 1, 2$, respectively governed by $\{\tilde{\theta}_j^k\}$, $k = 1, 2$, as follows: For the process $\{(X^1(j), Y^1(j))\}$, we define $\tilde{\theta}_j^1 = \theta(m) - \delta \Delta(m)$, for $n_m < j \leq n_{m+1}$, $m \geq 0$. The parameter sequence $\{\tilde{\theta}_j^2\}$ for $\{(X^2(j), Y^2(j))\}$ is similarly defined by $\tilde{\theta}_j^2 = \theta(m) + \delta \Delta(m)$, for $n_m < j \leq n_{m+1}$, $m \geq 0$. In the above, $\theta(m) \triangleq (\theta_1(m), \dots, \theta_N(m))^T$ is the value of the parameter update that is governed by the following recursion equations. For $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(m+1) &= \pi_i \left(\theta_i(m) + \sum_{j=n_m+1}^{n_{m+1}} a(j) \left(\frac{h(Y^1(j)) - h(Y^2(j))}{2\delta \Delta_{m,i}} \right) \right), \quad (10) \end{aligned}$$

$m \geq 0$. It will be shown in the proof of Theorem 4.1 that the sequence $\{n_m\}$ is an exponentially increasing sequence. Thus, using the above algorithms, subsequent parameter updates become less frequent as time progresses. The algorithm of Bhatnagar and Borkar (1998) (cf. $(N+1)$ -Simulation FDSA-2 below) on the other hand, uses coupled iterations with two timescales and updates the whole parameter at every instant (even though it requires $N+1$ parallel simulations for the same). We present this algorithm next.

$(N+1)$ -Simulation FDSA-2

Let $\{(X(n), Y(n))\}$ be governed by $\{\theta(n)\}$ (where $\theta(n) \triangleq (\theta_1(n), \dots, \theta_N(n))^T$ is the n th update of parameter θ) which is updated according to Equations (11) below. Let us also define N additional parallel simulations as follows: For $i = 1, \dots, N$, let $\{(X^i(n), Y^i(n))\}$ be governed by $\{\theta(n) + \delta e_i\}$. In the following, the sequences $\{Z(n)\}$ and $\{Z^i(n)\}$, $i = 1, \dots, N$, perform weighted averages of the cost function values and are defined as in the last two equations in (11) below. Let $Z(0) = Z^i(0) = 0$, $i = 1, \dots, N$. Then, for $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(n+1) &= \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z(n) - Z^i(n)}{\delta} \right] \right), \\ Z(n+1) &= Z(n) + b(n)(h(Y(n)) - Z(n)), \\ Z^i(n+1) &= Z^i(n) + b(n)(h(Y^i(n)) - Z^i(n)). \end{aligned} \quad (11)$$

It is clear that one requires $N+1$ parallel simulations in this manner. Finally, we present our next SPSA algorithm (SPSA-2) which requires only two parallel simulations as in SPSA-1 but which also allows for data aggregation over a fixed number L of epochs in between two successive parameter updates for better performance.

SPSA-2

Let $\{(X^-(l), Y^-(l))\}$ and $\{(X^+(l), Y^+(l))\}$ be the two parallel simulations. These depend on parameter sequences $\{\theta(n) - \delta\Delta(n)\}$ and $\{\theta(n) + \delta\Delta(n)\}$ respectively in the manner explained below: Let $L \geq 1$ be a given fixed integer. We extract double sequences $\{(X_m^-(n), Y_m^-(n))\}$ and $\{(X_m^+(n), Y_m^+(n))\}$, $n \geq 0$, $m = 0, 1, \dots, L-1$, from the two parallel simulations in the following manner. Write l as $l = nL + m$, where $n \geq 0$ and $m \in \{0, 1, \dots, L-1\}$. Now, set $X_m^-(n) \triangleq X^-(nL + m)$ and $Y_m^-(n) \triangleq Y^-(nL + m)$. Similarly, $X_m^+(n) \triangleq X^+(nL + m)$ and $Y_m^+(n) \triangleq Y^+(nL + m)$ respectively. Now, for $m = 0, 1, \dots, L-1$, $(X_m^-(n), Y_m^-(n))$ is governed by the parameter $\theta(n) - \delta\Delta(n)$. Similarly, for $m = 0, 1, \dots, L-1$, $(X_m^+(n), Y_m^+(n))$ is governed by the parameter $\theta(n) + \delta\Delta(n)$. We also define two double sequences $\{Z_m^-(n)\}$ and $\{Z_m^+(n)\}$, $n \geq 0$, $m = 0, 1, \dots, L-1$, in recursions (12) for averaging

the cost function. Let $Z_0^-(0) = Z_1^-(0) = \dots = Z_L^-(0) = 0$ and $Z_0^+(0) = Z_1^+(0) = \dots = Z_L^+(0) = 0$. Then, for $i = 1, \dots, N$, $n \geq 0$,

$$\theta_i(n+1) = \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z_L^-(n) - Z_L^+(n)}{2\delta\Delta_{n,i}} \right] \right), \quad (12)$$

where, for $m = 0, 1, \dots, L-1$,

$$\begin{aligned} Z_{m+1}^-(n+1) &= Z_m^-(n+1) + b(n)(h(Y_m^-(n+1)) - Z_m^-(n+1)), \\ Z_{m+1}^+(n+1) &= Z_m^+(n+1) + b(n)(h(Y_m^+(n+1)) - Z_m^+(n+1)), \end{aligned}$$

with $Z_0^-(n+1) = Z_L^-(n)$ and $Z_0^+(n+1) = Z_L^+(n)$. Note again that one requires only two parallel simulations in this manner as opposed to $N+1$ earlier. We observed in the numerical experiments that for $L=1$, SPSA-2 did not exhibit good performance when the parameter dimension is high. This could be due to the fact that in this case, the system does not adapt as quickly to the new parameter update before it changes again. By selecting $L > 1$, one can effectively take care of this problem by holding the parameter fixed for L instants, thus giving the system sufficient time to adapt to the new parameter update. The choice of L is completely arbitrary though. In the numerical experiments for instance, where we consider the parameter vectors to be 10 and 40-dimensional respectively, the value of L is chosen as 100.

3.2. Comparison of algorithms

We can classify the five algorithms broadly into two categories – those that update the parameter over time instants (or their multiples) of increasing separation n_m , $m \geq 1$, and those that update the parameter at regular intervals. Let us first consider the algorithms in the first category. These comprise $(N+1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and SPSA-1. As already stated earlier, $(N+1)$ -Simulation FDSA-1 requires $N+1$ parallel simulations at every instant but updates the whole parameter vector once every n_m instants. On the other hand, Two-Simulation FDSA-1 uses only two simulations and updates the parameter in cycles of n_{Nm} , $m \geq 1$, where N is the parameter dimension. SPSA-1, however, updates the full parameter every n_m instants and still requires only two parallel simulations for doing so. Thus SPSA-1 has the combined advantages of $(N+1)$ -Simulation FDSA-1 and Two-Simulation FDSA-1. Moreover SPSA-1 tracks trajectories of the ordinary differential equation (o.d.e.) (13) as does $(N+1)$ -Simulation FDSA-1. It was shown in Bhatnagar and Borkar (1997) that Two-Simulation FDSA-1 tracks trajectories of an o.d.e. that is similar to (13) but with a factor $1/N$ multiplying its RHS. This factor essentially serves to slow down the rate of convergence of the algorithm (9).

Our next set of algorithms, namely, $(N + 1)$ -Simulation FDSA-2 and SPSA-2 update parameters after every fixed number of instants. In particular, $(N + 1)$ -Simulation FDSA-2 updates the full parameter vector every instant while requiring $N + 1$ parallel simulations for the same. SPSA-2, on the other hand, requires only two parallel simulations and updates the parameter after every fixed number ' L ' of instants. This number ($L \geq 1$) is chosen arbitrarily. SPSA-2 thus allows for data aggregation in between successive parameter update epochs (for better performance) while requiring only two parallel simulations at any instant. It will be shown in Section 4 that SPSA-2 tracks the trajectories of the o.d.e. (13) on the slower timescale. A similar result was shown in Bhatnagar and Borkar (1998) for $(N + 1)$ -Simulation FDSA-2. In Section 5, we shall consider a simple queueing network with parameter vectors of dimensions 10 and 40 respectively. We found that both SPSA-1 and SPSA-2 exhibit significantly superior performance than the algorithms in Bhatnagar and Borkar (1997, 1998) mentioned here. We shall consider in Section 5 the following step-size sequences $\{a(n)\}$ and $\{b(n)\}$: $a(0) = b(0) = 1$, $a(n) = 1/n$ and $b(n) = 1/n^{2/3}$, $n \geq 1$. For these sequences, $n_0 = 1$, $n_{50} \approx 2.5 \times 10^4$, $n_{100} \approx 4.3 \times 10^5$ etc. Thus, a possible disadvantage in using SPSA-1 is for large systems/networks that require several updates of the parameter before convergence is achieved, since in using this scheme, successive parameter updates are held fixed over intervals of increasing sizes and thus the parameter is updated less often as time progresses. This is not the case with algorithm SPSA-2 where we hold the parameter fixed only for a fixed number L of epochs before updating it. This intuition is also confirmed in Section 5 where we show numerical experiments with parameters of dimensions 10 and 40 respectively. We observed that when the parameter dimension is 10, SPSA-1 outperforms SPSA-2. However, when the same is increased to 40, it is SPSA-2 that performs better than SPSA-1.

As already observed, the algorithms SPSA-1 and SPSA-2 are computationally superior than their corresponding variants. This is however achieved by generating N i.i.d. random variables $\Delta_{n,1}, \dots, \Delta_{n,N}$, that satisfy Condition (B). In particular, one could select these to be i.i.d., Bernoulli distributed (as we do in our numerical experiments in Section 6) namely, $\Delta_{n,i} = \pm 1$, w.p. $1/2$, $i = 1, \dots, N$. It will become clear in the convergence analysis in the next section that it is these randomizations and the particular form of the gradient estimates that are primarily responsible for both of these schemes using only two parallel simulations at any instant as against $N + 1$. Generating N i.i.d., Bernoulli random variables (or in general those satisfying Condition (B)) is far more computationally simple than generating N parallel simulations; the latter requires in particular simulating N independent parallel systems. It will be shown in the next two sections that algorithms

(10) and (12) asymptotically track the stable points of the o.d.e. (13).

Finally, in Bhatnagar *et al.* (1999b), the algorithm SPSA-1 was analyzed in the context of rate based feedback flow control in Available Bit Rate (ABR) service in Asynchronous Transfer Mode (ATM) networks. However, because of the finite state setting there, questions about stability of the scheme did not arise. This is however not the case here. Our state space is unbounded and hence we require Liapunov stability assumptions on the system to ensure tightness.

4. Convergence analysis

We now present the convergence analyses of algorithms SPSA-1 and SPSA-2. In order to save space, we skip certain straightforward steps in the proofs and refer the readers to Bhatnagar *et al.* (2000) for all the details.

4.1. Convergence analysis of SPSA-1

Consider now the stochastic approximation scheme SPSA-1. Recall that $\{(X^k(j), Y^k(j))\}$, $k = 1, 2$, are the two parallel simulations respectively governed by $\{\tilde{\theta}_j^k\}$, $k = 1, 2$, with $\tilde{\theta}_j^1 = \theta(m) - \delta\Delta(m)$ and $\tilde{\theta}_j^2 = \theta(m) + \delta\Delta(m)$ respectively, for $n_m < j \leq n_{m+1}$, $m \geq 0$. Also, the dynamics of the simulations $\{(X^k(j), Y^k(j))\}$, $k = 1, 2$, are governed by equations of type (3) and (4), with $\xi(j)$, $\eta(j)$ replaced with analogously defined $\xi^k(j)$, $\eta^k(j)$, $k = 1, 2$, respectively, independent of one another.

For this algorithm, we use the assumptions (A1), (A2) and (A3) (defined in Section 2) for our system. Let the filtration be represented by $\mathcal{F}_n = \sigma(X^k(j), Y^k(j), \xi^k(j), \eta^k(j), \tilde{\theta}_j^k, \bar{\Delta}_j^k, k = 1, 2; j \leq n)$, where $\tilde{\theta}_j = \theta(m)$ and $\bar{\Delta}_j = \Delta(m)$ for $n_m < j \leq n_{m+1}$, $m \geq 0$. For $m \geq 0$, define random variables $\{\mu_m^k\}$, $k = 1, 2$, for $f \in C_b(\mathcal{R}^{q+d})$ (the space of bounded and continuous functions on \mathcal{R}^{q+d}) by

$$\int f d\mu_m^k = \frac{\sum_{i=n_m+1}^{n_{m+1}} a(i) f(X^k(i), Y^k(i))}{\sum_{i=n_m+1}^{n_{m+1}} a(i)},$$

$m \geq 0, k = 1, 2$. Theorem 4.1 below proves stability of the algorithm SPSA-1. Its proof is given in the Appendix. Recall that E is the set where random variables $\Delta_{l,i}$, $l \in \{1, \dots, N\}$, $i \geq 0$, take values in (cf. Condition (B)).

Theorem 4.1. *Almost surely, $(\mu_m^1, \mu_m^2, \theta(m), \Delta(m))$, $m \geq 0$, converges to the compact set $\{(\mu_{\theta-\delta\Delta}, \mu_{\theta+\delta\Delta}, \theta, \Delta) \mid \theta \in C, \Delta \in E\}$.*

The final step is to show convergence of the algorithm (10) to the set of local minima. The o.d.e. technique is commonly used to prove convergence of stochastic

approximation algorithms. Here, we show that the algorithm (10) asymptotically converges to the stable points of the o.d.e. (13). For any function $H: \mathcal{R}^N \rightarrow \mathcal{R}$, let $\nabla H(x) \triangleq [\nabla_1 H(x), \dots, \nabla_N H(x)]^T$ represent the gradient of H at the point $x \in \mathcal{R}^N$. Let $\tilde{Z}^T(t) \triangleq (\tilde{Z}_1(t), \dots, \tilde{Z}_N(t)) \in \mathcal{R}^N$, with $\tilde{Z}_i(t)$, $i = 1, \dots, N$, satisfying the o.d.e.

$$\dot{\tilde{Z}}_i(t) = \tilde{\pi}_i(-\nabla_i J(\tilde{Z}(t))), t \geq 0, \tilde{Z}(0) \in C, \quad (13)$$

where for any bounded, continuous, real-valued function $v(\cdot)$,

$$\tilde{\pi}_i(v(y)) = \lim_{0 < \eta \rightarrow 0} \left(\frac{\pi_i(y + \eta v(y)) - \pi_i(y)}{\eta} \right).$$

For $x = (x_1, \dots, x_N)^T$, let $\tilde{\pi}(x) = (\tilde{\pi}_1(x_1), \dots, \tilde{\pi}_N(x_N))^T$. The operator $\tilde{\pi}(\cdot)$ forces the o.d.e. (13) to evolve within the constraint set C . Let $K \triangleq \{\theta \in C | \tilde{\pi}(\nabla J(\theta)) = 0\}$. We recall here a key result from Hirsch (1987) stated as Lemma 4.1 below. Consider an o.d.e. in \mathcal{R}^N

$$\dot{x}(t) = F(x(t)), \quad (14)$$

with the set $\bar{G} \triangleq \{x | F(x) = 0\}$ of stable points associated with it. Let \bar{G}^ϵ denote the ϵ -neighborhood of \bar{G} namely, $\bar{G}^\epsilon \triangleq \{x | \exists x' \in \bar{G} \text{ s.t. } \|x - x'\| \leq \epsilon\}$. For $T > 0$, $\gamma > 0$, say that $y(\cdot)$ is a (T, γ) -perturbation of (14) if there exist real numbers $0 = T_0 < T_1 < T_2 < \dots$, such that $T_{i+1} - T_i \geq T$, $\forall i$, and on each interval $[T_i, T_{i+1}]$, there exists a solution $x^i(\cdot)$ of (14) such that

$$\sup_{t \in [T_i, T_{i+1}]} |x^i(t) - y(t)| < \gamma.$$

The result from Hirsch (1987, p. 339) is as follows.

Lemma 4.1. For given $\epsilon > 0$, $T > 0$, there exists a $\bar{\gamma}$ such that for all $\gamma \in [0, \bar{\gamma}]$, any (T, γ) -perturbation of (14) converges to \bar{G}^ϵ .

For fixed $\eta > 0$, let $K^\eta \triangleq \{\theta \in C | \exists \theta' \in K \text{ s.t. } \|\theta - \theta'\| \leq \eta\}$ represent the set of points within a distance η of the set K . As a direct consequence of Lemma 4.1, for any given $\eta, T > 0$, $\exists \bar{\gamma} > 0$ s.t. $\forall \gamma \in [0, \bar{\gamma}]$, any (T, γ) -perturbation of (13) shall converge to K^η . Finally, Theorem 4.2 shows that given $\eta > 0$, there exists a $\bar{\delta} > 0$ such that the algorithm SPSA-1 for all $\delta \leq \bar{\delta}$, converges to K^η a.s. The proof of this theorem proceeds through several steps and is given in detail in Bhatnagar *et al.* (1999b).

Theorem 4.2. Given $\eta > 0$, $\exists \bar{\delta} > 0$ such that for any $\delta \in (0, \bar{\delta}]$, the algorithm (10) converges to K^η almost surely (a.s.).

Remark. Note that K is the set of all critical points of (13), and not just the set of local minima. However, points in K that are not local minima will be unstable equilibria and since our algorithm is of the gradient descent type, it will converge a.s. to the η -neighborhood of $K_0 \triangleq$ the set of local minima of $J(\cdot) \subset K$.

4.2. Convergence Analysis of SPSA-2

Consider the stochastic approximation scheme SPSA-2. We now assume (A1), (A2') and (A3) to prove convergence of SPSA-2. Recall that $\{(X^-(l), Y^-(l))\}$ and $\{(X^+(l), Y^+(l))\}$ are the two parallel simulations respectively governed by $\{\tilde{\theta}(l) - \delta \tilde{\Delta}(l)\}$ and $\{\tilde{\theta}(l) + \delta \tilde{\Delta}(l)\}$, where $\tilde{\theta}(l) = \theta([l/L])$ and $\tilde{\Delta}(l) = \Delta([l/L])$, and where $[l/L]$ represents integral part of l/L . In other words, if l has the form $l = nL + m$, where $m \in \{0, 1, \dots, L-1\}$ and n is an integer, then $[l/L] = n$. Note that by definition, $\{\tilde{\theta}(l)\}$ (resp. $\{\tilde{\Delta}(l)\}$) takes values in the compact set C (resp. E). Corresponding to $\{(X^-(l), Y^-(l))\}$, define the process $\{\mu_n^-\}$ by

$$\mu_n^-(A_1 \times A_2 \times B \times D) = \frac{1}{n} \sum_{m=0}^{n-1} I\{X^-(m) \in A_1, Y^-(m) \in A_2, \tilde{\theta}(m) \in B, \tilde{\Delta}(m) \in D\},$$

for Borel sets $A_1 \subset \mathcal{R}^q$, $A_2 \subset \mathcal{R}^d$, $B \subset C$ and $D \subset E$. Similarly, one can define $\{\mu_n^+\}$ as well. The following theorem establishes tightness of sequences $\{\mu_n^-\}$ and $\{\mu_n^+\}$. The proof follows in a somewhat similar manner as the proof of Theorem 4.1 and is not being presented in order to avoid repetition.

Theorem 4.3. Almost surely, $\{\mu_n^-\}$ and $\{\mu_n^+\}$ are tight sequences.

We now proceed with the rest of the analysis. Let for $k \geq 1$, $\mathcal{F}'_k \triangleq \sigma(\theta(0), \theta(1), \dots, \theta(k), \Delta(0), \Delta(1), \dots, \Delta(k-1))$. Then $\Delta(k)$ is independent of \mathcal{F}'_k , $\forall k \geq 1$. Define sequences $\{N_i^-(p), p \geq 1\}$, $i = 1, \dots, N$, as follows:

$$N_i^-(p) = \sum_{j=0}^p a(j) \left(\frac{J(\theta(j)) - \delta \Delta(j)}{\Delta_{j,i}} - E \left[\frac{J(\theta(j)) - \delta \Delta(j)}{\Delta_{j,i}} \middle| \mathcal{F}'_j \right] \right).$$

Then, we have

Lemma 4.2. For every $i = 1, \dots, N$, $\{N_i^-(p)\}$ converges a.s.

Proof. Follows in a similar manner as Lemma A.2 of Bhatnagar *et al.* (1999b). ■

A similar result holds for analogously defined sequences $\{N_i^+(p)\}$. Now define $\{s(n)\}$ as follows: $s(0) = 0$, $s(n) = \sum_{i=0}^{n-1} a(i)$, $n \geq 0$. Let $\Delta_{t,i} = \Delta_{n,i}$, for $t \in [s(n), s(n+1)]$, $n \geq 1$. Further, let $\Delta(t) = (\Delta_{t,1}, \dots, \Delta_{t,N})^T$. Recall that for any bounded, continuous, real-valued function $v(\cdot)$,

$$\tilde{\pi}_i(v(y)) = \lim_{0 < \eta \rightarrow 0} \left(\frac{\pi_i(y + \eta v(y)) - \pi_i(y)}{\eta} \right).$$

Also, for $x = (x_1, \dots, x_N)^T \in \mathcal{R}^N$, $\bar{\pi}(x) = (\bar{\pi}_1(x), \dots, \bar{\pi}_N(x))^T$. Consider the following o.d.e.: For $i = 1, \dots, N$,

$$\dot{\theta}_i(t) = \bar{\pi}_i \left(E \left[\frac{J(\theta(t) - \delta\Delta(t)) - J(\theta(t) + \delta\Delta(t))}{2\delta\Delta_{t,i}} \right] \right), \quad (15)$$

where the operator $E[\cdot]$ in (15) represents the expectation with respect to the common distribution of $\{\Delta_{t,i}\}$. Now, for $i = 0, 1, 2, \dots$, let $c(i) = b(i)L$. It is easy to see from (7) that

$$\sum_i c(i) = \infty, \sum_i c(i)^2 < \infty, a(i) = o(c(i)).$$

Here, we consider $\{a(i)\}$ and $\{c(i)\}$ to be the two step-size sequences (as opposed to $\{a(i)\}$ and $\{b(i)\}$ in SPSA-1). Define $\{t(n)\}$ as follows: $t(0) = 0$, $t(n) = \sum_{i=0}^{n-1} c(i)$, $n \geq 1$. Let $z^-(\cdot), z^+(\cdot) : [0, \infty) \rightarrow \mathcal{R}$ and $\bar{\theta}(\cdot) : [0, \infty) \rightarrow C$ denote the continuous functions obtained by setting $z^-(t(n)) = Z_L^-(n)$, $z^+(t(n)) = Z_L^+(n)$, $\bar{\theta}(t(n)) = \theta(n)$ respectively $\forall n$, with linear interpolation on $[t(n), t(n+1)]$, $n \geq 0$. Consider the system of o.d.e.'s

$$\begin{aligned} \dot{\theta}(t) &= 0, \\ \dot{z}^-(t) &= J(\theta(t) - \delta\Delta(t)) - z^-(t), \\ \dot{z}^+(t) &= J(\theta(t) + \delta\Delta(t)) - z^+(t). \end{aligned} \quad (16)$$

We now have the following theorem whose proof appears in the Appendix.

Theorem 4.4. For any $T, \delta > 0$, $(z^-(t(n) + \cdot), z^+(t(n) + \cdot), \theta(t(n) + \cdot))$ is a bounded (T, δ) -perturbation of (16) for n sufficiently large.

Define $\hat{z}^-(\cdot), \hat{z}^+(\cdot) : [0, \infty) \rightarrow \mathcal{R}$ and $\hat{\theta}(\cdot) : [0, \infty) \rightarrow C$ by $\hat{z}^-(s(n)) = Z_L^-(n)$, $\hat{z}^+(s(n)) = Z_L^+(n)$, $\hat{\theta}(s(n)) = \theta(n)$ respectively $\forall n$, with linear interpolation on intervals $[s(n), s(n+1)]$, $n \geq 0$.

Lemma 4.3. For any $T, \delta > 0$, $\hat{\theta}(s(n) + \cdot)$ is a bounded (T, δ) -perturbation of (15) for sufficiently large n .

Proof. Rewrite the first equation in (12) as follows: for $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(m+1) &= \pi_i \left(\theta_i(m) + a(m) E \left[\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \right]_{\mathcal{F}'_m} \right) + \eta(m). \end{aligned} \quad (17)$$

In the above, $\eta(m)$ is an error term that becomes asymptotically negligible (Bhatnagar *et al.*, 2000) by Lemma 4.1, Lemma 4.2 and Theorem 4.4. The algorithm (12) can then be viewed as a discretization of the o.d.e. (15). Now a standard argument as in Kushner and Clark (1978, pp.191–194) proves the claim. ■

Lemma 4.4. For any $\theta(m) \in C$, for all $i = 1, \dots, N$,

$$\lim_{\delta \downarrow 0} \left| E \left[\frac{J(\theta(m) - \delta\Delta(m)) - J(\theta(m) + \delta\Delta(m))}{2\delta\Delta_{m,i}} \right]_{\mathcal{F}'_m} - \nabla_i J(\theta(m)) \right| = 0.$$

Proof. Follows in a similar manner as Lemma A.5 of Bhatnagar *et al.* (1999b). ■

Recall that the set $K \triangleq \{\theta \in C \mid \bar{\pi}(\nabla J(\theta)) = 0\}$ is the asymptotically stable attractor set for the o.d.e. (13) with $J(\cdot)$ itself serving as the strict Liapunov function. Further, $K^\eta \triangleq \{\theta \in C \mid \|\theta - \theta_0\| < \eta, \theta_0 \in K\}$ represents the set of points in C that are within an η -distance from the set K . We now have

Lemma 4.5. Given $\eta > 0$, there exists $\delta_0 > 0$ such that for all $\delta \in (0, \delta_0]$, K^η is an asymptotically stable attractor set for the o.d.e. (15).

Proof. As already mentioned, $J(\cdot)$ itself serves as a strict Liapunov function for (13) outside the set K . Now by Lemmas 4.3 and 4.4, for sufficiently small δ , $J(\cdot)$ will also serve as a strict Liapunov function for (4.3) outside the set K^η . ■

Finally, we come to the main result of this section.

Theorem 4.5. Given $\eta > 0$, there exists $\delta_0 > 0$ such that for all $\delta \in (0, \delta_0]$, $\theta(n) \rightarrow K^\eta$ a.s.

Proof. Follows from Lemmas 4.1, 4.3 and 4.5. ■

This completes the convergence analysis of both the algorithms SPSA-1 and SPSA-2.

5. Numerical results

In this section, we demonstrate our algorithms SPSA-1 and SPSA-2 by means of a simple queueing system and numerically compare their performance with the algorithms in Bhatnagar and Borkar (1997, 1998) described in Section 3. We consider the two-node queueing network shown in Fig. 1.

There are two external arrival streams (one each) to the two nodes. Arrivals to the nodes from these streams follow independent Poisson processes with rates λ_1 and λ_2 . The service times are exponentially distributed with rates $\mu_1(\theta_1)$ and $\mu_2(\theta_2)$, respectively, where θ_1 and θ_2 are parameter vectors at the two nodes. The exact dependence of μ_1 and μ_2 on θ_1 and θ_2 , respectively is given below. A customer after service at Node 1 joins the queue at Node 2. After service at Node 2, a customer either departs with probability p or is fed back to Node 1

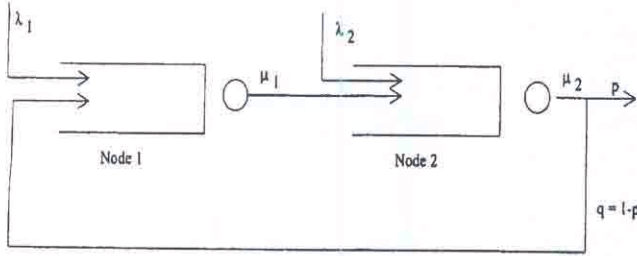


Fig. 1. Queuing network.

with the remainder probability of $1 - p$. Our aim is to find the optimum (joint) parameter vector (θ_1, θ_2) within the constraint set C which minimizes the sum of the stationary mean waiting times in the two queues. The constraint set C is defined as follows: Given $M > 0$, each component $\theta_i^j, i = 1, 2, j = 1, \dots, M$, takes values in the interval $[\theta_{i,\min}^j, \theta_{i,\max}^j]$ and so the set C is defined as

$$C = [\theta_{1,\min}^1, \theta_{1,\max}^1] \times \dots \times [\theta_{1,\min}^M, \theta_{1,\max}^M] \\ \times [\theta_{2,\min}^1, \theta_{2,\max}^1] \times \dots \times [\theta_{2,\min}^M, \theta_{2,\max}^M].$$

We assume that both θ_1 and θ_2 are vectors of the same dimension M . Note that $M = N/2$. Thus, $\theta_i \triangleq (\theta_i^1, \dots, \theta_i^M)^T, i = 1, 2$, and the whole parameter vector is represented as $\theta \triangleq (\theta_1^1, \dots, \theta_1^M, \theta_2^1, \dots, \theta_2^M)^T$. Let $\bar{\theta} \triangleq (\bar{\theta}_1^1, \dots, \bar{\theta}_1^M, \bar{\theta}_2^1, \dots, \bar{\theta}_2^M)^T$ represent the target/optimum parameter. The dependence of the service times on the parameters has the following form:

$$\mu_i(\theta_i) = \frac{\bar{\mu}_i}{\left(1 + \prod_{j=1}^M |\theta_i^j - \bar{\theta}_i^j|\right)}, \quad i = 1, 2,$$

where $\bar{\mu}_i, i = 1, 2$, is assumed to be constant. Note that the cost, which is the sum of the stationary mean waiting times in the two queues, will be minimized if the service rates are maximized, and which clearly occurs at $\theta = \bar{\theta}$. Thus we know that the optimum for our problem lies at $\bar{\theta}$. Let $\theta_i^j(0), i = 1, 2, j = 1, \dots, M$, represent the initial (starting) values of the parameter components.

For the simulation experiments, the following step-size sequences are chosen for all the five schemes: $a(0) = b(0) = 1, a(n) = 1/n, b(n) = 1/n^{2/3}, n \geq 1$. Also, we choose $L = 100$ in SPSA-2. For both SPSA-1 and SPSA-2, we choose random variables $\Delta_{n,i}, i = 1, \dots, N, n \geq 0$, to be i.i.d., Bernoulli distributed with $\Delta_{n,i} = \pm 1$ w.p. $1/2, i = 1, \dots, N, n \geq 0$. We consider the following set up for all the three algorithms.

$\lambda_1 = 0.2, \lambda_2 = 0.1, \bar{\mu}_1 = 87, \bar{\mu}_2 = 92, p = 0.4$. For $i = 1, 2, j = 1, \dots, M, \theta_{i,\min}^j = 0.1, \theta_{i,\max}^j = 0.6$, and $\bar{\theta}_i^j = 0.3$. Also, for $j = 1, \dots, M, \theta_1^j(0) = 0.2$ and $\theta_2^j(0) = 0.4$.

We consider two values of M for our experiments: $M = 5$ and $M = 20$. Thus the parameter vectors we con-

sider in the simulations have dimensions 10 and 40 respectively. We consider a total of 3×10^5 data aggregation epochs for all the five schemes. The corresponding total number of parameter updates for each algorithm (for $M = 5$ and 20 respectively) is shown in Table 1.

Note that whereas there are 3×10^5 parameter updates in $(N + 1)$ -Simulation FDSA-2, the corresponding number in SPSA-2 is 3×10^3 (since $L = 100$). Moreover, the corresponding number in SPSA-1 and $(N + 1)$ -Simulation FDSA-1 is only 92. This is so because for the step-size sequences considered in the experiments, the values n_{92} and n_{93} in $\{n_m, m \geq 1\}$ (defined after (7)) are: $n_{92} \approx 2.94 \times 10^5$ and $n_{93} \approx 3.09 \times 10^5$ respectively. Finally, in Two-Simulation FDSA-1, the number of parameter updates for $M = 5$ (mentioned in Table 1) is

$$9 \frac{2}{10}.$$

It is written in this manner to indicate that in addition to the nine times that the whole parameter is updated in 3×10^5 data aggregation epochs, the first two components of the parameter vector are also updated for a 10th time. Similarly, for $M = 20$, the number of parameter updates in Two-Simulation FDSA-1 is

$$2 \frac{12}{40}.$$

In what follows we compare the performance of our SPSA algorithms (SPSA-1 and SPSA-2) with the algorithms of Bhatnagar and Borkar (1997, 1998) described in Section 3, in terms of speed of convergence. We choose the Euclidean distance between the current parameter update and the target parameter value as the performance metric and plot that w.r.t. the number of data aggregation epochs for all the five schemes. The Euclidean distance $d(\theta, \bar{\theta})$ is defined by

$$d(\theta, \bar{\theta}) = \left(\sum_{j=1}^2 \sum_{i=1}^M (\theta_i^j - \bar{\theta}_i^j)^2 \right)^{1/2}.$$

We performed five independent replications of each experiment using different seeds. In Figs. 2 and 3, the mean

Table 1. Number of parameter updates in 3×10^5 epochs

Algorithm	Number of parameter updates for $M = 5$	Number of parameter updates for $M = 20$
$(N + 1)$ -FDSA-2	3×10^5	3×10^5
SPSA-2	3×10^3	3×10^3
SPSA-1	92	92
$(N + 1)$ -FDSA-1	92	92
Two-FDSA-1	$9 \frac{2}{10}$	$2 \frac{12}{40}$

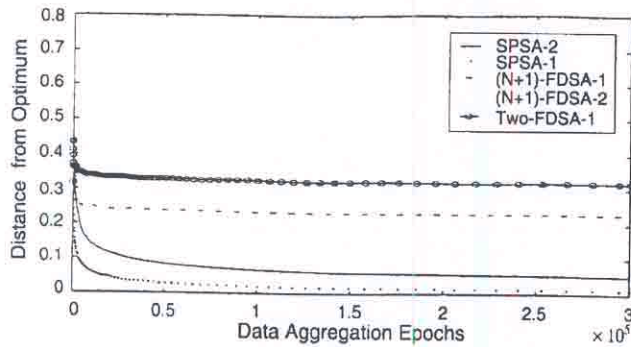


Fig. 2. 10-parameter vector.

trajectories from these experiments are plotted for all five schemes, for both the 10-dimensional and the 40-dimensional parameter cases respectively. The standard error from these replications for the five schemes was computed at the end of these simulations (after 3×10^5 data aggregation epochs), and is indicated in Table 2.

When we choose $L = 1$, SPSA-2 does not exhibit good performance for our model with high dimensional parameters. As already mentioned in Section 3, this is probably because of the fact that since SPSA-2 uses only two parallel simulations, the system is unable to adapt to the new parameter update before it changes again. Data aggregation over L epochs (for $L > 1$) on the other hand leads to additional averaging and hence improved performance. However, the choice of L is completely arbitrary. For our experiments, we select $L = 100$.

As expected, algorithms SPSA-1 and SPSA-2 show significantly better performance than the rest of the algorithms. We observe that using SPSA-1, for $M = 5$ (Fig. 2), the Euclidean distance between the current update and the optimum parameter becomes less than 0.10 (on an average of five replications) from the 28th parameter update onwards (after only 3183 data aggregation epochs). Using SPSA-2, the same is achieved from its 296th parameter update (after 2.96×10^4 data

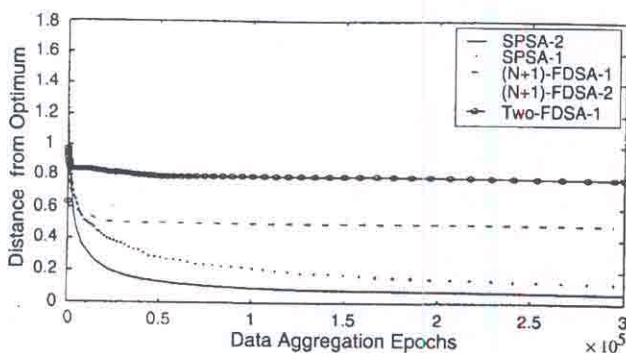


Fig. 3. 40-parameter vector.

Table 2. Distance after 3×10^5 data aggregation epochs

Algorithm	Mean \pm standard error (10-parameter vector)	Mean \pm standard error (40-parameter vector)
SPSA-1	0.011 \pm 0.001	0.121 \pm 0.03
SPSA-2	0.051 \pm 0.02	0.063 \pm 0.02
($N + 1$)-FDSA-1	0.227 \pm 0.03	0.492 \pm 0.06
($N + 1$)-FDSA-2	0.315 \pm 0.003	0.507 \pm 0.05
Two-FDSA-1	0.317 \pm 0.04	0.782 \pm 0.05

aggregation epochs). However, using SPSA-1 (after running the algorithm long enough), it is observed that for $M = 20$ (Fig. 3), the same distance becomes less than 0.10 after its 102nd parameter update (after nearly 4.75×10^5 data aggregation epochs). The same is achieved in SPSA-2 from its 829th update onwards (or after only 8.29×10^4 data aggregation points). It is clear from the above that SPSA-1 shows the best performance (better than even SPSA-2) when the parameter dimension is low. However, when the parameter dimension is significantly increased, it is SPSA-2 and not SPSA-1 that shows the best performance. This is probably due to the fact that SPSA-1 (and its variants) requires that the parameter be held fixed over intervals of increasing size. Also, higher dimensional parameters typically require several updates before convergence is achieved.

On a Sun Ultra10 Unix workstation, for $M = 5$, it took about 2 minutes using SPSA-1 for the Euclidean distance from optimum to become less than 0.05. SPSA-2 required about 3–4 minutes for the same. On the other hand, for $M = 20$, SPSA-1 took about 15 minutes, while SPSA-2 required only about 5 minutes for the same to happen. ($N + 1$)-Simulation FDSA-1 and ($N + 1$)-Simulation FDSA-2 (along with Two-Simulation FDSA-1) took orders of magnitude more time than SPSA-1 and SPSA-2. For $M = 20$, after almost 6.5×10^7 data aggregation epochs and running for nearly 21 hours, the separation from optimum (for one replication) of ($N + 1$)-Simulation FDSA-1 was about 0.44, while that of ($N + 1$)-Simulation FDSA-2 was about 0.42. As expected, Two-Simulation FDSA-1 showed the worst performance. For $M = 20$, after

$$8 \frac{20}{40},$$

parameter updates (or nearly 5×10^8 data aggregation epochs) and running for almost 18 hours, the Euclidean distance from optimum using Two-Simulation FDSA-1 was still about 0.70. Thus, our simulation experiments confirm that both algorithms SPSA-1 and SPSA-2 presented here perform orders of magnitude faster than the algorithms ($N + 1$)-Simulation FDSA-1, Two-Simulation FDSA-1 and ($N + 1$)-Simulation FDSA-2 of Bhatnagar and Borkar (1997, 1998).

6. Conclusions

We developed two Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms (SPSA-1 and SPSA-2) for simulation optimization of hidden Markov models. Both of these algorithms use only two parallel simulations (each) and are generalized variants of the two-timescale stochastic approximation algorithms of Bhatnagar and Borkar (1997, 1998) ($(N + 1)$ -Simulation FDSA-1, Two-Simulation FDSA-1 and $(N + 1)$ -Simulation FDSA-2) respectively. Whereas SPSA-1 updates the parameter over time instants of increasing separation as in Bhatnagar and Borkar (1997), SPSA-2 updates once after every fixed number of instants. The latter is a generalization of the algorithm in Bhatnagar and Borkar (1998) that updates the parameter at every instant; and it shows improved performance. The convergence analysis for both the algorithms was presented. We conducted numerical experiments with parameters of different dimensions on a two node queueing network model with feedback using both the SPSA algorithms, the Two-Simulation algorithm of Bhatnagar and Borkar (1997) and its $(N + 1)$ -Simulation analogue and the $(N + 1)$ -Simulation algorithm of Bhatnagar and Borkar (1998). We found that the SPSA algorithms converge orders of magnitude faster than the rest.

Acknowledgements

This research was supported in part by the NSF under Grant DMI-9713720, by the Semiconductor Research Corporation under Grant 97-FJ-491 and by DoD contract Number MDA90497C3015.

References

- Andradóttir, S. (1998) Simulation optimization, in *Handbook of Simulation*, Banks, J. (ed), Wiley, New York, NY, ch. 9.
- Bartusek, J. and Makowski, A. (1994) On stochastic approximation driven by sample averages: convergence results via the ODE method. Technical Report, Institute for Systems Research, University of Maryland, <http://www.isr.umd.edu/TechReports/ISR/1994/TR.94-4/>.
- Bhatnagar, S. and Borkar, V.S. (1997) Multiscale stochastic approximation for parametric optimization of hidden Markov models. *Probability in the Engineering and Informational Sciences*, **11**, 509–522.
- Bhatnagar, S. and Borkar, V.S. (1998) A two time scale stochastic approximation scheme for simulation based parametric optimization. *Probability in the Engineering and Informational Sciences*, **12**, 519–531.
- Bhatnagar, S., Fu, M.C. and Marcus, S.I. (1999a) Two timescale SPSA algorithms for rate-based ABR flow control, in *System Theory: Modeling, Analysis and Control*, Djafaris, T. and Schick, I. (eds), Kluwer Academic, Boston, MA, ch. 27.
- Bhatnagar, S., Fu, M.C. and Marcus, S.I. (1999b) Optimal multilevel feedback policies for ABR flow control using two timescale SPSA. Technical Report, Institute for Systems Research, University of Maryland, <http://www.isr.umd.edu/TechReports/ISR/1999/TR.99-18/>.
- Bhatnagar, S., Fu, M.C., Marcus, S.I. and Bhatnagar, S. (2000) Randomized difference two-timescale simultaneous perturbation stochastic approximation algorithms for simulation optimization of hidden Markov models. Technical Report, Institute for Systems Research, University of Maryland, <http://www.isr.umd.edu/TechReports/ISR/2000/TR.2000-13/>.
- Borkar, V.S. (1993) On white noise representations in stochastic realization theory. *SIAM Journal of Control and Optimization*, **31**, 1093–1102.
- Borkar, V.S. (1997) Stochastic approximation with two time scales. *Systems and Control Letters*, **29**, 291–294.
- Chong, E.K.P. and Ramadge, P.J. (1993) Optimization of queues using an infinitesimal perturbation analysis-based stochastic algorithm with general update times. *SIAM Journal of Control and Optimization*, **31**(3), 698–732.
- Chong, E.K.P. and Ramadge, P.J. (1994) Stochastic optimization of regenerative systems using infinitesimal perturbation analysis. *IEEE Transactions on Automatic Control*, **39**(7), 1400–1410.
- Elliott, R.J., Aggoun, L. and Moore, J.B. (1995) *Hidden Markov Models: Estimation and Control*, Springer-Verlag, New York.
- Fu, M.C. (1990) Convergence of a stochastic approximation algorithm for the $GI/G/1$ queue using infinitesimal perturbation analysis. *Journal of Optimization Theory and Applications*, **65**, 149–160.
- Fu, M.C. (1994) Optimization via simulation: a review. *Annals of Operation Research*, **53**, 199–248.
- Fu, M.C. and Hill, S.D. (1997) Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions*, **29**(3), 233–243.
- Hirsch, M.W. (1987) Convergent activation dynamics in continuous time networks. *Neural Networks*, **2**, 331–349.
- Ho, Y.-C. and Cao, X.-R. (1991) *Perturbation Analysis of Discrete Event Dynamical Systems*, Kluwer, Boston, MA.
- Kushner, H.J. and Clark, D.S. (1978) *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer Verlag, New York.
- Kushner, H.J. and Vazquez-Abad, F.J. (1996) Stochastic approximation methods for systems over an infinite horizon, *SIAM Journal of Control and Optimization*, **34**(2), 712–756.
- Kushner, H.J. and Yin, G.G. (1997) *Stochastic Approximation Algorithms and Applications*, Springer Verlag, New York.
- L'Ecuyer, P., Giroux, N. and Glynn, P.W. (1994) Stochastic optimization by simulation: numerical experiments with the $M/M/1$ queue in steady-state. *Management Science*, **40**(10), 1245–1261.
- L'Ecuyer, P. and Glynn, P.W. (1994) Stochastic optimization by simulation: convergence proofs for the $GI/G/1$ queue in steady state. *Management Science*, **40**(11), 1562–1578.
- Neveu, J. (1975) *Discrete Parameter Martingales*, North Holland, Amsterdam.
- Pflug, G.C. (1996) *Optimization of Stochastic Models: The Interface Between Simulation and Optimization*, Kluwer Academic, Boston, MA.
- Spall, J.C. (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, **37**(3), 332–341.
- Vazquez-Abad, F.J. and Kushner, H.J. (1992) Estimation of the derivative of a stationary measure with respect to a control parameter, *Journal of Applied Probability*, **29**, 343–352.

Appendix

Proof of Theorem 4.1. The proof proceeds through several steps. Let $\tilde{\mathcal{R}}^{q+d} = \mathcal{R}^{q+d} \cup \{\infty\}$ denote the one point

compactification of \mathcal{R}^{q+d} . For $f \in C_b(\mathcal{R}^{q+d})$, define sequences $\{Z_k(m), m \geq 1\}$, $k = 1, 2$, by

$$Z_k(m) = \sum_{j=0}^{m-1} b(j)^{-1} \left[\sum_{i=n_j+1}^{n_{j+1}} a(i)(f(X^k(i), Y^k(i)) - E[f(X^k(i), Y^k(i)) | \mathcal{F}_{i-1}]) \right]. \tag{A1}$$

Then $\{Z_k(m), \mathcal{F}_{n_m}\}$, $k = 1, 2$, are zero mean, square integrable martingale sequences (see Neveu (1975) for Martingales). Let us represent their quadratic variation processes by

$$\langle Z_k \rangle(m) = \sum_{j=0}^{m-1} E \left[(Z_k(j+1) - Z_k(j))^2 | \mathcal{F}_{n_j} \right] + E \left[Z_k(0)^2 \right], \tag{A2}$$

with $\langle Z_k \rangle(\infty) = \lim_{m \rightarrow \infty} \langle Z_k \rangle(m)$. Since f is a bounded and continuous function, it can be easily shown (cf. Theorem 4.1 of Bhatnagar *et al.* (2000)) that there exist constants $K_1, K'_1 > 0$ such that

$$\begin{aligned} \langle Z_k \rangle(m) &\leq K_1 \sum_{j=0}^{m-1} \frac{1}{b(j)^2} \sum_{i=n_j+1}^{n_{j+1}} a(i)^2 \\ &\leq K'_1 \sum_{j=0}^{m-1} \frac{1}{b(j)^2} \left(\frac{1}{n_j} - \frac{1}{n_{j+1}} \right), \end{aligned}$$

the latter inequality follows from the fact that $\sum_{i=0}^n a(i)^2 \approx 1 - 1/n$. Also note that $\sum_{i=0}^n a(i) \approx \ln(n)$, and $\sum_{i=0}^n b(i) \approx n^{1-\alpha}$. Now, from the definition of $\{n_m, m \geq 0\}$, $\sum_{i=0}^{n_{m+1}} a(i) \approx \sum_{i=0}^m b(i)$. Thus, $\ln(n_{m+1}) \approx m^{1-\alpha}$. Hence, $n_{m+1} \approx \exp(am^{1-\alpha})$ for some $a > 0$. Thus, $\langle Z_k \rangle(\infty) < \infty$. Hence, by Proposition V11.2.3(c) of Neveu (1975), $\{Z_k(m), m \geq 1\}$, $k = 1, 2$, are a.s. convergent martingale sequences. Now let us consider $\{Z_1(m)\}$. From the fact that $\sum_{j=n_m+1}^{n_{m+1}} a(j)/b(m) \rightarrow 1$, as $m \rightarrow \infty$ and from (6),

$$\int \left(f(x, y) - \int f(x', y') p_{\theta(m) - \delta\Delta}(x, y; dx', dy') \right) \times \mu_m^1(dx, dy) \rightarrow 0 \text{ a.s.} \tag{A3}$$

Now, since the above holds for all $f \in C_b(\mathcal{R}^{q+d})$, it follows that outside a set of measure zero, any limit point of $(\mu_m^1, \theta(m) - \delta\Delta(m))$, $m \geq 0$, must be of the form $(b\delta_\infty + (1-b)\mu, \theta - \delta\Delta)$, where δ_∞ denotes the Dirac measure at $\{\infty\}$ and is defined as follows:

$$\delta_\infty(x) = \begin{cases} 1 & \text{if } x = \{\infty\}, \\ 0 & \text{otherwise.} \end{cases} \tag{A4}$$

In the above, $b \in [0, 1]$, where when $b < 1$, μ must satisfy

$$\begin{aligned} \int f(x, y) \mu(dx, dy) &= \int \int f(x', y') p_{\theta - \delta\Delta}(x, y; dx', dy') \mu(dx, dy). \end{aligned}$$

Thus for $b < 1$, μ must be of the form $\mu_{\theta - \delta\Delta}$. For $b = 1$, μ is arbitrary and hence can be set to be $\mu_{\theta - \delta\Delta}$ itself. Now, note that if in the definition of the sequences $\{Z^k(m)\}$, $k = 1, 2$, the function $f(\cdot, \cdot)$ is replaced by the Liapunov function $V(\cdot, \cdot)$, the sequences $\{Z^k(m)\}$ continue to be Martingale sequences. Also, from Assumption (A2), the quadratic variation process of such a Martingale would converge as well. We then obtain (A3) with function V replacing f . Now define

$$\phi_{\theta - \delta\Delta}(x, y) \triangleq \int V(x', y') p_{\theta - \delta\Delta}(x, y; dx', dy') - V(x, y),$$

with $\phi_{\theta - \delta\Delta}(\{\infty\}) = -\epsilon_0, \forall \theta \in C$. Here ϵ_0 is the same as in Assumption (A3). Now, as a consequence of Assumption (A3), the map $(\theta, x, y) \rightarrow \phi_\theta(x, y) : C \times \tilde{\mathcal{R}}^{q+d} \rightarrow \mathcal{R}$ is upper semicontinuous and bounded from above. Further, if $(\mu_m^1, \theta(m) - \delta\Delta(m)) \rightarrow (b\delta_\infty + (1-b)\mu_{\theta - \delta\Delta}, \theta - \delta\Delta)$ along a subsequence, then from (A3) with f replaced by V , we have,

$$\begin{aligned} 0 &= \limsup_{m \rightarrow \infty} \int \phi_{\theta(m) - \delta\Delta(m)} d\mu_m^1 \\ &\leq (1-b) \int \phi_{\theta - \delta\Delta} d\mu_{\theta - \delta\Delta} - b\epsilon_0, \end{aligned} \tag{A5}$$

along the same subsequence. Now, from the definitions of $\phi_{\theta - \delta\Delta}$ and $\mu_{\theta - \delta\Delta}$ above,

$$\int \phi_{\theta - \delta\Delta} d\mu_{\theta - \delta\Delta} = 0.$$

Thus from (A5), we have $0 \leq -b\epsilon_0$, which cannot happen unless $b = 0$. A similar argument holds for the sequence $\{\mu_m^2\}$. Thus $\{\mu_m^k\}$, $k = 1, 2$, are tight sequences and have limit points of the form $\mu_{\theta - \delta\Delta}^1$ or $\mu_{\theta + \delta\Delta}^2$. Now by Lemma 2.1, the maps $\theta \rightarrow \mu_{\theta - \delta\Delta}^1$ and $\theta \rightarrow \mu_{\theta + \delta\Delta}^2$, $k = 1, 2$, are continuous. The claim now follows from the fact that any continuous image of a compact set is compact. ■

Proof of Theorem 4.4. Note that the algorithm SPSA-2 (cf. (12)) can be rewritten as follows: For $i = 1, \dots, N$,

$$\begin{aligned} \theta_i(n+1) &= \pi_i \left(\theta_i(n) + a(n) \left[\frac{Z_L^-(n) - Z_L^+(n)}{2\delta\Delta_{n,i}} \right] \right), \\ Z_L^-(n+1) &= Z_L^-(n) + c(n) \frac{1}{L} \sum_{m=0}^{L-1} (h(Y_m^-(n+1)) - Z_m^-(n+1)), \\ Z_L^+(n+1) &= Z_L^+(n) + c(n) \frac{1}{L} \sum_{m=0}^{L-1} (h(Y_m^+(n+1)) - Z_m^+(n+1)). \end{aligned} \tag{A6}$$

From $\{Z_m^-(n)\}$ and $\{Z_m^+(n)\}$, $n \geq 0$, $m \in \{0, 1, \dots, L-1\}$ defined in (12), we obtain sequences $\{Z^-(l)\}$ and $\{Z^+(l)\}$ in the obvious manner by writing l as $l = nL + m$ for appropriate n and m . Now from the second and third equations in (12), note that $Z^-(l+1)$ (resp. $Z^+(l+1)$) is the convex combination of $h(\cdot)$ and $Z^-(l)$ (resp. $Z^+(l)$). Thus $\{Z^-(l)\}$ (resp. $\{Z^+(l)\}$) are uniformly bounded sequences with upper bound depending on h and $\|Z^-(0)\|$ (resp. $\|Z^+(0)\|$). Now $\{Z_L^-(n)\}$ (resp. $\{Z_L^+(n)\}$) is just a subsequence of $\{Z^-(l)\}$ (resp. $\{Z^+(l)\}$) and hence is uniformly bounded (irrespective of the value of L) as well. Before we proceed further, let us look at the term

$$\frac{1}{L} \sum_{m=0}^{L-1} Z_m^-(n+1),$$

on the RHS of the second equation in (A6). We will show that it has the same asymptotic behaviour as $Z_L^-(n)$. A similar argument holds for the term

$$\frac{1}{L} \sum_{m=0}^{L-1} Z_m^+(n+1),$$

on the RHS of the third equation in (A6). First note that the terms $Z_0^-(n+1), Z_1^-(n+1), \dots, Z_{L-1}^-(n+1)$ are all governed by the same parameter update namely, $(\theta(n+1) - \delta\Delta(n+1))$. For notational simplicity let $(n+1)L = k$ in the rest of the proof. Now from the second equation in (12),

$$Z^-(k+1) = (1 - b(n))Z^-(k) + b(n)h(Y^-(k)).$$

Writing iteratively (cf. Bhatnagar *et al.* (2000)), one obtains upon simplification

$$\begin{aligned} \frac{1}{L} \sum_{m=0}^{L-1} Z^-(k+m) &= \frac{1}{L} \left[\left(\frac{1 - (1 - b(n))^L}{b(n)} \right) Z^-(k) \right. \\ &\quad + \sum_{i=1}^{L-2} (1 - (1 - b(n))^{L-i}) \\ &\quad \left. \times h(Y^-(k+i-1)) \right] \end{aligned} \tag{A7}$$

It is now easy to see using routine arguments (see Theorem 5.2 of Bhatnagar *et al.* (2000)) that

$$\left\| \frac{1}{L} \sum_{m=0}^{L-1} Z^-(k+m) - Z^-(k) \right\| \rightarrow 0, \quad \text{as } k \rightarrow \infty.$$

Consider now an algorithm analogous to (A6) but with

$$\frac{1}{L} \sum_{m=0}^{L-1} h(Y_m^-(n+1)) \text{ (resp. } \frac{1}{L} \sum_{m=0}^{L-1} h(Y_m^+(n+1))),$$

replaced by $J(\theta(n+1) - \delta\Delta(n+1))$ (resp. $J(\theta(n+1) + \delta\Delta(n+1))$), and with

$$\frac{1}{L} \sum_{m=0}^{L-1} Z_m^-(n+1) \text{ (resp. } \frac{1}{L} \sum_{m=0}^{L-1} Z_m^+(n+1)),$$

replaced by $Z_L^-(n)$ (resp. $Z_L^+(n)$). It can be easily shown as in Lemma 4.3 of Bhatnagar and Borkar (1998), that the latter algorithm would have analogous asymptotic behaviour as (A6). Finally, the first equation in (A6) can be rewritten as

$$\theta_i(n+1) = \pi_i \left(\theta_i(n) + c(n) \frac{a(n)}{c(n)} \left[\frac{Z_L^-(n) - Z_L^+(n)}{2\delta\Delta_{n,i}} \right] \right) \tag{A8}$$

Moreover, since $a(n) = o(c(n))$, applying standard arguments as in Borkar (1997) to (A8) and the second and third equations in (A6), one obtains the claim. ■

Biographies

Shalabh Bhatnagar received his Bachelors in Physics (Honors) from the University of Delhi, India in 1988, Masters and Ph.D. in Electrical Engineering from the Indian Institute of Science, Bangalore, in 1992 and 1998, respectively. Since October 1997, he has been a Research Associate at the Institute for Systems Research, University of Maryland, College Park. His areas of interest include stochastic approximation algorithms, Markov decision processes and their applications to optimization and control of discrete event systems.

Michael C. Fu is a Professor in the Robert H. Smith School of Business, with a joint appointment in the Institute for Systems Research, at the University of Maryland. He received degrees in Mathematics and EE/CS from MIT, and a Ph.D. in Applied Mathematics from Harvard University. His research interests include simulation optimization and sensitivity analysis, particularly with applications towards manufacturing systems, inventory control, and the pricing of financial derivatives. He teaches courses in applied probability, stochastic processes, discrete-event simulation, and operations management, and in 1995 was awarded the Maryland Business School's Allen J. Krowe Award for Teaching Excellence. He is a member of INFORMS, IEEE and a Senior Member of IIE. He serves on the Editorial Boards of *Operations Research*, *Management Science*, *INFORMS Journal on Computing*, *IIE Transactions*, and *Production and Operations Management*. He is co-author (with J.Q. Hu) of the book, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*.

Steven I. Marcus received a B.A. degree in Electrical Engineering and Mathematics from Rice University in 1971 and the S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology in 1972 and 1975, respectively. From 1975 to 1991, he was with the Department of Electrical and Computer Engineering at the University of Texas at Austin, where he was the L.B. (Preach) Meaders Professor in Engineering. He was Associate Chairman of the Department during the period 1984-89. In 1991, he joined the University of Maryland, College Park, where he was Director of the Institute for Systems Research until 1996. He is currently a Professor in the Electrical and Computer Engineering Department and the Institute for Systems Research. He is a Fellow of IEEE. His primary research interests are concerned with systems and control theory, stochastic and adaptive control, estimation, learning, and discrete event systems. Dr. Marcus is a Corresponding Editor of the *SIAM Journal on Control and Optimization*, Associate Editor of *Discrete Event Dynamic Systems*:

Theory and Applications, Associate Editor of *Acta Applicandae Mathematicae*, and Associate Editor of *Mathematics of Control, Signals, and Systems*.

Shashank Bhatnagar received his Bachelors, Masters and Ph.D. degrees in Physics from the University of Delhi, India, in 1983, 1985 and 1992,

respectively. From 1990 to 1998, he was a Senior Lecturer in the Department of Physics, University of Delhi. Since September 1998, he has been a member of the Department of Physics, University of Addis Ababa, Ethiopia, where he is currently an Associate Professor. His areas of interest include few body dynamics, quantum nonlinear filtering and stochastic control, and nuclear physics.