Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Discrete Parameter Simulation Optimization Algorithms with Applications to Admission Control with Dependent Service Times

Vivek Mishra , Shalabh Bhatnagar and N. Hemachandra

***Abstract*— We propose certain discrete parameter variants of well known simulation optimization algorithms. Two of these algorithms are based on the smoothed functional (SF) technique while two others are based on the simultaneous perturbation stochastic approximation (SPSA) method. They differ from each other in the way perturbations are obtained and also the manner in which projections and parameter updates are performed. All our algorithms use two simulations and two-timescale stochastic approximation. As an application setting, we consider the important problem of admission control of packets in communication networks under dependent service times. We consider a discrete time slotted queueing model of the system and consider two different scenarios – one where the service times have a dependence on the system state and the other where they depend on the number of arrivals in a time slot. Under our settings, the simulated objective function appears ill-behaved with multiple local minima and a unique global minimum characterized by a sharp dip in the objective function in a small region of the parameter space. We compare the performance of our algorithms on these settings and observe that the two SF algorithms show the best results overall. In fact, in many cases studied, SF algorithms converge to the global minimum.**

I. Introduction

Simulation based optimization approaches find wide applicability in a variety of diverse disciplines such as neural networks, supply chain management, computer networks etc. Many of these methods use some form of gradient search. In many scenarios studied in the literature, noisy observations of the objective function are assumed available based on which gradient estimates are obtained. In some instances such as the one considered in this paper, the objective function in fact corresponds to a long-run average performance metric. Examples include the problems of admission control or resource allocation in communication networks where the objective (that one wants to minimize) is (say) the mean waiting time in steady state of a customer. It is here that simulation based optimization approaches play a significant role. In [1], [2], for objective functions such as above, two-timescale stochastic approximation algorithms are proposed for simulation optimization. The overall idea here is that estimates of the objective function are obtained and aggregated along the faster timescale or step-size schedule while parameter updates along negative gradient directions are performed along the slower one.

Simulation optimization algorithms based on Kiefer-Wolfowitz (K-W) finite difference gradient estimates typically require $2N$ parallel simulations when the parameter dimension is N. In [14], a random directions version of K-W gradient estimates was proposed where all parameters are simultaneously perturbed most commonly by using i.i.d. symmetric Bernoulli random variables. This algorithm, also known as the simultaneous perturbation stochastic approximation (SPSA), requires only two function evaluations per iteration irrespective of the parameter dimension N. SPSA has been widely studied in many applications and is seen to perform well in general. Another algorithm based on random perturbations is the smoothed functional algorithm (SFA). Here, one approximates the gradient of the objective function by a convolution of the same with a multivariate normal distribution. By an integration by parts argument, one sees the same as a scaled convolution of the objective function itself with the (above) multivariate normal distribution. This requires only one simulation irrespective of the parameter dimension N. A two-sided estimate of the smoothed functional gradient as in [16] is seen to improve performance. In [2] and [3], certain two-timescale versions of SPSA and one-simulation SFA, respectively, have been proposed for simulation optimization. More recently, in [6], more SFA based algorithms have been developed, one of which estimates only the gradient using a two-sided gradient estimate and two others estimate both the gradient and Hessian in a Newton based scheme.

Both SPSA and SFA have been studied most often in the continuous parameter optimization framework. In [10], [11], discrete parameter variants of the SPSA algorithm have been proposed. In [4], an adaptation of two-timescale SPSA for discrete parameter optimization has been considered. The idea here is to form a closed convex hull of the underlying parameter set, perform parameter updates using the two-timescale algorithm in the continuous space formed by the above closed convex hull and obtain the corresponding policy updates by projecting the parameter update to the underlying discrete set. In [5], another discrete parameter SPSA algorithm is presented. Here, instead of forming a convex hull as above, the parameter itself is projected after each update epoch to the underlying discrete set. Thus the parameter updates are performed here in the discrete space.

In this paper, we develop certain discrete parameter variants of the two-simulation SFA and SPSA algorithms of [6] and [2] respectively. The variants that we propose for

V.Mishra is with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India. E-Mail: mishra@csa.iisc.ernet.in

S.Bhatnagar is with the Department of Computer Science and Automation, Indian Institue of Science, Bangalore, India. E-Mail: shalabh@csa.iisc.ernet.in

N.Hemachandra is with the Department of Industrial Engineering and Operations Research, Indian Institute of Technology Bombay, Mumbai, India. E-Mail: nh@iitb.ac.in

1-4244-1498-9/07/$25.00 ©2007 IEEE.

SPSA are similar in nature to the algorithms in [4] and [5] respectively. However, they differ in the manner in which projections are done. We use simple randomizations in policies here in place of direct projections to the discrete set (as in [4] and [5]) that result in improved performance. As an application setting, we consider the important problem of admission control in communication networks under dependent service times. We consider two different settings here. In the first, the service times depend on the system state while in the second, they depend on the arrival process. We consider a discrete time queueing framework for this purpose. This is quite different from the models considered in [4] and [5] where continuous time queueing framework is used. Moreover, the service times in the above references have been taken to be i.i.d. It is extremely difficult to obtain precise analytical solutions in the case of models that incorporate dependence. Our simulation based optimization methodology helps in obtaining the (locally) optimal parameter values and hence (locally) optimal policies within each specified class. We study experiments using these algorithms and observe that both the SFA based algorithms in fact converge to the globally optimal policy (within the considered class of policies) in many cases while those based on SPSA seem to converge to one of the locally optimal policies (in the above class) in most cases. This could possibly be the result of 'objective function smoothing' when SFA is used and is in agreement with the observations in [16]. We however did not study the performance of SPSA when perturbations other than Bernoulli are used and also when the gain parameter δ (below) is suitably adapted. As explained in [15], the latter may result in convergence to a global optimum in SPSA algorithms.

The rest of the paper is organized as follows: Section II describes the framework and algorithms. In Section III, we present the problem of admission control under dependent service times that we consider for the purpose of applications. In Section IV, we present our numerical experiments. Finally, concluding remarks are made in Section V.

II. Framework and Algorithms

Consider a Markov process $\{X_n^\theta\}$ parameterized by $\theta \in D$, $D \subset \mathcal{I}^N$ for some fixed $N \in \mathcal{I}^+$. Here $\mathcal{I}$ and $\mathcal{I}^+$ denote the sets of integers and positive integers respectively. We assume D to be a finite set (which, however, could be large). In particular, for simplicity, we assume that D has the form $D = \prod_{i=1}^N D^i$, where $D^i = \{d_i^0, \ldots, d_i^{n_i}\}$, $i = 1, \ldots, N$. Here, for any i, $d_i^0 \leq d_i^1 \leq \cdots \leq d_i^{n_i}$. Let $D_{i,\min} \stackrel{\triangle}{=} d_i^0$ and $D_{i,\max} \stackrel{\triangle}{=} d_i^{n_i}$ respectively. We also assume that X_n^θ, $n \geq 0$ take values in $\mathcal{R}^d$ for some $d \in \mathcal{I}^+$. For any fixed $\theta \in D$, we assume $\{X_n^\theta\}$ is ergodic with transition kernel $p_\theta(x, y)$, $x, y \in \mathcal{R}^d$. Let $h : R^d \to R$ be a given single-stage cost function that we assume is Lipschitz continuous. Our aim is to find a $\theta^* \in D$ such that

$$J(\theta^*) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^n h(X_i^{\theta^*}) = \min_{\theta \in D} J(\theta). \qquad (1)$$

Here $J(\cdot)$ denotes the long run average cost. Let $\bar{D}$ denote the closed convex hull of D. We describe below our algorithms. We denote the parameter updates as $\theta(n) \stackrel{\triangle}{=} (\theta_1(n), \ldots, \theta_N(n))^T$, $n \geq 1$. Algorithms SPSA-C and SFA-C (below) perform these updates within the convex set $\bar{D}$. For a given $\theta \in \bar{D}$, in order to obtain a sample X_m^θ of the state of the Markov process at an instant m, it was proposed in [4] to first project θ to D. Let $\bar{\theta}$ denote the projected parameter. The above sample would then correspond to $X_m^{\bar{\theta}}$. (Note that X_m^θ is not even defined for $\theta \in \bar{D} \backslash D$.)

We however propose and use an improved procedure based on simple randomizations for the above projections. For any $\theta = (\theta_1, \ldots, \theta_N)^T \in \mathcal{R}^N$, we define $\Gamma(\theta) = (\Gamma_1(\theta_1), \ldots, \Gamma_N(\theta_N))^T \in D$ as the projection of θ on to the set D as follows: Let θ_i be such that $d_i^j < \theta_i < d_i^{j+1}$ for some $d_i^j, d_i^{j+1} \in D^i$, $d_i^j < d_i^{j+1}$. Then,

$$\begin{aligned} \Gamma_i(\theta_i) &= d_i^j \text{ w.p. } (d_i^{j+1} - \theta_i)/(d_i^{j+1} - d_i^j) \\ &= d_i^{j+1} \text{ w.p. } (\theta_i - d_i^j)/(d_i^{j+1} - d_i^j). \end{aligned}$$

Also,

$$\begin{aligned} \Gamma_i(\theta_i) &= D_{i,\min} \text{ if } \theta_i < D_{i,\min} \\ &= D_{i,\max} \text{ if } \theta_i \geq D_{i,\max}. \end{aligned}$$

It is intuitively clear (as also experimentally observed by us) that a simple randomization as above would result in an improvement in performance over the projection methods in [4] and [5]. This is because a 'small' increment in the 'right' direction is not ignored when using randomized projections. On the other hand, one may obtain a big leap in the correct direction with a probability proportional to the size of the increment. Note from the way it is defined that $\Gamma(\theta) \in D$. For Algorithms SPSA-C and SFA-C, for any $\theta \in \mathcal{R}^N$, we also require the projection $\bar{\Gamma}(\theta) \stackrel{\triangle}{=} (\bar{\Gamma}_1(\theta_1), \ldots, \bar{\Gamma}_N(\theta_N))^T$ to the compact set $\bar{D}$ (the closed convex hull of D) and is defined in the usual manner. In SPSA-C and SFA-C, the projection $\bar{\Gamma}(\cdot)$ will be used to guide the parameter updates within the set $\bar{D}$ while $\Gamma(\cdot)$ will be used to identify the actual parameter value used in a simulation. Algorithms SPSA-D and SFA-D, on the other hand, do not use the projection operator $\bar{\Gamma}(\cdot)$ to the convex hull $\bar{D}$ but instead use $\Gamma(\cdot)$ (defined above) and thus directly update the parameter in the discrete space D itself.

For all our algorithms, we use two step-size sequences $\{a(n)\}$ and $\{b(n)\}$ that satisfy the conditions

$$\sum_n a(n) = \sum_n b(n) = \infty, \ \sum_n (a(n)^2 + b(n)^2) < \infty,$$
$$a(n) = o(b(n)).$$

A. The SPSA-C Algorithm

Let $\theta(m) := (\theta_1(m), \ldots, \theta_N(m))^T$ denote the parameter vector after the mth iteration. Let $\Delta_1(m), \ldots, \Delta_N(m)$ denote i.i.d. Bernoulli distributed symmetric ± 1-valued random variables. Set $\theta_i^1(m) = \Gamma_i(\theta_i(m) + \delta\Delta_i(m))$ and $\theta_i^2(m) =$

$\Gamma_i(\theta_i(m) - \delta\Delta_i(m))$ for $i = 1, \ldots, N$, where $\delta > 0$ is a given (small) constant. Let $\theta^j(m) \stackrel{\triangle}{=} (\theta^j_1(m), \ldots, \theta^j_N(m))$, $j = 1, 2$. Set $Z(0) = 0$.

Generate two parallel simulations $\{X_m^{\theta^1(m)}\}$ and $\{X_m^{\theta^2(m)}\}$ governed by parameter sequences $\{\theta^1(m)\}$ and $\{\theta^2(m)\}$, respectively. For $i = 1, \ldots, N$, $m \geq 0$, we have

$$\theta_i(m+1) = \bar{\Gamma}_i(\theta_i(m) + a(m)(\frac{Z(m+1)}{2\delta\Delta_i(m)})),$$

$$Z(m+1) = Z(m) + b(m)(h(X_m^{\theta^2(m)}) - h(X_m^{\theta^1(m)}) - Z(m)).$$

B. The SFA-C Algorithm

Let $\theta(m) := (\theta_1(m), \ldots, \theta_N(m))^T$ denote the parameter vector after the mth iteration. Let $\eta_1(m), \ldots, \eta_N(m)$ be independent $N(0,1)$-distributed random variables. Let $\beta > 0$ be a given (small) constant. Let $\theta^j(m) = (\theta^j_1(m), \ldots, \theta^j_N(m))^T$, $j = 1, 2$, where $\theta^1_i(m) = \Gamma_i(\theta_i(m) + \beta\eta_i(m))$ and $\theta^2_i(m) = \Gamma_i(\theta_i(m) - \beta\eta_i(m))$, $i = 1, \ldots, N$, respectively.

Generate two parallel simulations $\{X_m^{\theta^1(m)}\}$ and $\{X_m^{\theta^2(m)}\}$ governed by parameter sequences $\{\theta^1(m)\}$ and $\{\theta^2(m)\}$, respectively. For $i = 1, \ldots, N$, $m \geq 0$, we have

$$\theta_i(m+1) = \bar{\Gamma}_i(\theta_i(m) + a(m)Z_i(m+1)),$$

$$Z_i(m+1) = Z_i(m) + b(m)(\frac{\eta_i(m)}{2\beta}(h(X_m^{\theta^2(m)}) - h(X_m^{\theta^1(m)}) - Z(m)).$$

C. The SPSA-D and SFA-D Algorithms

These are the same as SPSA-C and SFA-C algorithms above except that in Steps 2 of the above algorithms, one replaces the projection map $\bar{\Gamma}$ with Γ. Thus these algorithms directly update the parameter vector in the discrete space (randomly) D instead of doing so in the convex hull $\bar{D}$.

Remark: In both the above algorithms, we observe as with [2], [4] that an extra averaging of the faster recursion (viz., the one governed by step-size $b(m)$) over a certain number $\bar{M}$ of instants, improves performance. We let $\bar{M} = 500$ in our experiments. We generate $N(0,1)$-distributed random variables in SFA using the standard Box-Muller algorithm.

We do not give the convergence analysis here for lack of space, see [6] for analysis of SFA for the continuous parameter case and [4]-[5] for analysis of discrete parameter SPSA using other (not randomized) projection maps. For continuous parameter optimization, in [8], performance comparisons between one-timescale SPSA and SFA algorithms are shown. It is observed in [8] that SPSA shows better performance than SFA over an objective function that is reasonably well behaved. For an ill-behaved objective function in a discrete parameter simulation optimization setting as with our experiments, we observe that SFA based algorithms converge more often to the global minimum as compared to SPSA based ones. It is argued in [16] that the SFA algorithm (viz., the continuous parameter version) converges to the globally optimal solution much like simulated annealing because it performs optimization over a smoothed functional and not the original function. The original function may have several local minima and may not be well behaved in general. Our results are in agreement with the observations in [16].

In the next section, we study applications of the proposed methodology to the important problem of admission control under dependent service times. We consider two different scenarios for service time dependence – in the first, the service times are a given function of the queue length while in the second, they depend on the number of packets that arrive in a given time slot.

III. Applications to Admission Control with Dependent Service Times

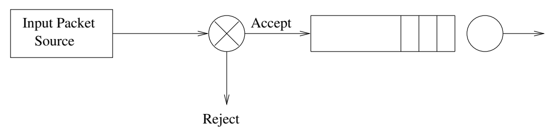

Fig. 1. The admission control model

We consider the problem of admission control in communication networks. Admission control has been widely studied over the past several years, see [7], [13], for some representative work. In [5], a continuous time queueing model in a stochastic setting is considered. Many references in the literature consider service times to be independent. We however consider the case where these depend on the system state or arrival process. Such dependence may arise naturally in many scenarios. For instance, in [9], a finite buffer queue where service times can take one of two random values depending on whether the queue length is below or above a given threshold is used to model the cell discarding scheme arising for voice traffic in ATM networks. In [12], the stationary distribution associated with a continuous time multi-class queueing network of single server nodes is obtained where the service of a customer belonging to a class depends not only on queue contents but also on the residual work loads of customers at the node. We however use simulation based optimization techniques for admission control in a dependent queue.

We consider two different settings here. In the first, the service times are a deterministic function of the current queue length, while in the second, they depend in a probabilistic manner on the number of arrivals in a given time slot. We consider threshold type feedback policies in both settings. In particular, we consider a one-level policy structure for the former setting and a four-level structure for the latter. The form of the policy is guided by the underlying Markov process in either case. We compare the performance of SPSA and SFA algorithms in computing an optimal policy within the prescribed class of policies. Our basic model is shown in Fig. 1. We consider a slotted system here. There is a single node that is fed with packets whose arrivals at each instant are geometrically distributed with some fixed parameter $p \in [0, 1)$. Both arrival and service times take values from a discrete set. We accept or reject a packet depending on the admission policy under consideration. We now explain the two settings considered here.

A. The One-Level Admission Policy

Here the service time S_n at instant n, when the number of packets in queue as seen by an arriving packet is q_n, is given by

$$S_n = (q_n - \frac{B}{2})^2 + 1$$

where B is the buffer size (at the node) that we assume is an even positive integer so that S_n is also a positive integer. We consider the following form for the admission policy: At instant n, given $L \in \mathcal{I}^+$, we have

- if $q_n \leq L$
  accept incoming packet,
- else
  reject incoming packet.

Thus, the parameter to be optimized corresponds to $\theta = L$. One can see here that $\{q_n\}$ is ergodic Markov for any given θ and thus SFA and SPSA algorithms can be applied.

B. The Four-Level Admission Policy

Let A_n denote the number of packets that arrive at the nth instant. The generic service time of a packet served at time n from this batch (of A_n packets) is given by

$$S_n = \lfloor (A_n - \overline{L})^2 + A_n U + 1 \rfloor$$

where, $\overline{L} = \left\lfloor \frac{p}{1-p} \right\rfloor$ and U is a random variable that has the distribution $U(0, 1)$. Here $\bar{L}$ denotes the integer part of the mean number of arrivals in a time slot. The admission policy in this case is as follows: Let X_n be a 'scaling' function defined according to

$$X_n = \begin{cases} 0 & A_n \leq 1 \\ 1 & 2 \leq A_n \leq 3 \\ 2 & A_n = 4 \\ 3 & A_n \geq 5 \end{cases}$$

Then, for $i = 0, 1, 2, 3$,

- if $(X_n = i)$
  - if $q_n \leq L_i$
    accept incoming packet,
  - else
    reject incoming packet.

Note that X_n plays the role of clustering into four levels, the various values of A_n. The admission control policy assigns threshold L_i for the ith such level, $i = 0, 1, 2, 3$. The parameter θ then corresponds to $\theta = (L_0, L_1, L_2, L_3)^T$. Note that because of the dependence of service times on the arrival process, for any given θ, $\{(q_n, A_n)\}$ is now Markov and the above class of policies is meaningful. One may consider any number of levels for the above policies (not necessarily four). Our objective function appears ill-behaved and seems to have unique global minimum but multiple local minima. Recall that the objective function here corresponds to the long-run average over single-stage costs and thus is not analytically known a priori in many cases even if the single-stage cost has an analytical expression. In the next section, we show a few plots of the objective function estimate obtained via simulation over a large number of runs.

IV. Numerical Results

We show numerical results using our algorithms for the two admission policy settings described in the previous section. We assume a buffer size of $B = 100$. The cost of acceptance of an arriving packet is the queue length as seen by it. Further, the cost of rejecting a packet is fixed at 75 or three-fourths of the buffer size. The step-sizes $\{a(n)\}$ and $\{b(n)\}$ for both algorithms for each of the two settings below are chosen as

$$b(n) = \frac{1}{\lceil \frac{n}{20} \rceil^{2/3}}, a(n) = \frac{1}{\lceil \frac{n}{20} \rceil^{3/4}}$$

respectively, for $n \geq 1$ and $a(0) = b(0) = 1$.

A. The One-Level Policy Setting

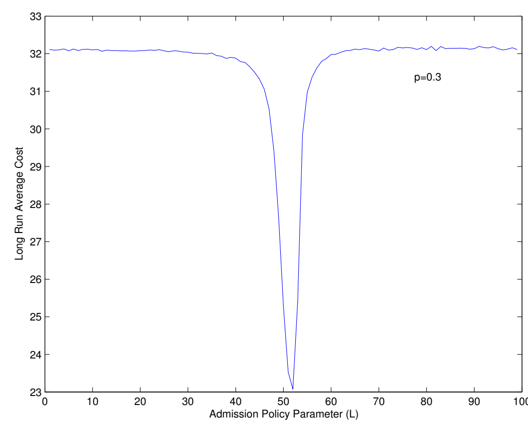

Fig. 2. The Objective Function for the One-Parameter Case with $p = 0.3$

The simulated long-run average cost objective of the system when $p = 0.3$ is plotted in Fig. 2. Similar plots were obtained for other values of p. Here, for each integer value of L in the range from 1 to 100, the corresponding objective function value was simulated using 5×10^5 simulation runs. The above plots exhibit the existence of a global minimum around $L = 51$. The function falls sharply within a narrow band around the point $L = 51$ at which global minimum is attained. Also, there seem to be multiple local minima with the function being almost flat in a large portion of the parameter space. We run all four algorithms for a total of 5×10^7 simulation runs. We obtain a total of 10^5 parameter updates using both algorithms.

The initial value of acceptance policy (L) is set at 75. For the SPSA (resp. SFA) algorithms we set $\delta = 1.0$ (resp. $\beta = 1.0$) as the setting parameters. These choices seem to give the best results overall. Using the converged parameter values, we obtain sample trajectories with fifty different initial seeds for 10^5 simulation runs. The mean and standard error of the average cost for the above runs for different values of p for the 'C' and 'D' algorithms are shown in Tables I and II. We observe in our experiments that SFA settles to the global minimum in most cases. Note here that in all the entries in Tables I-II, the mean values of SFA algorithms shown are lower (in some cases by a high margin) as compared to the corresponding values for SPSA algorithms. However, in many cases, the standard error in SFA algorithms is much higher as compared to SPSA. This is again because of the nature of the objective function (explained above) with the result that when SFA is used, convergence to the global minimum is achieved with many different starting seeds, while with some other seeds, convergence only to a local minimum is attained. On the other hand, SPSA algorithms do not have a high value of standard deviation as they seem to converge in most cases only to local minima. This also explains the high mean values in the SPSA algorithms as compared to SFA in the entries above. Note that in cases when $p \geq 0.4$, the standard error is quite high in SFA algorithms as compared to the same with lower p-values where this difference is less. This is because the difference in costs between the unique global minimum and the other local minima (in the flat region) when $p \geq 0.4$ is also much higher than when $p < 0.4$. The plots of convergence of L with the number of simulation runs, when $p = 0.3$, for the 'C' and 'D' algorithms are shown in Fig. 3. A key observation is that in both figures, the SFA algorithms converge to the unique global minimum ($L = 51$). As expected, SPSA-D and SFA-D show more fluctuations in their trajectories as compared to their 'C'-counterparts. Amongst 'C' and 'D' algorithms, the former algorithms show better results.

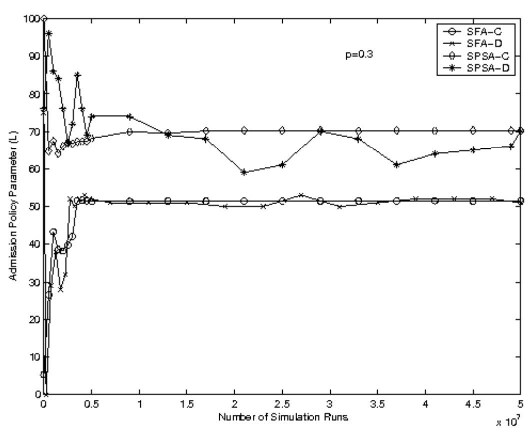

Fig. 3. Convergence of Algorithms for One-Parameter Case

B. The Four-Level Policy Setting

Note that since the parameter θ here corresponds to $\theta = (L_0, L_1, L_2, L_3)^T$, it is not possible to plot the long-run average cost as a function of θ. In order to get an idea, however, we plot here the long-run average cost for only those θ for which $L_0 = L_1 = L_2 = L_3 = L'$. We vary

TABLE I

OPTIMAL AVERAGE COSTS OBTAINED USING SPSA-C AND SFA-C: ONE-PARAMETER CASE

p	SPSA-C	SFA-C
0.1	8.13 ± 0.69	6.73 ± 1.45
0.2	18.05 ± 1.82	16.20 ± 2.96
0.3	31.77 ± 1.17	26.97 ± 4.35
0.4	49.63 ± 1.04	45.01 ± 5.57
0.5	74.62 ± 0.98	67.38 ± 6.68
0.6	112.26 ± 0.70	103.04 ± 7.73
0.7	174.62 ± 0.79	165.67 ± 8.68
0.8	298.74 ± 3.97	289.76 ± 9.14
0.9	674.69 ± 0.88	664.13 ± 9.58

TABLE II

OPTIMAL AVERAGE COSTS OBTAINED USING SPSA-D AND SFA-D: THE ONE-PARAMETER CASE

p	SPSA-D	SFA-D
0.1	8.20 ± 0.46	7.08 ± 1.38
0.2	18.25 ± 1.35	15.54 ± 2.95
0.3	31.78 ± 1.25	25.08 ± 3.52
0.4	49.66 ± 1.61	42.27 ± 5.06
0.5	74.82 ± 0.72	67.39 ± 6.10
0.6	111.91 ± 1.88	102.99 ± 6.92
0.7	174.48 ± 1.70	164.79 ± 7.04
0.8	299.85 ± 0.47	292.54 ± 8.23
0.9	674.79 ± 0.91	673.01 ± 4.92

L' from 1 to 100 and plot the simulated average cost for $p = 0.4$ in Fig. 4.

We run all algorithms for 10^8 simulation runs for this setting. We set the initial value of the parameter vector as $(50, 50, 50, 50)^T$ for all algorithms. As before for the SPSA (resp. SFA) algorithms, we set $\delta = 1.0$ (resp. $\beta = 1.0$). With the converged values of the parameter updates, for 10^5 iterations, we run simulations with fifty different initial seeds and obtain the mean and standard error of the long-run average cost over these (cf. Tables III-IV). As with the previous setting, we observe from our experiments that both SFA algorithms show better performance as compared to corresponding SPSA ones as with many initial seeds, SFA algorithms seem to converge to global minimum while SPSA

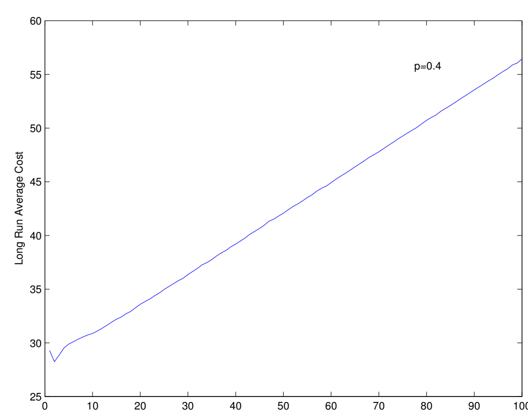

Fig. 4. The Objective Function for the Four-Parameter Case for p=0.4

TABLE III
OPTIMAL AVERAGE COSTS OBTAINED USING SPSA-C AND SFA-C: THE FOUR-PARAMETER CASE

p	SPSA-C	SFA-C
0.1	0.13 ± 0.00	0.13 ± 0.00
0.2	0.71 ± 0.03	0.74 ± 0.06
0.3	13.43 ± 5.06	9.51 ± 3.83
0.4	42.14 ± 7.86	28.20 ± 10.64
0.5	72.60 ± 5.14	58.60 ± 13.09
0.6	109.78 ± 2.42	81.21 ± 18.22
0.7	170.15 ± 6.47	153.14 ± 15.76
0.8	297.10 ± 3.69	280.85 ± 12.61
0.9	676.11 ± 2.39	675.87 ± 0.56

TABLE IV
OPTIMAL AVERAGE COSTS OBTAINED USING SPSA-D AND SFA-D: THE FOUR-PARAMETER CASE

p	SPSA-D	SFA-D
0.1	0.13 ± 0.00	0.13 ± 0.00
0.2	0.75 ± 0.03	0.75 ± 0.05
0.3	13.82 ± 3.57	9.85 ± 3.07
0.4	44.52 ± 6.48	31.39 ± 12.53
0.5	74.50 ± 2.97	52.86 ± 13.31
0.6	112.64 ± 7.26	73.14 ± 8.73
0.7	174.76 ± 2.99	150.27 ± 17.37
0.8	299.92 ± 2.54	286.28 ± 11.09
0.9	675.91 ± 2.36	674.60 ± 1.11

ones seem to converge to local minima. As with the previous setting, this is also the reason for the high standard error in SFA algorithms that results due to convergence to a global minimum with some seeds and to one of the local minima with some other seeds. As before, the 'C'-algorithms are seen to show better results as compared to their 'D'-counterparts.

In order to validate our observation on high standard error in the above experiments for SFA algorithms, we conducted experiments for the one-parameter setting where the service times are simply selected to be i.i.d. and have the distribution $U(0,1)$. We observe here that SFA algorithms show better performance in both mean and standard error over SPSA algorithms. We do not show the results of these experiments for lack of space.

V. CONCLUSIONS

We developed four discrete parameter variants of simulation optimization algorithms. Two of these are based on the simultaneous perturbation stochastic approximation (SPSA) method and two others are based on the smoothed functional (SF) technique. These differ in the parameter space in which updates are performed as also the perturbations used viz., Gaussian vrs. Bernoulli. We proposed a different method for projection to the discrete space that is based on a simple randomization and that results in an improved performance. We considered the problem of admission control under dependent service times for two different settings – one where the service times are a function of the system state and the other where they depend on the number of arrivals in a time slot. The objective function was observed to be ill-behaved with multiple local minima and a unique global minimum characterized by a sharp drop in a small region of the parameter space. Our numerical results show that SFA algorithms (based on Gaussian perturbations) converge to the global minimum in many cases. This is in agreement with the observations in [16] for the case of one-timescale continuous parameter optimization using SFA. SFA-C shows the best results overall. Also, as expected, in general the 'C'-algorithms perform better than their 'D'-counterparts. In [6], certain smoothed functional estimates of the Hessian have recently been developed and used in a Newton based scheme. The discrete parameter variants of these algorithms could also be developed and their performance tested on similar settings.

Acknowledgements

S. Bhatnagar was supported in his research through grants SR/S3/EE/43/2002-SERC-Engg. and SR/S3/EECE/011/2007 from the Department of Science and Technology, Government of India.

REFERENCES

[1] 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.
[2] Bhatnagar, S., Fu, M.C., Marcus, S.I. and Bhatnagar, S. (2001) "Two timescale algorithms for simulation optimization of hidden Markov models", *IIE Transactions*, 33(3):245-258.
[3] Bhatnagar, S. and Borkar V. (2003) "Multiscale Chaotic SPSA and Smoothed Functional Algorithms for Simulation Optimization", *Simulation*, 79(10):568-580.]
[4] Bhatnagar, S., and Reddy, I.B.B. (2005) "Optimal threshold policies for admission control in communication networks via discrete parameter stochastic approximation", *Telecommuncation Systems* 29 (1):9-31.
[5] Bhatnagar, S. and Kowshik, H.J. (2005) "A discrete parameter stochastic approximation algorithm for simulation optimization", *Simulation*, 81(11):757-772.
[6] Bhatnagar, S. (2008) "Adaptive Newton-based multivariate smoothed functional algorithms for simulation optimization", *ACM TOMACS*, to appear.
[7] Cheng, R.-G., Chang, C.-J. and Lin, L.-F. (1999) "A QoS provisionig neural fuzzy connection admission controller for multimedia high speed networks", *IEEE/ACM Trans. on Network.*, 7(1):111-121.
[8] Chin, D.C. (1997) "Comparative study of stochastic algorithms for system optimization based on gradient approximations", *IEEE Trans. Systems, Man, and Cybernetics - Part B:Cybernetics*, 27(2):244-249.
[9] Choi, B.D. and Choi, D.I. (1996) "Queueing system with queue length dependent service times and its application to cell discarding scheme in ATM networks", *IEE Proceedings - Communications*, 143(1):5-11.
[10] Gerencsér, L., Hill, S. D., and Vágó, Z. (1999) "Optimization over Discrete Sets via SPSA," in Proc. of the 38th IEEE Conf. on Decision and Control, Phoenix, AZ, pp. 1791-1795.
[11] Hill, S. D., Gerencsér, L., and Vágó, Z. (2004) "Stochastic Approximation on Discrete Sets Using Simultaneous Difference Approximations," in Proc. of the 2004 American Control Conf., Boston, MA, pp. 2795-2798.
[12] Ivnitski, V.A. (2001) "Network of single-server queues in dependent service times", *Queueing Systems*, 37:363-377.
[13] Kelly, F.P., Key, P.B. and Zachary, S. (2000) "Distributed admission control", *IEEE Journal on Selected Areas in Communications*, 18:2617-2628.
[14] Spall, J.C. (1992) "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation", *IEEE Trans. Autom. Control* 37:332-341.
[15] Spall, J.C. (2003) *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, Hoboken, NJ.
[16] Styblinski, M.A. and Tang, T.-S. (1990) "Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing", *Neural Networks*, 3:467-483.