Comparative Study of Stochastic Algorithms for System Optimization Based on Gradient Approximations

Daniel C. Chin

***Abstract*—Stochastic approximation (SA) algorithms can be used in system optimization problems for which only noisy measurements of the system are available and the gradient of the loss function is not. This type of problem can be found in adaptive control, neural network training, experimental design, stochastic optimization, and many other areas. This paper studies three types of SA algorithms in a multivariate Kiefer–Wolfowitz setting, which uses only noisy measurements of the loss function (i.e., no loss function gradient measurements). The algorithms considered are: the standard finite-difference SA (FDSA) and two accelerated algorithms, the random-directions SA (RDSA) and the simultaneous-perturbation SA (SPSA). RDSA and SPSA use randomized gradient approximations based on (generally) far fewer function measurements than FDSA in each iteration. This paper describes the asymptotic error distribution for a class of RDSA algorithms, and compares the RDSA, SPSA, and FDSA algorithms theoretically (using mean-square errors computed from asymptotic distributions) and numerically. Based on the theoretical and numerical results, SPSA is the preferable algorithm to use.**

***Index Terms*—Asymptotic normality, convergence rate, gradient approximation, Kiefer–Wolfowitz algorithm, optimization, stochastic approximation.**

I. Introduction

In virtually all areas of engineering and the physical and social sciences, one encounters problems involving the optimization of some mathematical objective function (e.g., as in optimal control, system design and planning, model fitting, and performance evaluation from system test data). Typically, the solution to this optimization problem corresponds to a vector of parameters such that the gradient of the objective function (with respect to the system parameters being optimized) is zero. Over the last several years, there has been a growing interest in recursive optimization algorithms that do not depend on direct gradient information or measurements. Rather, these algorithms are based on and *approximation* to the gradient formed from (generally noisy) measurements of the objective function. This interest has been motivated, for example, by problems in the adaptive control and statistical identification of complex systems, the optimization of processes by large Monte Carlo simulations, the training of recurrent neural networks, and the design of complex queuing and discrete-event systems.

Overall, such algorithms exhibit certain convergence properties of gradient-based algorithms while requiring only objective (say, loss) function measurements. A main advantage of such algorithms is that they do not require the detailed knowledge of the functional relationship between the parameters being adjusted (optimized) and the loss function being minimized that is a requirement in gradient-based algorithms. Such a relationship can be notoriously difficult to develop in problem areas such as nonlinear feedback controller design. Further, in areas such as Monte Carlo optimization

Manuscript received October 10, 1993; revised January 2, 1995 and December 23, 1995. This work was supported by a JHU/APL Independent Research and Development Grant.

The author is with the Applied Physics Laboratory, Johns Hopkins University, Laurel, MD 20723-6099 USA (e-mail: Daniel.Chin@jhuapl.edu).

Publisher Item Identifier S 1083-4419(97)00150-7.

or recursive statistical parameter estimation, there may be large computational savings in calculating the loss function relative to that required in calculating gradients. Because of the inherent randomness in the data and search algorithms here, all algorithms will be viewed from the perspective of stochastic approximation (SA).

Let us elaborate on the distinction between algorithms based on direct gradient information or measurements and algorithms based on gradient approximation from measurements of the loss function. Examples of the former include Robbins–Monro SA [13], steepest descent and Newton–Raphson [1, ch. 8] neural network backpropagation [14], perturbation analysis [8], and likelihood ratio methods [12]. Examples of the approximation-based methods using loss function measurements are given below but include as an early prototype the Kiefer–Wolfowitz finite-difference SA (FDSA) algorithm [9]. The gradient-based algorithms rely on direct measurements of the gradient of the loss function with respect to the parameters being optimized. These measurements typically yield an estimate of the gradient since the underlying data generally include noise. Because it is not usually the case that one would obtain direct measurements of the gradient (with or without noise) naturally in the course of operating or simulating a system, one must have knowledge of the underlying system input-output relationships in order to calculate the gradient estimate (using the chain rule) from basic system output measurements. In contrast, the approaches based on gradient approximation use only the sample values of the loss function, which can be observed without knowledge of the system input–output relationships.

The purpose of this paper is to compare three different types of SA algorithms of the Kiefer–Wolfowitz form. These are iterative procedures to find the minimum, θ^*, of a loss function $L\colon R^p \rightarrow R^1$, $p \geq 1$, when the function $L(\cdot)$ can be observed only in the presence of (unknown) noise and its gradient, $g(\theta)$, is unknown and has to be approximated from the measurements on $L(\cdot)$, where

$$g(\theta) \equiv \frac{\partial L(\theta)}{\partial \theta}. \tag{1.1}$$

We are interested in finding the minimizing θ^* such that $g(\theta^*) = 0$. The SA iterative procedure usually has the following form

$$\theta_{k+1} = \theta_k - a_k g_k(\theta_k) \tag{1.2}$$

where the subscripts k and $k+1$ represent the number of iterations, a_k is a positive scalar gain and $g_k(\cdot)$ will be approximated.

FDSA uses a finite difference equation to approximate each individual element of the gradient. Let $\tilde{\theta}_k$ and $\tilde{g}(\cdot)$ denote the FDSA approximated estimates and gradients, u_ℓ be a unit vector the direction of the ℓth coordinate in R^p, and $\tilde{y}_k$ denote the FDSA measurement. For a given positive scale c_k, the two-sided FDSA gradient uses the measurements $\tilde{y}_k^{(\ell\pm)}$ defined by

$$\tilde{y}_k^{(\ell+)} = L(\tilde{\theta}_k + c_k u_\ell) + \epsilon_k^{(\ell+)}$$

and

$$\tilde{y}_k^{(\ell-)} = L(\tilde{\theta}_k - c_k u_\ell) + \epsilon_k^{(\ell-)} \tag{1.3}$$

which are evaluated at the design levels $\tilde{\theta}_k \pm c_k u_\ell$, where $\epsilon_k^{(\ell\pm)}$ are the noises associated with the measurements, assumed to satisfy $E(\epsilon_k^{(\ell+)} - \epsilon_k^{(\ell-)} \mid \mathcal{L}) = 0$ a.s. $\forall k$, where $\mathcal{L} \equiv (\tilde{\theta}_0, \ldots, \tilde{\theta}_k)$. Then the two-sided FDSA gradient is

$$\tilde{g}_k(\tilde{\theta}_k) = \frac{1}{2c_k}\begin{bmatrix} \tilde{y}_k^{(1+)} - \tilde{y}_k^{(1-)} \\ \vdots \\ \tilde{y}_k^{(p+)} - \tilde{y}_k^{(p-)} \end{bmatrix}. \tag{1.4}$$

The single-sided FDSA would use $\tilde{y}_k = L(\tilde{\theta}_k) + \epsilon_k$ to replace $\tilde{y}_k^{(\ell-)}\,\forall\,\ell$ and omit the 2 in the denominator. A total of $2p$ measurements of L are required for an approximation of a double-sided $\tilde{g}_k$, or $p+1$ measurements for a single-sided. It is not worthwhile to use the single-sided formula because it has double the noise level (comparing to the double-sided $\tilde{g}_k$ directly) and has a slower convergence rate ($O(c_k)$ vs. $O(c_k^2)$ as indicated in [10, p. 51]). Reference [2] showed that under a set of conditions the FDSA iterations will converge almost surely; [3] and [16] gave conditions for the asymptotic distribution of the errors of the estimates; a comprehensive discussion can be found in [10].

FDSA tends to be very hard to use for large systems. For example, an application using neural networks for process control commonly involves the estimation of hundreds of parameters (weights). Using two measurements for each parameter, FDSA requires hundreds of measurements for a single iteration. For an experiment or operation of a process, it may be extremely costly (time, money, fuel, labor, etc.) and difficult to make the required large number of measurements. Therefore alternatives have been proposed that aim to use fewer measurements to achieve a solution to (1.1). This paper addresses two basic types of alternative algorithms: random-directions SA (RDSA) and simultaneous-perturbation SA (SPSA), both of which can be shown to be more efficient than FDSA.

RDSA and SPSA both use only a pair of measurements to approximate all elements of a gradient at each iteration. Each measurement is evaluated at the design levels $\theta_k \pm c_k\delta_k$, where δ_k represents the random perturbations which are generated from a statistical distribution which satisfies the requirements of RDSA or SPSA. To make it clear, I will use d_k to represent the RDSA perturbations and Δ_k to represent the SPSA perturbations. If $(\mathring{\theta}, \mathring{g}, \mathring{y})$ represent the items used in RDSA and $(\hat{\theta}, \hat{g}, \hat{y})$ represent the items used in SPSA corresponding to the similar items in FDSA, then the RDSA gradient is approximated using

$$\mathring{g}_k(\mathring{\theta}_k) = \frac{1}{2c_k} d_k \left[\mathring{y}_k^{(+)} - \mathring{y}_k^{(-)}\right] \tag{1.5}$$

where $\mathring{y}_k{}^{(\pm)} = L\left(\mathring{\theta}_k \pm c_k d_k\right) + \epsilon_k^{(\pm)}$, and the SPSA gradient is approximated using

$$\hat{g}_k(\theta_k) = \frac{1}{2c_k}\begin{bmatrix} \Delta_{k1}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix}\left(\hat{y}_k^{(+)} - \hat{y}_k^{(-)}\right) \tag{1.6}$$

where $\hat{y}_k^{(\pm)} = L\left(\hat{\theta}_k \pm c_k\Delta_k\right) + \epsilon^{(\pm)}$.

The SPSA algorithm is presented in [17], [18], [19]; the RDSA algorithm is discussed by [10, p. 59], using random perturbations distributed uniformly on a p-dimensional sphere with radius 1 (it should use radius p instead[1]). There are several other kinds of

[1] It is incorrect to use a radius of 1. The proof of Theorem 2.3.6 in ([10, p. 60] mistakenly states that "$\beta_n \rightarrow 0$ wp1 as $n \rightarrow \infty$" where β_n is expected difference between measurement and true $L(\cdot)$ values and n is the number of iterations. The proof of the convergence will hold for radius p with (2.3.19) in [10] adjusted to reflect the different radius.

distributions for the perturbations that have since been used in the RDSA type of SA, such as the Cauchy and Normal (0, 1) applied in [20] for a global optimization problem. An RDSA expressed in kernel functions is discussed in [11]. The extension they made to using only *one* measurement per iteration tended to produce estimates having more bias than those of standard RDSA [10, p. 51].

For some loss functions, simple averaging of gradients may stabilize the convergence process and provide smaller mean-square errors (see Section III). For example, [18] and [19] showed that there are sometimes advantages to using (1.6) with several conditionally (on $\hat{\theta}_k$) independent simultaneous perturbation approximations averaged at each iteration. That is, $g_k(\cdot)$ in (1.2) can be replaced by

$$g_k(\hat{\theta}_k) = q^{-1}\sum_{i=1}^{q} g_k^{(i)}(\hat{\theta}_k), \qquad q \ge 1 \tag{1.7}$$

where each $g_k^{(i)}(\cdot)$ is generated as in (1.6) based on a new pair of measurements, which are conditionally (on $\hat{\theta}_k$) independent. This averaging idea could also benefit RDSA in the same way as for SPSA. The averaging technique used for RDSA in [15] and [20] is a running average (across iterations) and is different from (1.7). The estimation error distribution associated with this type of average has not yet been established.

Although RDSA and SPSA have a superficial similarity, they differ in several critical ways. For example, the gradient approximation forms in the two approaches differ somewhat and the convergence results for the algorithms place different requirements on the distributions of the random perturbations. The convergence theories require the distribution of the SPSA perturbations to have a finite second inverse moment and that of RDSA to have a unit second moment and a finite fourth moment.

The conditions for the almost sure convergence and the asymptotic error distribution of the estimates have been established for FDSA as mentioned before, and for SPSA in [18], [19]. The conditions for the almost sure convergence for RDSA using perturbation distributions $N(0,\,1)$ have been given in [7]. To compare all three algorithms using the mean-square errors computed from asymptotic distributions, this paper will derive the results for RDSA for a *general* class of distributions for the perturbations. Also, numerical studies will be given for a demonstration of the properties of the three algorithms.

The remainder of this paper is organized as follows: Section II discusses the convergence conditions and distribution of the errors in their estimates, Section III compares the accuracy and efficiency of the algorithms, Section IV presents numerical studies, and Section V states some conclusions.

II. Convergence Conditions and Distributions of the Estimation Errors

A. Convergence Conditions

Under certain conditions, all three algorithms, FDSA, SPSA, and RDSA, will converge almost surely if the derivatives of L are equicontiuous and bounded. These conditions have been established as mentioned in Section I. The FDSA algorithm sequence will converge under the following conditions:

C1: $a_k,\ c_k > 0\,\forall\, k; a_k \rightarrow 0, c_k \rightarrow 0$ as $k \rightarrow \infty$, $\sum_{k=0}^{\infty} a_k = \infty$, $\sum_{k=0}^{\infty}(\frac{a_k}{c_k})^2 < \infty$.

C2: $\text{Sup}_k \|\tilde{\theta}_k\| < \infty$ a.s.

C3: θ^* is an asymptotically stable solution of the differential equation $dx(t)/dt = -g(x)$.

C4: Let $D(\theta^*) = [x_0\colon \lim_{t\rightarrow\infty} x(t \mid x_0) = \theta^*]$ where $x(t \mid x_0)$ denotes the solution to the differential equation of C3 based on initial conditions x_0 (i.e., $D(\theta^*)$ is the domain of attraction). There exists a compact $S \subseteq D(\theta^*)$ such that $\tilde{\theta}_k \in S$ infinitely often for almost all sample points.

These conditions are the conditions A1 and A3–A5 of [19] which are also required convergence conditions for SPSA and RDSA. In addition, SPSA and RDSA require their random sequences to satisfy certain conditions to assure convergence. SPSA has $\Delta_{k\ell}\forall\ell$ acting as divisors and requires these quantities to satisfy the conditions in C5 (the condition A2 in [19]),

C5: For some $\alpha_0,\ \alpha_1,\ \alpha_2 > 0$ and $\forall\, k$, $E\epsilon_k^{(\pm)^2} \le \alpha_0$, $EL(\hat{\theta}_k \pm c_k\Delta_k)^2 \le \alpha_1$, and $E(\Delta_{k\ell})^{-2} \le \alpha_2$, for $\ell = 1, 2, \ldots, p$.

RDSA has $d_{k\ell}\forall\,\ell$ acting as multiplier and requires the $d_{k\ell}$ to satisfy the conditions

C6: For some $\alpha_0,\ \alpha_1 > 0$ and $\forall\, k$, $E(d_k d_k^T) = I$, $E(\epsilon_k^{(\pm)^2}) \le \alpha_0$, and $Ed_{k\ell}^2 L(\hat{\theta}_k \pm c_k d_k)^2 \le \alpha_1$, for $\ell = 1, 2, \ldots, p$.

B. Asymptotic Normality of $\overset{\circ}{\theta}_k$

To develop the means of comparing the relative performance of RDSA, SPSA, and FDSA in Section III, this section will discuss the asymptotic distribution for RDSA (to augment the known results for FDSA and SPSA that were mentioned in Section I). The discussion will focus on the basic RDSA algorithm; it can be expanded to an averaged RDSA ($q \ge 1$) with simple modifications.

If we strengthen Condition C6 to

C6$'$: For some $\delta,\ \alpha_0,\ \alpha_1,\ \alpha_2 > 0$ and $\forall\, k$, $E(\epsilon_k^{(\pm)^{2+\delta}}) \le \alpha_0$, $E(d_{k\ell}L(\hat{\theta}_k \pm c_k d_k))^{2+\delta} \le \alpha_1$, and $E(d_{k\ell})^{4+\delta} \le \alpha_2$, for $\ell = 1, 2, \ldots, p$,

and let $H(\cdot)$ denote the Hessian matrix for $L(\theta)$ and σ and τ be such that $E(\epsilon_k^{(+)} - \epsilon_k^{(-)})^2 \rightarrow \sigma^2$ and $Ed_{k\ell}^4 \rightarrow \tau$ (in many practical settings [e.g., i.i.d. noise], these convergences as $k \rightarrow \infty$, can be replaced by equalities for all k). Let $0 < \alpha \le 1$, $\gamma \ge \alpha/6$, $\beta = \alpha - 2\gamma$, $a > 0$, $c > 0$, $a_k = a/k^\alpha$, and $c_k = c/k^\gamma$. Then, the conditions for the asymptotic normality for RDSA can be stated in the following proposition:

Proposition 1: Assume that the above condition C6$'$ and conditions C1–C4 hold and $\beta > 0$. Let P be orthogonal with $PH(\theta)P^T = a^{-1}\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$. Then

$$k^{\beta/2}(\overset{\circ}{\theta}_k - \theta^*) \underset{k\rightarrow\infty}{\overset{\text{dist}}{\rightarrow}} N(\overset{\circ}{\mu}, P\overset{\circ}{M}P^T) \tag{2.1}$$

where $\overset{\circ}{M} = \frac{1}{4}a^2c^{-2}\sigma^2\text{diag}[(2\lambda_1 - \beta_+)^{-1}, \ldots, (2\lambda_p - \beta_+)^{-1}]$ with $\beta_+ = \beta$ if $\alpha = 1$ and $\beta_+ = 0$ If $\alpha < 1$, and

$$\overset{\circ}{\mu} = \begin{cases} (aH(\theta^*) - \beta^+ I/2)^{-1}T & \text{if } \gamma = \alpha/6 \\ 0 & \text{if } \gamma > \alpha/6 \end{cases}$$

the ℓth element of T is $\frac{1}{6}ac^2\tau[L_{\ell\ell\ell}^{(3)}(\theta^*) + 3\sum_{i=1,i\neq\ell}^{p} L_{ii\ell}^{(3)}(\theta^*)]$.

Proof: Let us define $b_k(\overset{\circ}{\theta}_k) \equiv E[\overset{\circ}{g}_k(\overset{\circ}{\theta}_k) - g(\overset{\circ}{\theta}_k) \mid \overset{\circ}{\theta}_k]$. Similar to the proof of the Proposition 2 in [19], it is sufficient to establish the result if conditions (2.2.1), (2.2.2), and (2.2.3) of [5] hold. In the same situation as in SPSA, there is a $\bar{\theta}_k$ on the line segment between $\overset{\circ}{\theta}_k$ and θ^* for large enough k such that

$$E[\overset{\circ}{g}_k(\overset{\circ}{\theta}_k) \mid \overset{\circ}{\theta}_k] = H(\bar{\theta}_k)(\overset{\circ}{\theta}_k - \theta^*) + b_k(\overset{\circ}{\theta}_k) \tag{2.2}$$

under the condition $Ed_k d_k^T = I$. In the notation of [5], we can show that

$$\begin{aligned}&\mathring{\theta}_{k+1} - \theta^* \\ &\quad = (I - k^{-\alpha}\Gamma_k)(\mathring{\theta}_k - \theta^*) + k^{-(\alpha+\beta)/2}\Phi_k V_k + k^{-\alpha-\beta/2}T_k\end{aligned} \tag{2.3}$$

where $\Gamma_k = aH(\bar{\theta}_k)$, $V_k = k^{-\gamma}[\mathring{g}_k(\mathring{\theta}_k) - E(\mathring{g}_k(\mathring{\theta}_k) \mid \mathring{\theta}_k)]$, $\Phi_k = -aI$, and $T_k = -ak^{\beta/2}b_k(\mathring{\theta}_k)$.

Using the techniques in the proof of Proposition 2 of [19] one can show that Γ_k converges almost surely and that

$$\begin{aligned}&k^{2\gamma}b_{k\ell}(\mathring{\theta}_k) - \frac{1}{6}c^2L^{(3)}(\theta^*)E[d_{k\ell}(d_k \otimes d_k \otimes d_k)] \to 0 \qquad \text{a.s.} \\ &E\big(V_k V_k^T \mid \mathring{\theta}_k\big) \to \frac{1}{4}c^{-2}\sigma^2 I \qquad \text{a.s.} \qquad (2.4) \\ &T_{k\ell} \to -\frac{1}{6}ac^2\tau \\ &\quad\times\left(L_{\ell\ell\ell}^{(3)}(\theta^*) + \sum_{i=1,i\neq\ell}^{p}\left[L_{\ell ii}^{(3)}(\theta^*) + L_{i\ell i}^{(3)}(\theta^*) + L_{ii\ell}^{(3)}(\theta^*)\right]\right) \\ &\qquad\qquad \text{a.s. } \ell = 1, 2, \ldots, p,\end{aligned}$$

where $\otimes$ is the Kronecker vector product. These show that conditions (2.2.1) and (2.2.2) of [5] hold. The proof of condition (2.2.3) of [5] is the same as that given for SPSA in [19]. □

The value of β is bounded between 0 and 2/3 in Proposition 1, because it relates to the values of α and γ which are limited. That means the best possible convergence rate for RDSA is $k^{-1/3}$. Likewise, this limitation will apply to SPSA and FDSA for the same reason.

III. Relative Accuracy and Efficiency Among SA Algorithms

A. Relative Accuracy

Relative accuracy is measured by the size of mean-square errors (m.s.e.) computed from the asymptotic distribution for each algorithm, utilizing the same number of measurements for each algorithm. First, we will generalize (2.1) to include the averaged RDSA gradient using (1.5). The averaging asymptotic distribution for RDSA is $N(\mathring{\mu}, q^{-1}P\mathring{M}P^T)$, where q is the number of averaged gradients [see (1.7)]. Under the mild assumption that the m.s.e. for $k^{\beta/2}(\mathring{\theta}_k - \theta^*)$ corresponds to the asymptotic m.s.e. of $k^{\beta/2}(\mathring{\theta}_k - \theta^*)$, we have that the m.s.e. for RDSA satisfies

$$k^\beta E\|\mathring{\theta}_k - \theta^*\|^2 \underset{k\to\infty}{\longrightarrow} \frac{1}{q}\text{tr}P\mathring{M}P^T + \mathring{\mu}^T\mathring{\mu}. \tag{3.1}$$

The corresponding limits for the m.s.e. of FDSA and SPSA (using the asymptotic error distributions stated in [19]) are $\text{tr}P\tilde{M}P^T + \tilde{\mu}^T\tilde{\mu}$ and $q^{-1}(\text{tr}P\hat{M}P^T) + \hat{\mu}^T\hat{\mu}$, where $N(\tilde{\mu}, P\tilde{M}P^T)$ and $N(\hat{\mu}, q^{-1}P\hat{M}P^T)$ are the FDSA and SPSA asymptotic error distributions.

Two properties found in [6] for FDSA can be easily shown to extend to RDSA and SPSA. One is that the optimal value for a used in computing a_k (see notation in Section II) in (1.2) is a function of the second derivatives of L only. The other is that the asymptotic optimal α is 1 and γ is a known function of the highest order nonzero derivatives of L. Based on these results, we may compare RDSA, SPSA, and FDSA using the same value for a and the optimal values for α and γ. This also implies that a single value of β can be chosen so as to be optimal for all three algorithms.

The remaining question is to determine what values for c to use when comparing the three algorithms. As usual, let the symbols $\mathring{c}$, $\hat{c}$, and $\tilde{c}$ represent the values of c used in RDSA, SPSA, and FDSA. Then, (3.1) can be rewritten (factoring $\mathring{c}^{-2}$ out of $\text{tr}P\mathring{M}P^T$ and $\tau^2\mathring{c}^4$ out of $\mathring{\mu}^T\mathring{\mu}$) as

$$E\|\mathring{\theta}_k - \theta^*\|^2 \simeq \left(\frac{n}{2q}\right)^{-\beta}\left(\frac{s}{q}\mathring{c}^{-2} + \tau^2 t_1^2\mathring{c}^4\right) \tag{3.2}$$

where $s = \mathring{c}^2\text{tr}P\mathring{M}P^T$ and $t_1^2 = \tau^{-2}\mathring{c}^{-4}\mathring{\mu}^T\mathring{\mu}$ (both of which do not depend on $\mathring{c}$), n is the number of measurements ($n = 2qk\,\forall\, k$), and "$\simeq$" means "is asymptotic to as $n \to \infty$." Similarly, the m.s.e. formula for SPSA is

$$E\|\hat{\theta}_k - \theta^*\|^2 \simeq \left(\frac{n}{2q}\right)^{-\beta}\left(\frac{\rho^2 s}{q}\hat{c}^{-2} + \xi^4 t_1^2\mathring{c}^4\right) \tag{3.3}$$

where ρ^2 and ξ^2 are the second and second-inverse moments of the distribution of the random perturbations of SPSA. let $t_2^2 = \tilde{c}^{-4}\tilde{\mu}^T\tilde{\mu}$ be the same expression as t_1^2 without the cross terms of the third partial derivatives of L; then the m.s.e. formulas for FDSA is

$$E\|\tilde{\theta}_{k'} - \theta^*\|^2 \simeq \left(\frac{n}{2p}\right)^{-\beta}\left(s\tilde{c}^{-2} + t_2^2\tilde{c}^4\right) \tag{3.4}$$

where k' is the FDSA iteration ($n = 2pk'$). Then, the accuracy of RDSA, SPSA, and FDSA can be compared using the m.s.e. either with $\mathring{c} = \hat{c} = \tilde{c}$ or with values of $\mathring{c}$, $\hat{c}$, and $\tilde{c}$ that minimize the m.s.e. in (3.2), (3.3), and (3.4).

If the loss function L is known then we can calculate the optimal a and the associated s, t_1, and t_2. Therefore, we can easily find the values $\mathring{c}$, $\hat{c}$, and $\tilde{c}$ that minimize the m.s.e. in (3.2), (3.3), and (3.4). Substituting these values back into (3.2), (3.3), and (3.4), we will get the optimal m.s.e. Then, the asymptotic optimal m.s.e. for RDSA, SPSA, and FDSA are, respectively,

$$3n^{-2/3}(\tau s t_1), 3n^{-2/3}(\rho^2\xi^2 s t_1), \quad \text{and} \quad 3n^{-2/3}(p s t_2). \tag{3.5}$$

These optimal m.s.e. formulas are functions of the total number of measurements, the moments of the random perturbations and the elements of the third partial derivatives. The different numbers of averaging at each iteration will make no differences on the asymptotic m.s.e., while the total number of measurements used is same for all three algorithms. [6] made the same conclusion for FDSA (these results do not hold, however, when L is not known; see Section III-C). In (3.5), τ, ρ, and ξ are defined by the types of perturbation used in RDSA or SPSA and the ratio of t_1 to t_2 is determined by the loss function. Among the three algorithms, FDSA is the only one having the dimensional factor p. Unless t_1 is p times larger than t_2, RDSA and SPSA have a smaller m.s.e. than FDSA has. The fourth moment, τ, for the distribution of the random perturbations of RDSA influences the accuracy of RDSA (the large fourth moment is the reason that using the uniform distribution on a sphere with radius of p is less accurate than using $N(0, 1)$, which, in turn, is less accurate than using Bernoulli (±1). In SPSA, ρ^2 and ξ^2 may offset each other and yield the smallest optimal m.s.e. among the algorithms. For example, in a dimension 15 problem, if we are using Bernoulli (±1) for SPSA and $N(0, 1)$ for RDSA, the ratios for the optimal m.s.e. of SPSA,

TABLE I
MEAN VALUE OF 100 REPLICATIONS OF $\|\theta_k - \theta^*\|^2/\|\theta_0 - \theta^*\|^2$ WITH $p = 15$

	NOISY MEASUREMENTS		NOISE-FREE MEASUREMENTS	
	$n = 900$	$n = 3000$	$n = 900$	$n = 3000$
FDSA	1.20	0.77	0.072	0.042
SPSA	0.41	0.14	0.075	0.014
RDSA	1.02	0.33	0.325	0.162

RDSA, and FDSA are 1.0 : 2.1 : 6.1 (assuming $t_1 = t_2$ for the loss function used).[2]

B. Relative Efficiency

Another way of looking at the results is to consider the efficiency of the algorithms, expressed in terms of the amount of data needed to obtain a given accuracy. In Section III-A, if we use $\mathring{n}$, $\hat{n}$, and $\tilde{n}$ to represent the number of measurements needed of RDSA, SPSA, and FDSA to achieve the same level of accuracy, then the optimal m.s.e. for the algorithms are $3\mathring{n}^{-2/3}(\tau s t_1)^{2/3}$, $3\hat{n}^{-2/3}(\rho^2\xi^2 s t_1)^{2/3}$, and $3\tilde{n}^{-2/3}(p s t_2)^{2/3}$. Assuming these optimal m.s.e. are equal to each other, we can solve for the ratio of the number of measurements for these algorithms. For example, if Bernoulli (±1) and $N(0, 1)$ are used for the distributions of the random perturbations of SPSA and RDSA, then $\hat{n} : \mathring{n} : \tilde{n} = 1.0 : 3.0 : p$ (assuming $t_1 = t_2$).

C. Practical Considerations

In general, the details of the loss function L are not known and the optimal values for a and c have to be empirically determined. In practice, we will compare RDSA, SPSA, and FDSA assuming they are using the same values for a and c. In this case, [19] concluded that SPSA generally is more accurate than FDSA, especially for large dimension problems. This conclusion can be extended to the comparison between RDSA, SPSA, and FDSA, except that RDSA may require a larger dimension than the one SPSA requires to perform better than FDSA. The comparison between RDSA and SPSA will use (3.2) and (3.3) directly. The first terms of (3.2) and (3.3) are different by the second moment of SPSA random perturbations which may allow us to choose a distribution of the random perturbations for SPSA to have a smaller m.s.e. using the given a and c. In addition, the averaged gradient for both SPSA and RDSA may reduce the m.s.e. for some loss functions using the selected a and c.

IV. NUMERICAL STUDIES

This section presents the results of studies that compare all three types of algorithms. Related studies have been reported in other papers. The asymptotic error distributions of the SPSA estimates were demonstrated in [4]. The comparison of the m.s.e. of SPSA and FDSA using fixed values of a and c in [4] and [19] have shown that SPSA has a smaller m.s.e. value, as predicted in Section III. The RDSA using a uniform distribution on a unit p-dimensional sphere is studied in [4] and has very large m.s.e. in all of the cases studied (recall from Section I that, although this distribution is considered in [10] it does not satisfy convergence conditions).

The studies in this section will focus on an empirical comparison of the m.s.e. for RDSA, SPSA, and FDSA at the optimal asymptotic condition for each algorithm. The loss function used in these studies is based on the commonly used squared norm, plus an exponential penalty function that penalizes large positive components of θ. Such a penalty may be suitable for many problems, such as 1) optimization of traffic flow in an urban network while strongly penalizing excessive congestion on any one thoroughfare and 2) controlling the operational cost in a electronic system using superconductivity, because it becomes extremely inefficient above a certain temperature. The loss function is as follows

$$L(\theta) = \|\theta\|^2 + \sum_{\ell+1}^{p} e^{\theta(\ell)/p} \tag{4.1}$$

where $\theta(\ell)$ denotes the ℓth component of θ. This loss function has nonzero direct third-partials of L, has zero values in the cross third-partial terms (implying that $t_1 = t_2$) and allows us to compare the theoretical results between SPSA, RDSA, and FDSA.

Table I was generated using Bernoulli (± 1) for SPSA and $N(0,1)$ for RDSA, and using the optimal values for $\alpha(= 1)$, $\beta(= 1/6)$, $a(= 0.5)$, and c. Where noisy measurement are obtained, we use $\sigma^2 = 7.5$. The optimal values of c, computed from (3.2), (3.3), and (3.4) are $\hat{c} = \tilde{c} = 25.22$, and $\mathring{c} = 17.49$. These same values of the c's are used for the noise-free case, since the optimal values for c are not defined in that case. In the table, the initial value, θ_0, is set to -0.01 in every component ($-0.033\,259$ for every component is the true minimizing parameter, θ^*), and the estimates of θ are generated using both 900 and 3000 measurements. Each tabulated value is the average over 100 replications of the algorithm, each replication using the number of measurements as indicated by $n = 900$ and $n = 3000$. The averaging technique discussed in (1.7) was not used, because it does not affect the optimal asymptotic results.

Table I shows that SPSA has the smallest relative m.s.e. among the three SA algorithms; the relative m.s.e. value for RDSA is smaller than that of FDSA for the noisy-measurement case, and it is just the opposite for the noise-free case. In the noisy-measurement case, all three algorithms diverge from the truth in early iterations because of imprecise gradient approximations (noise in every measurement); asymptotically they all converge to θ^* without much of a problem. Among the three algorithms, SPSA is the fastest one to come back into the initial circle (the p-dimensional circle with center at true θ^* and radius $\|\theta_0 - \theta^*\|$). At 900 measurements, the m.s.e. of SPSA has already been reduced to 41% of its initial value, while RDSA and FDSA are still on the boundary or outside of the initial circle.

In Table I at $n = 3000$, the ratios for the m.s.e. of SPSA, RDSA, and FDSA are 1.0 : 2.4 : 5.6 for the noisy-measurement case and 1.0 : 11.5 : 3.0 for the noise-free case. The predicted m.s.e. ratios are 1.0 : 2.1 : 6.1 (see Section III). The reason for the inexact matching appears to be that the estimates at 3000 measurements have not

[2] The m.s.e. ratios of SPSA (or RDSA) to FDSA will change when $t_1 \neq t_2$. The change direction depends on the loss function, in particular on the third partials of L.

reached asymptotic for all of the algorithms, especially for RDSA in the noise-free case. For the noise-free case, the ratios among the algorithms are 1.0 : 2.1 : 4.8 using 9000 measurements with only ten replications. Given the variability of the ratios, 100 replications is a fairly small sample. Nevertheless, the results of Table I tend to confirm the theoretical considerations discussed above.

In another study where the initial values of θ were chosen farther away from θ^*, the results favor FDSA over SPSA or RDSA in small sample cases (this is not generally true; see [4] and [19]). Using the same setup for the runs tabulated in Table I except that the initial values are set to a value of 1 for every component of θ, at 3000 measurements the ratios for the m.s.e. of SPSA, RDSA, and FDSA are 1.0 : 1.2 : 0.5. But, at 30 000 measurements for ten replications the ratios are 1.0 : 2.0 : 4.0, approaching the ratios predicted in Section III.

V. Conclusion

Among the three SA algorithms considered for the gradient-free stochastic optimization (i.e., Kiefer–Wolfowitz type of problem), we have shown that SPSA is generally the best one to use. RDSA and SPSA may have smaller mean-square errors than FDSA has in large dimensional problems (see Section III). Theoretically, there are no differences in performance between RDSA and SPSA when they both use the Bernoulli (± 1) distribution for their random perturbations (although such a distribution had never previously been reported for RDSA). For fixed a and c, the RDSA m.s.e. is bounded (below) by a variance term [from (3.2)] for all choices of random perturbations, but the SPSA m.s.e. is not.

The numerical studies in Section IV and in other papers mentioned in that section reinforce the theoretical results. These studies compared RDSA, SPSA, and FDSA using various loss functions (squared norm with penalty function in (4.1) here, log-likelihood (in [4] and [19]), Euclidean norm square (in [4]), and others), and using perturbations which have the distributions $N(0,1)$, uniform on a sphere, and Cauchy for RDSA and Bernoulli (± 1) for SPSA. In all of the studies, SPSA had the smallest observed m.s.e. Also, the observed ratios of the m.s.e. are usually close to the predicted ones when the sample sizes are reasonably large. In summary, therefore, we have found that SPSA is the preferable algorithm to use in both theory and practice.

Acknowledgment

The author is grateful to J. Maryak, Applied Physics Laboratory, Johns Hopkins University, for reviewing this paper.

References

[1] M. Bazaraa and C. M. Shetty, *Nonlinear Programming*. New York: Wiley, 1979.

[2] J. R. Blum, "Multidimensional stochastic approximation methods," *Ann. Math. Stat.*, vol. 25, pp. 737–744, 1954.

[3] D. L. Burkholder, "On a class of stochastic approximation procedures," *Ann. Math. Stat.*, vol. 27, pp. 1044–1059, 1956.

[4] D. C. Chin, "Comparative study of several multivariate stochastic approximation algorithms," in *Proc. Stat. Comput. Section ASA*, Anaheim, CA, 1990, pp. 223–228.

[5] V. Fabian, "On asymptotic normality in stochastic approximation," *Ann. Math. Stat.*, vol. 39, pp. 1327–1332, 1968.

[6] ——, "Stochastic approximation," in *Optimizing Methods in Statistics*, J. J. Rustagi, Ed. New York: Academic, 1971, pp. 439–470.

[7] L. Goldstein, "Minimizing noisy functionals in hilbert space: An extension of the Kiefer-Wolfowitz procedure," *J. Theoretical Probability*, vol. 1, no. 2, pp. 189–204, 1988.

[8] Y. C. Ho and X. B. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*. Boston, MA: Kluwer, 1991.

[9] J. Kiefer and J. Wolfowitz, "Stochastic estimation of a regression function," *Ann. Math. Stat.*, vol. 23, pp. 462–466, 1952.

[10] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Berlin, Germany: Springer-Verlag, 1978.

[11] B. Polyak and A. Tsybakov, "On stochastic approximation with arbitrary noise (the KW-Case)," *Advances Soviet Math.*, vol. 12, pp. 107–113, 1992.

[12] M. I. Reiman and A. Weiss, "Sensitivity analysis via likelihood ratios," in *Proc. 1986 Winter Simulation Conf.*, 1986, pp. 285–289.

[13] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 29, pp. 400–407, 1951.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*. Cambridge, MA: MIT Press, vol. 1, pp. 318–362, 1986.

[15] A. Ruszczynski and W. Syski, "Stochastic approximation method with gradient averaging for unconstrained problems," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 1097–1105, 1983.

[16] J. Sacks, "Asymptotic distribution of stochastic approximation procedures," *Ann. Math. Stat.*, vol. 29, pp. 373–405, 1958.

[17] J. C. Spall, "A stochastic approximation algorithm for generating maximum likelihood parameter estimates," in *Proc. American Control Conf.*, 1987, pp. 1544–1548.

[18] ——, "A stochastic approximation algorithm for large-dimensional systems in the Kiefer-Wolfowitz setting," in *Proc. IEEE Conf. Decision Control*, 1988, pp. 1544–1548.

[19] ——, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automat. Contr.*, vol. 37, no. 3, pp. 332–341, 1992.

[20] M. A. Styblinski and T. S. Tang, "Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing," *Neural Networks*, vol. 3, pp. 467–483, 1990.