Some Experience with Stochastic Approximation Algorithms in Large-Scale Systems

Thomas M. Webster, COMPUTATIONAL ENGINEERING, INC.
14504 Greenview Drive, Suite 500, Laurel, Maryland 20708

KEY WORDS: Stochastic Approximation, Multivariate, Large-Scale Systems

1 Introduction

Stochastic approximation is a widely applicable recursive procedure for finding roots of equations in the presence of noisy observations. Spall (1987, 1988a) has previously demonstrated that an SA procedure based on simultaneous perturbations about the estimate (SPSA) has the potential to be significantly more efficient than the standard multivariate algorithms that are based on finite difference gradient approximations (Kiefer and Wolfowitz (1952), Blum (1954)).

The basic problem consists of finding the root θ^* of the gradient equation

$$g(\theta) = \frac{\partial L(\theta)}{\partial \theta} = 0 \tag{1.1}$$

where $L(\theta)$ is a function of the parameter vector $\theta \epsilon R^p$. $L(\theta)$ is a differentiable function, $L : R^p \rightarrow R^1$, and p represents the dimension of the parameter vector. When L and g are observed directly, there are several methods for finding θ^*, including steepest descent, Newton-Raphson, or scoring. In the case where L is observed (computed) in the presence of noise, an SA algorithm is appropriate.

SA algorithms based on finite difference methods (FDSA) require $2p$ observations of L at each iteration. When FDSA is applied to a higher order problem, say $p = 20$, the computational burden of forming an approximate gradient can be significant. In contrast, the simultaneous perturbation algorithm (SPSA) requires only two observations of L at each iteration regardless of the size of p.

This paper will show that, under certain conditions, the computational savings of SPSA at each iteration will more than offset the added number of iterations generally required to reach convergence. Additionally, this paper will discuss SPSA algorithm enhancement techniques such as gradient averaging and gain sequence selection/modification that further improve the relative performance of SPSA (with respect to FDSA).

2 The SPSA Algorithm

The SPSA methodology was presented in preliminary form by Spall (1987) in which the technique of forming the SA gradient approximation through simultaneous perturbations of the parameter estimate was discussed. Spall (1988a) followed up his initial study by considering the performance of SPSA in the presence of observation noise. This paper presents the SPSA methodology, but it will not repeat the theoretical material from the above-mentioned references.

The SPSA methodology as applied to a standard first order minimization problem is given by

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \hat{g}_k(\hat{\theta}_{k-1}) \tag{2.1}$$

where

k = iteration count
a_k= gain sequence $(a_k = \frac{A}{(k+1)^\alpha})$
$\hat{g}_k$= SPSA gradient approximation.

Note that the difference between (2.1) and the method of steepest descent is that the exact gradient is replaced by the SPSA gradient approximation.

The SPSA gradient $\hat{g}_k$ is given by

$$\hat{g}_k = \begin{bmatrix} \frac{y_k^{(+)} - y_k^{(-)}}{2(\Delta(1)_k)/c_k} \\ \vdots \\ \frac{y_k^{(+)} - y_k^{(-)}}{2(\Delta(p)_k)/c_k} \end{bmatrix} \tag{2.2}$$

$$y_k^{(+)} = L[\hat{\theta}_{k-1} + \Delta_k/c_k] + \varepsilon_k^{(+)} \qquad (2.3)$$

$$y_k^{(-)} = L[\hat{\theta}_{k-1} - \Delta_k/c_k] + \varepsilon_k^{(-)} \qquad (2.4)$$

where

$\hat{\theta}_{k-1}$ = initial/previous parameter estimate
L = observed function value at the perturbed estimate
Δ_k = vector of random (Bernoulli) perturbations to $\hat{\theta}_{k-1}$, $\Delta(i)_k = \pm\delta$
c_k = scale factor for $\Delta_k (c_k = (1+k)^\gamma)$
ε_k = independent (Gaussian) noise $N(0, \sigma^2)$

Each element $\hat{\theta}(i)_{k-1}$ in the parameter vector is perturbed by the corresponding element in the Δ_k vector. The corresponding gradient element $\hat{g}(i)_k$ is then formed as the divided difference of the likelihood function values computed at the perturbed estimates over the size of the interval of perturbation.

Since all elements of $\hat{\theta}_{k-1}$ are perturbed simultaneously, only two function evaluations are required to form $\hat{g}_k$. The formulation of the standard finite difference SA (FDSA) gradient is similar to (2.2), except that each element of the parameter estimate $\hat{\theta}_{k-1}$ is perturbed separately. Thus, $2p$ function evaluations are required in forming the FDSA gradient.

The values of the likelihood function L represent the measurement observation. Observation noise is introduced (simulated) by considering a non-zero scale factor to the Gaussian noise terms ε_k^+, ε_k^-. In the numerical study, values of $\sigma = 0$ (no noise), 20 and 40 are used to evaluate the effect of observation noise on the performance of the SPSA and FDSA algorithms.

SPSA Gradient Averaging. This paper considers using (2.1) with several independent SPSA gradients averaged at each iteration. In particular, $\hat{g}_k$ in (2.2) is replaced by

$$\hat{g}_k = \frac{1}{q}\sum_{i=1}^{q} \hat{g}_k^i \qquad (2.5)$$

where each $\hat{g}_k^i$ is generated as in (2.2). It will be shown in Section 4 that gradient averaging can enhance the performance of the SPSA algorithm, particularly when measurements are observed in the presence of non-zero noise.

3 Defining the Minimization Problem

The problem used to evaluate the performance of the various minimization algorithms is the maximum likelihood estimate (MLE) of the covariance of an indirectly observed random variable. That is, a random variable z, with (known) mean zero and unknown variance Σ, is measured to obtain observations x_i, where $x_i = z + w_i$. The w_i's are random variables with mean zero and known covariance P_i. The equation to be minimized is given by

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{N}[\log \det(\Sigma(\theta) + P_i) \qquad (3.1) \\ &+ x_i^T(\Sigma(\theta) + P_i)^{-1}x_i] \end{aligned}$$

The value p determines the size (number of estimated parameters) of the minimization problem to be evaluated. A 5-dimensional system represents a moderately sized minimization problem that is small enough to support a large number of computer studies without requiring excessive computer system resources. A 20-dimensional system demonstrates the applicability of the SPSA methodology to high order systems.

4 Numerical Studies

Earlier work on the SPSA methodology and its performance relative to the FDSA approach concentrated primarily on the theoretical considerations necessary to justify the applicability of SPSA to the solution of maximum likelihood estimation problems (Spall (1987, 1988a)). The numerical analyses for these papers represent a starting point for the analysis to be discussed here.

In Spall (1988a), the performance of SPSA (relative to FDSA) was evaluated for the *observation plus noise* problem. The technique of gradient averaging was introduced as a means to improve SPSA performance, and contributed significantly to the relative superiority of SPSA over FDSA in handling noisy measurements. The same study is discussed in this paper, but results are presented with a different focus and additional detail.

In all numerical studies performed, the basis for comparing the relative merits of the SPSA and FDSA algorithms is the value of the normalized function $\overline{L}$ for an equivalent number of function evaluations. $\overline{L}$ is defined by

$$\overline{L} = L(\hat{\theta}_k) - L(\theta^*) \qquad (4.1)$$

where

$L(\hat{\theta}_k)$ = function value at the current estimate
$L(\theta^*)$ = function value at the point of convergence (previously determined).

4.1 SPSA vs FDSA Given Noise-Free Observations.

This numerical study attempts to verify the results of Spall (1987) for the 5- and 20-dimensional problems, thus to further demonstrate the superiority of SPSA over FDSA when both algorithms are applied to higher order problems. When the SPSA and FDSA algorithms were applied to the 5-dimensional problem, the a_k and c_k sequences were initialized (identically) with $A = 1500, \alpha = 0.7501$, and $\gamma = 0.25$.

The SPSA algorithm converged in 170 iterations (340 function evaluations), while the FDSA algorithm required 90 iterations (900 function evaluations) to converge. For FDSA, the value of $\overline{L}$ at 340 function evaluations corresponded to an equivalent value of $\overline{L}$ reached by SPSA in 168 function evaluations. These observations lead to the conclusion that SPSA is two to three times as efficient as FDSA for the noise-free 5-dimensional problem.

The comparative study was repeated for the 20-dimensional problem, except that the stepsize coefficient (A) in the a_k gain sequence had to be lowered to 500. (SPSA diverged for higher values of A). Both algorithms were permitted to run for a total of 1000 function evaluations. Because of the reduced magnitude of the gain, neither algorithm converged. The value of $\overline{L}$ for the SPSA algorithm was approximately 0.1, while it was roughly 24 for the FDSA algorithm. Moreover, SPSA required only 180 function evaluations to reach the value of $\overline{L}$ that was reached in 1000 function evaluations for FDSA (an advantage of more than 5 to 1).

These results not only succeed in verifying the superiority of SPSA over FDSA as applied to a noise-free minimization problem (for a given gain sequence), but also suggest that the difference is magnified for higher order problems. However, it also points out that higher order problems may tend to be slower to converge.

4.2 SPSA vs FDSA for *Observation Plus Noise* Problem.

This numerical study considers the relative performance of SPSA vs. FDSA when the likelihood function is observed in the presence of noise. Again, the 5- and 20-dimensional problems are considered, with the focus being placed on demonstrating the applicability of SPSA for higher order problems. This study also considers the techniques of gradient averaging and, to a lesser extent, gain sequence selection as means for accelerating the rate of convergence.

The analysis was accomplished by performing a set of 48 computer simulations (plus a set of 15 additional sensitivity analysis runs) to cover the following set of combinations:

two problem sizes	(5 and 20)
four algorithms	(SPSA-1, SPSA-2, SPSA-4, and FDSA)
three levels of noise	($\sigma = 0, 20, 40$
two a_k gain sequences	($\alpha = .7501$, $\alpha = 1.0$)

The large and moderate scaled problems were evaluated for FDSA and SPSA, where three levels of SPSA gradient averaging were considered.

The noise levels $\sigma = 20$ and $\sigma = 40$ represent low and moderate values of observation noise associated with the likelihood function value. Selecting the $\sigma = 0$ case provides a baseline for comparison with the other cases. Two choices of the a_k gain sequence were considered in this study.

$$\begin{aligned} a_k &= 500/(1+k)^{1.0} \\ a_k &= 400/(1+k)^{0.7501}. \end{aligned}$$

The first a_k gain sequence starts off with a larger initial stepsize, but contains a standard $1/k$ rate of decay. The second gain sequence possesses a smaller initial stepsize, but it also decays more slowly than its counterpart. The value $\alpha = 0.7501$ is selected because of regularity conditions appearing in Spall (1988a).

SPSA vs FDSA in the Presence of Noise. Earlier, it was shown that SPSA consistently outperformed FDSA (for a given gain sequence) when no observation noise was present. The results in Tables 4.1 through 4.4 indicate that the same is true in the presence of observation noise. Of the twelve FDSA results that appear in Table 4.1 through 4.4, there are two instances where FDSA

Noise	Algorithm	$\overline{L}$	SPSA Eval.
	SPSA-1	1.58	145
$\sigma = 0$	SPSA-2	2.02	260
	SPSA-4	3.12	480
	FDSA	6.40	
	SPSA-1	1.60	360
$\sigma = 20$	SPSA-2	2.49	160
	SPSA-4	3.95	800
	FDSA	4.46	
	SPSA-1	11.09	>1200
$\sigma = 40$	SPSA-2	5.15	120
	SPSA-4	6.60	360
	FDSA	10.62	

Table 4.1: SPSA vs FDSA for 5-dimensional Problem and Rapidly Decaying a_k Gain Sequence

Algorithm	$\overline{L}$ (includes sensitivity analysis results)				SPSA Eval.
SPSA-1	0.011				200
SPSA-2	0.410				360
SPSA-4	0.200				650
FDSA	0.900				
SPSA-1	2.160				250
SPSA-2	1.190	2.32	1.850		250
SPSA-4	0.650	0.61	1.810	1.66	225
FDSA	3.180	7.09	8.320		
SPSA-1	7.600				80
SPSA-2	4.650	9.61	4.220	10.53	370
SPSA-4	2.670	6.63	1.476	5.41	115
FDSA	13.430	34.83	22.690	0.914	

Table 4.2: SPSA vs FDSA for 5-dimensional Problem with Slowly Decaying a_k Gain Sequence

Noise	Algorithm	$\overline{L}$	SPSA Eval.
	SPSA-1	28	88
$\sigma = 0$	SPSA-2	22	80
	SPSA-4	22	80
	FDSA	70	
	SPSA-1	40	80
$\sigma = 20$	SPSA-2	34	60
	SPSA-4	33	80
	FDSA	95	
	SPSA-1	64	30
$\sigma = 40$	SPSA-2	55	40
	SPSA-4	54	40
	FDSA	149	

Table 4.3: SPSA vs FDSA for 20-dimensional Problem and Rapidly Decaying a_k Gain Sequence

Noise	Algorithm	$\overline{L}$	SPSA Eval.
	SPSA-1	1.22	120
$\sigma = 0$	SPSA-2	1.69	120
	SPSA-4	3.51	220
	FDSA	39.70	
	SPSA-1	12.20	120
$\sigma = 20$	SPSA-2	7.00	120
	SPSA-4	7.70	220
	FDSA	44.10	
	SPSA-1	116.00	>1200
$\sigma = 40$	SPSA-2	45.00	200
	SPSA-4	30.00	100
	FDSA	91.00	

Table 4.4: SPSA vs FDSA for 20-dimensional Problem and Slowly Decaying a_k Gain Sequence

appears to perform as well as SPSA-1. However, in these cases, along with all the others, there are gradient averaging results that indicate SPSA will always outperform FDSA to a significant degree. Furthermore, there is evidence in Table 4.2 that suggests the apparent FDSA superiority may have been spurious. The sensitivity studies for SPSA and FDSA indicate that a wide range of $\overline{L}$ values can result, especially for FDSA, depending on the nature of the random stream of measurements that are generated.

The extent to which SPSA outperforms FDSA can be seen in the data from the last column of Tables 4.1 through 4.4. This column reveals that the SPSA computer runs generally required far fewer function evaluations to reach the same level of performance as corresponding FDSA runs. Although there is a fair amount of variation in individual runs, particularly for the 5-dimensional problem, averaged results seem to indicate that SPSA is roughly four times more efficient than FDSA on the 5-dimensional problem. For the 20-dimensional problem, SPSA appears to be roughly 10 to 15 times more efficient. Figure 4.1 presents a graphic illustration of the relative performance of SPSA to FDSA for a selected set of computer runs. The values of $\overline{L}$ are plotted at intervals of 40 function evaluations. In this example, the FDSA algorithm outperforms the SPSA-1 algorithm, but both gradient averaging cases yield significantly better results.

SPSA vs FDSA as a Function of Problem Size. The advantage of SPSA over FDSA is greater for the 20-dimensional problem. That is, the FDSA computer runs generally required 10 to 15 times the number of function evaluations to

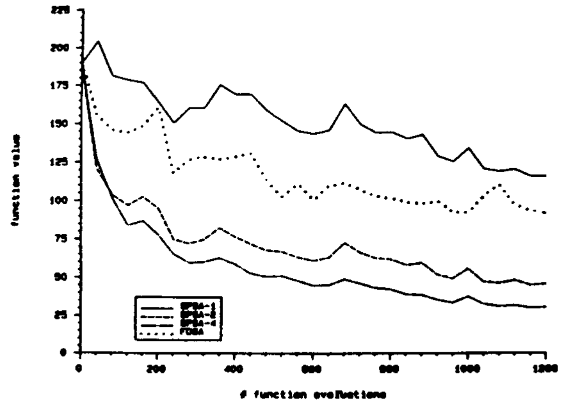

Figure 4.1: Gradient Averaging Improves SPSA Performance ($p = 20, \sigma = 40$)

reach the same values of $\overline{L}$ for the 20-dimensional problem. The SPSA to FDSA advantage for the 5-dimensional problem is only about four-to-one.

Moreover, SPSA is more consistent than FDSA for higher order problems. There is much less variation in the number of function evaluations in the 20-dimensional examples than for the 5-dimensional examples. The reason for this might be that, while there is always a chance for a bad perturbation about θ, the chances of a bad step resulting from 20 perturbations of θ is less than that of getting a bad step from perturbing a 5-dimensional θ vector.

Finally, the results on Tables 4.1 through 4.4 indicate that problem size may also contribute to the rate of convergence for SPSA and FDSA algorithms. Consider, for instance, the problems for which the slowly decaying gain sequence was considered, (i.e., Table 4.2 vs. 4.4). In all SPSA and FDSA examples, the 5-dimensional case clearly outperformed the 20-dimensional case. This result had been observed in the earlier noise-free study and seems to hold true for general classes of minimization problems (Vandergraft (1976)).

Effects of Gradient Averaging. Two factors that contribute to the effectiveness of SPSA gradient averaging for the *observation plus noise* problem are problem size and levels of observed noise. Gradient averaging tended to be effective more often when applied to the 20-dimensional problem than it was for the 5-dimensional problem for varying levels of observation noise. This suggests that gradient averaging should be considered for higher order problems, even when the expected noise contribution might be negligible.

Gradient averaging was also effective for the 5- and 20-dimensional problems for high levels of observation noise ($\sigma = 40$). Figure 4.1 contains a plot of $\overline{L}$ for SPSA-1, SPSA-2, SPSA-4, and FDSA for the 20-dimensional problem where the observation noise was high. As was mentioned earlier, it appears that FDSA outperforms SPSA-1 in this example. But on closer inspection, both curves are rather erratic. It might also be reasonable to conclude that the two algorithms are indistinguishable, given the variation in values of $\overline{L}$ (which is due to the observation noise).

This is not the case for the SPSA-2 and SPSA-4 plots. Not only are the values for $\overline{L}$ significantly lower than SPSA-1 and FDSA at the conclusion of 1200 function evaluations, but the plots are far smoother than the other two (with SPSA-4 being better than SPSA-2). This means the SPSA-4 not only provides a better estimate of the parameter vector θ than the other algorithms, but that one can be more confident of the parameter values that are obtained whenever the SPSA-4 algorithm is terminated. The improved results may be due to the tendency for the observed noise contributions to cancel each other out as the SPSA gradients are averaged. This also suggests that higher levels (than 4) of SPSA gradient averaging might be effective when considering high levels of observation noise.

Effects of Rapidly vs Slowly Decaying Gain Sequence. The analysis of the SPSA and FDSA algorithms in the presence of observation noise was conducted for two choices of the a_k gain sequence. In Spall (1988b), it was shown that the slowly decaying gain sequence gave optimal results for SPSA and FDSA in the noise-free case. This turned out to be the case in the noise free results shown in Tables 4.1 through 4.4. It also turned out to be the case for all noise levels in the 20-dimensional problem. However, this was not the case for the 5-dimensional problem for noise levels of 20 and 40. The reason for this does not lie with the similarity of the gain sequences in question, but rather that both algorithms succeed in reaching a point such that the noise level interferes with any further advance of the algorithm after that point.

Noise Threshold for the 5-dimensional Problem. Spall (1988a) proved that the SPSA algorithm would converge, even in the case of

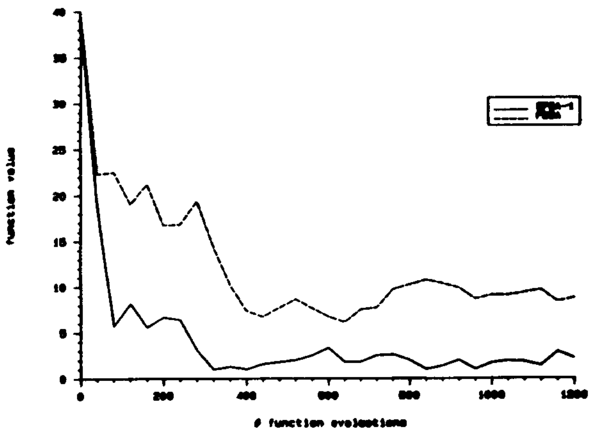

Figure 4.2: Rate of Convergence Slows at "Noise Threshold"

noisy observations. But when many of the computer runs for the 5-dimensional problems were analysed, it was almost always the case that the value of $\overline{L}$ at 1200 function evaluations was not the lowest value of $\overline{L}$ over the course of the iterations. It appears that some sort of threshold is reached early on, and that the algorithm simply bounces around from that point onward. Figure 4.2 provides an illustration of this occurrence for two examples. From the SPSA and FDSA examples shown in Figure 4.2, it appears that both algorithms succeed in reaching a minimum point at around 400 function evaluations. From that point on, they seem to wander about with no apparent movement in the direction of convergence. It is expected that the 20-dimensional problems would exhibit a similar pattern if they had been allowed to continue for more than 1200 function evaluations. From these results, it must be concluded that it may not be feasible to consider SPSA as an algorithm of choice to carry a minimisation problem to convergence given noisy observations (of the function).

5 Conclusion

Spall (1987) presented the argument that the SPSA algorithm was superior to the more standard FDSA algorithm and supported his claim with a numerical study for a low order noise-free minimisation problem. A similar conclusion was reached in this paper for a higher order problem.

The Spall (1988a) argument that maintains the superiority of SPSA over FDSA in the *observation plus noise* problem was also found to be appropriate based on the results of this study. The key to SPSA's superior performance appears to be in the way SPSA with gradient averaging handles very high levels of observation noise. Still, there are some concerns that remain. Spall (1988a) proves that an SPSA algorithm applied to an *observation plus noise* problem will converge asymptotically. Yet, the (preliminary) results from this study suggest that attaining the designed result may not be practical from a numerical standpoint. Moreover, the convergence rate of SPSA (and FDSA) slows down significantly when it is applied to higher order minimisation problems. While this observation is nothing new, dealing with its consequences is a problem that must be overcome if SPSA is to serve as a viable algorithm for solving high order problems.

6 References

Blum, J.R. [1954], "Multidimensional Stochastic Approximation Methods," *Ann. Math. Stat.* , 25, 737-744.

Kiefer, J., and J. Wolfowits [1952], "Stochastic Approximation of a Regressive Function," *Ann. Math. Stat.* , 23, 462-466.

Spall, J.C. [1987], "A Stochastic Approximation Technique for Generating Maximum Likelihood Parameter Estimates," *Proceedings of the 1987 American Control Conference*, 1161-1167.

Spall, J.C. [1988-a], "A Stochastic Approximation Algorithm for Large-Dimensional Systems in the Kiefer-Wolfowits Setting," *27th IEEE Proceedings on Decision and Control*, December, 1988, 1544-1548.

Spall, J.C. [1988-b], "Bayesian Error Isolation for Models of Large-Scale Systems," *IEEE Trans. Auto. Control*, AG-33, 341-347.

Vandergraft, James S. [1976], "A Note on Convergence Rates for Iterative Methods Applied to Ill-Conditioned Linear Systems," Technical Report TR-436, Computer Science Center, University of Maryland, College Park.