Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

DEVELOPMENTS IN STOCHASTIC OPTIMIZATION ALGORITHMS WITH GRADIENT APPROXIMATIONS BASED ON FUNCTION MEASUREMENTS

James C. Spall

The Johns Hopkins University
Applied Physics Laboratory
Laurel, Maryland 20723-6099
(E-mail: james.spall@jhuapl.edu)

ABSTRACT

There has recently been much interest in recursive optimization algorithms that rely on measurements of only the objective function, not requiring measurements of the gradient (or higher derivatives) of the objective function. The algorithms are implemented by forming an approximation to the gradient at each iteration that is based on the function measurements. Such algorithms have the advantage of not requiring detailed modeling information describing the relationship between the parameters to be optimized and the objective function. To properly cope with the noise that generally occurs in the measurements, these algorithms are best placed within a stochastic approximation framework. This paper discusses some of the main contributions to this class of algorithms, beginning in the early 1950s and progressing until now.

Key words: Optimization, stochastic approximation, gradient estimation, recursive procedure

1. 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 an 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 queueing 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 required 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 or recursive statistical parameter estimation, there may be large computational savings in calculating a loss function relative to that required in calculating a gradient. 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 measurements and algorithms based on gradient approximations from measurements of the loss function. Examples of the former include Robbins-Monro SA (Robbins and Monro, 1951), steepest descent and Newton-Raphson (Bazaraa and Shetty, 1979, Chap. 8), neural network back propagation (Rumelhart, et al., 1986), and infinitesimal perturbation analysis (IPA)-based optimization for discrete-event systems (Glasserman, 1991). Examples of approximation-based methods using loss function measurements are given below, but include as an early prototype the Kiefer-Wolfowitz finite-difference SA algorithm (Kiefer and Wolfowitz, 1952). 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

This work was partially supported by the JHU/APL IRAD Program and U.S. Navy Contract N00039-94-C-0001.

gradient since the underlying data generally include added noise. Because it is not usually the case that one would obtain direct measurements of the gradient (with or without added 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 approximations require only conversion of the basic output measurements to sample values of the loss function, which does not require knowledge of the system input-output relationships.

Because of the fundamentally different information needed in implementing these two general types of algorithms, it is difficult to construct meaningful methods of comparison. As a general rule, however, the gradient-based algorithms will be faster to converge than those based on gradient approximations when speed is measured in number of iterations. Intuitively, this is not surprising given the additional information required for the gradient-based algorithms. In particular, the rate of convergence—measured in terms of the deviation of the parameter estimate from the true optimal parameter vector—is typically of order $k^{-1/2}$ for the gradient-based algorithms and of order $k^{-\beta}$ for some $0<\beta<1/2$ for the algorithms based on gradient approximations, where k represents the number of iterations (Fabian, 1971). In practice, of course, many other factors must be considered in determining which algorithm is most appropriate for a given circumstance. Two examples of why this is true are: (1) In cases where it is not possible to obtain reliable knowledge of the system input-output relationships, the gradient-based algorithms may be either infeasible (if no system model is available) or undependable (if a poor system model is used) and (2) The total computational burden to achieve effective convergence depends not only on the number of iterations required, but also on the computation needed per iteration, which is typically greater in gradient-based algorithms. Thus, for both of the reasons above, one cannot say in general that the IPA-based search algorithm (as an example) is superior to a gradient approximation-based algorithm even though the IPA algorithm requires only one system run per iteration while the approximation-based algorithm requires multiple system runs per iteration. As a general rule, however, if direct gradient information is conveniently and reliably available, it is generally to one's advantage to use this information in the optimization process. The focus in this review is the case where such information is not readily available.

In the remainder of this write-up we attempt to trace the historical development of algorithms based on gradient approximations from function measurements, and to discuss the "when, what, and who" for significant original contributions that have been made. These contributions are given in a list in Section 3 after some notation and basic concepts are described in Section 2. The list in Section 3 is likely to be incomplete, and the author welcomes suggestions for corrections or additions. Note, however, that the focus here is on developments that represent significant methodological and/or theoretical advances. In particular, not generally included are developments that are focused on specific applications and/or hardware implementation. Also excluded here are algorithms that are not based on gradient approximations (such as genetic algorithms, evolutionary programming, simulated annealing, random sampling, etc.) and algorithms that require direct gradient measurements (as discussed above). These are worthy topics for another write-up—and another writer! (Two reviews that tend to focus mainly on these other approaches are L'Ecuyer, 1991, and Fu, 1994).

2. BACKGROUND

Consider the problem of minimizing a (scalar) differentiable loss function $L(\theta)$ where $\theta \in \mathbb{R}^p$, $p \geq 1$. A typical example of $L(\theta)$ would be some measure of mean-square error for the output of a process as a function of some design parameters θ. For most cases of practical interest, this is equivalent to finding the minimizing θ^* such that

$$g(\theta^*) \equiv \frac{\partial L}{\partial \theta}\bigg|_{\theta=\theta^*} = 0 \ .$$

It is assumed that measurements of $L(\theta)$ are available at various values of θ (actually, the algorithms have a slightly weaker requirement in that they only need measurements of the difference in two values of the loss function, as opposed measuring the loss functions themselves). These measurements may or may not include added random noise. No direct measurements (either with or without noise) of $g(\theta)$ are assumed available, such as are required in the well-known Robbins-Monro (1951) SA algorithm.

The recursive procedure we consider is in general SA form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \ , \qquad (1)$$

where $\hat{\theta}_k$ represents the estimate of θ at the k^{th} iteration, $a_k > 0$ represents a scalar gain coefficient, and $\hat{g}_k(\hat{\theta}_k)$ represents an approximation of $g(\hat{\theta}_k)$ based on the above-mentioned measurements of $L(\theta)$ at values of θ that are perturbed from the nominal value $\hat{\theta}_k$. Under appropriate conditions, $\hat{\theta}_k$ will converge (in some stochastic sense) to θ^* (see, e.g., Kushner and Clark, 1978).

The most critical aspect in implementing (1) is the gradient approximation $\hat{g}_k(\hat{\theta}_k)$ at each k. This review discusses the three general forms that appear to have attracted the most attention. In forming $\hat{g}_k(\hat{\theta}_k)$ we let $y(\cdot)$ denote a measurement of $L(\cdot)$ at a design level represented by the dot (i.e., $y(\cdot) = L(\cdot) + noise$) and c_k be some (usually small) positive number (in general, a dot as a function argument represents a specific value of θ that we will not specify here in order to avoid excess notation). "One-sided" gradient approximations involve measurements $y(\hat{\theta}_k)$ and $y(\hat{\theta}_k + perturbation)$ for each component of $\hat{g}_k(\hat{\theta}_k)$ while "two-sided" approximations involve the measurements $y(\hat{\theta}_k \pm perturbation)$. The three general forms are:

Finite difference (FD), where each component of $\hat{\theta}_k$ is perturbed one-at-a-time and corresponding measurements $y(\cdot)$ are obtained; each component of $\hat{g}_k(\hat{\theta}_k)$ is formed by differencing corresponding measurements of $y(\cdot)$ and then dividing by the difference interval. This is the "standard" approach to approximating gradient vectors and is motivated directly from the definition of a gradient as a vector of p partial derivatives, each constructed as a limit of the ratio of a change in the function value over a corresponding change in one component of the argument vector. Typically, the i^{th} component of $\hat{g}_k(\hat{\theta}_k)$ (i=1, 2, ..., p) for a two-sided FD approximation is given by

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k e_i) - y(\hat{\theta}_k - c_k e_i)}{2c_k},$$

where e_i denotes a unit vector in the i^{th} direction (an obvious analogue holds for the one-sided version; likewise for the RD and SP forms below).

Random directions (RD), where all components of $\hat{\theta}_k$ are randomly perturbed together in two separate directions to obtain two measurements $y(\cdot)$, and each component of $\hat{g}_k(\hat{\theta}_k)$ is formed from the product of the corresponding component of the perturbation vector times the difference in the two measurements. For two-sided RD, we have

$$\hat{g}_{ki}(\hat{\theta}_k) = d_{ki}\frac{y(\hat{\theta}_k + c_k d_k) - y(\hat{\theta}_k - c_k d_k)}{2c_k},$$

where $d_k = (d_{k1}, ..., d_{kp})^T$ is a vector of user-specified random variables satisfying certain conditions.

Simultaneous perturbation (SP), which also has all elements of $\hat{\theta}_k$ randomly perturbed together to obtain two measurements $y(\cdot)$, but each component of $\hat{g}_k(\hat{\theta}_k)$ is formed from a ratio involving the individual components of the perturbation vector and the difference in the two corresponding measurements. For two-sided SP, we have

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)}{2c_k\Delta_{ki}},$$

where the distribution of the user-specified random perturbations for SP, $\Delta_k = (\Delta_{k1}, ..., \Delta_{kp})^T$, must satisfy conditions different from those of RD (although certain distributions satisfy both sets of conditions).

Algorithm (1) with one of the gradient approximations will be referred to as FDSA, RDSA, or SPSA, as appropriate. Note that the number of loss function measurements $y(\cdot)$ needed in FD grows with p, while with RD and SP only two measurements are needed independent of p. This, of course, provides the potential for RDSA or SPSA to achieve a large savings (over FDSA) in the total number of measurements required to estimate θ when p is large. This potential is only realized if the number of iterations required for effective convergence to θ^* does not increase in a way to cancel the measurement savings per gradient approximation. Some of the references in Section 3 address this issue (especially Spall (1992) and Chin (1993)), demonstrating when this potential can be realized. Although the RD and SP gradient approximation forms have certain similarities, the performance of the RDSA and SPSA algorithms will generally be quite different, as demonstrated in some of the references below. Note, however, that in one important special case the RDSA and SPSA algorithms coincide: namely, when the components of the perturbation vector are symmetric Bernoulli distributed (e.g., $\pm$ 1 with each outcome having probability ½). In general, however, the SPSA formulation has been shown to be more efficient than RDSA, as discussed in Section 3. Theoretically, this follows from the fact that SPSA has a lower asymptotic mean-square error than RDSA for the same number of measurements $y(\cdot)$.

Let us close with definitions of some terms used in Section 3. "Gradient averaging" refers to the averaging of several (say, q) gradient approximations at any given iteration; "gradient smoothing" refers to averaging gradients across iterations (analogous to "momentum" in the neural network parlance).

3. LIST OF KEY DEVELOPMENTS AND ASSOCIATED REFERENCES

References	No. of Function Measurements $y(\cdot)$ per Iteration	Measurement Noise in $y(\cdot)$ Considered?	Comments
Kiefer and Wolfowitz (1952)	2 (p=1)	Yes	First FDSA algorithm. Limited to scalar setting. Convergence in probability shown.
Blum (1954)	p+1	Yes	Multivariate extension of Kiefer and Wolfowitz (1952) FDSA. One-sided finite difference gradient approximation. Shows almost sure (a.s.) convergence.
Sacks (1958)	$2p$	Yes	Shows asymptotic normality of multivariate two-sided FDSA method. Normality result useful in quantifying accuracy of SA estimate.
Ermoliev (1969)	2	Yes	Apparently first paper to consider a special case of RDSA-type algorithm: one-sided RDSA form with uniformly distributed perturbations. Includes analysis of bias in gradient approximation. No convergence analysis or theoretical or numerical comparisons with standard FDSA algorithm.
Fabian (1967, 1971)	$>2p$	Yes	Papers present several different methods for accelerating convergence of FDSA-type algorithms. Methods are based on taking additional measurements to explore loss function surface in greater detail. The 1971 paper discusses stochastic analogue to second-order algorithms of generic Newton-Raphson form (this algorithm uses $O(p^2)$ measurements of $y(\cdot)$).
Polyak and Tsypkin (1973)	Depends on algorithm	Yes	Performs general analysis of FDSA- and RDSA-type algorithms, including a demonstration of a.s. convergence.
Kushner and Clark (1978)	2	Yes	Considers two-sided RDSA form with spherically uniform distributed perturbations. Theoretical analysis shows no improvement over FDSA; this finding appears to result from an error in choice of RD perturbation distribution, as discussed in Chin (1993).

References	No. of Function Measurements $y(\cdot)$ per Iteration	Measurement Noise in $y(\cdot)$ Considered?	Comments
Ermoliev (1983)	2	Yes	Extends Ermoliev (1969) to include constraints; special treatment for convex $L(\cdot)$. No convergence theory or comparative numerical analysis.
Spall (1987)	2	No	Introduces SPSA (two-sided) form. Apparently first paper to consider general perturbation distributions (vs. uniform distributions for RDSA above). Analysis of bias in gradient approximation; numerical study for special case of symmetric Bernoulli (random binary) perturbations shows performance superior to FDSA. No theoretical convergence analysis.
Spall (1988)	$2q$ for any $q \geq 1$	Yes	Extends Spall (1987) to include measurement noise; also proves a.s. convergence of SPSA algorithm; numerical analysis of potential benefits of gradient averaging ($q>1$).
Polyak and Tsybakov (1990, 1992)	1 or 2	Yes	Papers present approach similar to RDSA based on kernel functions. Show a.s. convergence in general noise settings.
Styblinski and Tang (1990); Chin (1994)	$2q$ for any $q \geq 1$	No	Styblinski and Tang uses modified version of one- and two-sided RDSA algorithms for global optimization. Considers Gaussian- and Cauchy-distributed perturbations. Both across-iteration smoothing and gradient averaging ($q>1$) considered. Extensive numerical analysis, including demonstration of superiority of RDSA to simulated annealing; no theoretical convergence analysis. Chin substitutes SPSA for RDSA in algorithm of Styblinski and Tang and numerically illustrates superior performance.
Spall (1992)	$2q$ for any $q \geq 1$	Yes	Extends SPSA theory in Spall (1987, 1988) to include asymptotic normality (à la Sacks (1958)). First paper to show theoretical advantage of two-measurement approaches (SPSA in particular, which includes RDSA with symmetric Bernoulli perturbations as special case) over classical FDSA approach. Also includes theoretical analysis on when gradient averaging ($q>1$) is beneficial, and extensive numerical analysis.

References	No. of Function Measurements $y(\cdot)$ per Iteration	Measurement Noise in $y(\cdot)$ Considered?	Comments
Spall and Cristion (1992, 1994a,b)	$2q$ (for any $q \geq 1$)	Yes	Papers show use of SPSA in closed-loop control problems (where, e.g., the function $L(\cdot)$ changes over time). Allows for optimal control without constructing model of system dynamics. Convergence (a.s.) to optimal controller shown under certain conditions. Across-iteration gradient smoothing considered in 1994a paper. Numerical analysis and comparison with FDSA for closed-loop estimation. Considers polynomial and neural net examples where p ranges from 70 to 400.
Chin (1993)	Depends on algorithm	Yes	Extends theoretical and numerical comparison of SPSA and FDSA in Spall (1992) to include RDSA. Shows theoretical and numerical superiority of SPSA for general perturbation distributions.
Yakowitz (1993)	$2p$	Yes	Alternate global optimization approach (vs. Styblinski and Tang (1990) and Chin (1994)) using two-sided FDSA algorithm. Shows both a.s. convergence and asymptotic normality. Extensions to RDSA and/or SPSA seem feasible.
Cauwenberghs (1993, 1994); Alspector, et al. (1993)	$2q$ for any $q \geq 1$	No	Focus on constant gain ($a_k = a$) implementation of SPSA/RDSA algorithms with symmetric Bernoulli perturbation distribution (SPSA/RDSA equivalent in this case). Both open-loop identification and closed-loop control problems considered. Techniques for hardware implementation in feed-forward and recurrent neural networks.
Spall (1994)	$3q$ for any $q \geq 1$	Yes	Extends SPSA to include second-order effects for purposes of algorithm acceleration (in the spirit of Fabian (1971) above). Estimates both gradient and inverse Hessian at each iteration (with number of measurements independent of p, as indicated at left) to produce an SA analogue of the deterministic Newton-Raphson algorithm.

REFERENCES

Alspector, J., R. Meir B. Yuhas and A. Jayakumar, 1993. A Parallel Gradient Descent Method for Learning in Analog VLSI Neural Networks. *Advances in Neural Information Processing Systems,* 5:836-844. San Mateo, California: Morgan Kaufman.

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

Blum, J. R. 1954. Multidimensional Stochastic Approximation Methods. *Annals of Mathematical Statistics* 25:737-744.

Cauwenberghs, G. 1993. A Fast Stochastic Error-Descent Algorithm for Supervised Learning and Optimization. *Advances in Neural Information Processing Systems*, 5:244-251. San Mateo, California: Morgan Kaufman.

Cauwenberghs, G. 1994. Analog VLSI Autonomous Systems for Learning and Optimization. Ph.D. dissertation, California Institute of Technology.

Chin, D. C. 1993. Performance of Several Stochastic Approximation Algorithms in the Multivariate Kiefer-Wolfowitz Setting. *Proceedings of the 25th Symposium on the Interface Computing Science and Statistics*, 289-295.

Chin, D. C. 1994. A More Efficient Global Optimization Algorithm Based on Styblinski and Tang. *Neural Networks* 7: 573-574.

Ermoliev, Y. 1969. On the Method of Generalized Stochastic Gradients and Quasi-Fejer Sequences. *Cybernetics* 5: 208-220.

Ermoliev, Y. 1983. Stochastic Quasigradient Methods and their Application to System Optimization. *Stochastics* 9:1-36.

Fabian, V. 1967. Stochastic Approximation of Minima with Improved Asymptotic Speed. *Annals of Mathematical Statistics* 38:191-200.

Fabian, V. 1971. Stochastic Approximation. *Optimizing Methods in Statistics,* ed. J. J. Rustigi, 439-470. New York: Academic Press.

Fu, M. C. 1994. Optimization via Simulation: A Review to appear in *Annals of Operations Research.*

Glasserman, P. 1991. *Gradient Estimation via Perturbation Analysis.* Boston: Kluwer.

Kiefer, J. and J. Wolfowitz 1952. Stochastic Estimation of a Regression Function. *Annals of Mathematical Statistics* 23: 462-466.

Kushner, H. J. and D. S. Clark. 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* New York: Springer-Verlag.

L'Ecuyer, P. 1991. An Overview of Derivative Estimation. *Proceedings of the Winter Simulation Conference, ed.* B. L. Nelson, W. D. Kelton, and G. M. Clark, 207-217.

Polyak, B. T. and Y. Z. Tsypkin. 1973. Pseudogradient Adaptation and Training Algorithms. *Automation and Remote Control* 34:377-397.

Polyak, B. T. and A. B. Tsybakov. 1990. Optimal Rates of Convergence for the Global Search Stochastic Optimization. *Problemy Peredachi Informatsii.* 26:45-53; English transl. in *Problems of Information Transmission* 26:126-133, 1990.

Polyak, B. T. and A. B. Tsybakov. 1992. On Stochastic Approximation with Arbitrary Noise (the K-W case). *Advances in Soviet Mathematics* 12:107-113.

Robbins, H. and S. Monro. 1951. A Stochastic Approximation Method. *Annals of Mathematical Statistics* 29:400-407.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing*. 1:318-362. Cambridge: MIT Press.

Sacks, J. 1958. Asymptotic Distributions of Stochastic Approximation Procedures. *Annals of Mathematical Statistics* 26:373-405.

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

Spall, J. C. 1988. A Stochastic Approximation Algorithm for Large-Dimensional Systems in the Kiefer-Wolfowitz Setting. *Proceedings of the IEEE Conference on Decision and Control* 1544-1548.

Spall, J. C. 1992. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control* 37:332-341.

Spall, J. C. 1994. A Second Order Stochastic Approximation Algorithm Using Only Function Measurements. *Proceedings of the IEEE Conference on Decision and Control*, to appear.

Spall, J. C. and J. A. Cristion. 1992. Direct Adaptive Control of Nonlinear Systems Using Neural Networks and Stochastic Approximation. *Proceedings of the IEEE Conference on Decision and Control* 878-883.

Spall, J. C. and J. A. Cristion. 1994a. Nonlinear Adaptive Control Using Neural Networks: Estimation with a Smoothed Simultaneous Perturbation Gradient Approximation. *Statistica Sinica* 4:1-27.

Spall, J. C. and J. A. Cristion. 1994b. Model-Free Control of General Stochastic Discrete-Time Systems. *IEEE Transactions on Automatic Control*, submitted (preliminary form in 1993 *Proceedings of the IEEE Conference on Decision and Control* 2792-2797).

Styblinski, M. A. and T. S. Tang. 1990. Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing. *Neural Networks* 3:467-483.

Yakowitz, S. 1993. A Globally Convergent Stochastic Approximation. *SIAM Journal on Control and Optimization* 31:30-40.

AUTHOR BIOGRAPHY

JAMES C. SPALL has been with the Johns Hopkins University Applied Physics Laboratory since 1983, where he is a project leader for several research efforts focusing on problems in statistical modeling and control. For the year 1990, he received the R. W. Hart Prize as principal investigator of the most outstanding Independent Research and Development project at JHU/APL. In 1991, he was appointed to the Principal Professional Staff of the laboratory. Dr. Spall has published numerous research papers in the areas of statistics and control, including articles on subjects such as Kalman filtering, optimization, parameter estimation, and neural networks. He also served as editor and coauthor of the book *Bayesian Analysis of Time Series and Dynamic Models*. He is a member of IEEE, the American Statistical Association, and Sigma Xi, and a fellow of the engineering honor society Tau Beta Pi.