Proceedings of the Congress on Evolutionary Computation, 6 – 9 July 1999, Washington, DC, pp. 1398–1405

Theoretical Comparisons of Evolutionary Computation and Other Optimization Approaches[1]

James C. Spall, Stacy D. Hill, and David R. Stark

The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723-6099 U.S.A.

E-mail: james.spall@jhuapl.edu
Fax: 240-228-6519
Phone: 240-228-4960

Abstract– This paper is a first step to formal comparisons of several leading optimization algorithms, establishing guidance to practitioners for when to use or not use a particular method. The focus in this paper is four general algorithm forms: random search, simultaneous perturbation stochastic approximation, simulated annealing, and evolutionary computation. We summarize the available theoretical results on rates of convergence for the four algorithm forms and then use the theoretical results to draw some preliminary conclusions on the relative efficiency. Our aim is to sort out some of the competing claims of efficiency and to suggest a structure for comparison that is more general and transferable than the usual problem-specific numerical studies. Much work remains to be done to generalize and extend the results to problems and algorithms of the type frequently seen in practice.

1. Introduction

To address the shortcomings of classical deterministic algorithms, a number of powerful optimization algorithms with embedded randomness have been developed. The population-based methods of evolutionary computation are only one class among many of these available *stochastic* optimization algorithms. Hence, a user facing a challenging optimization problem for which a stochastic optimization method is appropriate meets the daunting task of determining which algorithm is appropriate for a given problem. This choice is made more difficult by the large amount of "hype" and dubious claims that are associated with some popular algorithms. An inappropriate approach may lead to a large waste of resources, both from the view of wasted efforts in implementation and from the view of the resulting suboptimal solution to the optimization problem of interest.

Hence, there is a need for objective analysis of the relative merits and shortcomings of leading approaches to stochastic optimization. This need has certainly been recognized by others, as illustrated in the recent 1998 IEEE International Conference on Evolutionary Computation, where one of the major subject divisions in the conference was devoted to comparing algorithms. Nevertheless, virtually all comparisons have been numerical tests on specific problems. Although sometimes enlightening, such comparisons are severely limited in the *general* insight they provide. On the other end of the spectrum are the "No Free Lunch (NFL) Theorems" (Wolpert and McReady, 1997), which simultaneously considers all possible loss functions and thereby draw conclusions that have limited practical utility since one always has at least *some* knowledge of the nature of the loss function being minimized. Our aim in this preliminary paper is to lay a framework for a *theoretical* comparison of efficiency applicable to a broad class of practical problems where some (incomplete) knowledge is available about the nature of the loss function. We will consider four basic algorithm forms—evolutionary strategies, random search, simulated annealing, and simultaneous perturbation stochastic approximation (SPSA)—in the context of continuous variable optimization. The basic optimization problem corresponds to finding an optimal point θ^*:

$$\theta^* = \arg\min_{\theta\in D} L(\theta) ,$$

where $L(\theta)$ is the loss function to be minimized, D is the domain over which the search will occur, and θ is a p-dimensional (say) vector of parameters. We are mainly interested in the typical case where θ^* is a *unique* global minimum.

[1] A more-complete version of this paper is available upon request. This work was partially supported by the JHU/APL IRAD Program and U.S. Navy contract N00024-98-D-8124.

Although stochastic optimization algorithms other than the four above certainly exist, we are restricting ourselves to the four general forms in order to be able to make tangible progress (note that there are various specific implementations of each of these general algorithm forms). These four algorithms are general-purpose optimizers with powerful capabilities for serious multivariate optimization problems. Further, they have in common the requirement that they only need measurements of the objective function, not requiring the gradient or Hessian of the loss function.

Critical to the approach of this paper will be the known theoretical analysis on the rate of convergence of each of the candidate algorithms. Our approach will be built as much as possible on *existing* theory characterizing the rates of convergence for the algorithms to perform the comparative analysis. There appears to be no previous analysis putting the theoretical results on a common basis for performing an objective comparison. Of course, this approach is limiting in that many algorithms have little—or possibly no—theoretical justification. Nonetheless, it is our expectation that performing a formal theoretical comparison of the chosen algorithms will shed some light on relative performance of other similar algorithms as well, even if the similar algorithms lack the same current level of theoretical justification.

After a brief discussion in Section 2 about the recent "NFL" Theorems, we discuss in Sections 3 through 6 the known convergence rate results on the four algorithms under consideration. Finally, Section 7 offers some preliminary assessment of the relative efficiency based on the theoretical results of the previous sections.

2. No Free Lunch (NFL) Theorems and Their Relationship to Rate of Convergence

Wolpert and Macready (1997) present a formal analysis of search algorithms for optimization. One approach is to compare the performance of algorithms as one runs over the set of optimization problems; the other is to compare performance for a particular problem as one runs over a specified collection of algorithms. The essence of the NFL Theorems is that the expected performance of any pair of optimization algorithms across all possible problems is identical. Of course, these results do not reflect the "usual" types of prior information that might be available to the algorithms and thus may not adequately reflect the performance of algorithms as they are actually applied.

Obviously, for a specific choice of L, one algorithm may be more efficient than another. Within the NFL framework, however, what is of interest is the expected efficiency over all possible loss functions (a finite sum since the domain for θ has a finite number of elements), *not* the efficiency for a specific loss function. According to the NFL Theorem, the expected efficiency over all algorithms is the same, since according to the main result $\sum_L P(\Phi(y_1, y_2, \ldots, y_n) \in S(\theta^*) \mid L, n, a)$ is independent of the algorithm a, where y_k is the the k^{th} value of the loss function, is $\Phi(y_1, y_2, \ldots, y_n)$ a measure of the performance of a after n iterations, and $S(\theta^*)$ is some "small" neighborhood of the optimum θ^*. A self-evident implication of the NFL Theorem is: If a_1 has a faster rate of convergence than a_2 for one set of problems, then there is a set of problems for which a_2 has a faster rate of convergence than a_1.

All the foregoing assumes that the domain and range for $L(\theta)$ are countable (and finite) sets. Hence, NFL results, though interesting, have unclear implications for optimization in continuous parameter domains. The NFL Theorem implies that an algorithm, a_1 say, is uniformly more efficient than a_2 only if a_1 uses (even implicitly) more information about the structure of loss functions than a_2. For example, simulated annealing uses only the current and most recent values of the loss function, and cannot be expected to be more efficient than algorithms that rely on more detailed knowledge of functions (such as their local shape).

3. Simple Global Random Search

We first establish a rate of convergence result for the simplest random search method where we repeatedly sample over the domain of interest, $D \subseteq R^p$. This can be done in "batch" or recursive form by simply laying down a number of points in D and taking as our estimate of θ^* that value of θ yielding the lowest L value

It is well known that the random search algorithm above will converge in some stochastic sense under modest conditions (e.g., Solis and Wets, 1981). A typical convergence theorem is of the form (proof in Spall, 1999):

Theorem 3.1. Suppose that θ^* is the unique minimizer of L on the domain D and that $L(\theta^*) > -\infty$. Suppose further that for any $\varepsilon > 0$ and $\forall$ k, there exists a $\delta(\varepsilon) > 0$ such that

$$P\left(\theta_{new}(k): L(\theta_{new}(k)) < L(\theta^*) + \varepsilon\right) \geq \delta(\varepsilon) \qquad (3.1)$$

Then, for the random search algorithm, $\hat{\theta}_k \to \theta^*$ a.s. as $k \to \infty$.

While the above theorem establishes convergence of the simple random search algorithm, it is also of interest to examine the *rate* of convergence. The rate is intended to tell the analyst how close $\hat{\theta}_k$ is likely to be to θ^* for a given cost of search. The cost of search here will be expressed in terms of number of loss function evaluations. Knowledge of the rate is critical in practical applications as simply knowing that an algorithm will eventually converge begs the question of whether the algorithm will yield a practically acceptable solution in any reasonable period. To evaluate the rate, let us specify a "satisfactory region" $S(\theta^*)$ representing some neighborhood of θ^* providing acceptable accuracy in our solution (e.g., $S(\theta^*)$ might represent a hypercube about θ^* with the length of each side representing a tolerable error in each coordinate of θ). An expression related to the rate of convergence of Algorithm A is then given by

$$P(\hat{\theta}_k \in S(\theta^*)) = 1 - [1 - P(\theta_{\text{new}}(k) \in S(\theta^*))]^k \qquad (3.2)$$

We will use this expression in Section 7 to derive a convenient formula for comparison of efficiency with other algorithms.

4. Simultaneous Perturbation Stochastic Approximation (SPSA)

The next algorithm we consider is SPSA. This algorithm is designed for continuous variable optimization problems. Unlike the other algorithms here, SPSA is fundamentally oriented to the case of *noisy* function measurements and most of the theory is in that framework. This will make for a difficult comparison with the other algorithms, but Section 7 will attempt a comparison nonetheless. The SPSA algorithm works by iterating from an initial guess of the optimal θ, where the iteration process depends on a highly efficient "simultaneous perturbation" approximation to the gradient $g(\theta) \equiv \partial L(\theta)/\partial\theta$.

Assume that measurements $y(\theta)$ of the loss function are available at any value of θ:

$$y(\theta) = L(\theta) + noise\ .$$

For example, in a Monte Carlo simulation-based optimization context, $L(\theta)$ may represent the mean response with input parameters θ, and $y(\theta)$ may represent the outcome of one simulation experiment at . θ. In some problems, exact loss function measurements will be available; this corresponds to the *noise* = 0 setting (and in the simulation example, would correspond to a deterministic—non-Monte Carlo—simulation). Note that no direct measurements (with or without noise) of the gradient are assumed available.

It is assumed that $L(\theta)$ is a differentiable function of θ and that the minimum point θ^*corresponds to a zero point of the gradient, i.e.,

$$g(\theta^*) = \left.\frac{\partial L(\theta)}{\partial\theta}\right|_{\theta=\theta^*} = 0. \qquad (4.1)$$

In cases where more than one point satisfies (4.1), then the algorithm may only converge to a local minimum (as a consequence of the basic recursive form of the algorithm, there is generally not a risk of converging to a maximum or saddlepoint of $L(\theta)$, i.e., to nonminimum points where $g(\theta)$ may equal zero). Extensions of SPSA to global optimization are discussed in Chin (1994) and Maryak and Chin (1999), but we will not discuss these ideas further due to the relative immaturity of the theoretical foundation.

The SPSA procedure is in the general recursive SA form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \qquad (4.2)$$

where $\hat{g}_k(\hat{\theta}_k)$ is the estimate of the gradient $g(\theta) \equiv \partial L/\partial\theta$ at the iterate $\hat{\theta}_k$ based on the above-mentioned measurements of the loss function and $a_k > 0$ is a "gain" sequence. This iterate can be shown to converge under reasonable conditions (e.g., Spall, 1992; Dippon and Renz, 1997). The core gradient approximation is

$$\hat{g}_k(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)}{2c_k}\begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \cdot \\ \cdot \\ \cdot \\ \Delta_{kp}^{-1} \end{bmatrix}, \qquad (4.3)$$

where c_k is some "small" positive number and the user-generated p-dimensional random perturbation vector, $\Delta_k = (\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp})^T$, contains $\{\Delta_{ki}\}$ that are independent and symmetrically distributed about 0 with finite inverse moments $E(|\Delta_{ki}|^{-1})$ for all k, i. One particular distribution for Δ_{ki} that satisfies these conditions is the symmetric Bernoulli ±1 distribution; two common distributions that do *not* satisfy the conditions (in particular, the critical finite inverse moment condition) are uniform and normal. The essential basis for efficiency of SPSA in multivariate problems is apparent in (4.3), where only two measurements of the loss function are needed to estimate the p-dimensional gradient vector for any p; this contrasts with the standard finite difference method of gradient approximation, which requires $2p$ measurements.

Most relevant to the comparative analysis goals of this paper is the asymptotic distribution of the iterate. This was derived in Spall (1992), with further developments in Chin (1997), Dippon and Renz (1997), and Spall (1998). Essentially, it is known that under appropriate conditions,

$$k^{\beta/2}(\hat{\theta}_k - \theta^*) \xrightarrow{dist} N(\mu, \Sigma) \text{ as } k \to \infty\ , \qquad (4.4)$$

where $\beta > 0$ depends on the choice of gain sequences (a_k and c_k), μ depends on both the Hessian and the third derivatives of $L(\theta)$ at θ^* (note that in general, $\mu \neq 0$ in contrast to many well-known asymptotic normality results in estimation), and Σ depends on the Hessian matrix at θ^* and the variance of the noise in the loss measurements. Given the restrictions on the gain sequences to ensure convergence and asymptotic normality, the fastest allowable value for the rate of convergence of $\hat{\theta}_k$ to θ^* is $k^{-1/3}$. This contrasts with the fastest allowable rate of $k^{-1/2}$ for gradient-based algorithms such as Robbins-Monro SA.

Unfortunately, (4.4) is not directly usable in our comparative studies since the other three algorithms being considered here appear to have convergence rate results only for the case of noise-free loss measurements. The authors are unaware of any comparable asymptotic distribution result for SPSA in the noise-free case (note that it is *not* appropriate to simply let the noise level go to zero in (4.4) in deriving a result for the noise-free case; it is likely that the rate factor β will also change if an asymptotic distribution exists).

5. Simulated Annealing (SAN) Algorithms

The SAN method [(Metropolis et al. (1953) and Kirkpatrick et al. (1983)] was originally developed for optimization over finite sets. The Metropolis method produces a sequence that converges in probability to the set of global minima of the loss function as T_k , the *temperature*, converges to zero. Geman and Hwang (1986) present a SAN algorithm for continuous parameter optimization. Their algorithm produces a *continuous-time* stochastic process—a diffusion process—whose probability distributions converge weakly to the uniform probability distribution concentrated on the (global) minima of the loss function, as the temperature decreases to zero.

More recently, Gelfand and Mitter (1993), obtained discrete-time recursions for Metropolis-type SAN algorithms that, in the limit, optimize continuous parameter loss functions:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\left(g\left(\hat{\theta}_k\right) + \xi_k\right) + b_k W_k\,,\ k > 0, \qquad (5.1)$$

where $\{W_k\}$ is an i.i.d. sequence of p-dimensional, standard Gaussian random vectors such that W_k is independent of $\{\xi_0, \xi_1, \ldots, \xi_{k-1}\}$ for each $k \geq 1$, and a_k and b_k are suitably chosen sequences. In particular, following Yin (1998), assume that $a_k = a/k$, $b_k = (b/(k^\gamma \log(k^{1-\gamma} + B_0))^{1/2}$, where B_0, a, and b are positive constants, $0 < \gamma < 1$. Under the foregoing assumptions, the sequence in (5.1) is a non-stationary Markov chain. Let $p_k(y \mid x)$ denote its one-step transition probability density at the epoch of the k^{th} transition; then

$$\text{Prob}\left(\hat{\theta}_{k+1} \in A \,\middle|\, \hat{\theta}_k = x\right) = \int_A p_k(y|x)dy.$$

Furthermore (Gelfand and Mitter (1993)),

$$p_k(y \mid x) = q_k(x,y)s_k(x,y) + r_k(x)\delta(y-x)$$

were, $\delta(\bullet)$ is the Dirac-delta function, $q_k(x,\bullet)$ is the Gaussian density function with mean x and variance $b_k^2\sigma_k^2(x)$, and $s_k(x,y) = \exp(-[L(y) - L(x)]/T_k\ )$, if $L(y) > L(x)$, otherwise $s_k(x, y) = 1$. The function r_k is a normalizing term, thus $r_k(x) = 1 - \int_{R^p} q_k(x,y)s_k(x,y)dy$, for all $x \in R^p$. The variance term is given by $\sigma_k^2(x) = \max\{1, a_k^\tau\|x\|\ \}$, where τ is fixed, $0 < \tau < 1/4$, and $T_k(x) = b_k^2\sigma_k^2(x)/(2a_k)$.

The function $s_k(x, y)$ is the *acceptance probability* in the usual Metropolis SAN. If the temperature sequence T_k were independent of the state, the sequence $\{\hat{\theta}_k\}$ would reduce to the classical SAN algorithm. (Strictly speaking, to qualify as a classical SAN algorithm, the $\hat{\theta}_k$'s would also be required to lie in a finite set, in which instance $p_k(\ \bullet \mid \bullet\ )$ would be taken to be some one-step transition density on that set.) It is easy to show that for k sufficiently large, (5.1) *is* a Metropolis SAN if almost all $\hat{\theta}_k$ lie in some fixed compact set for all $k > K$, for some $K > 0$. In this sense, (5.1) is equivalent to classical SAN for large enough k.

The sequence $\{\hat{\theta}_k\}$ converges in probability to the global minimum of the loss function. To be specific, suppose that $L(\theta)$ has a unique minimum at θ^* and let $S(\theta^*)$ be a neighborhood of θ^*. Gelfand and Mitter (1993) show that $P\left(\hat{\theta}_k \in S\left(\theta^*\right)\right) \to 1$ *as* $k \to \infty$. Hence, the weak convergence of (5.1) implies that it eventually is a classical SAN.

6. Evolutionary Computation

There are three general approaches in Evolutionary Computation (EC), namely Evolutionary Programming (EP), Evolutionary Strategies (ES) and Genetic Algorithms (GA). All three approaches work with a population of candidate solutions and randomly alter the solutions over a sequence of generations according to evolutionary operations of competitive selection, mutation and sometimes recombination (reproduction).

Global convergence results can be given for a broad class of problems [see for example Eiben, Aarts and Van Hee (1991) and Bäck, Hoffmeister, and Schwefel, (1991)], but the same cannot be said for convergence *rates*. The mathematical complexity of analyzing EC convergence rates is apparently great. Therefore convergence rate results that exist are for certain restricted classes of fitness functions that have some special properties that can be taken advantage of and usually with simplified ECs. Most of the convergence rate results available are for EC algorithms using mutation and selection only, or using recombination and selection only. Both Beyer (1995) and Rudolph (1997a) examine ES algorithms that include selection, mutation and recombination. The function analyzed in both cases is the classic spherical fitness function $L(\theta) = ||\theta||^2$ whose exact solution is of course known. Convergence rates based on the spherical fitness function are somewhat useful, if you assume that the sphere approximates a local basin of attraction. The most practically useful convergence rates for EC algorithms seem to be for the class of strongly convex fitness functions. The following theorem due to Rudolph (1997b) is an extension of a more general result by Rappl (1989).

Theorem 6.1 Let $\hat{\theta}_{k1}$, $\hat{\theta}_{k2}$, ..., $\hat{\theta}_{kN}$ be the sequence of populations of size N generated by some ES at generation k. If $E[L_k^* - L(\theta^*)] < \infty$ and

$$\mathrm{E}[L_{k+1}^* - L(\theta^*) \mid \hat{\theta}_{k1}, \hat{\theta}_{k2}, \ldots, \hat{\theta}_{kN}] \leq c\,[L_k^* - L(\theta^*)] \text{ a. s.}$$

for all $k \geq 0$ where

$L_k^* = \min\{L(\hat{\theta}_{k1}), L(\hat{\theta}_{k2}), \ldots, L(\hat{\theta}_{kN})\}$ and $c \in (0,1)$ is called the **convergence rate**. The ES algorithm converges a.s. geometrically fast to the optimum of the objective function. An algorithm has a **geometric rate of convergence** if and only if $E[L_k^* - L(\theta^*)] = O(r^k)$ with $r \in (0,1)$.

The condition $E[L_{k+1}^* - L(\theta^*) \mid \hat{\theta}_{k1}, \hat{\theta}_{k2}, \ldots, \hat{\theta}_{kN}] \leq c \left[L_k^* - L\left(\theta^*\right)\right]$ implies that the sequence decreases monotonically on average. This condition is needed since in the ES that will be considered below, the fitness value of the best parent in the current generation may be worse than the fitness value of the best parent of the previous generation, but on average this will not be the case. Rudolph (1997b) shows that a (1,λ)-ES using selection and mutation only (where the mutation probability is selected from a uniformly distributed distribution on the unit hyperball), with certain classes of fitness functions, satisfies the assumptions of the theorem. One such class is the (K,Q)-strongly convex functions.

A precise definition of (K,Q)-strongly convex functions may be found in Rudolph (1997b). Every quadratic function is (K,Q)-strongly convex, for example, if the Hessian matrix is positive definite. In the case of twice differentiable functions, fairly simple tests are available for verifying that a function is (K,Q)-strongly convex, from Nemirovsky and Yudin (1983). Other tests are possible that only assume the existence of the gradient $g(\theta)$ [see Göpfert (1973)].

The convergence rate result for a (1,λ)-ES using selection and mutation only on a (K,Q)-strongly convex fitness function is geometric with a rate of convergence

$$c = (1 - M_{\lambda,p}^2 / Q^2)$$

where $M_{\lambda,p}$ is equal to the expected value of the maximum of λ independent identically distributed Beta random variables. The computation of $M_{\lambda,p}$ is apparently very complicated since it depends on both the number of offspring λ and the problem dimension p. Asymptotic approximations are fortunately available. Assuming p is fixed and $\lambda \to \infty$ then

$$M_{\lambda,p} \approx (2\,p^{-1}\log\lambda)^{1/2}.$$

To extend this convergence rate from a (1,λ)–ES to a (N, λ)–ES, note that each of the N parents generate λ / N offspring. Then the convergence rate for the (N, λ) – ES where offspring are only obtained by mutation is $c \leq [\,1 - (2p^{-1}\log\{\lambda/N\}) / Q^2]$ for (K,Q)-strongly convex functions.

7. Comparative Analysis

This section represents our preliminary attempt at interpreting the specific algorithm results in Sections 3 to 6 above and attempting to draw conclusions on what the results are saying regarding the relative performance of the four algorithms. We will address the rate of convergence by focusing on the question:

> *With some high probability* $1 - \rho$ *(*ρ *a small number), how many* $L(\bullet)$ *function evaluations, say n, are needed to achieve a solution lying in some "satisfactory set"* $S(\theta^*)$ *containing* θ^*?

With the random search algorithm in Section 3, we have a closed form solution for use in questions of this sort while with the SPSA, SAN, and EC algorithms of Sections 4 through 6, we must apply the existing asymptotic results, assuming that they apply to the finite-sample question above. For each of the four algorithms, we will outline below an analytical expression useful in addressing the question.

Random Search

We can then use (3.2) to answer the question above. Setting the left-hand side of (3.2) to $1 - \rho$ and supposing that there is a constant sampling probability $P^* = P(\theta_{\text{new}}(k) \in S(\theta^*))\ \forall\ k$, we have

$$n = \frac{\log\rho}{\log(1 - P^*)} \tag{7.1}$$

Although (7.1) may appear benign at first glance, this expression will grow rapidly as p gets large (due to P^* approaching 0). Hence, (7.1) shows the extreme inefficiency of simple random search in higher-dimensional problems as illustrated in Example 7.1 below. Note that while (7.1) is in terms of the iterate $\hat{\theta}_k$, a result related to the rate of convergence for $L(\hat{\theta}_k)$ is given in Pflug (1996, p. 24); this result is in terms of extreme value distributions and also confirms the inefficiency of simple random search algorithms in high-dimensional problems.

Example 7.1. Let $D = [0, 1]^p$ (the p-dimensional hypercube with minimum and maximum θ values of 0 and 1 for each component) and suppose uniform sampling on D is used to generate $\theta_{\text{new}}(k)$ $\forall$ k. We want to guarantee with probability 0.90 that each element of θ is within 0.04 units of the optimal. Let the (unknown) true θ, θ^*, lie in $(0.04, 0.96)^p$. The individual components of θ^* are θ_i^*. Hence,

$$S(\theta^*) = [\theta_1^* - 0.04, \theta_1^* + 0.04] \times [\theta_2^* - 0.04, \theta_2^* + 0.04] \times \ldots \times [\theta_p^* - 0.04, \theta_p^* + 0.04] \subset D$$

and $P^* = 0.08^p$. How many loss evaluations (= number of iterations) are required to ensure that we land in $S(\theta^*)$ with probability of 0.90 (i.e., $\rho = 0.10$)? The table below provides the answer.

p	1	2	5	10
n	28	359	7.0×10^5	2.1×10^{11}

The table illustrates the explosive growth in the number of loss evaluations needed as p increases. ❑

Simultaneous Perturbation Stochastic Approximation

As mentioned in Section 4, there is no known asymptotic normality result in the case of noise-free measurements of $L(\theta)$ (it is *not* the limit of the known asymptotic normality result for the noisy case as the noise level goes to zero). Nonetheless, a *conservative* representation of the rate of convergence is available by assuming a noisy case with small levels of noise. Then we know from (4.4) that the approximate distribution of $\hat{\theta}_k$ with optimal decay rates for the gains a_k and c_k is $N(\theta^* + \mu/k^{1/3}, \Sigma/k^{2/3})$. In principle, then, one can use this distribution to compute the probabilities associated with arbitrary sets $S(\theta^*)$, and these probabilities will be directly a function of k. In practice, this may not be easy and so inequalities such as in Tong (1980, Chap. 2) can be used to provide bounds on $P(\hat{\theta}_k \in S(\theta^*))$ in terms of the marginal probabilities of the $\hat{\theta}_k$ elements. For purposes of insight, we can consider a case where the covariance matrix is diagonal. If $S(\theta^*)$ is a hypercube of the form $[S_1^-, S_1^+] \times [S_2^-, S_2^+] \times \ldots \times [S_p^-, S_1^+]$, then $P(\hat{\theta}_k \in S(\theta^*))$ is a product of the marginal normal probabilities associated with each element of $\hat{\theta}_k$ lying in its respective interval $[S_i^-, S_i^+]$, $i = 1, 2, \ldots, p$. Then we can find the k such that the product of probabilities equals $1 - \rho$. To illustrate more specifically, suppose further that $\Sigma = \sigma^2 I$, the $\mu/k^{1/3}$ term in the mean is negligible, that $S(\theta^*)$ is centered around θ^*, and that $\delta s \equiv s_i^+ - s_i^-$ $\forall$ i. Then for a specified ρ, we seek the n such that $P(\hat{\theta}_k \in S(\theta^*)) = P(\hat{\theta}_{ki} \in [s_i^-, s_i^+])^p = 1 - \rho$. From standard $N(0, 1)$ distribution tables, there exists a displacement factor, say $d(p)$, such that the probability contained within ± $d(p)$ units contains probability amount $(1 - \rho)^{1/p}$; we are interested in the k such that $2d(p)\sigma/k^{1/3} = \delta s$. From the fact that SPSA uses two $L(\theta^*)$ evaluations per iteration, the value n to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is then

$$n = 2\left(\frac{2d(p)\sigma}{\delta s}\right)^3.$$

Unfortunately, the authors are unaware of any convenient analytical form for determining $d(p)$, which would allow a "clean" analytical comparison with the efficiency formula (7.1) above (a closed-form approximation to normal probabilities of intervals is given in Johnson and Kotz, 1970, pp. 55-57, but this approximation does not yield a closed form for $d(p)$). To compare with the random search algorithm, consider Example 7.1 given above and consider a loss function producing a σ such that the same number of function measurements in the $p = 1$ case (28) is used for both random search and SPSA (so $\delta s = 0.08$ and $\sigma = 0.0586$). We then have the following results (for direct comparison with the results in Example 7.1):

p	1	2	5	10
n	28	48	78	106

Relative to the simple random search algorithm, this table illustrates the large gains in efficiency possible in higher-dimensional problems by using SPSA (~10^9 -fold reduction in loss evaluations in the 10-dimensional problem). This gain partly results from the fact that SPSA operates under more restrictive conditions than the random search algorithm (i.e., for formal convergence, SPSA assumes a unimodal, several-times-differentiable loss function) and partly from the fact that SPSA works with *implicit* gradient information via its efficient gradient approximation (of course, to maintain a fair comparison, SPSA, like the other algorithms here, only explicitly uses loss evaluations, no direct gradient information).

Simulated Annealing

Like SPSA, SAN has an asymptotic normality result. Hence the method for characterizing the rate of convergence for SPSA may also be used here. Let $H(\theta^*)$ denote the Hessian of $L(\theta)$ evaluated at θ^* and let I_p denote the $p \times p$ identity matrix. Yin (1998) showed that for $b_k = (b/(k^\gamma \log(k^{1-\gamma} + B_0))^{1/2}$,

$$(\log(k^{1-\gamma} + B_0))^{1/2}(\hat{\theta}_k - \theta^*) \longrightarrow N(0,\Sigma) \text{ in distribution}$$

where $\Sigma H + H^T\Sigma + (b/a)I = 0$.

Again, we shall consider the case where the covariance matrix is diagonal. Assume also that $S(\theta^*)$ is a hypercube of the form $[S_1^-, S_1^+] \times [S_2^-, S_2^+] \times \ldots \times [S_p^-, S_p^+]$ centered around θ^*, and that $\delta s \equiv s_i^+ - s_i^-$, $\forall$ i. The (positive) constant B_0 is assumed small enough that it can be ignored. At each iteration after the first, SAN must evaluate $L(\theta)$ only once per iteration. So the value n to achieve the desired probability for $\hat{\theta}_k \in S(\theta^*)$ is

$$\log n^{1-\gamma} = \left(\frac{2d(p)\sigma}{\delta s}\right)^2$$

Again, to compare this procedure with the other algorithms, consider a loss function producing a σ such that the same number of function measurements in the $p = 1$ case ($n = 28$) is used for both random search and SAN (so $\delta s = 0.08$ and $\sigma = 0.031390$). Also, for convenience, take $\gamma = 1/2$. We then have the following results:

p	1	2	5	10
n	28	31	172	438

This table suggests that SAN is less efficient in high-dimensional problems than SPSA (although it compares favorably with the random search algorithm). The gradient approximations in the two algorithms may explain their relative efficiency. The "Metropolis-type approximation appears to be much farther away from an exact gradient-based algorithm than a finite-difference approximation" [(Gelfand and Mitter (1993, p. 128)]. By contrast, SPSA, recall, utilizes a (highly efficient) finite-difference-like approximation to the gradient.

Evolutionary Computation

As discussed in Section 6, the rate-of-convergence results for EC algorithms are not as well developed as for the other three algorithms of this paper. Theorem 6.1 gives a general bound on $E[L(\hat{\theta}_k) - L(\theta^*)]$ for application of a (N,λ)-ES to strongly convex functions. A more explicit form of the bound is available for the $(1,\lambda)$-ES. Unfortunately, even in the optimistic case of an explicit numerical bound on $E[L(\hat{\theta}_k) - L(\theta^*)]$, we cannot readily translate the bound into a probability calculation for $\hat{\theta}_k \in S(\theta^*)$, as used above (and, conversely, the asymptotic normality result on $\hat{\theta}_k$ for SPSA and SAN cannot be readily translated into one on $L(\hat{\theta}_k)$ since $\partial L/\partial\theta = 0$ at θ^*—see, e.g., Serfling, 1980, pp. 122-124—although Lehmann, 1983, pp. 338-339 suggests a possible means of coping with this problem via higher-order expansions). So, in order to make *some* reasonable comparison, let us suppose that we can associate a set $S(\theta^*)$ with a given deviation from $L(\theta^*)$, i.e., $S(\theta^*) = S(\theta^*, \varepsilon) = \{\theta: L(\hat{\theta}_k) - L(\theta^*) \le \varepsilon\}$ for some prespecified tolerance $\varepsilon > 0$. As presented in Rudolph (1997b), $E[L(\hat{\theta}_k) - L(\theta)] \le c^k$ for sufficiently large k, where c is the convergence rate in Section 6. Then by Markov's inequality,

$$1 - P(\hat{\theta}_k \in S(\theta^*)) \le \frac{E[L(\hat{\theta}_k) - L(\theta^*)]}{\varepsilon} \le \frac{c^k}{\varepsilon}$$

indicating that $P(\hat{\theta}_k \in S(\theta^*))$ is bounded below by the ES bounds mentioned in Section 6.

For EC algorithms there are λ evaluations of the fitness function for each generation k so that $n = \lambda k$, where

$$k = \frac{\log\rho - \log(1/\varepsilon)}{\log\left[1 - \frac{2}{pQ^2}\log(\lambda/N)\right]}$$

To compare the (N,λ)-ES algorithm with random search, SPSA, and SAN algorithms assume that the fitness function is restricted to the (K,Q)-strongly convex functions as is discussed in Section 6. Also let $\lambda = 14$, $N = 7$, $\varepsilon = 8.3$, $Q = 4$, and $\rho = 0.1$. The variables were constrained here so that for $p = 1$, we would have $n = 28$ as is the case for the other algorithms. We then have the following results for comparison with the performance of the other algorithms above.

p	1	2	5	10
n	28	59	150	300

This performance for ES is quite good. However the restriction to strongly convex fitness functions gives the ES in this setting a strong structure not available to the algorithms above. It remains unclear what practical theoretical conclusions can be drawn on a broader class of problems. More advanced sensitivity studies for various λ, N, and Q have not yet been completed. Ideally, in the long run, a more general rate-of-convergence theory will provide a more broadly applicable basis for comparison.

References

Bäck, T., Hoffmeister, F., and Schwefel, H.-P. (1991), "A Survey of Evolution Strategies," in *Proceedings of the Fourth International Conference on Genetic Algorithms* (R.K. Belew and L.B. Booker eds), pp. 2-9.

Beyer, H.-G. (1995), "Toward a Theory of Evolution Strategies: On the Benefits of Sex– the ($\mu/\mu,\lambda$) Theory," *Evolutionary Computation*, vol. 3, pp. 81-111.

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

Chin, D. C. (1997), "Comparative Study of Stochastic Algorithms for System Optimization Based On Gradient Approximations," *IEEE Transactions on Systems, Man, and Cybernetics—B*, vol. 27, pp. 244-249.

Culberson, J. C. (1998), "On the Futility of Blind Search: An Algorithmic View of 'No Free Lunch'," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 109-127.

Dippon, J. and Renz, J. (1997), "Weighted Means in Stochastic Approximation of Minima," *SIAM Journal on Control and Optimization,* vol. 35, pp. 1811-1827.

Eiben, A. E., Aarts, E. H. L., and van Hee, K.M. (1991), "Global Convergence of Genetic Algorithms: A Markov Chain Analysis," in *Parallel Problem Solving from Nature* (H.-P. Schwefel and R. Männer eds.), Springer, Berlin and Heidelberg, pp. 4-12.

Fabian, V. (1968), "On Asymptotic Normality in Stochastic Approximation," *Annals of Mathematical Statistics,* vol. 39, pp. 1327-1332.

Gelfand, S. and Mitter, S.K. (1993), "Metropolis-Type Annealing Algorithms for Global Optimization in R^d," *SIAM Journal of Control and Optimization,* vol. 31, pp. 111-131.

Geman, S. and Hwang, C.-R. (1986), "Diffusions for Global Optimization," *SIAM Journal of Control and Optimization,* vol. 24, pp. 1031-1043.

Göpfert, A (1973), *Mathematische Optimierung in allgemeinen Vektorraumen,*. Leipzig, Teubner.

Johnson, N.I. and Kotz, S. (1970), *Continuous Univariate Distributions—1,* Houghton Mifflin, Boston.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983), "Optimization by Simulated Annealing," *Science,* vol. 220, pp. 671-680.

Lehmann, E.L. (1983), *Theory of Point Estimation,* Wiley, New York.

Maryak, J.L. and Chin, D.C. (1999), "Efficient global optimization using SPSA," *Proceedings of the American Control Conference,* in press.

Metropolis, N., Rosenbluth, A., Rosenbluth, M. Teller, A. and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics,* vol. 21, pp. 1087-1092.

Nemirovsky, A.S. and Yudin, D.B (1983), *Problem Complexity and Method Efficiency in Optimization, Chichester,* Wiley.

Pflug, G. Ch. (1996), *Optimization of Stochastic Models,* Kluwer Academic, Boston.

Rappl, G. (1989), "On linear convergence of a class of random search algorithms," *Zeitschrift für angewandt Mathematik und Mechanik (ZAMM),* vol. 69, pp. 37-45.

Rudolph, G. (1994), "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks,* vol. 5, No. 1, pp. 96-101.

Rudolph, G. (1997a), *Convergence Properties of Evolutionary Algorithms, Kovac,* Hamburg.

Rudolph, G. (1997b), "Convergence Rates of Evolutionary Algorithms for a Class of Convex Objective Functions," *Control and Cybernetics,* vol. 26, pp. 375-390.

Serfling, R.J. (1980), *Approximation Theorems of Mathematical Statistics,* Wiley, New York.

Solis, F. J. and Wets, J. B. (1981), "Minimization by Random Search Techniques," *Mathematics of Operations Research,* vol. 6, pp. 19-30.

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

Spall, J.C. (1998), "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *Proceedings of the IEEE Conference on Decision and Control,* pp. 3872-3879.

Spall, J. C. (1999), *Introduction to Stochastic Search and Optimization,* Wiley, New York, in preparation.

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

Tong, Y.L. (1980), *Probability Inequalities in Multivariate Distributions,* Academic, New York.

Wolpert, D. H. and Macready, W. G. (1997), "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation,* vol. 1, pp. 67-82.

Yin, G. G. (1998), "Rates of Convergence for a Class of Global Stochastic Optimization Algorithms," to appear in *SIAM Journal on Optimization.*