Formal Basis for Algorithm Comparisons in Stochastic Optimization

James C. Spall (james.spall@jhuapl.edu), 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.

Abstract—**This paper establishes a framework for 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 evolution strategies. 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 contribute towards sorting 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**

Keywords—**Stochastic optimization; randomized algorithms; rate of convergence; random search; simultaneous perturbation stochastic approximation (SPSA); simulated annealing; evolutionary computation.**

I. INTRODUCTION

Many powerful optimization algorithms with embedded randomness have been developed. The population-based methods of evolutionary computation, for example, are one class among many of the available *stochastic* optimization algorithms. 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 some dubious claims that have been made about 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, for example, in recent conferences on evolutionary computation, where numerous sessions are devoted to comparing algorithms. Nevertheless, virtually all comparisons have been numerical tests on specific problems. For example, a large fraction of the Schwefel (1995) book is devoted to numerical comparisons. Although sometimes enlightening, such comparisons are severely limited in the *general* insight they provide. Some comparisons for *noisy* evaluations of a simple spherical loss function are given in Arnold (2002, Chap. 6); however, some of the competitors were implemented in nonstandard forms, making the results difficult to interpret for an analyst using a more conventional implementation. Spall (2003) also has a number of comparisons (theoretical and numerical) for the cases of noise-free and noisy loss evaluations. On the other end of the spectrum are the "no free lunch (NFL)" theorems (Wolpert and Macready, 1997), which simultaneously consider 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 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—random search, simultaneous perturbation stochastic approximation (SPSA), simulated annealing (SAN), and the evolution strategy form of evolutionary computation (EC). The basic optimization problem corresponds to finding an optimal point θ^*:

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

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

This work was partially supported by the JHU/APL IRAD Program and U.S. Navy contract N00024-03-D-6606.

Although many stochastic optimization approaches other than the four above 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.

Central 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.

In Sections 2 through 5, we discuss the known convergence rate results on the four algorithm forms under consideration. Section 6 then uses these results to provide a theoretical framework for comparison. We demonstrate these results in analyzing the relative efficiency as the problem dimension increases.

II. Simple Global Random Search

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

Our primary interest is 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 the above simple random search algorithm is then given by

$$P(\hat{\theta}_k \in S(\theta^*)) = 1 - [1 - P(\theta_{\text{new}}(k) \in S(\theta^*)]^k \qquad (2.1)$$

We will use this expression in Section 6 to derive a convenient formula for comparison of efficiency with other algorithms.

III. Simultaneous Perturbation Stochastic Approximation

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 6 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:

$$y(\theta) = L(\theta) + \text{noise} .$$

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.,

$$g(\theta^*) = \left.\frac{\partial L(\theta)}{\partial\theta}\right|_{\theta=\theta^*} = 0. \qquad (3.1)$$

In cases where more than one point satisfies (3.1), there exists theory that ensures that the algorithm will converge to the global minimum (Maryak and Chin, 2001).

The SPSA procedure has the general recursive stochastic approximation (SA) form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) , \qquad (3.2)$$

where $\hat{g}_k(\hat{\theta}_k)$ is the simultaneous perturbation estimate of the gradient $g(\theta)$ at the iterate $\hat{\theta}_k$ based on the above-mentioned measurements of the loss function and $a_k > 0$ is a "gain" sequence. The essential basis for efficiency of SPSA is that 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), Spall (2000), and elsewhere. Essentially, it is known that under appropriate conditions,

$$k^{\beta/2}(\hat{\theta}_k - \theta^*) \xrightarrow{\text{dist}} N(\mu, \Sigma) \text{ as } k \to \infty , \qquad (3.3)$$

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, (3.3) is not directly usable in our comparative studies here since the other algorithms being considered here appear to have formal results for convergence rates only for the case of *noise-free* loss measurements. The authors are unaware of any general asymptotic distribution result for the noise-free case (note that it is *not* appropriate to simply let the noise level go to zero in (3.3) 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). Some partial results, however, are available that are related to the rate of convergence. Gerencsér and Vágó (2001) established that the noise-free SPSA algorithm has a geometric rate of convergence when *constant* gains $a_k = a$ are used. In particular, for functions having bounded third derivatives, they show for sufficiently small a,

$$\limsup_{k\to\infty} \frac{\left\|\hat{\theta}_k - \theta^*\right\|}{\eta^k} = 1 \text{ a.s.}$$

for some $0 < \eta < 1$.

IV. SIMULATED ANNEALING ALGORITHM

The SAN method (Metropolis et al., 1953; Kirkpatrick et al., 1983) was originally developed for optimization over discrete finite sets. The Metropolis SAN 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 at an appropriate rate (Hajek, 1988).

Gelfand and Mitter (1993) present a SAN method for continuous parameter optimization. They obtained discrete-time recursions (which are similar to a stochastic approximation algorithm) for Metropolis-type SAN algorithms that, in the limit, optimize continuous parameter loss functions. Spall (2003, Sect. 8.6) summarizes this connection of SAN to SA in greater detail. Suppose that $\hat{\theta}_k$ is such a Metropolis-type SAN sequence for optimizing L. To define this sequence, let $q_k(x, \cdot)$ be the p-dimensional Gaussian density function with mean x and variance $b_k^2\sigma_k^2(x)I_p$, where $\sigma_k^2(x) = \max\{1, a_k^{\tau}\|x\|\}$, τ is fixed in the range $0 < \tau < 1/4$, and $a_k = a/k$ with $a > 0$. (Observe that $\sup\{\sigma_k^2(x), x \in A\} \to 1$ as $k \to \infty$ for any bounded set A.) Also, let $s_k(x, y) = \exp\left(-\left[L(y) - L(x)\right]/T_k\right)$, if $L(y) > L(x)$, and $s_k(x, y) = 1$ otherwise, where $T_k(x) = b_k^2\sigma_k^2(x)/(2a_k)$. The function $s_k(x, y)$ is the *acceptance probability*, as in the usual Metropolis algorithm.

Let $\{W_k\}$ be an independent identically distributed (i.i.d.) sequence of p-dimensional standard Gaussian random vectors and let the sequence $\xi_0, \xi_1, \ldots$ be defined by setting

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\left(g(\hat{\theta}_k) + \xi_k\right) + b_k W_k \quad \text{a.s.,} \quad k > 0. \tag{4.1}$$

The reason for introducing this form for the recursion is to show that $\hat{\theta}_k$ converges in probability to the set of global minima of L.

Furthermore, like SPSA, SAN has an asymptotic normality result (but unlike SPSA, this result applies in the noise-free case). In particular, following Yin (1999), 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, $0 < \gamma < 1$. Let $H(\theta^*)$ denote the Hessian of $L(\theta)$ evaluated at θ^* and let I_p denote the $p \times p$ identity matrix. Yin (1999) showed that

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

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

V. EVOLUTIONARY COMPUTATION

There are some results on rates of convergence for EC algorithms, but, unfortunately, many of the results are not useful in the practical characterization of the rates. Based on results in Rudolph (1994) and elsewhere, Spall (2003, Sect. 10.5) and Stark and Spall (2003) discuss how it is possible to cast the binary bit-based GA in the framework of Markov chains. Unfortunately, the dimension of the transition matrix grows very rapidly with increases in the number of bits in the representation of the chromosomes in the algorithms and/or with increases in the number of candidate solutions in the population.

One of the more computationally useful convergence rates for EC algorithms applies in a particular class of convex loss functions. The following theorem due to Rudolph (1997) is an application of a more general result by Rappl (1989). The theorem is the starting place for the specific convergence rate result that will be used for comparison in Section 6.

Definition 5.1. An algorithm has a *geometric rate of convergence* if and only if $E[L_k^* - L(\theta^*)] = O(\eta^k)$ where $\eta \in (0, 1)$ defines the convergence rate.

Theorem 5.1 (Rudolph 1997). Let $\bar{\Theta}_k \equiv \{\hat{\theta}_{k1}, \hat{\theta}_{k2}, \ldots, \hat{\theta}_{kN}\}$ be the sequence of populations of size N generated by some ES at generation k ($\hat{\theta}_{ki}$ represents the i^{th} estimate for θ from the population of N elements). If $E[L_k^* - L(\theta^*)] < \infty$ and $E[L_{k+1}^* - L(\theta^*) | \bar{\Theta}_k] \le \eta[L_k^* - L(\theta^*)]$ a.s. for all $k \ge 0$

where $L_k^* = \min\{L(\hat{\theta}_{k1}), L(\hat{\theta}_{k2}), \ldots, L(\hat{\theta}_{kN})\}$, then the ES algorithm converges a.s. geometrically fast to the optimum of the objective function.

The condition $E[L_{k+1}^* - L(\theta^*) | \bar{\Theta}_k] \le \eta[L_k^* - L(\theta^*)]$ implies implies that the sequence decreases monotonically on average. This condition is needed since in the $(1, \lambda)$-ES that will be considered below, the loss value of the best parent in the current generation may be worse than the loss value of the best parent of the previous generation, although on average this will not be the case. Rudolph (1997) shows that a $(1, \lambda)$-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 loss functions, satisfies the assumptions of the theorem. One such class is the (K, q)-*strongly convex* functions:

Definition 5.2. Let $L: \Theta \to \mathbb{R}^1$. Then L is called (K, q)-strongly convex on Θ if for all $x, y \in \Theta$ and for each $\alpha \in [0, 1]$ the inequalities

$$\frac{K}{2}\alpha(1-\alpha)\|x-y\|^2 \le \alpha L(x) + (1-\alpha)L(y) - L(\alpha x + (1-\alpha)y) \le \frac{G}{2}\alpha(1-\alpha)\|x-y\|^2$$

hold with $0 < K \le G \equiv Kq < \infty$.

For example, every quadratic function is (K, q)-strongly convex 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).

The convergence rate result for a $(1, \lambda)$-ES using only selection and mutation on a (K, q)-strongly convex loss function is geometric with a rate of convergence $\eta = (1 - M_{\lambda,p}^2 q^2)$, where $M_{\lambda,p} = E[B_{\lambda:\lambda}] > 0$ and where $B_{\lambda:\lambda}$ denotes the maximum of λ i.i.d. Beta random variables. The computation of $M_{\lambda,p}$ is complicated since it depends on both the number of offspring λ and the problem dimension p. Asymptotic approximations are available. Assuming p is fixed and $\lambda \to \infty$ then $M_{\lambda,p} \approx (2p^{-1}\log\lambda)^{1/2}$. To extend this convergence rate from a $(1, \lambda)$-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

$$\eta \le 1 - \frac{2p^{-1}\log(\lambda/N)}{q^2}$$

for (K, q)-strongly convex functions.

VI. Comparative Analysis

A. Problem Statement and Summary of Efficiency Theory for the Four Algorithms

This section uses the specific algorithm results in Sections 2 to 5 above in drawing conclusions on the relative performance of the four algorithms. There are obviously many ways one can express the rate of convergence, but it is expected that, to the extent they are based on the theory outlined above, the various ways will lead to broadly similar conclusions. We will address the rate of convergence by focusing on the question:

With some high probability $1-\rho$ (ρ a small number), how many $L(\cdot)$ 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 2, we have a closed form solution for use in questions of this sort while with the SPSA, SAN, and ES algorithms of Sections 3 through 5, 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. After we have discussed the analytical expressions, we present a comparative analysis in a simple problem setting for varying p. To maintain a fair comparison, the algorithms here explicitly use only loss evaluations, no direct gradient information.

Random Search. We can use (2.2) to answer the question above. Setting the left-hand side of (2.2) to $1-\rho$ and supposing that there is a constant sampling probability $P^* = P(\theta_{new}(k) \in S(\theta^*))$ for all k, we have

$$n = \frac{\log\rho}{\log(1-P^*)}. \qquad (6.1)$$

Although (6.1) may appear benign at first glance, this expression grows rapidly as p gets large due to P^* approaching 0. (A numerically stable approximation that is useful with small P^* is given in Spall, 2003, p. 62.) Hence, (6.1) shows the extreme inefficiency of simple random search in higher-dimensional problems as illustrated in the study below.

Simultaneous Perturbation Stochastic Approximation. As mentioned in Section 3, there is no known asymptotic normality result in the case of noise-free measurements of $L(\theta)$. 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 (3.4) 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, due to the correlation in Σ, 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, consider a case where the covariance matrix Σ is diagonal. If $S(\theta^*)$ is a hypercube of the form $[s_1^-, s_1^+] \times [s_2^-, s_2^+] \times ... \times [s_p^-, s_p^+]$, 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 $[s_i^-, s_i^+]$, $i = 1, 2, \ldots, p$. Such diagonal covariance matrices arise when the loss function is separable in each of the components of θ. 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 $\delta s \equiv s_i^+ - s_i^-$ for all i (i.e., $s_i^+ - s_i^-$ does not depend on 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 $\pm d(p)$ 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

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

Simulated Annealing. Because SAN, like SPSA, has an asymptotic normality result, the method above for characterizing the rate of convergence for SPSA may also be used here. Again, we shall consider the case where the covariance matrix is diagonal ($\Sigma = \sigma^2 I$). Assume also that $S(\theta^*)$ is a hypercube of the form $[s_1^-, s_1^+] \times [s_2^-, s_2^+] \times ... \times [s_p^-, s_p^+]$ centered around θ^*, and that $\delta s \equiv s_i^+ - s_i^-$, for all 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

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

Evolution Strategy. As discussed in Section 5, the rate-of-convergence results for algorithms of the evolutionary computation type are not as well developed as for the other three algorithms of this paper. Theorem 5.1 gives a general bound on $E[L(\hat{\theta}_k) - L(\theta^*)]$ for application of a (N, λ)-ES form of EC algorithm to (K, q)-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. 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., $S(\theta^*) = \{\theta: L(\hat{\theta}_k) - L(\theta^*) \le \varepsilon\}$ for some prespecified tolerance $\varepsilon > 0$ (note that $S(\theta^*)$ is a function of ε). As presented in Rudolph (1997), $E[L(\hat{\theta}_k) - L(\theta)] \le \eta^k$ for sufficiently large k, where η is the convergence rate in Section 5. Then by Markov's inequality,

$$1 - P(\hat{\theta}_k \in S(\theta^*)) \le \frac{E[L(\hat{\theta}_k) - L(\theta^*)]}{\varepsilon} \le \frac{\eta^k}{\varepsilon}, \qquad (6.2)$$

indicating that $P(\hat{\theta}_k \in S(\theta^*))$ is bounded below by the ES bounds mentioned in Section 5. For EC algorithms in general (and ES in particular), there are λ evaluations of the loss function for each generation k so that $n = \lambda k$, where

$$k = \frac{\log\rho - \log(1/\varepsilon)}{\log\left[1 - \frac{2}{pq^2}\log(\lambda/N)\right]}. \qquad (6.3)$$

B. Application of Convergence Rate Expressions for Varying p

We now apply the results above to demonstrate relative efficiency for varying p for random search, SPSA, SAN and ES. Let $\Theta = [0, 1]^p$ (the p-dimensional hypercube with minimum and maximum θ values of 0 and 1 for each component). We want to guarantee with probability 0.90 that each element of θ is within 0.04 units of the optimal. Let the (unknown) optimal θ, θ^*, lie in $(0.04, 0.96)^p$. The individual components of θ^* are θ_i^*. Hence,

$$\begin{aligned} S(\theta^*) &= [\theta_1^* - 0.04, \theta_1^* + 0.04] \\ &\quad \times [\theta_2^* - 0.04, \theta_2^* + 0.04] \\ &\quad \times ... \times [\theta_p^* - 0.04, \theta_p^* + 0.04] \subset \Theta \end{aligned}$$

Table I is a summary of relative efficiency for the setting above for $p = 2$, 5, and 10; the efficiency is normalized so that all algorithms perform equally at $p = 1$, as described below. The numbers in Table I are the ratios of the number

of loss measurements for the given algorithm over the number for the best algorithm at the specified p; the highlighted values 1.0 indicate the best algorithm for each of the values of p. To establish a fair basis for comparison, we fixed the various parameters in the expressions above (e.g., σ in SPSA and SAN, λ for the ES, etc.) so that the algorithms produced identical efficiency results for $p = 1$ (requiring $n = 28$ measurements to achieve the objective outlined above). We then use these parameter settings as p increases. Of course, in practice, algorithm parameters are typically tuned for each new problem, including changes in p. Rather, they point towards general efficiency trends as a function of problem dimension in the absence of problem-specific tuning.

For the random sampling algorithm, suppose uniform sampling on Θ is used to generate $\theta_{new}(k)$ for all k. Then, $P^* = 0.08^p$. For SPSA, we fix σ such that the same number of function measurements in the $p = 1$ case ($n = 28$) is used for both random search and SPSA (so $\delta s = 0.08$ and $\sigma = 0.0586$). Likewise, for SAN, we fix σ to achieve the same objective (so $\delta s = 0.08$ and $\sigma = 0.031390$). Also, for convenience, take $\gamma = 1/2$. To compare the (N, λ)-ES algorithm with the random search, SPSA, and SAN algorithms, it is assumed that the loss function is restricted to the (K, q)-strongly convex functions discussed in Section 5. 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 have the same n (= 28) as realized for the other algorithms. Table I summarizes the performance comparison results.

Table I illustrates the explosive growth in the relative (and absolute) number of loss evaluations needed as p increases for the random search algorithm. The other algorithms perform more comparably, but there are still some non-negligible differences. For example, at $p = 5$, SAN will take 2.2 times more loss measurements than SPSA to achieve the objective of having $\hat{\theta}_k$ inside $S(\theta^*)$ with probability 0.90. Of course, as p increases, all algorithms take more measurements; the table only shows *relative* numbers of function evaluations (considered more reliable than absolute numbers).

TABLE I
RATIOS OF LOSS MEASUREMENTS NEEDED RELATIVE TO BEST ALGORITHM AT EACH p FOR $1 \le p \le 10$

	$p = 1$	$p = 2$	$p = 5$	$p = 10$
Random Search	1.0	11.6	8970	2.0×10^9
SPSA	1.0	1.5	1.0	1.0
SAN	1.0	1.0	2.2	4.1
ES (from (6.2) and (6.3))	1.0	1.9	1.9	2.8

The performance for ES is quite good. The restriction to strongly convex loss functions (from (6.2) and (6.3)), however, gives the ES in this setting a strong structure not available to the other algorithms. It remains unclear what practical theoretical conclusions can be drawn on a broader class of problems.

REFERENCES

[1] Arnold, D. V. (2002), *Noisy Optimization with Evolution Strategies*, Kluwer, Boston.

[2] 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.

[3] 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.

[4] Gerencsér, L. and Vago, Z. (2001), "The Mathematics of Noise-Free SPSA," *Proceedings of the IEEE Conference on Decision and Control*, 4–7 December 2001, Orlando, FL, pp. 4400–4405.

[5] Hajek, B. (1988), "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, vol. 13, pp. 311–329.

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

[7] Maryak, J. L. and Chin, D. C. (2001), "Global Random Optimization by Simultaneous Perturbation Stochastic Approximation," in *Proceedings of the American Control Conference*, pp. 756–762.

[8] 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.

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

[10] 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.

[11] Rudolph, G. (1994), "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks*, vol. 5, pp. 96–101.

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

[13] Schwefel, H.-P. (1995), *Evolution and Optimum Seeking*, Wiley, New York.

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

[15] Spall, J. C. (2000), "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *IEEE Transactions on Automatic Control*, vol. 45, pp. 1839–1853.

[16] Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization*, Wiley, Hoboken, NJ.

[17] Stark, D. R. and Spall, J. C. (2003), "Rate of Convergence in Evolutionary Computation," in *Proceedings of the American Control Conference*, pp. 1932–1937.

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

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

[20] Yin, G. (1999), "Rates of Convergence for a Class of Global Stochastic Optimization Algorithms," *SIAM Journal on Optimization*, vol. 10, pp. 99–120.