A Convergence Analysis of Unconstrained and Bound Constrained Evolutionary Pattern Search

William E. Hart
Sandia National Laboratories
Applied Mathematics Department
P. O. Box 5800, MS 1110
Albuquerque, NM 87185-1110
Phone: 505-844-2217 Fax: 505-845-7442
wehart@cs.sandia.gov
http://www.cs.sandia.gov/~wehart/

March 31, 1999

Abstract

We present and analyze a class of evolutionary algorithms for unconstrained and bound constrained optimization on $\mathbf{R}^n$: evolutionary pattern search algorithms (EPSAs). EPSAs adaptively modify the step size of the mutation operator in response to the success of previous optimization steps. The design of EPSAs is inspired by recent analyses of pattern search methods. We show that EPSAs can be cast as stochastic pattern search methods, and we use this observation to prove that EPSAs have a probabilistic weak stationary point convergence theory. This work provides the first convergence analysis for a class of evolutionary algorithms that guarantees convergence almost surely to a stationary point of a nonconvex objective function.

1 Introduction

We consider the application of evolutionary algorithms (EAs) to solve unconstrained minimization problems

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & x \in \mathbf{R}^n \end{array} \tag{1}$$

where $f : \mathbf{R}^n \to \mathbf{R}$, as well as bound constrained problems

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & x \in \mathbf{R}^n \\ & l \leq x \leq u, \end{array} \tag{2}$$

where $l, u \in \mathbf{R}^n$, and $l < u$. These problems have been solved using EAs like evolutionary programming (EP), evolutionary strategies (ESs) and genetic algorithms (GAs). In this paper we introduce and analyze evolutionary pattern search algorithms (EPSAs), a class of EAs that can be used to solve problems (1) and (2). EPSAs are *adaptive EAs* like EPs and ESs, which modify the mutation *step length* during optimization. However, EPSAs have an absolute step length that is used to generate a finite set of offsets, while EPs and ESs generate mutations by adding a continuous random variable that is scaled by a step length parameter. Also, EPSAs adaptively modify the mutation step length using a global step length parameter instead of the per-individual step length parameter commonly used by ESs and EPs.

Like most other EAs, EPSAs are direct search methods, methods that do not use either the derivative or an approximation to the derivative of f to perform optimization. EPSAs are, however, distinguished from other EAs by the fact that they can be cast as stochastic pattern search methods. Pattern search methods are direct search methods that examine a pattern of exploratory moves in search of points with lower functional values [36, 25]. We extend the definition of pattern search to provide a correspondence between pattern search and EPSAs. In particular, the exploratory moves in the pattern are allowed to be stochastically selected.

The main result of this paper is to demonstrate how the correspondence between stochastic pattern search and EPSAs can be exploited to provide a convergence theory for EPSAs. We generalize the convergence analysis for pattern search to stochastic pattern search. Formally, this convergence theory guarantees that for an unconstrained continuously differentiable function the sequence of best points found by a stochastic pattern search method, $\{x_k^*\}$, has the property that

$$P\left(\liminf_{k\to\infty} \|g(x_k^*)\| = 0\right) = 1,$$

where $g(x)$ is the gradient of f at x. This means that the set sequences that do not converge has probability zero of occuring, and this convergence is said to occur *almost surely* [9]. Further, for a bound constrained continuously differentiable function a subsequence of $\{x_k^*\}$ converges almost surely to a constrained stationary point (equivalently, a Karush-Kuhn-Tucker point for problem (2) [13]). These results can be extended to prove convergence almost surely on continuous nondifferentiable functions to limit points where the gradient

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

does not exist or where the gradient is not continuous [36, 37]. Preliminary analyses of EPSAs and stochastic pattern search methods are given in Hart [16, 17]. Our present analysis extends the analysis in Hart [16] to (a) allow the mutation step length to be expanded and contracted by an arbitrary rational value, (b) allow a broader set of possible mutation steps, and (c) consider the application of EPSAs to bound constrained problems.

This convergence theory for EPSAs is the first proof of probabilistic convergence to a stationary point for adaptive EAs on nonconvex continuously differentiable functions. Although this convergence theory does not guarantee convergence to a *global* optimum, it does capture an important aspect of EAs: how they refine and focus their search to converge upon a final estimate of the global optimum. Furthermore, Rudolph's recent analysis which shows that ESs fail to converge to a global optimum almost surely [32] suggests that simultaneously ensuring convergence to a global optimum and convergence to a stationary point is not possible.

In the next section we provide more background about EAs and pattern search methods. Section 3 defines stochastic pattern search algorithms, describes their convergence properties and proves the convergence of these methods. Section 4 defines EPSAs and demonstrates their correspondence with stochastic pattern search. Finally, Section 5 discusses these results and points to ongoing work with EPSAs.

2 Background

Let **R**, **Q**, **Z** and **N** denote the sets of real, rational, integer and natural numbers, respectively. All norms will be Euclidean vector norms or the associated operator norm. Let $\Omega = \{x \in \mathbf{R}^n \mid l \leq x \leq u\}$. We define $L_\Omega(y) = \{x \in \Omega \mid f(x) \leq f(y)\}$ and for convenience let $L(y) = L_{\mathbf{R}^n}(y)$. We use the notation $X = [YZ]$ to denote that the matrix X is partitioned into the matrices Y and Z. If y is a vector and A is a matrix, then we define $y \in A$ to mean that y is a column of A.

2.1 Pattern Search

Pattern search methods were first formally defined by Torczon [36], and they provide a general framework for describing a wide variety of direct search methods. This class of direct search methods has been used since the 1950s for optimization in a wide variety of applications [40]. In fact, many of the arguments for using EAs [6, 14, 12, 34] have been used to promote direct search methods like pattern search: they can be used when the function is nondifferentiable, they are effective when the function is highly nonlinear and multimodal, and they are insensitive to noise (or inaccuracies) in the function.

In a general sense, pattern search methods sample the objective function from a given *pattern* of points that represents offsets from the current best point. If there is a better point in this pattern, then it is accepted as the new iterate and the sampling is repeated about it. If not, then the scale of the pattern is reduced (e.g. by halving it), and the function is again sampled about the best point.

Pattern search methods use an *exploratory moves algorithm* to conduct a series of exploratory moves about the current iterate before identifying a new iterate. A *pattern matrix* is used to define the set of possible exploratory moves. The pattern matrix is decomposed into a nonsingular basis matrix $B \in \mathbf{R}^{n\times n}$ and a generating matrix $C_k \in \mathbf{Q}^{n\times p}$, $p > n+1$; the index k denotes the iterations of the pattern search algorithm. Each C_k is constrained to include a subset of core search directions that span $\mathbf{R}^n$. Given a pattern matrix $P_k = BC_k$, there are p possible exploratory moves in $\Delta_k P_k$, where Δ_k is a step-length parameter. Conceptually, the generating matrix defines the pattern of directions that are searched, while the basis matrix rotates and scales the search directions to determine the coordinate system used during the search. We refer the reader to Lewis and Torczon [36, 25, 23] for a complete description of pattern search methods. The framework for stochastic pattern search that we describe in Section 3 is directly related to Lewis and Torczon's framework. A key element of pattern search methods is that the restrictions made on the pattern, the contraction factor and the expansion factors guarantee that all of the iterates lie on a scaled, translated and rotated integer lattice. Consequently, pattern search methods can be viewed as adaptive grid search methods [7].

Lewis and Torczon's analysis provides a set of conditions which, if satisfied, guarantee a weak stationary point convergence [25, 23]. For unconstrained pattern search methods, they show that

$$\liminf_{k\rightarrow\infty} \|g(x_k^*)\| = 0.$$

Let

$$p_i(t) = \begin{cases} l_i & \text{if } t < l_i \\ t & \text{if } l_i \leq t \leq u_i \\ u_i & \text{if } t < u_i \end{cases}$$

and consider the projection of $x \in \mathbf{R}^n$ onto the feasible region of problem (2),

$$Q(x) = \sum_{i=1}^{n} p_i(x_i) e_i,$$

where e_i is the ith standard basis vector. It is well known that x is a stationary point of (2) if and only if $q(x) = Q(x - g(x)) - x = 0$. In the bound constrained theory, the quantity $q(x)$ plays the role of $g(x)$, providing a continuous measure of how close x is to a constrained stationary point. The framework for constrained pattern search methods imposes additional restrictions on the core search directions to guarantee that

$$\liminf_{k\rightarrow\infty} \|q(x_k^*)\| = 0.$$

2.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) are a class of adaptive search algorithms that are inspired by the mechanisms of natural evolution. Unlike many other optimization methods, EAs process a collection of points in each iteration. The main search mechanisms in EAs are competitive selection and genetic operators; competitive selection is used to bias the EAs search towards promising regions of the search space, and the genetic operators are used to generate new points in the space.

The pseudo-code in Figure 1 describes the steps executed in a canonical EA that minimizes an unconstrained or bound constrained objective function on $\mathbf{R}^n$. The *population* used by an EA consists of an N-tuple of points $x_i \in \mathbf{R}^n$. Each point x_i, called an *individual* of the population, is a feasible point to the problem, and the value $f(x_i)$ is said to be the *fitness* of the point. Let $X_t = \{x_1^t, \ldots, x_N^t\}$ be the N points in a population at time t, and let $\overline{X}$ and $\hat{X}$ be sets of N points used by the EA. Let unif() return a uniform random value in $[0,1]$, and let uint(N) return a uniform random integer in $1, \ldots, N$. Let $D = \mathbf{R}^n$.

```
Select an initial population $\{x_1^0, \ldots, x_N^0\}$, $x_i^0 \in \mathbf{R}^n$
$x_0^* = \arg\min\{f(x_1^0), \ldots, f(x_N^0)\}$
Repeat $t = 0, 1, \ldots$
  $\overline{X} = \text{selection}(X_t)$
  for $i = 1 : N$
    if (unif() $< \chi$) then
      $\hat{x}_i = \text{crossover}(\overline{x}_{\text{uint}(N)}, \overline{x}_{\text{uint}(N)})$
    else
      $\hat{x}_i = \overline{x}_{\text{uint}(N)}$
  for $i = 1 : N$
    $z_i = \text{mutation}(\hat{x}_i)$
  if (enforcing-bounds && feasible($z$)) then $\hat{x} = z$
  $X_{t+1} = \text{compose}(X_t, \hat{X})$
  $x_{t+1}^* = \arg\min\{f(x_t^*), f(x_1^{t+1}), \ldots, f(x_N^{t+1})\}$
Until some stopping criterion is satisfied
```

Figure 1: **A Canonical Evolutionary Algorithm**

The pseudo-code in Figure 1 relies on five subroutines: selection : $D^N \rightarrow D^N$, crossover : $D \times D \rightarrow D$, mutation : $D \rightarrow D$, compose : $D^N \times D^N \rightarrow D^N$, and feasible : $D \rightarrow \{0, 1\}$. These routines are typically randomized. We omit a detailed discussion of these methods since our analysis only requires the specification of general properties for each of them. For further details see Fogel [11, 12] for descriptions of EP, Bäck and Schwefel [3] and Bäck, Hoffmeister and Schwefel [1] for descriptions of ES, and Goldberg [14] and Davis [6] for descriptions of GAs. The **selection** operator performs a stochastic competitive selection; if $\overline{X} = \text{selection}(X_t)$, then $\overline{X}$ contains a multiset of the points in X_t, often with a bias towards points that minimize $f(x_i^t) - f(x^*)$. This subset of points is then used to generate new points using the crossover and mutation operations. The **crossover** operator combines two points to form a third. For example, if we have points x and y, then a coordinate-wise crossover operator would generate points from the coordinate values in x and y, such as

$$\text{crossover}(\{x_1,\ldots,x_n\},\{y_1,\ldots,y_n\}) \rightarrow \{x_1,x_2,y_3,y_4,\ldots,y_{n-1},x_n\}.$$

The **mutation** method modifies a single point, typically by modifying each dimension of the point with a fixed probability. For example, a normally distributed random variable could be added to each dimension of the point with probability μ. The compose method selects the set of points that are included in the subsequent population from the previous population and the newly generated points. Finally, the feasible function determines whether a point is feasible with respect to bound the constraints.

The principle feature distinguishing EP, ES and real-coded GAs is the choice of operators used to perform the evolutionary search. EP utilizes mutation to generate new points. ES and GAs utilize both mutation and crossover, although crossover is usually applied with a higher probability in GAs than in ES. The mutation operators commonly used in EP and ES employ a mechanism for adapting the step length.

For continuous domains, convergence analyses of EAs can be distinguished in two ways. First, whether or not the analysis pertains to adaptive EAs. Second, whether the analysis proves the convergence of the entire population or of the best point found in each iteration. In most cases, convergence analyses of EAs concern population convergence. For example, analyses of real-coded GAs [39, 15, 10, 26, 27] focus on how crossover processes schemata, which are subsets of the search domain that are implicitly represented by the distribution of points in a population. Unfortunately, these analyses have not provided provable convergence guarantees for real-coded GAs. In fact, Rudolph's [30] analysis of binary-coded GAs makes it clear that "The schema theorem [21] does not imply that the Canonical GA will converge to the global optimum in static optimization problems."

Convergence analyses of EP, ES and real-coded GAs that focus on the role of mutation have been more successful at providing provable convergence guarantees. Qi and Palmeri [26] consider a real-coded GA with an infinite population size and show that the distribution of points generated with selection alone converges in distribution to the distribution concentrated at the optimum, and that the mean fitness of the populations of a real-coded GA with selection and mutation converges to the fitness of the optimal point. For EP, ES and real-coded GAs with finite populations, a proof of convergence the best point to a point with near-optimal fitness can be shown using the Borel-Cantelli Lemma [1, 2, 30]. This proof requires that the mutation operator be applied with nonzero probability such that the joint distribution of possible new points has nonzero probability everywhere. Unfortunately, the assumptions used by these convergence proofs make them only of theoretical interest. In particular, practical EAs adapt the scale of search used by the mutation operator to adapt to local characteristics of the objective function.

EPs and ESs are the two main classes of adaptive EAs that modify the mutation scale, and these methods have been the subject of much analysis. Although many of these analyses consider EPs and ESs with fixed mutation scale, Beyer [4], Rappl [28, 29] and Rudolph [31] have examined the convergence of adaptive EAs. These results focus on the convergence of the population to the local minima of simple convex problems. In particular, these analyses focus on how the step length should be adapted to maximize the rate of convergence of the population. Rudolph [32] also examines how the use of adaptive methods to modify the mutation scale can limit an ESs ability to perform global optimization.

Our analysis differs from prior analyses of adaptive EAs in several respects. We analyze adaptive EAs that perform mutation with an absolute step length and which select mutation steps from a finite set of mutation offsets. Further, our analysis focuses on the convergence of the best point in the population. As a result, our analysis does not assume that the entire population is clustered about the best point. Finally, our analysis considers convergence for nonconvex problems, and thus provides a broader justification for self-adaptation than previous analyses.

3 Stochastic Pattern Search

3.1 Overview

Randomness can be introduced into pattern search methods in several different ways. A simple example would be to simply shuffle the order in which the exploratory moves algorithm considers the exploratory moves. Since each move will be considered once and since there are a bounded number of exploratory moves, this randomized exploratory moves algorithm is guaranteed to return a decreasing step if one exists. Consequently, the convergence theory for pattern search methods is immediately applicable to this class of stochastic pattern search methods.

In this section, we define a class of stochastic pattern search methods in which the exploratory moves algorithm is only probablistically guaranteed to terminate. In these pattern search algorithms, the exploratory moves algorithm contains an iterative loop, indexed by h. In each iteration, the exploratory moves algorithm may randomly consider an exploratory move from a generating matrix C_k^h. Although this matrix may vary in each iteration, the set of core exploratory moves remains the same for all h. The simple stochastic pattern search method shown in Figure 2 illustrates this type of method. The inner for-loop represents the exploratory moves algorithm. In each iteration the exploratory moves defined by the coordinate vectors $\pm e_i$ can always be sampled, which represent a set of core exploratory moves. Additionally, the algorithm samples from among the $2 \cdot 3^n$ steps in $\{0,1,-1\}^n$ and $\{0,h,-h\}^n$ in a random fashion, and the matrix C_k^h changes in each iteration because the vectors in $\{0,h,-h\}^n$ are included in this set. Using exploratory moves defined by vectors in $\{0,h,-h\}^n$ may not be of practical interest, but our analysis of evolutionary pattern search in Section 4 provides a well-motivated use of this flexibility in stochastic pattern search.

Because the exploratory moves algorithm is only probablistically guaranteed to terminate, it is possible that an "unlucky" sequence of exploratory moves could fail to ever provide a simple decrease. To ensure that the exploratory moves algorithm terminates with high probability, the probability of selecting each of the core exploratory moves must be greater than or equal to some constant greater than zero. Further, the exploratory moves algorithm terminates if all of the core exploratory moves have been sampled without finding a decreasing step. These conditions ensure that all of the core exploratory moves are eventually tried

```
Given $x_0$ and $\Delta_0$.
For $t = 1, 2, \ldots$
  For $h = 1, 2, \ldots$
    Randomly select $s \in \{0, 1, -1\}^n \bigcup \{0, h, -h\}^n$
    If($f(x_t + \Delta_t s) < f(x_t)$)
      $s_t = \Delta_t s$
      Break
    Else If ($s == e_i$ for some $i$)
      If(all vectors $\pm e_i$ have been generated)
        $s_t = \{0\}^n$
        Break
  EndFor
  $x_{t+1} = x_t + s_t$
  If ($f(x_t) <= f(x_t + s_t)$)
    $\Delta_{t+1} = \Delta_t / 2$
  Else
    $\Delta_{t+1} = \Delta_t$
EndFor
```

Figure 2: **A simple stochastic pattern search algorithm**

almost surely. Although the exploratory moves algorithm can run an arbitrarily long time, the expected length in which it terminates is bounded.

In this section, we define a class of stochastic pattern search methods that extends the definitions of pattern search provided by Lewis and Torczon [36, 25, 23]. Specifically, we generalize the definitions of positive basis pattern search methods and bound constrained pattern search methods. These algorithms allow stochastic selection of exploratory moves as we have described. Further, they allow the noncore search directions in the generating matrix C_k^h to be rational vectors. This generalization allows the noncore exploratory moves to search on the most refined integer lattice considered by the pattern search so far.

3.2 Definitions

The following definition of stochastic pattern search extends the abstract description of pattern search methods provided by Lewis and Torczon [36, 25, 23]. The definitions of the main loop and step-length update algorithm are virtually the same as standard pattern search methods. The definition of the pattern generalizes the definition of the generating matrix, and the definition of the exploratory moves algorithm is extended to include stochastically selected exploratory moves. The restrictions required to define a bound constrained stochastic pattern search method are explicitly noted. In Section 3.3 we discuss the differences between this method and standard pattern search methods.

From the theory of positive linear dependence [5] we have the following definitions. A *positive span* of a set of vectors $\{a_1, \ldots, a_r\}$ is the cone

$$\{a \in \mathbf{R}^n \mid a = c_1 a_1 + \ldots + c_r a_r, c_i \geq 0 \text{ for all } i\}.$$

The set $\{a_1, \ldots, a_r\}$ is called *positively dependent* if one of the a_i's is a nonnegative combination of the others; otherwise the set is *positively independent.* A *positive basis* is a positively independent set whose positive span is $\mathbf{R}^n$.

The (Bound Constrained) Stochastic Pattern Search Method Figure 3 defines the main elements of a stochastic pattern search method for bound constrained and unconstrained minimization. To define a stochastic pattern search method, it is necessary to specify the basis matrix B, the initial value of C_0^0, the exploratory moves algorithm, and the algorithms for updating C_k^h and Δ_k.

```
Let $x_0 \in \mathbf{R}^n$ and $\Delta_0 > 0$ be given.
For $k = 0, 1, \ldots$
   Compute $f(x_k)$.
   Determine a step $s_k$ using an exploratory moves algorithm.
   Compute $\rho_k = f(x_k) - f(x_k + s_k)$.
   If $\rho_k > 0$, then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
   Update $C_{k+1}^0$ and $\Delta_{k+1}$.
```

Figure 3: **Stochastic Pattern Search**

The Updates Figure 4 defines the algorithm that updates the step size, Δ_k. This algorithm reduces the step size if the exploratory moves algorithm has examined all feasible core trial steps and they fail to produce a simple decrease. If the exploratory moves algorithm returns a core trial step that gives a simple decrease, then this algorithm either increases the step size or preserves the current step size.

After k iterations, the value of Δ_k is

$$\Delta_k = \Delta_0 \theta^{a_0} \lambda_1^{a_1} \ldots \lambda_L^{a_L} = \Delta_0 \tau^{r_k},$$

where $a_i \in \mathbf{Z}$ and $r_k \in \mathbf{Z}$; the a_i are simply values that arise in the expression of Δ_k. Let $r_{\max}^k = \max_{i=1,\ldots,k} r_k$ and $r_{\min}^k = \min_{i=1,\ldots,k} r_k$. We define $\overline{\tau}_k = \tau_n^{(-r_{\min}^k)} \tau_d^{(r_{\max}^k)}$.

Suppose $\tau \in \mathbf{Q}$, such that $\tau > 1$ and $\tau = \tau_n/\tau_d$, where $\tau_n, \tau_d \in \mathbf{N}$ and τ_n and τ_d are relatively prime. Let $\theta = \tau^{\omega_0}$ and $\lambda_k \in \Lambda = \{\tau^{\omega_1}, \dots, \tau^{\omega_L}\}$, where $\{\omega_0, \omega_1, \dots, \omega_L\} \subset \mathbf{Z}$, $L = |\Lambda| < \infty$, $\omega_0 < 0$, and $\omega_i \geq 0$, $i = 1, \dots, L$.

If $\rho_k \leq 0$ then $\Delta_{k+1} = \theta \Delta_k$.
If $\rho_k > 0$ and s_k is a core trial step then $\Delta_{k+1} = \lambda_k \Delta_k$.

Figure 4: **Updating Δ_k**

The Pattern To define a pattern we need two components, a *basis matrix* and a *generating matrix.* A basis matrix can be any nonsingular matrix $B \in \mathbf{R}^{n \times n}$. A generating matrix is a matrix $C_k^h \in \mathbf{Z}^{n \times p}$, where $p > n + 1$, h is the iteration number of the exploratory moves algorithm and k is the iteration number of the stochastic pattern search algorithm; the value $p \equiv p_{k,h}$, but we drop the indeces for simplicity. A generating matrix is partitioned into components

$$C_k^h = \left[\Gamma_k \ \frac{\Delta_0}{\Delta_k \overline{\tau}_k} L_k^h \ 0 \right],$$

where L_k^h and Γ_k are integral matrices with n rows, and 0 in the last column of C_k^h is a single column of zeros. We call Γ_k the core generating matrix, and we require that $\Gamma_k \in M$, where M is a finite set of integral matrices, each of which form a positive basis for $\mathbf{R}^n$. It follows that Γ_k must have at least $n + 1$ columns. For bound constrained pattern search, $\Gamma_k = [M_k \ \ -M_k]$, where M_k is a diagonal integer matrix (so Γ_k is a maximal positive basis [23]). A *pattern* is defined by the columns of the matrix $P_k^h = BC_k^h$. Because both B and C_k^h have rank n, the columns of P_k^h span $\mathbf{R}^n$.

Given $\Delta_k \in \mathbf{R}$, $\Delta_k > 0$, we define a *trial step* s_k^h to be any vector of the form $s_k^h = \Delta_k B c_k^h$, where c_k^h denotes a column of C_k^h. The core trial steps are defined by $\Delta_k B \Gamma_k$. Note that Bc_k^h determines the direction of the step, while Δ_k serves as a step length parameter. For bound constrained pattern search, a trial step s_k^h is *feasible* if $x_k + s_k^h \in \Omega$. At iteration k, a *trial point* is any point of the form $x_k + s_k^h$, where x_k is the current iterate.

The Exploratory Moves Pattern search methods proceed by conducting a series of *exploratory moves* about the current iterate before identifying a new iterate. The stochastic pattern search methods we consider differ from non-stochastic methods in that the exploratory moves algorithm is only probablistically guaranteed to terminate. Consequently, there is no fixed number of iterations for the stochastic exploratory moves algorithm; the algorithm terminates when a decreasing step has been found or after all feasible core trial steps have been examined. In addition, the stochastic nature of the exploratory moves algorithm enables the generating matrix C_k^h to vary with each iteration h of the algorithm. The following two sets of conditions are placed on the exploratory moves s_k generated by an exploratory moves for unconstrained and bound constrained problems respectively.

Condition 1 *Unconstrained Stochastic Pattern Search*

1. $s_k = s_k^h \in \Delta_k P_k^h$, $h = 0, 1, 2, \ldots$

2. *If* $\min\{f(x_k + y) \mid y \in \Delta_k B \Gamma_k\} < f(x_k)$ *then* $f(x_k + s_k) < f(x_k)$.

3. *The exploratory moves algorithm terminates and returns* $s_k = 0$ *if each of the core steps defined by* $\Delta_k B \Gamma_k$ *has been examined without identifying a decreasing step. At each iteration of the exploratory moves algorithm, the probability of selecting each of the core steps is greater than or equal to a constant* $\nu > 0$.

Condition 2 *Bound Constrained Stochastic Pattern Search*

1. $s_k = s_k^h \in \Delta_k P_k^h$, $h = 0, 1, 2, \ldots$

2. $x_k + s_k \in \Omega$

3. *If* $\min\{f(x_k + y) \mid y \in \Delta_k B \Gamma_k, x_k + y \in \Omega\} < f(x_k)$ *then* $f(x_k + s_k) < f(x_k)$.

4. *The exploratory moves algorithm terminates and returns* $s_k = 0$ *if each of the feasible core steps defined by* $\Delta_k B \Gamma_k$ *has been examined without identifying a decreasing step. At each iteration of the exploratory moves algorithm, the probability of selecting each of the feasible core steps is greater than or equal to a constant* $\nu > 0$.

The addition of Condition 1.3 and 2.4 is the difference between these conditions and the hypotheses on exploratory moves for unconstrained and bound constrained pattern search [36, 25]. This restriction uses the observation that the convergence theory for pattern search relies solely on the ability of the core generating matrix Γ_k to generate a decreasing step. Since the other trial steps are not critical to ensure the convergence of the pattern search method, it is safe to terminate the exploratory moves algorithm after all of the feasible core steps have been examined.

3.3 Comparison

There are three main differences between the stochastic pattern search methods that we have defined and the pattern search methods defined by Lewis and Torczon [36, 25]. First, these methods randomly select trial steps to find an improving point. Condition 1 and 2 both restrict the probability of selecting a core trial step above a fixed probability for every iteration. This requirement ensures that the sequence of calls to the exploratory moves algorithm terminates almost surely. Clearly, the expected length of the exploratory moves algorithm depends on the precise value of ν.

Next, the step length may only be increased when the exploratory moves algorithms finds a core trial step that provides a simple decrease. Thus both reductions and expansions of the step length are tied to the search characteristics of the core trial steps. Consequently, the restrictions on the core search steps ensure that the sequence of improving trial steps eventually provides a "gradient related" search that can be used to provide a stationary-point convergence guarantee [36].

Finally, the noncore elements of the generating matrix are allowed to assume fractional values. The fractional columns of the generating matrix are integer vectors scaled by $\frac{\Delta_0}{\Delta_k \overline{\tau}_k}$. This enables new iterates to lie on the lattice scaled by $\Delta_0/\overline{\tau}_k$. Thus if Δ_k increases, the pattern search method can still search with a scale that is bounded by the smallest value of Δ_k.

3.4 Convergence Analysis

Our main results are Theorems 1 and 2, which prove a stationary-point convergence for unconstrained and bound constrained stochastic pattern search respectively. The proofs of these theorems are given in the next section. Recall that $L_\Omega(y) = \{x \in \Omega \mid f(x) \leq f(y)\}$ and $L(y) = L_{\mathbf{R}^n}(y)$.

Theorem 1 *Let $L(x_0)$ be compact and suppose that $f : \mathbf{R}^n \to \mathbf{R}$ is continuously differentiable on an open neighborhood of $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the unconstrained stochastic pattern search method,*

$$P\left(\liminf_{k\to\infty} \|g(x_k)\| = 0\right) = 1.$$

Theorem 2 *Let $L_\Omega(x_0)$ be compact and suppose that $f : \mathbf{R}^n \to \mathbf{R}$ is continuously differentiable on an open neighborhood of $L_\Omega(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the bound constrained stochastic pattern search method,*

$$P\left(\liminf_{k\to\infty} \|q(x_k)\| = 0\right) = 1.$$

This convergence guarantee is weak, since it only implies that the gradient is sampled infinitely often near a stationary point. Thus it is possible for $\limsup_{k\to\infty} \|g(x_k)\| > 0$ for an unconstrained stochastic pattern search method. However, the sequence of iterates generated by a pattern search method is monotone nonincreasing and bounded below on a compact set, so $\lim_{k\to\infty} f(x_k) = \hat{f}$ for some fixed value $\hat{f}$. Note that this is a "global" convergence analysis since it guarantees convergence to a stationary point from any starting point.[1]

Although Theorems 1 and 2 require that f is continuously differentiable on an open set containing the compact sets $L(x_0)$ and $L_\Omega(x_0)$ respectively, these results could be proven when f is simply continuously differentiable on $L(x_0)$ and $L_\Omega(x_0)$. However, the assumption that f is continuously differentiable on a set slightly larger than $L(x_0)$ or $L_\Omega(x_0)$ makes the proof shorter with little expense of generality (see Lewis and Torczon [23] for further discussion of this point).

[1] This terminology is unfortunate in that convergence to a *global* minimizer of the function is *not* implied. However, "locally convergent" is reserved for another use for nonlinear optimization [8].

3.5 Proof of Theorems 1 and 2

The proof of Theorem 1 follows the same basic structure as the proof for the positive basis pattern search methods described by Lewis and Torczon [23], and the proof of Theorem 2 follows the same basic structure as the proof for bound constrained pattern search methods described by Lewis and Torczon [25]. The main differences are that (a) the noncore steps are allowed to search a more refined lattice and (b) we show how the randomness in stochastic pattern search guarantees convergence almost surely.

Lemma 1 illustrates the manner in which Δ_k represents a step length. Unlike the pattern search methods described by Lewis and Torczon [36, 25, 23], Δ_k may only represent a step length for core trial steps.

Lemma 1 *There exists a constant $\eta > 0$, independent of k, such that for any core trial step s_k^i we have $\|s_k^i\| \geq \eta\Delta_k$.*

Proof. The columns of the core generating matrix satisfy the same conditions as the columns in standard pattern search methods, so this follows from Lemma 3.1 in [36]. ∎

The following theorem demonstrates that the iterates generated by stochastic pattern search methods remain on a translated, scaled integer lattice, G. This lattice is scaled by $\tau_n^{(r^k_{\min})}\tau_d^{(-r^k_{\max})}\Delta_0$, and rotated by B.

Theorem 3 *Any iterate x_t produced by a stochastic pattern search method can be expressed as*

$$x_t = x_0 + \tau_n^{(r^k_{\min})}\tau_d^{(-r^k_{\max})}\Delta_0 B \sum_{k=0}^{t-1} z_k$$

for some $z_k \in \mathbf{Z}^n$, $k = 0, \ldots, t-1$.

Proof. The stochastic pattern search algorithm guarantees that any iterate x_t is of the form

$$x_t = x_0 + \sum_{k=0}^{t-1} s_k.$$

The trial steps are of the form $s_k = \Delta_k B c_k$, so we have

$$x_t = x_0 + \sum_{k=0}^{t-1} \Delta_k B c_k.$$

Now recall that $\tau = \tau_n/\tau_d$, where $\tau_n, \tau_d \in \mathbf{N}$ are relatively prime. Let I_Γ be the set of iterations for which c_k is a core step, and let I_L be the set of iterations for which c_k is not a core step. Then we have

$$\begin{aligned} x_t &= x_0 + \Delta_0 B \left(\sum_{k \in I_\Gamma} \tau^{r_k} c_k + \sum_{k \in I_L} l_k/\bar{\tau}_k \right) \\ &= x_0 + \tau_n^{(r^k_{\min})}\tau_d^{(-r^k_{\max})}\Delta_0 B \sum_{k=0}^{t-1} z_k, \end{aligned} \quad (3)$$

for some $l_k, z_k \in \mathbf{Z}^n$. ■

The following lemma guarantees that each of the exploratory moves terminate with probability one.

Lemma 2 *Let A be the set of sequences of trial steps for which each exploratory move terminates. Then $P(A) = 1$.*

Proof. Recall that the exploratory moves algorithm terminates if a simple decrease is found or if all $\gamma_k = |\Gamma_k| \leq 2n$ core trial steps have been sampled. Thus it suffices to show that all core trial steps will eventually be sampled almost surely. Let $R_{r,i,j}$ be the set of sequences of trial steps for which the exploratory moves algorithm does not sample the jth core trial step in steps $r+1$ through $r+i$, and let $R_{r,i}$ be the set of sequences of trial steps for which the exploratory moves algorithm does not sample one or more of the core trial steps in all of the $r+1$ through $r+i$ steps. Note that $R_{r,i} = \bigcup_{j=1}^{\gamma_k} R_{r,i,j}$, where Γ_k defines the core trial steps. Consequently,

$$P(R_{r,i}) \leq \sum_{j=1}^{\gamma_k} P(R_{r,i,j}) \leq 2n(1-\nu)^i.$$

Let R_r be the set of sequences of trial steps for which the exploratory moves algorithm does not sample one or more of the core trial steps in all of the trial steps following the rth trial step. Note that $R_r = \bigcap_{i=1}^{\infty} R_{r,i}$, and that $R_{r,i} \supset R_{r,i+1}$. Thus we have

$$P(R_r) = \lim_{i \to \infty} P(R_{r,i}) \leq 2n(1-\nu)^i = 0.$$

Now $P(A) = 1 - P(A^c)$ and $A^c = \bigcup_{r=1}^{\infty} R_r$. Since $R_r \subset R_{r+1}$, we have $P(A^c) = \lim_{r\to\infty} P(R_r) = 0$, so $P(A) = 1$. ■

Theorem 4 uses the previous results to demonstrate that the sequence of step lengths approaches zero with probability one for the unconstrained and bound constrained cases. This proof is analogous to the proofs of Theorem 3.3 in [36] and Theorem 5.5 in [25].

Theorem 4 *For unconstrained (bound constrained) stochastic pattern search, suppose that $L(x_0)$ ($L_\Omega(x_0)$) is compact. Then $P(\liminf_{k\to\infty} \Delta_k = 0) = 1$.*

Proof. Suppose that each exploratory move terminates. Now suppose that $0 < \Delta_{\min} \leq \Delta_k = \tau^{r_k}\Delta_0$ for all k. The hypothesis that $\Delta_{\min} \leq \Delta_k$ for all k means that the sequence $\{\tau^{r_k}\}$ is bounded away from zero. We also know that the sequence $\{\Delta_k\}$ is bounded above because all iterates x_k lie inside $L(x_0)$ ($L_\Omega(x_0)$), which is compact; Lemma 1 then guarantees that the upper bound $\Delta_{\max} < \infty$ for $\{\Delta_k\}$. Thus the sequence $\{\tau^{r_k}\}$ is bounded above, from which it follows that the sequence $\{\tau^{r_k}\}$ is a finite set. Equivalently, the sequence $\{r_k\}$ is bounded above and below. Let

$$r_{\min} = \min_{0\leq k\leq\infty} r_k \qquad r_{\max} = \max_{0\leq k\leq\infty} r_k. \tag{4}$$

Then Equation (3) holds for the bounds given in Equation (4), so x_k lies in a translated, scaled integer lattice G for all k. The intersection of $L(x_0)$ ($L_\Omega(x_0)$) with the lattice G is finite, so there must exist a point x_* for which $x_k = x_*$ for infinitely many k.

However, this is a contradiction since we cannot revisit a point on the lattice infinitely many times. We accept a new step s_k if and only if $f(x_k) > f(x_k + s_k)$, so there exists N such that for all $k \geq N$, $x_k = x_*$. This implies that $\rho_k = 0$ for $k \geq N$, but this implies that $\Delta_k \to 0$, which gives a contradiction to our assumption that $\Delta_k \geq \Delta_{\min} > 0$.

From Lemma 2 we know that the set of sequences for which each exploratory move terminates has measure one, so $P\left(\liminf_{x\to\infty} \Delta_k = 0\right) = 1$. ■

The previous results provide the basis for the convergence of unconstrained and bound constrained pattern search methods. Specifically, they guarantee that the sequence of iterates is implicitly constrained to a translated, scaled and rotated integer lattice, and that the step size asymptotically converges to zero almost surely. The following proposition follows directly from Proposition 3.4 in Torczon [36] and Corollary 5.4 in Lewis and Torczon [23]. This proposition uses Theorems 3 and 4, and it uses the fact that the core steps provide a simple decrease in the objective function if one exists. We omit the details of this proof since they are virtually identical to the lengthy proofs of Proposition 3.4 in Torczon [36] and Corollary 5.4 in Lewis and Torczon [23]. The principle difference is that these proofs implicitly rely on the search behavior of the pattern search with respect to the core trial steps. Consequently, we can apply these results to the stochastic pattern search methods that we have defined because (a) they apply the same restrictions to the core search directions and (b) they tie changes in the step length Δ_k to the search among the core search directions.

Proposition 1 *For unconstrained (bound constrained) stochastic pattern search, assume that $L(x_0)$ ($L_\Omega(x_0)$) is compact, that f is continuously differentiable on $L(x_0)$ ($L_\Omega(x_0)$), and that there exists a sequence $\{x_k\}$ generated by the stochastic pattern search algorithm for which $\liminf_{k\to\infty} \|g(x_k)\| \neq 0$ ($\liminf_{k\to\infty} \|q(x_k)\| \neq 0$). Then there exists a constant $\Delta_{\min} > 0$ such that for all k, $\Delta_k > \Delta_{\min}$.*

We are now ready to prove Theorems 1 and 2.

Proof. The proof is by contradiction. For an unconstrained (bound constrained) stochastic pattern search algorithm, suppose that $P\left(\liminf_{k\to\infty} \|g(x_k)\| = 0\right) < 1$ ($P\left(\liminf_{k\to\infty} \|q(x_k)\| = 0\right) < 1$). Then there exists a set of sequences of iterates with measure greater than zero for which $\liminf_{k\to\infty} \|g(x_k)\| \neq 0$ ($\liminf_{k\to\infty} \|q(x_k)\| \neq 0$). For these sequences, we know from Proposition 1 that there exists $\Delta_{\min} > 0$ such that $\Delta_k > \Delta_{\min}$. But this contradicts Theorem 4. ■

4 Evolutionary Pattern Search Algorithms

Consider the pseudo code in Figure 5 which defines EPSAs. Various conditions are placed upon this class of EAs to ensure that they can be cast as stochastic pattern search methods. Mild restrictions are placed upon the **selection** and **compose** functions to ensure that (a) the best point in the population is selected with probability of at least $\pi > 0$ in each iteration and (b) the best point in $X_t \bigcup \hat{X}$ is is always included in X_{t+1}. The **crossover** function is also restricted to generate a point such that crossover$(x, y) \in \{x_1, y_1\} \times \{x_2, y_2\} \times \ldots \times \{x_n, y_n\}$, which is consistent with most standard crossover operators (e.g. two-point crossover).

The call to uint(j) uniformly generates an integer from 1 to j. The χ is the probability of crossover and μ is the probability of mutating a point in the population by adding a mutation offset. The vector $\eta \in \{0,1\}^m$ serves as a counter to determine whether all of the mutation offsets have been generated on the best point in the population. The contraction factor for Δ_t is $\theta = (\tau_n/\tau_d)^{\kappa_0}$, $\tau_n, \tau_d \in \mathbf{N}$, $\tau_n > \tau_d$ and $\kappa_0 < 0$. The expansion factor is $\lambda_t \in \{1, \theta^{\kappa_1}, \ldots, \theta^{\kappa_d}\}$, $\kappa_i \in \mathbf{Z}$ and $\kappa_i > 0$.

To highlight the relationship between EPSAs and stochastic pattern search algorithms, we consider the revised EPSAs defined in Figures 6 and 7. Figure 6 defines an exploratory moves algorithm that is comprised of the iterations of an EPSA in which no improvement in the objective function is found. Figure 7 defines the main loop of the EPSA that uses this exploratory moves algorithm. Together, these routines perform exactly the same search as the EPSA in Figure 5. Each iteration of the loop in the exploratory moves algorithm corresponds to a set of moves of a stochastic pattern search algorithm with generating matrix C_k^h. The restriction on the replacement strategy ensures that the best individual found is kept for further processing. We will see that these restrictions enable EPSAs to be viewed as a stochastic pattern search method with respect to the best individual in the each generation. Finally, note that step 15.a in Figure 5 and step 10.a in Figure 6 are only used for bound-constrained EPSAs. Otherwise, these conditions are always true.

4.1 EPSAs as Stochastic Pattern Search

In what follows we describe how the three central components of stochastic pattern search – updating step length, the generating matrix, and the exploratory moves algorithm – are implemented by EPSAs. This demonstrates that EPSAs can be described as stochastic pattern search methods.

Updating the Step Length Steps 8 through 11 of Figure 7 perform the update to the step length Δ_k. Because of the restrictions on θ and λ_k, this update is exactly as given in Figure 4. When the inner loop fails to generate a simple decrease on f, the step length is decreased by a factor of θ. Otherwise, if the decrease is due to an offset of the best point in the population then the step length may be increased by a factor of λ_k. The step length is not allowed to expand when any decreasing step is generated because only the mutation offsets of the best point in the population correspond to core trial steps in a stochastic pattern search method.

```
Given $\Delta_0 \in \mathbf{Q}^{>0}$
Given $S = \{s_1, \ldots, s_m\}$, where $s_i \in \mathbf{Z}^n$ forms a positive basis
Let $\eta = \{0\}^m$
Select an initial population $X_0 = \{x_1^0, \ldots, x_N^0\}$, $x_i^0 \in \mathbf{Q}^n$
$x_0^* = \arg\min\{f(x_1^0), \ldots, f(x_N^0)\}$
Repeat $t = 0, 1, \ldots$
    $\overline{X} = \text{selection}(X_t)$
    For $i = 1 : N$
        If $(\text{unif}() < \chi)$ then
            $\hat{x}_i = \text{crossover}(\overline{x}_{\text{uint}(N)}, \overline{x}_{\text{uint}(N)})$
        Else
            $\hat{x}_i = \overline{x}_{\text{uint}(N)}$
    For $i = 1 : N$
        If $(\text{unif}() < \mu)$ then
            $j = \text{uint}(m)$
            If $(\hat{x}_i + \Delta_t \cdot s_j$ is feasible$)$
                If $(\hat{x}_i == x_t^*)$ $\eta_j = 1$
                $\hat{x}_i = \hat{x}_i + \Delta_t \cdot s_j$
    $X_{t+1} = \text{compose}(X_t, \hat{X})$
    $x_{t+1}^* = \arg\min\{f(x_1^{t+1}), \ldots, f(x_N^{t+1})\}$
    If $(f(x_{t+1}^*) < f(x_t^*))$
        If $(\exists s \in S$ s.t. $x_{t+1}^* = x_t^* + s)$ $\Delta_{t+1} = \Delta_t * \lambda_t$
        $\eta = \{0\}^m$
    ElseIf $(|\eta| == m)$
        $\eta = \{0\}^m$
        $\Delta_{t+1} = \Delta_t \theta$
    Else
        $\Delta_{t+1} = \Delta_t$
Until some stopping criterion is satisfied
```

Figure 5: **Pseudo Code for EPSAs**

```
Repeat $h = 0, 1, \ldots$
  $\overline{X} = \text{selection}(X_t)$
  For $i = 1 : N$
    If ($\text{unif}() < \chi$) then
      $\hat{x}_i = \text{crossover}(\overline{x}_{\text{uint}(N)}, \overline{x}_{\text{uint}(N)})$
    Else
      $\hat{x}_i = \overline{x}_{\text{uint}(N)}$
  For $i = 1 : N$
    If ($\text{unif}() < \mu$) then
      $j = \text{uint}(m)$
      If ($\hat{x}_i + \Delta_t \cdot s_j$ is feasible)
        If ($\hat{x}_i == x_t^*$) $\eta_j = 1$
        $\hat{x}_i = \hat{x}_i + \Delta_k \cdot s_j$
  $X_{t+1} = \text{compose}(X_t, \hat{X})$
  $x_{t+1}^* = \arg\min\{f(x_1^{t+1}), \ldots, f(x_N^{t+1})\}$
  $t = t + 1$
Until ($f(x_t^*) < f(x_{t-1}^*)$) or ($|\eta| == m$)
```

Figure 6: **Pseudo Code for the EPSA exploratory moves algorithm.**

```
Given $\Delta_0 \in \mathbf{Q}^{>0}$
Given $S = \{s_1, \ldots, s_m\}$, where $s_i \in \mathbf{Z}^n$ forms a positive basis
Let $\eta = \{0\}^m$
Select an initial population $X_0 = \{x_1^0, \ldots, x_N^0\}$, $x_i^0 \in \mathbf{Q}^n$
$x_0^* = \arg\min\{f(x_1^0), \ldots, f(x_N^0)\}$
$t = 0$
Repeat $k = 0, 1, \ldots$
    Determine a population $X_{t+1}$ using the EPSA exploratory moves algorithm
    If($|\eta| == m$)
        $\Delta_{k+1} = \Delta_k \theta$
    ElseIf ($\exists s \in S$ s.t. $x_{t+1}^* = x_t^* + s$)
        $\Delta_{k+1} = \Delta_k \lambda_k$
    $\eta = \{0\}^m$
Until some stopping criterion is satisfied
```

Figure 7: **Pseudo Code for the main loop of EPSAs**

The Generating Matrix Pattern search methods are designed to generate test patterns from a single solution. To cast EPSAs within this framework, the individuals generated at each generation are viewed as patterns with respect to the best individual in the population. Figure 8 illustrates the pattern vectors generated by individuals created by crossover and mutation. Points A and B represent individuals created by mutation from the best individual, x_1. Point C represents an individual created by crossover from x_1 and x_i. Point D represents an individual created by mutation of x_i, and point E represents an individual created by crossover from x_1 and x_i followed by mutation. The solid arrow lines represent the pattern vectors, which are implicitly scaled by the basis matrix.

The generating matrix C_k^h is constructed using all possible individuals that can be generated in the current generation t (as indexed by k and h). Let $f(x_1^t) \leq f(x_j^t)$, $j = 2, \ldots, N$, for all t. Let $\text{lcd}(q_1, \ldots, q_d)$ denote the least common denominator of q_i, $q_i \in \mathbf{Q}$. Then let $\gamma = \text{lcd}(1/\Delta_0, x_{k,j}^0, s_{i,j}, i = 1, \ldots, m, k = 1, \ldots, N, j = 1, \ldots, n)$, which represents the least common denominator of the current points, the initial step length and the mutation offset vectors.

The basis matrix implicitly used by EPSAs is $B = \frac{1}{\gamma^3} I$, where I is the identity matrix. The generating matrix $\Gamma_k = \gamma^3 [s_1 \ldots s_m]$ defines the mutation offsets around the best individual in each generation. The columns of L_k^h are implicitly defined by taking each individual $\hat{x}$ that can be generated by an EPSA at generation t and constructing the vector $\gamma^3 \tau_n^{(-r_{\min}^k)} \tau_d^{(r_{\max}^k)} (\hat{x} - x_1^t)/\Delta_0$. Thus the trial step generated by a

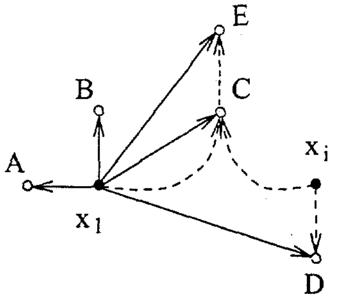

Figure 8: Illustration of pattern vectors that are offsets from the best individual, x_1. Solid arrows represent the vectors in the generating matrix and dashed arrows represent the genetic operations in an EPSA that generate new solutions.

column l_k^h in L_k^h is

$$\Delta_k B \frac{\Delta_0}{\overline{\tau}_k \Delta_k} l_k^h = B \frac{\Delta_0}{\overline{\tau}_k} \gamma^3 \tau_n^{(-r_{\min}^k)} \tau_d^{(r_{\max}^k)} (\hat{x} - x_1^t)/\Delta_0 = (\hat{x} - x_1^t). \tag{5}$$

The number of columns in C_k^h is bounded by the maximum number of points that can be generated from crossover and mutation by a finite population of size N. Thus there are $O(mN^2)$ distinct columns in C_k^h.

The following lemma confirms that the columns of L_k^h are in $\mathbf{Z}^n$.

Lemma 3 $L_k^h \in \mathbf{Z}^{n \times z}$ *for some* $z > 1$ *and for all* $h = 0, 1, \ldots$ *and* $k = 0, 1, \ldots$.

Proof. Let $l_k^h = \gamma^3 \tau_n^{(-r_{\min}^k)} \tau_d^{(r_{\max}^k)} (\hat{x} - x_1^t)/\Delta_0$ be a column of L_k^h. Consider the i-th dimension of the vector $\hat{x} - x_1^t$. Because crossover is coordinate-wise, the i-th dimension of $\hat{x}$ is equal to

$$x_j^0 + \Delta_{h_1} s_{g_1}(j) + \ldots + \Delta_{h_t} s_{g_t}(j).$$

for some j, sequence $h_1, \ldots, h_t$ and sequence $g_1, \ldots, g_t$. Here, $s_{g_t}(j)$ refers to the jth element of the vector s_{g_t}. Similarly, the i-th dimension of x_1^t is equal to

$$x_{j'}^0 + \Delta_{h'_1} s_{g'_1}(j') + \ldots + \Delta_{h'_t} s_{g'_t}(j').$$

for some j', sequence $h'_1, \ldots, h'_t$ and sequence $g'_1, \ldots, g'_t$.

Now $\gamma/\Delta_0 \in \mathbf{Z}$, $\gamma s_{ij} \in \mathbf{Z}$, $\forall i, j$, and $\gamma x_j^0 \in \mathbf{Z}$, $\forall j$. Also, we know that $\tau_n^{(-r_{\min}^k)} \tau_d^{(r_{\max}^k)} \Delta_j \in \mathbf{Z}$, $\forall j$. It follows that l_k^h is an integer vector. Since l_k^h was chosen arbitrarily, $L_k^h \in \mathbf{Z}^{n \times z}$ for some $z > 1$ and for all $h = 0, 1, \ldots$ and $k = 0, 1, \ldots$. ■

Exploratory Moves As required by the exploratory moves hypotheses (Conditions 1 and 2), the exploratory moves algorithm described in Figure 6 only terminates if a solution is found that generates a simple decrease, or if all (feasible) core steps defined by $\Delta_k B \Gamma_k$ have been examined.

Equation 5 shows that all possible feasible points that can be generated by EPSAs are captured in patterns in C_k^h. The core generating matrix represents the mutation offsets from the best point in the population. The restriction on the selection strategy ensures that the point x_1^t is included in $\overline{X}$ with probably at least π. Thus the probability that each of these mutation steps can occur on x_1^t is greater than or equal to $\pi(1-\chi)\mu/m > 0$. Thus, the exploratory moves algorithm implicitly defined by EPSAs satisfies the hypothesis on exploratory moves.

4.2 Generalizations

The formulation of EPSAs given in Figures 5, 6 and 7 was designed to provide the simplest modification to the canonical EA in Figure 1 that captures both crossover and mutation operators. There are many other related classes of EAs that can be cast as stochastic pattern search. For example, this description of EPSAs can be extended to include EAs that perform mutation by adding each mutation offset with a fixed probability. If the set of mutation offsets are the unit vectors and their complements, this form of mutation is analogous to the mutation performed in real-valued GAs. Using this form of mutation extends the set of possible points that can be generated in each generation, but the analysis remains the same because the same set of core trial steps is preserved in this class of EPSAs. Hart and Hunter [17, 20] consider the empirical impact of using different forms of mutation in EPSAs.

Another way that this framework can be modified is by using a hybrid form of randomization. Since an explicit check is made to determine whether a mutation is an offset from the best point in the population, the mutation steps applied to the best point can be selected without replacement. This can be achieved using a shuffled list of indices into the list of mutation offsets. This modification preserves the basic properties of the EPSA while (a) increasing the rate at which all of the trial points about the best point are selected and (b) allowing other random trial steps to be selected in a less structured manner.

5 Discussion

The central contribution of this work is the development of a convergence theory for a class of EAs that guarantees convergence almost surely to a stationary point for any continuously differentiable function. This is the first stationary-point convergence proof for any class of EAs that optimizes a general class of nonconvex continuous functions. Consequently, this convergence theory also provides a rigorous justification for the use of adaptive EAs in a broad range of problems; previously, methods for adapting the mutation operator have been analyzed for specific classes of functions (e.g. see Rappl [29]).

Torczon [35, 36] notes that the convergence analysis for pattern search methods can be easily extended to handle cases when the function f is nondifferentiable. This is reassuring since these methods are often applied to nondifferentiable functions. Let X^* include the set of stationary points of the function f in $L(x_0)$, the set of all points in $L(x_0)$ where f is nondifferentiable, and the set of all points in $L(x_0)$ where the derivative of f exists but is not continuous. Then Theorems 1 and 2 can be extended to ensure convergence to a point in X^*.

The convergence theory for EPSAs is qualitatively different from proofs of convergence for EAs that converge almost surely to a global optimum. The difference stems from the fact that the problem of finding a solution x^* for which $f(x^*)$ is the global optimum is ill-conditioned [38]. Consequently, these proofs of convergence to the global optimum only guarantee that the estimate of the optimal fitness, $f(x_t^*)$, converges to $f(x^*)$. Although the convergence theory for EPSAs only guarantees convergence to *some* stationary point $\bar{x}^*$, it can guarantee that the sequence of improving solutions x_t^* weakly converges to $\bar{x}^*$.

Since EAs are typically cast as global optimization methods, it is natural to consider the relevance of this form of local convergence. Like EPSAs, EAs such as EPs and ESs employ adaptive mechanisms for adjusting the mutation step length. It is well recognized that these adaptive mechanisms are important for effective refinement of points during the evolutionary search. However, Rudolph [32] has recently shown that the self-adaptive mechanism used by ESs limits the global search performed by these EAs; for nonconvex problems the probability that an ES converges to a globally optimal point may be less than one. Consequently, the local nature of the EPSA's convergence theory is not inconsistent with our current understanding of EAs for continuous problem domains.

Additionally, since convergence to a global optimum requires convergence to *some* local optima, the dynamics and performance of EPSAs can provide insight into the behavior of EAs as they approach the global optimum. For example, the convergence analysis of pattern search methods highlights the fact that the rate of convergence of EPSAs is likely to decrease as the dimension of the problem increases [36]. Also, pattern search methods can provide insight into the requirements for stopping rules. Although direct search methods cannot guarantee that they will terminate at (or near) a stationary point, a variety of stopping rules have been successfully applied for these methods. Stopping rules based on weaker analyses of EAs (like the Borel-Cantelli lemma) force EAs to terminate only after a very large number of generations.

Finally, recent experimental work with EPSAs confirms that their empirical performance is comparable with EPs. Hart and Hunter [20] evaluate the impact of a variety of algorithmic parameters on the performance of EPSAs. The performance of the EPSA design that their experiments recommend is at least as good as EPs on a suite of standard global optimization test functions. Further, Hart [19] confirms this result on a challenging drug docking application. These results confirm that EPSAs can perform a global search that is comparable to other EAs.

It has long been recognized that pattern search methods do not enjoy fast local convergence properties [36, 40], so it is reasonable to expect that EPSAs converge slowly. Although crossover has proved a valuable search mechanism for a variety of problems, the expected length of the inner loop of an EPSA could be increased by a factor of approximately $1/(1-\chi)$ when the step length needs to be reduced [18]. Hart and Hunter [17, 19, 20] show how using crossover can lead to a trade-off between the rate of convergence of EPSAs and the degree of global search. Using crossover in an EPSA appears to decrease its rate of convergence to a final solution, but it may find a better solution overall.

The connection that we have established between stochastic pattern search and evolutionary algorithms provides insight into the fundamental relationship between evolutionary algorithms and direct search. The extensions that we have made to pattern search highlight the general nature of this framework. Since the analysis of direct search methods is more mature than the analysis of EAs, the connection we have made will further our understanding of the basic theoretical properties of EAs. We conclude by mentioning several extensions that this analysis suggests:

- Although our focus in this paper is unconstrained and bound constrained optimization, the analysis of pattern search can be extended to linearly constrained and nonlinearlly constrained optimization problems [22, 24]. We expect that EPSAs can be generalized to match these convergence analyses. Lewis and Torczon's analysis for linearly constrained problems is also interesting because it suggests how the core search steps can be adapted to accomodate constraint boundaries only when the current iterate is near the constraint boundary. We expect that this can be used to improve the empirical efficiency of bound-constrained EPSAs.

- The analysis of EPSAs in this paper focuses on the relative success of mutation offsets from the best point in each population. Consequently, the mutation step length is uniform across the entire population. If crossover is not used in an EPSA, this restriction can be relaxed, which may improve the performance of these methods by allowing the mutation step length to be locally adapted throughout the search space.

- Although our analysis of EPSAs indicated how EPSAs use an implicit basis matrix, these methods can be adapted to use an explicit basis matrix that rotates and scales the search offsets. For example, if variables are known to differ by several orders of magnitude, then this can be taken into account by the appropriate choice of the basis matrix. Previously, researchers have examined problems for which coordinate transformations were shown to provide measurable changes in the performances of EAs (e.g., see [33]). Thus the use of a different basis matrix could be interesting, particularly when the basis matrix can be used to incorporate *prior* knowledge of an objective function into the EA.

- The convergence theory for sequential EPSAs should be extensible to related parallel EAs. For example, consider an island model EA for which each population begins with the same step size and reduces the step size by the same fraction. For this EA, the combined populations can be shown to lie on a common translated, scaled integer lattice. This is an important property of pattern search methods. If we view the steps taken by all processors as patterns with respect to the best individual on all processors, then the convergence theory applies to this island model EA.

Acknowledgements

We thank John DeLaurentis, Virginia Torczon, Michael Lewis, Bruce Hendrickson and Juan Meza for their helpful discussions. We also thank Margaret Wright for detailed comments on an earlier version of this paper. This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

References

[1] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A survey of evolution strategies. In R. K. Belew and L. B. Booker, editors, *Proc. of the Fourth Intl. Conf. on Genetic Algorithms*, pages 2–9, San Mateo, CA, 1991. Morgan-Kaufmann.

[2] T. Bäck, G. Rudolph, and H.-P. Schwefel. Evolutionary programming and evolution strategies: Similarities and differences. In *Proc. of Second Annual Conf. on Evolutionary Programming*, pages 11–22, 1993.

[3] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[4] H.-G. Beyer. Toward a Theory of Evolution Strategies: Self-Adaptation. *Evolutionary Computation*, 3:311–347, 1995.

[5] C. Davis. Theory of positive linear dependence. *American Journal of Mathematics*, pages 733–746, 1954.

[6] L. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[7] J. E. Dennis and V. J. Torczon. Derivative-free pattern search methods for multidisciplinary design problems. In *The fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 922–932, 1994.

[8] J. J. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.

[9] J. L. Doob. *Stochastic Processes*. John Wiley and Sons, Inc., New York, 1953.

[10] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval schemata. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 187–202. Morgan-Kauffmann, San Mateo, CA, 1993.

[11] D. B. Fogel. An introduction to simulated evolutionary optimization. *IEEE Trans Neural Networks*, 5(1):3–14, 1994.

[12] D. B. Fogel. *Evolutionary Computation.* IEEE Press, Piscataway, NJ, 1995.

[13] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization.* Academic Press, 1981.

[14] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Co., Inc., 1989.

[15] D. E. Goldberg. The theory of virtual alphabets. In H.-P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature*, pages 13–22, New York, 1990. Springer-Verlag.

[16] W. E. Hart. A generalized stationary point convergence theory for evolutionary algorithms. In T. Baeck, editor, *Proc 7th Intl Conf on Genetic Algorithms (ICGA97)*, pages 127–134, San Francisco, CA, 1997. Morgan Kaufmann.

[17] W. E. Hart. A stationary point convergence theory for evolutionary algorithms. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 325–342, San Fransico, CA, 1997. Morgan Kaufmann Publishers, Inc.

[18] W. E. Hart. On the application of evolutionary pattern search algorithms. In *Proc Evolutionary Programming VII*, pages 303–312, New York, 1998. Springer.

[19] W. E. Hart. Comparing evolutionary programs and evolutionary pattern search algorithms: A drug docking application. In *Proc. Genetic and Evolutionary Computation Conf*, 1999. (to appear).

[20] W. E. Hart and K. Hunter. A performance analysis of evolutionary pattern search with generalized mutation steps. In *Proc Conf Evolutionary Computation*, 1999. (to appear).

[21] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1976.

[22] M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM J Opt*, 1998. (to appear).

[23] M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. *Mathematical Programming*, 1998. (submitted).

[24] R. M. Lewis and T. V. J. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM J Opt*, 1999. (submitted).

[25] R. M. Lewis and V. J. Torczon. Pattern search methods for bound constrained minimization. *SIAM J Optimization*, 1997. (to appear).

[26] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space part I: Basic properties of selection and mutation. *IEEE Trans. on Neural Networks*, 5(1):102–119, 1994.

[27] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space part II: Analysis of the diversification role of crossover. *IEEE Trans. on Neural Networks*, 5(1):120–129, 1994.

[28] G. Rappl. *Konvergenzraten von Random Search Verfahren zur globalen Optimierung.* PhD thesis, HSBw München, Germany, 1984.

[29] G. Rappl. On linear convergence of a class of random search algorithms. *Zeitschrift f. angew. Math. Mech.*, 69(1):37–45, 1989.

[30] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Trans Neural Networks*, 5(1):96–101, 1994.

[31] G. Rudolph. Local convergence rates of simple evolutionary algorithms with cauchy mutations. *IEEE Trans Evolutionary Computation*, 1(4):249–258, 1998.

[32] G. Rudolph. Self-adaptation and global convergence: A counter example. In *Proc CEC99*, 1999. (to appear).

[33] R. Salomon. Performance degradation of genetic algorithms under coordinate rotation. In L. J. Fogel, P. Angeline, and T. Bäck, editors, *Proc. of the Fifth Annual Conf. on Evolutionary Programming*, pages 155–161, Cambridge, Massachusetts, 1996. MIT Press.

[34] H.-P. Schwefel. *Evolution and Optimum Seeking.* John Wiley & Sons, New York, 1995.

[35] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM J. Optimization*, 1:123–145, 1991.

[36] V. Torczon. On the convergence of pattern search methods. *SIAM J Optimization*, 7(1):1–25, Feb 1997.

[37] V. Torczon, February 1999. Personal Communication.

[38] A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science.* Springer-Verlag, 1989.

[39] A. H. Wright. Genetic algorithms for real parameter optimization. In G. J. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 205–218. Morgan-Kauffmann, San Mateo, CA, 1991.

[40] M. Wright. Direct search methods: Once scorned, now respected. In *Proc 1995 Dundee Biennial Conf in Numerical Analysis*, pages 191–208, Harlow, United Kingdom, 1996. Addison Wesley Longman.