Soft Comput (2005) 9: 225–235
DOI 10.1007/s00500-004-0376-5

FOCUS

William E. Hart

Rethinking the design of real-coded evolutionary algorithms: Making discrete choices in continuous search domains

Published online: 4 October 2004

Abstract Although real-coded evolutionary algorithms (EAs) have been applied to optimization problems for over thirty years, the convergence properties of these methods remain poorly understood. We discuss the use of discrete random variables to perform search in real-valued EAs. Although most real-valued EAs perform mutation with continuous random variables, we argue that EAs using discrete random variables for mutation can be much easier to analyze. In particular, we present and analyze two simple EAs that make discrete choices of mutation steps.

Keywords Convergence · Evolutionary algorithms · Self-adaptation · Real-coded · Evolution Strategies

1 Introduction

Real-coded evolutionary algorithms (EAs) optimize a function defined on $\mathbf{R}^n$ using vectors of floating point numbers [20]. Real-coded representations have been used by Evolutionary Programming (EP) and Evolution Strategies (ES) since the 1960s, and they have been widely applied with Genetic Algorithms since the early 1990's. Despite their popularity, the convergence properties of real-coded EAs are essentially unknown. Although a wide range of mutation and recombination operators have been developed for these EAs (e.g., see [20]), the impact of these operators on the convergence properties of real-coded EAs has been largely unexplored [25].

One of the particular challenges for real-coded EAs is the need to perform both global and local search at different scales of resolution. The design and application of EAs is motivated by their ability to effectively search broadly to find near-optimal points. However on continuous domains, local refinement of solutions is also needed to help ensure that the final population contains locally-optimal points. Most real-coded EAs perform local refinement with a mutation operator. In particular, adaptive methods that dynamically rescale step length parameters used for mutation have proven particularly effective [8].

Unfortunately, the analysis of adaptive real-coded EAs has proven quite challenging. We argue that one of the reasons is that the formulation of common real-coded EAs makes them difficult to analyze. In particular, commonly used mutation operators generate new points from a continuous distribution. Consequently, it is quite difficult to characterize how the EAs search progresses, even from one iteration to the next! For example, Qi and Palmieri [21] use a discrete time stochastic process to model the time-evolution of the probability density functions that characterize the distribution of the entire population in a real-coded EA. However, this analysis only provides broad insight into how mutation and selection interact in their model.

In this paper, we consider real-coded EAs that perform mutation using steps generated by a discrete random variable. These EAs have a finite number of mutation steps to choose from, so we call them discrete-choice real-coded EAs (DCRC-EAs). For example, if the current point is x, then a new point x' is generated by adding $s \in \mathbf{Q}^n$, where s is selected from a *finite* set of possible mutation steps. The use of discrete choices in a mutation operator significantly simplifies the search dynamics of real-coded EAs. Additionally, it also provides mathematical structure that can be leveraged to more effectively characterize their convergence properties. For example, the probability density function that characterizes the distribution of the entire population has a finite number of possible states in any iteration.

We argue that discrete-choice mutation is a design principle that can be employed to ensure robust

W. E. Hart
Sandia National Laboratories, P. O. Box 5800,
MS 1110, Albuquerque, NM 87185-1110, USA
Tel.: +1-505-8442217
Fax: +1-505-8457442
E-mail: wehart@sandia.gov

convergence for real-coded EAs. To illustrate this, we describe convergence theories for two DCRC-EAs. First, we consider a self-adaptive $(1, \lambda)$-ES. If we discretize the step length update and the mutation steps, then we can show that this ES converges on symmetric, unimodal one-dimensional problems. Second, we consider an explicitly adaptive $(1 + \lambda)$-EPSA. Evolutionary pattern search algorithms (EPSAs) ensure that all mutation steps about a point are sampled before the step length is reduced. EPSAs use a finite, "well-distributed" set of mutation steps to ensure that they weakly converge to stationary points of continuously differentiable functions. These two analyses are simplifications of more general convergence theories [14, 15, 16, 18]. However, our focus is on illustrating how the use of discrete-choice mutation in these EAs provides mathematical structure that can be leveraged to demonstrate robust convergence properties.

The next two sections describe convergence theories for the $(1, \lambda)$-ES and $(1 + \lambda)$-EPSA. We have made an effort to keep the notation in these analyses similar to our previous work [14, 15, 16, 18]. Consequently, some of the notation used in these analyses is inconsistent. However, each of these analyses is self-contained.

2 Self-adaptive (1, λ)-ES

The distinguishing feature of self-adaptive EAs is that the control parameters are evolved by the EA (e.g., see [3, 4, 26]). The idea behind this approach is that individuals with well-scaled step lengths will evolve more rapidly and thus there is evolutionary pressure to both optimize an individual's real parameters as well as its step length (in general, the step length may be represented by one or more parameters). Self-adaptation is a central feature of EAs like evolution stategies (ES) and evolutionary programming (EP), which are applied to continuous design spaces. Although several authors have developed convergence theories for explicitly adaptive EAs, Auger [2], Beyer [5, 6] and Hart et al. [19] appear to have developed the only theoretical investigations of self-adaptive EAs.

We consider the convergence properties of the self-adaptive $(1, \lambda)$-ES, described in Fig. 1. This ES generates λ new points in each iteration and selects the best point generated for the next iteration. This ES typically updates the mutation scale σ_t^i with a log-normal random variable, D_t^i, and the new points x_t^i with a normal random variable, B_t^i. However, the use of these continuous random variables significantly complicates the analysis of this ES. Beyer [4, 5] notes that these EAs can be described by an inhomogeneous Markovian process, and that the stochastic evolution of the system can be expressed by Chapman-Kolmogorov equations. However, he further notes that a direct treatment of these equations is generally quite difficult when using log-normal and normal random variables. Thus Beyer treats the (μ, λ)-ES as a dynamical system from which simpler dynamical systems are derived and validated.

Given x_0, σ_0
For $t = 1, \ldots$
  For $k = 1 : \lambda$
    $\sigma_t^k = \sigma_{t-1} \cdot D_t^k$
    $x_t^k = x_{t-1} + \sigma_t^k \cdot B_t^k$
  End
  $j = \arg\min_{k=1:\lambda} f(x_t^k)$
  $x_t = x_t^j$
  $\sigma_t = \sigma_t^j$
End

Fig. 1 The self-adaptive $(1, \lambda)$-ES for one-dimensional problems. D_t^k and B_t^k are random variables described in the text

The dynamics of the $(1, \lambda)$-ES can be significantly simplified by considering discrete random variables D_t^k and B_t^k. If both of these random variables are discrete, then there are a finite number of possible individuals that can be generated in each iteration. Consequently, the expected behavior of the EA can be well-characterized from one iteration to the next without resorting to approximations of the underlying stochastic process.

The remainder of this section describes the convergence properties of the $(1, \lambda)$-ES with discrete random variables. We demonstrate that this ES converges almost surely to the global optimum of a symmetric, unimodal objective function. If Y and Y_t are random variables, then we say that the sequence $\{Y_t\}_{t \geq 0}$ converges almost surely to Y if $P\{\lim_{t \to \infty} Y_t = Y\} = 1$. We write this as $Y_t \overset{a.s}{\to} Y$. See Grimmett and Stirzaker [12] for a thorough discussion of stochastic convergence.

2.1 A discrete ES

At any iteration, we consider a $(1, \lambda)$-ES that updates the step length σ_t^k by (1) contracting σ_t by $\gamma < 1$, (2) simply setting it equal to σ_t or (3) expanding σ_t by $\eta > 1$. These updates are generated by a discrete random variable D_t^k, which generates the values γ, 1, and η with fixed probabilities v_1, v_2, and v_3 respectively. The step length σ_t^k is used to generate the point x_t^k by simply adding or subtracting this value: $x_t^k = x_t \pm \sigma_t^k$. Thus the random variable B_t^k generates -1 and $+1$ with equal probability.

Let Algorithm A denote the self-adaptive $(1, \lambda)$-ES that employs these discrete random variables. In each iteration, Algorithm A can generate at most six possible new individuals (x_{t+1}, σ_{t+1}):

$$\begin{aligned} x_{t+1} &= x_t \pm \gamma\sigma_t \quad & \sigma_{t+1} &= \gamma\sigma_t \\ x_{t+1} &= x_t \pm \sigma_t \quad & \sigma_{t+1} &= \sigma_t \\ x_{t+1} &= x_t \pm \eta\sigma_t \quad & \sigma_{t+1} &= \eta\sigma_t \end{aligned}$$

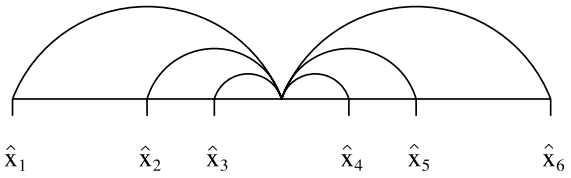

Fig. 2 Illustration of the six possible points that can be generate from a point x_t by Algorithm A. Note that the step length is expanded by η for points $\hat{x}_1$ and $\hat{x}_6$, and it is contracted by γ for points $\hat{x}_3$ and $\hat{x}_4$

Let $\hat{\sigma}_i$ and $\hat{x}_i$ refer to the i-th possibility, which are illustrated in Fig. 2. Generically, each of these has some probability of being generated, $\rho_i > 0$, which is a product of the probabilities of D_t^k and B_t^k (e.g., $\rho_3 = v_1/2$).

We model the way that the new points are generated in Algorithm A by ranking them in decreasing order according to their proximity to the optimum, determined by their $f(x)$ values. The values of ρ_i are used to calculate the ranked probability for each point. The ranked probablity p_i represents the probability that the point x_{t+1}, generated from x_t, is at the position $\hat{x}_i$. The population size λ that is used in this calculation represents the number of samples taken at a particular iteration of Algorithm A. For example, suppose we have a ranking where the order $\{\hat{x}_1, \hat{x}_6, \hat{x}_2, \hat{x}_5, \hat{x}_3, \hat{x}_4\}$ reflects the distance from the optimum from farthest to nearest. Then the ranked probabilities are calculated as follows:

$$
\begin{aligned}
p_1 &= (\rho_1)^\lambda \\
p_6 &= (\rho_6 + \rho_1)^\lambda - (\rho_1)^\lambda \\
p_2 &= (\rho_2 + \rho_6 + \rho_1)^\lambda - (\rho_6 + \rho_1)^\lambda \\
p_5 &= (\rho_5 + \rho_2 + \rho_6 + \rho_1)^\lambda - (\rho_2 + \rho_6 + \rho_1)^\lambda \\
p_3 &= (\rho_3 + \rho_5 + \rho_2 + \rho_6 + \rho_1)^\lambda - (\rho_5 + \rho_2 + \rho_6 + \rho_1)^\lambda \\
p_4 &= 1 - (\rho_3 + \rho_5 + \rho_2 + \rho_6 + \rho_1)^\lambda
\end{aligned}
$$

2.2 Search dynamics

We consider the search dynamics of Algorithm A when applied to a one-dimensional, symmetric, unimodal objective function. Formally, we consider functions that satisfy the following assumption:

Assumption 1. *The function $f : \mathbf{R} \to \mathbf{R}$ has the property that*

1. There exists a unique global minimum $x^* = 0$,
2. f is strictly monotonically increasing for $x \in (x^*, \infty)$.
3. $f(x) = f(-x), \forall x$.

Figure 3 illustrates some functions that are consistent with Assumption 1. Assumption 1 requires that f be unimodal, but it is quite weak otherwise. In particular, Assumption 2 does *not* require that f be continuous, and the global optimum can be at an isolated point. Note that we assume that $x^* = 0$ only for convenience, since if an EA converges on a function that satisfies this condition, then we can show convergence for any other function h with nonzero global minimizer by considering the convergence of the function $f(x) = h(x + x^*)$.

Let X_t^λ and Σ_t^λ be random variables that describe the distribution of the values of x_t and σ_t respectively when a population of size λ is used by Algorithm A on a function that satisfies Assumption 1. There are various metrics for demonstrating that Algorithm A converges to x^*. In the next section we will demonstrate that $\Sigma_t^\lambda \xrightarrow{a.s.} 0$ and $X_t^\lambda \xrightarrow{a.s.} 0$. For now, though, we consider the expected behavior of X_t^λ and Σ_t^λ from one iteration to the next. In particular, we wish to show that the expected value of X_{t+1}^λ is closer to x^* than x_t is to x^λ. Formally, this is equivalent to

$$E(|X_{t+1}^\lambda| : X_t^\lambda = x_t, \Sigma_t^\lambda = \sigma_t) \leq |X_t^\lambda| \ . \tag{1}$$

Similarly, we wish to show that the expected value of Σ_t^λ is less than σ_t. Unfortunately, the following two examples illustrate that we cannot guarantee that the expected behavior of X_t^λ and Σ_t^λ is improving from any particular point in the search.

Example 1 Let $f(x) = |x|$, $\gamma = 0.75$, $\eta = 1.25$, and let $v_1 = v_2 = v_3 = 1/3$. Now suppose that $x_t = 100$ and $\sigma_t = 10$. Figure 4a illustrates the search dynamics in this case. It is clear that the probability that $x_{t+1} = \hat{x}_i$ is greater than the probability that $x_{t+1} = \hat{x}_{i+1}$ in all cases. Thus, it is easy to show that $E(|X_{t+1}^\lambda| : X_t^\lambda = x_t, \Sigma_t^\lambda = \sigma_t) \leq |X_t^\lambda|$. Now as $\lambda \to \infty$ the probability that $x_{t+1} = \hat{x}_1$ goes to one. But this step is generated after expanding the step length. Consequently, the expected value of $\Sigma_{t+1}^\lambda > \sigma_t$ for sufficiently large values of λ.

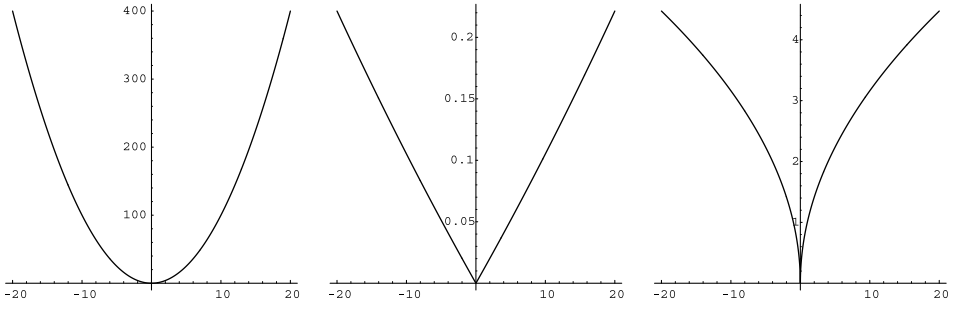

Fig. 3 Examples of functions that satisfy Assumption 1

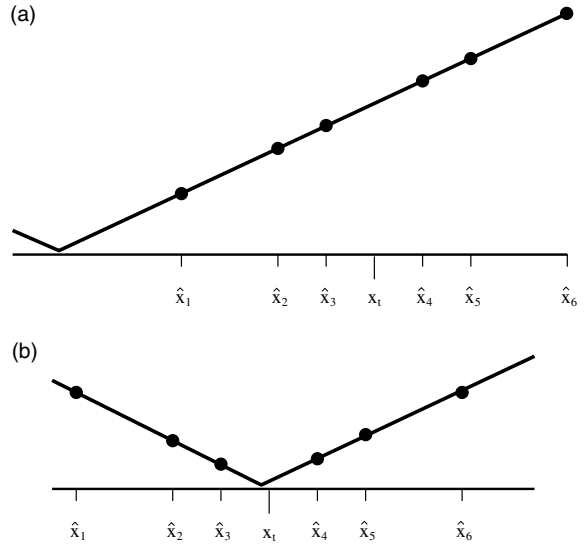

Fig. 4 Examples of the search dynamics of Algorithm A: (**a**) $x_t \gg \sigma_t$, and (**b**) $x_t \ll \sigma_t$

Example 2 Now consider the same algorithmic parameters as in Example 1, but let $x_t = 100$ and $\sigma_t = 1000$. Figure 4b illustrates the search dynamics in this case. Note that *every* possible value for X_{t+1}^{λ} is farther from x^* than x_t. Thus the expected value of X_{t+1}^{λ} is greater than x_t. However, it is clear that the event that $x_{t+1} = \hat{x}_3$ has the greatest probability. Since the step length is contracted in this case, it follows that the expected value of $\Sigma_{t+1}^{\lambda} < \sigma_t$ for sufficiently large values of λ.

These two examples clearly illustrate that we cannot expect the values of X_t^{λ} and Σ_t^{λ} to decrease independently. However, it does appear that these variables converge in a complementary fashion: when X_t^{λ} is expected to increase, Σ_t^{λ} is expected to decrease and visa versa. These observations led us to consider the convergence of the random variable $Z_t^{\lambda} = |X_t^{\lambda}| + W_{\gamma,\eta}\Sigma_t^{\lambda}$, for a constant $W_{\gamma,\eta}$ described below. In particular, we can show that $E(Z_{t+1}^{\lambda} : X_t^{\lambda} = x_t, \Sigma_t^{\lambda} = \sigma_t) \leq Z_t^{\lambda}$, which implies that Algorithm A decreases Z_t^{λ} on average. This confirms our intuition that either X_t^{λ} or Σ_t^{λ} are expected to decrease in any given iteration.

2.3 Analysis of Z_t^{λ}

In this section we describe a general convergence theory for Algorithm A when applied to functions that satisfy Assumption 2. We make the following additional assumption concerning the parameterization of Algorithm A

Assumption 2. *Algorithm A has the property that*

1. $1/2 < \gamma < 1 < \eta < 1/\gamma$.
2. $x_0 \in \mathbf{R} \setminus \mathbf{Q}$ *and* $\sigma_0, \gamma, \eta \in \mathbf{Q}$.

The first part of this assumption specifies that the step lengths do not contract too quickly and that the step lengths do not expand too much relative to the rate of contraction. The second part of this assumption is used to simplify our analysis by eliminating the possibility of ties.

Lemma 1. *Suppose that Algorithm A satisfies Assumption 2 and f satisfies Assumption 1. Then from any iteration* (x_t, σ_t), *the points* $\hat{x}_1, \ldots, \hat{x}_6$ *that can be generated all have distinct function values.*

Proof. Given x_t and σ_t, suppose towards a contradiction that there exist $\hat{x}_i$ and $\hat{x}_j$ such that $f(\hat{x}_i) = f(\hat{x}_j)$. Because f is symmetric and strictly monotonically increasing for $x > 0$, it follows that $\hat{x}_i = -\hat{x}_j$.

Note that $\sigma_t \in \mathbf{Q}$ for all t, since $\sigma_0 \in \mathbf{Q}$ and the step lengths are only contracted and expanded by rational factors. For some $b_i, b_t \in \{-1, 1\}$ and $d_i \in \{\gamma, 1, \eta\}$, we can rewrite $\hat{x}_i$ as follows:

$$\hat{x}_i = x_t + \sigma_t b_i d_i = x_0 + \sum_{k=1}^{t} \sigma_k b_k + \sigma_t b_i d_i = x_0 + \bar{\sigma}_i,$$

where $\bar{\sigma}_i$ is a rational value that reflects the offset of $\hat{x}_i$ from x_0. We can rewrite $\hat{x}_j$ in a similar manner, so we can rewrite the expression $\hat{x}_i = -\hat{x}_j$ as follows:

$$x_0 + \bar{\sigma}_i = -x_0 - \bar{\sigma}_j.$$

This implies that $x_0 = -(\bar{\sigma}_i + \bar{\sigma}_j)/2$, which is a rational value. But this contradicts Assumption 2, which specifies that x_0 is irrational. Thus we cannot have two points $\hat{x}_i$ and $\hat{x}_j$ with equal values. □

The crux of the convergence analysis in this section is that the value of Z_t^{λ} decreases in expectation from any given iteration (x_t, σ_t). This is stated formally in the following theorem, though we defer the proof of this theorem until the end of this section. We use the value

$$W_{\gamma,\eta} = \xi \frac{\gamma}{1-\gamma} + (1-\xi)\frac{1}{\eta - 1},$$

for any $\xi \in (0, 1)$, to define Z_t^{λ} for the remainder of our analysis.

Theorem 1. *Suppose that Algorithm A satisfies Assumption 2, and suppose that f satisfies Assumption 1. Then there exists* $\lambda_0 > 0$ *such that for all* $\lambda \geq \lambda_0$, $E(Z_{t+1}^{\lambda} : X_t^{\lambda} = x_t, \sum_t^{\lambda} = \sigma_t) < Z_t^{\lambda}$.

The following two lemmas will be used throughout our analysis.

Lemma 2. *If Algorithm A satisfies Assumption 2, then*

$$\frac{\gamma}{1-\gamma} < W_{\gamma,\eta} < \frac{1}{\eta - 1}.$$

Proof. It is clear that $W_{\gamma,\eta}$ is between these two extremes, so it suffices to show that

$$\frac{\gamma}{1-\gamma} < \frac{1}{\eta - 1}.$$

We can rewrite this to get

$$0 < \frac{1-\gamma\eta}{(1-\gamma)(\eta-1)}.$$

Both terms in the denominator are positive, and the numerator is positive since $\eta < 1/\gamma$. □

Lemma 3. *Let $A > 0$. Then we can rewrite the function $h(d,A) = |d-A| - |d|$ as follows*:

$$h(d,A) = \begin{cases} A, & d \le 0 \\ A - 2d, & 0 \le d \le A \\ -A, & d \ge A \end{cases}.$$

Proof. The three cases in the definition of $h(d,A)$ follow by considering when $d-A$ and d are positive and negative. □

We now consider how $E(Z^\lambda_{t+1} : X^\lambda_t = x_t, \Sigma^\lambda_t = \sigma_t)$ compares with Z^λ_t. Given x_t and σ_t, we know that $\hat{\sigma}_i = d_i\sigma_t$ and $\hat{x}_i = x_t + b_i d_i \sigma_t$, where $d_i \in \{\gamma, 1, \eta\}$ and $b_i \in \{-1, 1\}$. Thus we have

$$\begin{aligned} &E(Z^\lambda_{t+1} : X^\lambda_t = x_t, \Sigma^\lambda_t = \sigma_t) \\ &\quad = E(|X^\lambda_{t+1}| + W_{\gamma,\eta}\Sigma^\lambda_{t+1} : X^\lambda_t = x_t, \Sigma^\lambda_t = \sigma_t) \\ &\quad = \sum_{i=1}^{6} p_i\big(|\hat{x}_i| + W_{\gamma,\eta}\hat{\sigma}_i\big) \\ &\quad = \sum_{i=1}^{6} p_i(|x_t + d_i b_i \sigma_t| + W_{\gamma,\eta} d_i \sigma_t), \end{aligned}$$

where the probabilities p_i reflect the probability of generating $\hat{x}_i$ from x_t with population size λ. Now consider the inequality

$$\sum_{i=1}^{6} p_i(|x_t + d_i b_i \sigma_t| + W_{\gamma,\eta} d_i \sigma_t) \le \big(|x_t| + W_{\gamma,\eta}\sigma_t\big),$$

which is equivalent to

$$\sum_{i=1}^{6} p_i\big(|x_t + d_i b_i \sigma_t| - |x_t| + W_{\gamma,\eta}\sigma_t(d_i - 1)\big) \le 0. \tag{2}$$

Let $g(x_t, \sigma_t, \lambda)$ be the left hand side of (2), and let $\bar{a}_j$ and $\bar{p}_j$ refer to the term in g that corresponds to the j-th best rank. Thus we can write g abstractly as

$$g(x_t, \sigma_t, \lambda) = \sum_{i=1}^{6} \bar{p}_i \bar{a}_i,$$

noting that the values $\bar{p}_i$ depend on λ. The following proposition demonstrates that the term in g corresponding to the best point has a strictly negative coefficient. Thus for sufficiently large λ this term dominates the value of g.

Proposition 1. *Suppose that Algorithm A satisfies Assumption 2, and suppose that f satisfies Assumption 1. Let $g(x_t, \sigma_t, \lambda) = \sum_{i=1}^{6} \bar{a}_i\bar{p}_i$. Then there exist constants $C < 0$ and $D > 0$, independent of λ, such that $g(x_t, \sigma_t, \lambda) < \sigma_t(C\bar{p}_1 + 5D(1-\bar{p}_1))$*

Proof. Without loss of generality, suppose that $x_t \ge x^* = 0$. From Lemma 1 we know that there is a unique best point amongst the six possible points. Thus we can write g as

$$g(x_t, \sigma_t, \lambda) = \sum_{i=1}^{6} \bar{a}_i\bar{p}_i = \bar{a}_1\bar{p}_1 + \sum_{i=2}^{6} \bar{a}_i\bar{p}_i \le \bar{a}_1\bar{p}_1 + (1-\bar{p}_1)\sum_{i=2}^{6} \bar{a}_i.$$

It follows from Lemma 3 that $|x_t \pm \sigma_t b_i d_i| - |x_t| \le \sigma_t\eta$, so $\bar{a}_i \le \sigma_t(\eta + W_{\gamma,\eta}(\eta-1))$ for all i. Thus we have $D = \eta + W_{\gamma,\eta}(\eta-1)$.

The following three cases show that there exists $C < 0$ such that $\bar{a}_1 \le \sigma_t C$. We consider the case where $0 \le x_t$ and $\hat{x}_1$, $\hat{x}_2$ or $\hat{x}_3$ is the best point; the analysis when $x_t \le 0$ follows similarly.

Case 1: If $\hat{x}_1 = x_t - \sigma_t\eta$ is the best point, we know that $f(x_t - \sigma_t) > f(x_t - \sigma_t\eta)$. Consequently, from the symmetry of f it follows that $x_t > \sigma_t(\eta+1)/2$. From Lemma 3 we have $|x_t - \sigma_t\eta| - |x_t| < -\sigma_t$ for $x_t > \sigma_t(\eta+1)/2$. Thus

$$\begin{aligned} \bar{a}_1 &= |x_t - \sigma_t\eta| - |x_t| + W_{\gamma,\eta}\sigma_t(\eta-1) < \sigma_t(-1 + W_{\gamma,\eta}(\eta-1)) \\ &< \sigma_t\left(-1 + \left(\frac{1}{\eta-1}\right)(\eta-1)\right) = 0. \end{aligned}$$

This last inequality follows from Lemma 2. Thus we have $\bar{a}_1 < \sigma_t C_1$, where $C_1 = -1 + W_{\gamma,\eta}(\eta-1) < 0$.

Case 2: If $\hat{x}_2 = x_t - \sigma_t$ is the best point, we know that $f(x_t - \sigma_t\gamma) > f(x_t - \sigma_t)$. Consequently, from the symmetry of f it follows that $x_t > \sigma_t(1+\gamma)/2$. From Lemma 3 we have $|x_t - \sigma_t| - |x_t| < -\gamma\sigma_t$ for all $x_t > \sigma_t(1+\gamma)/2$. Thus

$$\bar{a}_1 = |x_t - \sigma_t| - |x_t| < -\gamma\sigma_t < 0.$$

Thus we have $\bar{a}_1 < \sigma_t C_2$, where $C_2 = -\gamma < 0$.

Case 3: If $\hat{x}_3 = x_t - \sigma_t\gamma$ is the best point, we know that $f(x_t + \sigma_t\gamma) > f(x_t - \sigma_t\gamma)$. Now $x_t \ge 0$, so from Lemma 3 we have $|x_t - \sigma_t\gamma| - |x_t| \le \sigma_t\gamma$ for all $x_t \ge 0$. Thus

$$\begin{aligned} \bar{a}_1 &= |x_t - \sigma_t\gamma| - |x_t| + W_{\gamma,\eta}\sigma_t(\gamma-1) \\ &\le \sigma_t(\gamma + W_{\gamma,\eta}(\gamma-1)) < \sigma_t\left(\gamma + \left(\frac{\gamma}{1-\gamma}\right)(\gamma-1)\right) = 0. \end{aligned}$$

This last inequality follows from Lemma 3. Thus $\bar{a}_1 < \sigma_t C_3$, where $C_3 = \gamma + W_{\gamma,\eta}(\gamma-1) < 0.$,

To conclude, let $C = \max_i C_i < 0$. □

Proposition 1 is the main result use to prove Theorem 1.

Proof. **[Theorem 1]** Note that $g(x_t, \sigma_t, \lambda) = E(Z^\lambda_{t+1} : X^\lambda_t = x_t, \Sigma^\lambda_t = \sigma_t) - Z^\lambda_t$. It follows from Proposition 1 that there exist constants $C < 0$ and $D > 0$, independent of λ, for which $g(x_t, \sigma_t, \lambda) < \sigma_t(C\bar{p}_1 + 5D(1-\bar{p}_1))$. We know

that $\bar{p}_1 \geq 1-(1-\kappa)^{\lambda}$, for some constant κ. Thus $\lim_{\lambda\to\infty}\bar{p}_1 = 1$, so there exists λ_0 such that for all $\lambda \geq \lambda_0$, $\bar{p}_1 > \dfrac{-5D(1-\bar{p}_1)}{C}$.

Thus for $\lambda \geq \lambda_0$, $E(Z_{t+1}^{\lambda} : X_t^{\lambda} = x_t, \Sigma_t^{\lambda} = \sigma_t) - Z_t^{\lambda} = g(x_t, \sigma_t, \lambda) < 0$. □

2.4 Super-martingale analysis

Our analysis in the previous section demonstrates that Z_t^{λ} is expected to shrink from one iteration to the next. By itself, this does not necessarily guarantee that X_t^{λ} and Σ_t^{λ} converge as we desire. However, we can use Theorem 1 to demonstrate that Z_t^{λ} almost surely converges to some random variable Z_{∞}^{λ}.

The crux of this result is that Z_t^{λ} is a super-martingale. A random process Y_t is a super-martingale if $E[|Y_t|] < \infty$ and $E[Y_{t+1}|\mathscr{F}_t'] \leq Y_t$, where $\mathscr{F}_t'$ is the family of σ-algebras that describe the events underlying Y_t [12]. Intuitively, a super-martingale is a stochastic process that decreases on average. Let $(\Omega, \mathscr{F}, P)$ be the probability space describing the random events that underly X_t^{λ} and Σ_t^{λ}. $\mathscr{F}_t$ is a sequence of σ-algebras, and we have $E(Z_{t+1}^{\lambda} : \mathscr{F}_t) = E(Z_{t+1}^{\lambda} : X_t^{\lambda} = x_t, \Sigma_t^{\lambda} = \sigma_t)$.

Theorem 2. *Suppose that Algorithm A satisfies Assumption 2 and suppose that f satisfies Assumption 1. Then there exists $\lambda_0 > 0$ such that for all $\lambda \geq \lambda_0$, Z_t^{λ} is a super-martingale with respect to the σ-algebras $\mathscr{F}_t$. Furthermore, there exists a random variable Z_{∞}^{λ} such that $Z_t^{\lambda} \xrightarrow{a.s.} Z_{\infty}^{\lambda}$.*

Proof. From Theorem 1 we know that there exists λ_0 such for all $\lambda \geq \lambda_0$, $E(Z_{t+1}^{\lambda} : \mathscr{F}_t) < Z_t^{\lambda}$. To conclude that Z_t^{λ} is a super-martingale with respect to $\mathscr{F}_t$, note that $E(Z_t^{\lambda}) < \infty$. This follows from the fact that there are a finite number of states that can be reached by Algorithm A after t iterations, and Z_t^{λ} is finite for each of these states. Since Z_t^{λ} is a non-negative super-martingale, it follows that there exists a random variable Z_{∞}^{λ} such that $Z_t^{\lambda} \xrightarrow{a.s.} Z_{\infty}^{\lambda}$. □

Theorem 2 ensures that X_t^{λ} and Σ_t^{λ} almost surely generate a convergent sequence. This result confirms the observation noted above: X_t^{λ} and Σ_t^{λ} converge in a complementary fashion. Furthermore, this result by itself suggests that the ES is behaving in an interesting manner. For example, Theorem 2 confirms that $Z_t^{\lambda}(\omega)$ has a limit point for all random events $\omega \in \Omega$. Thus, Theorem 2 demonstrates that the asymptotic dynamics of Algorithm A can be exactly characterized.

However, this result is not sufficient to demonstrate that both X_t^{λ} and Σ_t^{λ} converge to zero. Although such an analysis is beyond the scope of this paper, we can apply our results in Hart et al. [18] to prove this result.

Theorem 3. *Suppose that Algorithm A satisfies Assumptions 2 and suppose that f satisfies Assumption 1. Then for all $\lambda \geq \lambda_0$, $\Sigma_t^{\lambda} \xrightarrow{a.s.} 0$.*

Let $\lambda_1 \geq \lambda_0$ be the smallest integral value of λ for which $p\log\eta + q\log\gamma > 0$, where $p = 1-\left(1-\frac{v_3}{2}\right)^{\lambda}+\left(\frac{v_3}{2}\right)^{\lambda}$ and $q = \left(\frac{1+v_1}{2}\right)^{\lambda}-\left(\frac{1-v_1}{2}\right)^{\lambda}$. Then for all $\lambda \geq \lambda_1$, $X_t^{\lambda} \xrightarrow{a.s.} 0$.

Proof. These results follow directly from Theorems 2 and 3 [18]. The assumptions on Algorithm A required for these theorems are slightly different than those we require, but these assumptions are simply needed to ensure that there exists Z_{∞}^{λ} for which $Z_t^{\lambda} \xrightarrow{a.s.} Z_{\infty}^{\lambda}$. Furthermore, these theorems do not depend on the specific value of $W_{\gamma,\eta}$ used to formulate Z_t^{λ}, though Theorem 2 [18] does require that $W_{\gamma,\eta}$ has an irrational value. □

These convergence results for Algorithm A confirm common empirical observations that self-adaptive ESs converge to locally optimal points. The fact that Algorithm A is a DCRC-EA is exploited extensively throughout this analysis. Although this analysis does assume limitations on the step length expansion and contraction ratios, we believe that these limitations provide insight into how the search dynamics of a $(1,\lambda)$-ES need to be constrained in order to ensure robust convergence. For example, the requirement for λ_1 in Theorem 4 appears to relate directly to empirical behavior; we have observed experiments in which this condition was violated and Algorithm A failed to converge.

3 Explicitly adaptive (1+λ)-EPSA

Explicitly adaptive EAs have been developed by a number of authors, and convergence theories have been developed for a variety of explicitly adaptive formulations [1, 11, 14, 15, 16, 22, 23, 24, 29]. Recent analyses of EPSAs [14, 15, 16] have examined their convergence behavior on problems of the form

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & x \in \Omega = \{y \in \mathbf{R}^n : l \leq Ay \leq u\}, \end{array} \tag{3}$$

where $l, u \in (\mathbf{R} \bigcup \{\pm\infty\})^m$ and $A \in \mathbf{Q}^{m\times n}$. These results show that if the sequence of best points generated in each iteration, $\{x_t^*\}$, lies in a compact set, then for any continuously differentiable nonlinear function there exists a subsequence that converges to a stationary point of the objective function f. These results are particularly distinguished by their ability to exactly capture the analytic behavior of a class of adaptive EAs on *nonconvex, multimodal* problems. Although some convergence theories have been developed for EAs on nonlinear problems [1, 11, 24], EPSAs are the only class of EAs whose *local* convergence properties have been well-characterized on a broad class of nonlinear optimization problems. Consequently, the analysis of EPSAs provides valuable insight into the search dynamics needed to effectively refine points within an EA's search.

Select an initial point $p_0 = (x_0, \Delta_0, D_0)$, where $x_0 \in \Omega, \Delta_0 \in \mathbf{R}^{>0}, D_0 = \mathcal{D}$
For $t = 1, \ldots$
  For $k = 1 : \lambda$
    Select $s_t^k \in D_t$
    $x_t^k = x_{t-1} + s_t^k \Delta_{t-1}$
  End
  $j = \arg\min_{k=1:\lambda} \ f(x_t^k)$
  if $(f(x_t^j) < f(x_t))$
    $x_{t+1} = x_t^j, \quad \Delta_{t+1} = \theta_t \Delta_t$, and $\quad D_{t+1} = \mathcal{D}$
  else if $(D_t \setminus \{s_t^1, \ldots, s_t^\lambda\} = \emptyset)$
    $x_{t+1} = x_t, \quad \Delta_{t+1} = \phi_t \Delta_t$, and $\quad D_{t+1} = \mathcal{D}$
  else
    $x_{t+1} = x_t, \quad \Delta_{t+1} = \Delta_t$, and $\quad D_{t+1} = D_t \setminus \{s_t^1, \ldots, s_t^\lambda\}$
End

Fig. 5 Algorithm B. An explicitly adaptive $(1 + \lambda)$-EPSA. Note that θ_t and ϕ_t can be selected arbitrarily such that they represent powers of $\tau > 1$ (see Sect. 3). For example, we can have $\theta_t = \tau$ and $\phi_t = \tau^{-1}$ for all t

In this section, we consider the convergence properties of Algorithm B the explicitly adaptive $(1 + \lambda)$-EPSA described in Fig. 5. This EPSA generates λ new points with mutation in each iteration and replaces the current point if a better point is generated. The mutation steps used by Algorithm B are restricted to lie in a finite *pattern* defined by $\mathcal{D}$. We require that $\mathcal{D}$ forms a *positive spanning set* [7]: any point in Ω can be generated by a positive linear combination of the vectors in the pattern. Two examples of simple patterns are shown in Fig. 6. In each iteration, $D_t \subseteq \mathcal{D}$ is the set of mutation steps that have not been tested about x_t.[1] The order in which these steps are selected may be fixed, or they may be selected from a random distribution. Further, such a randomized selection method can be adaptive so long as the probability of selecting each step is greater than a fixed value $\gamma > 0$.

The mutation step length, Δ_t, is only reduced if all mutation steps in $\mathcal{D}$ lead to points with higher function values than x_t. When an improving point is generated, Δ_t may be increased. Let $\tau \in \mathbf{Q}$ such that $\tau > 1$. Algorithm B expands and contracts Δ_t by multiplying or dividing by integral powers of τ. More formally, we increase Δ_t by multiplying by $\theta_t = \tau^{\kappa_t}$ where $\kappa_t \in \{0, 1, \ldots, \kappa_{max}\}$, $\kappa_{max} \in \mathbf{N}$. This includes the case where Δ is not increased. Similarly, we contract Δ_t by multiplying by $\phi_t = \tau^{\kappa_t}$, where $\kappa_t \in \{-\kappa_{\max}, \ldots, -1\}$.

Algorithm B controls step length updates more carefully than traditional $(1 + \lambda)$-ES methods. The motivation for this mechanism is to provide more explicit control of the convergence of the step length parameters. In particular, Algorithm B ensures that the step lengths are not contracted so rapidly that the algorithm fails to generate interesting limit points. The key to this analysis is the fact that this EA uses a discrete random variable to generate mutation steps from a finite set of rational search directions, $\mathcal{D}$. As a consequence, we can show that some subsequence of the step lengths $\{\Delta_t\}$ must converge to zero.

[1] Note that Algorithm B selects mutation steps from D_t with replacement. Selecting steps without replacement would certainly be more efficient, though that complicates the definition of Algorithm B and is dissimilar to many EA implementations. However, our analysis of Algorithm B also applies when steps are selected without replacement.

3.1 Refining subsequences

Consider the points generated by Algorithm B, $\{x_k\}$, and the associated sequence of step lengths, $\{\Delta_k\}$. We make the following general assumption about Algorithm B

Assumption 3. *The following are true for Algorithm B:*

1. The sequence $\{x_k\}$ lies in a compact set,
2. $\forall s \in \mathcal{D}$, $s \in \mathbf{Q}^n$ and $|\mathcal{D}| < \infty$.

Assumption 3.1 is a standard assumption for the analysis of nonlinear optimizers on continuous domains [2]. A reasonable sufficient condition for this to hold is that $L_\Omega(y) = \{x \in \Omega : f(x) \leq f(y)\}$ is compact. Our analysis does not make this assumption because we allow discontinuities and even $f(x) = \infty$ for some x, so $L_\Omega(y)$ may not be closed. However, we could assume that the set is bounded or precompact [9].

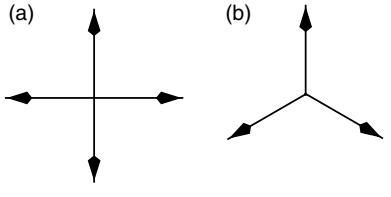

Fig. 6 Illustration of two simple patterns of mutation offsets for EPSAs: (**a**) a pattern of 2n coordinate-wise unit-length, and (**b**) a pattern of n + 1 unit-length offsets generated from a regular simplex

If the sequence $\{x_k\}$ lies in a compact set, there exist convergent subsequences of this sequence. The following definition describes the type of "interesting" convergent subsequences that we will consider in our analysis.

Definition 1. *We say that a convergent subsequence $\{x_k\}_{k\in K}$ (for some set of indices K) is a refining subsequence if*

1. $\Delta_{k_i} > \Delta_{k_{i+1}}$ *for all* $k_i \in K$,
2. $\lim_{k\in K} \Delta_k = 0$, *and*
3. For each x_{k_i}, $f(x_{k_i}) \leq f(x_{k_i} + \Delta_{k_i} s)$ *for all* $s \in \mathscr{D}$.

The following theorem provides the main result of our convergence analysis.

Theorem 4. If Algorithm B satisfies Assumption 3, then there exists a refining subsequence of $\{x_k\}$ with probability one.

To prove Theorem 4, we show that the points generated by Algorithm B lie on a mesh, which implies that the set of possible points at iteration t is finite. It follows that there exists a subsequence of $\{\Delta_t\}$ that converges to zero with probability one.

We now define a mesh that contains the points $\{x_t\}$. Note that Δ_t is generated by expansions and contractions of Δ_0, so we can write $\Delta_t = \Delta_0 \tau^{r_t}$, for some $r_t \in \mathbf{Z}$. Let $r_t^{\max} = \max_{j=0,\ldots,t} r_j$ and $r_t^{\min} = \min_{j=0,\ldots,t} r_j$; since $r_0 = 0$, it follows that $r_t^{\max} \geq 0 \geq r_t^{\min}$. Now $\tau = \tau_n/\tau_d$ where $\tau_n, \tau_d \in Z^{>0}$. We define $\bar{\tau}_t = (\tau_d)^{r_t^{\max}} (\tau_n)^{-r_t^{\min}}$; $\bar{\tau}_t$ is a lower bound on the greatest common denominator of $\Delta_1/\Delta_0, \ldots, \Delta_t/\Delta_0$. Consider the mesh

$$M_t = \left\{ x_0 + \frac{\Delta_0}{\bar{\tau}_t} \sum_{s\in\mathscr{D}} z_s s : z_s \in Z^{>0} \right\}. \tag{4}$$

The set M_t is a mesh defined by the lattice spanned by the directions in $\mathscr{D}$ and scaled by the smallest step length seen so far. Note that $\bar{\tau}_t$ is nondecreasing, which implies that $M_t \subseteq M_{t+1}$. As $\bar{\tau}_t$ increases, the number of points in the mesh increases because smaller fractional points are added to the mesh. Fig. 7 illustrates the mesh M_t and M_{t+1} after a step length has contracted.

The following lemma confirms that the sequence $\{x_t\}$ generated by Algorithm B lies on the meshes $\{M_t\}$.

Lemma 4. For all t, $x_t \in M_t$.

Proof. Clearly $x_0 \in M_0$. By induction we assume that $x_t \in M_t$. If $x_{t+1} = x_t$, then $x_{t+1} \in M_t \subseteq M_{t+1}$. Otherwise, $x_{t+1} = x_t + \Delta_t \bar{s}$ for some $\bar{s} \in \mathscr{D}$. Now $\bar{\tau}_{t+1} = \rho_t \bar{\tau}_t$, where $\rho_t \in \{\tau_d, \ldots, \tau_d^{\kappa_{\max}}, \tau_n, \ldots, \tau_n^{\kappa_{\max}}, 1\}$. Thus we have

$$\begin{aligned} x_{t+1} &= x_t + \Delta_t \bar{s} = x_t + \frac{\Delta_0}{\bar{\tau}_{t+1}} \tau^{r_t} \bar{\tau}_t \rho_t \bar{s} \\ &= x_0 + \frac{\Delta_0}{\bar{\tau}_{t+1}} \Big(\sum_{s\in\mathscr{D}\setminus\{\bar{s}\}} z_s s + (z_{\bar{s}} + \tau^{r_t} \bar{\tau}_t \rho_t) \bar{s} \Big). \end{aligned}$$

But from the definition of $\bar{\tau}_t$, we know that $\tau^{r_t} \bar{\tau}_t \in \mathbf{Z}$. Thus if $z'_s = z_s$ for all $s \in \mathscr{D} \setminus \{\bar{s}\}$ and $z'_{\bar{s}} = z_{\bar{s}} + \bar{\tau}^{r_t} \bar{\tau}_t \rho_t$, then $x_{t+1} = x_0 + \frac{\Delta_0}{\bar{\tau}_{t+1}} \sum_{s\in\mathscr{D}} z'_s s \in M_{t+1}$. □

The following lemma shows that the step lengths are bounded above for all t.

Lemma 5. The step length Δ_t is bounded above by a positive constant independent of t.

Proof. Let Ω' be the compact set that contains the points generated by the EPSA. Since Ω' is bounded, the diameter $d = \max_{a,b\in\Omega'} |a - b|$ is finite. Suppose towards a contradiction that Algorithm B generates a point $x_t \neq x_{t-1}$ with step length greater or equal than $d\tau^{2\kappa_{\max}}$. Now x_t must have been generated via a mutation step with a step length greater or equal to $d\tau^{\kappa_{\max}}$. But if a point has a step length greater than d then any mutation steps about that point will be unsuccessful because they lie outside of Ω'. Thus we have a contradiction, and therefore all step lengths are bounded above by a positive constant independent of t. □

We now use Lemmas 4 and 5 to prove the following proposition, which provides the key result for the proof of Theorem 4. This proposition demonstrates that with probability one some subsequence of the step lengths converges to zero. The basic idea behind this proof is that if the step lengths were bounded away from zero in all cases, then there is a mesh M_∞ that contains all of the points $\{x_t\}$. But this would contradict the fact that Algorithm B forces the step lengths to decrease when all mutation steps in $\mathscr{D}$ have been generated.

Proposition 2. *If the points sampled by Algorithm B lie in a compact set Ω' then*

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

Proof. Suppose that there exists a non-positive integer $\underline{\rho}$ such that $0 < \Delta_0 \tau^{\underline{\rho}} \leq \Delta_t$ for all $t \geq 0$. This implies that $r_t \geq \underline{\rho}$ for all $t \geq 0$. Further, we know from Lemma 5 that there is a non-negative integer $\bar{\rho}$ such that $r_t \leq \bar{\rho}$. Let $\bar{\tau}_\infty = (\tau_d)^{\bar{\rho}} (\tau_n)^{-\underline{\rho}}$, and note that $\bar{\tau}_t \leq \bar{\tau}_\infty$. Consequently, it follows that for all k, $x_k \in M_\infty$, where

$$M_\infty = \left\{ x + \frac{\Delta_0}{\bar{\tau}_\infty} \sum_{s\in\mathscr{D}} z_s s : z_s \in \mathbf{Z}^{>0} \right\}.$$

It follows that the intersection of M_∞ and Ω' is finite. Thus there must exist a point $\hat{x}$ and an index N such that for all $k > N$, $x_k = \hat{x}$. Since each mutation step is selected with a probability bounded away from zero (independent of t), it follows that with probability one there exist infinitely many points in $\{x_k\}$, $k > N$, for which all mutation steps are generated (unsuccessfully) and the step length is contracted. But this implies that $\Delta_k \to 0$ with probability one, which gives a contradiction. □

Given Proposition 2, we prove Theorem 4. A refining subsequence is guaranteed to exist (with probability one) because we have subsequences for which $\Delta_t \to 0$ and for which all mutation steps have been generated.

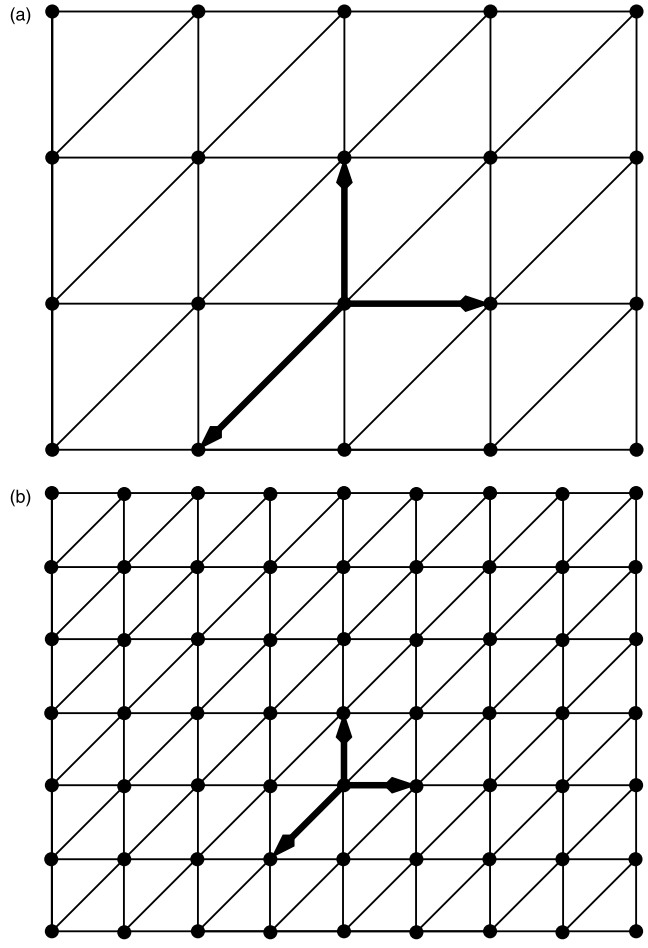

Fig. 7 An example of **a** the mesh M_t generated by the mutation steps $D = \{e_1, e_2, -e_1 - e_2\}$, and **b** the mesh M_{t+1} after the step length is halved

Proof **(Theorem 4).** Consider the subsequence $\{x_k\}_{k\in K}$ that consists of the points for which the step length is contracted after unsuccessfully generating all steps in $\mathscr{D}$, and for which $\Delta_k > \Delta_{k+1}$. From Proposition 1 we know that with probability one $\{x_k\}_{k\in K}$ is an infinite sequence. It follows that it has a convergent subsequence in Ω'. Let this convergent subsequence be $\{x_k\}_{k\in K'}$, and note that $\lim_{k\in K'} \Delta'_k = 0$. Thus we have shown that there exists a refining subsequence of $\{x_k\}$ with probability one. □

Theorem 4 ensures that there exist "interesting" convergent subsequences of the points generated by Algorithm B for which we can prove convergence. However, this provides a weak convergence theory because we can only prove convergence for a subsequence of $\{x_k\}$. We focus on convergent subsequences because the conditions required to ensure that the entire sequence converges are rather restrictive (e.g., see Torczon [28]).

3.2 Analysis of limit points

Given that Algorithm B generates a refining subsequence with probability one, we can apply the convergence theory for EPSAs [15, 16] to describe the type of limit points that Algorithm B can generate. In particular, the smoothness of f at the limit point of a refining subsequence determines the properties of f at that point. The following theorem describes limit points of refining subsequences for general, nonsmooth functions. A natural generalization of the notion of a gradient for nonsmooth functions is the generalized directional derivative [6]. The generalized directional derivative of f at x in the direction s is

$$f^o(x;s) = \limsup_{y\to x, t\downarrow 0} \frac{f(y+ts) - f(y)}{t}.$$

Note that if $f^o(x;s) \geq 0$, then f is increasing in the direction s. Thus a local minimum of a nonsmooth function is defined by a point where $f^o(x;s') \geq 0$ for all $s' \in \mathbf{R}^n$. Recall that f is Lipshitz if $|f(x) - f(y)| \leq C|x - y|$ for some C independent of x and y.

Theorem 5. [Theorem 2 [16]] Let $\hat{x}$ *be the limit of a refining subsequence* $\{x_k\}_{k\in K}$. *If* f *is Lipshitz in the neighborhood of* $\hat{x}$ *then there exists a positive spanning set* $S \subseteq \mathscr{D}$ *such that for all* $s \in S$, $f^o(\hat{x};s) \geq 0$ *if* $x_k + \Delta_k s$ *is feasible for infinitely many* $k \in K$.

Note that Algorithm B does not consider subsets of $\mathscr{D}$ that might be positively spanning (e.g. by updating Δ_t earlier). Thus this theorem shows that for all $s \in \mathscr{D}$, the directional derivative of f at $\hat{x}$ in the direction s is positive. If the refining subsequence converges to a nondifferentiable point, $\bar{x}$, then it may be possible for $f^o(\bar{x};\epsilon s) \geq 0$ for every direction s in a given positive spanning set and for all $\epsilon > 0$ [27].

The next theorem extends the previous result when f is strictly differentiable at a limit point $\hat{x}$. A point x is strictly differentiable if $\bigtriangledown f(x)$ exists and $\bigtriangledown f(x)^T w = \lim_{y\to x, t\downarrow 0} \frac{f(y+tw)-f(y)}{t}$ for all $w \in \mathbf{R}^n$ [6].

Theorem 6. [Theorem 3 [16]] *Let* $\Omega = \mathbf{R}^n$, *and let* $\hat{x}$ *be the limit of a refining subsequence of* $\{x_k\}$. *If* f *is Lipshitz in the neighborhood of* $\hat{x}$ *and* f *is strictly differentiable at* $\hat{x}$, *then* $\bigtriangledown f(\hat{x}) = 0$.

If f is continuously differentiable then for any point x, f is Lipshitz in the neighborhood of x and f is strictly differentiable at x [6]. Thus in this case $\bigtriangledown f(\hat{x}) = 0$ for a limit point $\hat{x}$ of any refining subsequence. However, Theorem 6 is more general because it is applicable to functions for which continuity properties vary across the search domain.

The result given by Theorem 2 can be extended to bound constrained problems if the set $\mathscr{D}$ contains the unit vectors and their opposites: $\{\pm e_1, \ldots, \pm e_n\}$. This restriction ensures that Algorithm B can effectively search along the bound constraint as well as away from the bound constraint. Recall that a first-order constrained stationary point $\hat{x}$ for problem (3) is a Karush-Kuhn-Tucker (KKT) point, where there is no first-order direction of improvement [10].

Theorem 7. Let Ω *be a bound-constrained domain, and let* $\hat{x}$ *be the limit of a refining subsequence of* $\{x_k\}$. *If* $\{\pm e_1, \ldots, \pm e_n\} \subseteq \mathscr{D}$, f *is Lipshitz in the neighborhood of* $\hat{x}$, *and* f *is strictly differentiable at* $\hat{x}$, *then* $\hat{x}$ *is a KKT point*.

Proof. This result follows directly from Theorem 4 [16] since if $\{\pm e_1, \ldots, \pm e_n\} \subseteq \mathscr{D}$ then these search directions are ϵ-conforming directions for any value of ϵ. Consequently, this theorem ensures that $\hat{x}$ is a KKT point. □

These convergence results for Algorithm B ensure convergence on a much broader range of problems than the results for Algorithm A. Although both methods are DCRC-EAs, the careful step length control in Algorithm B provides additional mathematical structure that ensures convergence on a wide range of nonlinear problems. Further, we exploit the fact that Algorithm B is a DCRC-EA only to demonstrate the existence of interesting limit points. Subsequently, the analysis in this section simply follows from the properties of that subsequence.

4 Discussion

Our analysis of Algorithms A and B clearly leverages the fact that their mutation operators make discrete choices. For both methods, the simplified search dynamics allow us to mathematically describe key properties of these methods. As a consequence, our analysis does not approximate the underlying stochastic processes of these EAs. The convergence theories we have described directly reflect the search behavior of these methods. Furthermore, these analyses are applicable to classes of problems, and thus they ensure that convergence is not particular to a specific problem domain.

Although convergence theories have been developed for DCRC-EAs, we have witnessed that EA practitioners are reluctant to use mutation operators that make discrete choices. Despite this, our experience is that these EAs can be effective in practice. For example, empirical studies of EPSAs on test problems and a real-world application confirm that these methods perform a global search that is comparable to standard self-adaptive EAs [13, 19]. Similarly, the preliminary numerical experiments in Hart et al. [18] suggest that self-adaptive ESs like Algorithm A are not inherently less effective than standard $(1, \lambda)$-ES formulations.

There are several theoretical issues that need to be addressed to better assess the practical utility of DCRC-EAs. For example, the lower bound on λ in Theorems 2 and 3 needs to be reconsidered. Although we have demonstrated that practical values of λ are feasible (i.e. $\lambda < 6$), the argument used in our analysis is broad and thus the lower bound on λ is weak. Thus, it is unclear whether $\lambda = 2$ is generally allowable, and how the problem structure (e.g. the Lipshitz constant) impacts the minimal allowable value of λ. Additionally, our convergence analysis for self-adaptive ES can be directly generalized to multi-dimensional problems. We have taken a preliminary step in this direction by demonstrating convergence for self-adaptive ESs on separable, unimodal problems of the form $g(\bar{x}) = \sum_{i=1}^{n} g_i(x_i)$, where g_i are one-dimensional unimodal functions [17]. However, these results need to be generalized to a much broader range of multi-dimensional problems. We also need to generalize these convergence theories to allow for other evolutionary operators, particularly crossover. For example, standard crossover operators can be used with EPSAs without loss of generality in the convergence theory, but no such analysis has been developed for DCRC-EAs. Finally, it remains to be seen whether analyses of the rate of convergence can be developed for DCRC-EAs like these. In fact, the discretizations used in DCRC-EAs may make it more difficult to analyze convergence rates than standard self-adaptive EAs, for which the well-developed analytic techniques for continuous distributions can be applied [5].

Acknowledgements We thank John DeLaurentis and Lauren Ferguson for their collaborations on the analysis of self-adaptive EAs, and anonymous reviewers for their critical feedback. 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. Agapie A (2001) Theoretical analysis of mutation-adaptive evolutionary algorithms. Evol Comput 9(2): 127–146
2. Auger A, Le Bris C, Schoenaver M (2003) Dimension independent convergence rate for non-isotropic $(1,\lambda)$ – ES. (eds.) E. Cantu-paz et al. Pro Genetic and Evolutionary Algorithms Conf 512–524
3. Audet C, Dennis JE (2000) Analysis of generalized pattern searches. Technical Report TR00-07, Rice University, Department of Computational and Applied Mathematics
4. Bäck T, Schwefel H-P (1993) An overview of evolutionary algorithms for parameter optimization. Evol Comput 1(1): 1–23
5. Beyer H-G (1995) Toward a theory of evolution strategies: self-adaptation. Evol Comput 3(3): 311–347
6. H-G Beyer (2001) The Theory of Evolution Strategies. Springer, Berlin Heidelberg New york
7. Clarke FH (1990) Optimization and Nonsmooth Analysis, vol5. SIAM classics in applied mathematics Philadelphia, PA
8. Davis C (1954) Theory of positive linear dependence. American J Math, pp 733–746
9. Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. IEEE Trans Evol Comput 3(2): 124–141
10. Folland GB (1984) Real Analysis - Modern Techniques and Their Applications. Wiley
11. Gill PE, Murray W, Wright MH (1981) Practical Optimization. Academic
12. Greenwood GW, Zhu QJ (2001) Convergence in evolutionary programs with self-adaptation. Evol Comput 9(2):147–158
13. Grimmett GR, Stirzaker DR (1992) Probability and Random Processes, Second Edition. Oxford University Press, Oxford

14. Hart WE (1999) Comparing evolutionary programs and evolutionary pattern search algorithms: A drug docking application. In: Proceedings of Genetic and evolutionary computation Conference pp 855–862
15. Hart WE (2001) A convergence analysis of unconstrained and bound constrained evolutionary pattern search. Evol Comput 9(1):1–23
16. Hart WE (2001) Evolutionary pattern search algorithms for unconstrained and linearly constrained optimization. IEEE Trans Evol Comput 5(4): 388–397
17. Hart WE (2003) Locally-adaptive and memetic evolutionary pattern search algorithms. Evol Comput 11(1):29–52
18. Hart WE, DeLaurentis JM (2003) Convergence of a discretized self-adaptive evolutionary strategy on multi-dimensional problems. IEEE Trans Evol Comput (submitted)
19. Hart WE, DeLaurentis JM, Ferguson LA (2003) On the convergence of an implicitly self-adaptive evolutionary algorithm on one-dimensional unimodal problems. IEEE Trans Evol Comput (to appear)
20. Hart WE, Hunter K (1999) A performance analysis of evolutionary pattern search with generalized mutation steps. In: Proc Cong Evolutionary Computation, pp 672–679
21. Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial Intelligence Review, 12:265–319
22. Qi X, Palmieri F (1994) 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 Netw 5(1): 102–119
23. Rudolph G (1994) Convergence of non-elitist strategies. In: Proceedings of the First IEEE Conference on Evolutionary Computation, Vol1. IEEE Press Piscataway, NJ,pp 63–66
24. Rudolph G (1997) Convergence properties of evolutionary algorithms. Kovač
25. Rudolph G (1997) Convergence rates of evolutionary algorithms for a class of convex objective functions. Control and Cybernetics, 26(3):375–390
26. Rudolph G (1999) Self-adaptation and global convergence: A counter example. In: Proceedings of Congress Evolutionary Computation, pp 646–651
27. Saravanan N, Fogel DB, Nelson KM (1995) A comparison of methods for self-adaptation in evolutionary algorithms. BioSystems, 36(2):157–166
28. Torczon V (1991) On the convergence of the multidirectional search algorithm. SIAM J Optimization, 1(1):123–145
29. Torczon V (1997) On the convergence of pattern search methods. SIAM J Optimization, 7(1):1–25
30. Yin G, Rudolph G, Schwefel H-P (1996) Analyzing $(1, \lambda)$ evolution strategy via stochastic approximation methods. Evol Comput 3(4):473–489