Evolutionary Programming and Evolution Strategies: Similarities and Differences

Thomas Bäck* Günter Rudolph† Hans-Paul Schwefel‡

University of Dortmund
Department of Computer Science · Chair of Systems Analysis
P.O. Box 50 05 00 · 4600 Dortmund 50 · Germany

Abstract

Evolutionary Programming and *Evolution Strategies*, rather similar representatives of a class of probabilistic optimization algorithms gleaned from the model of organic evolution, are discussed and compared to each other with respect to similarities and differences of their basic components as well as their performance in some experimental runs. Theoretical results on global convergence, step size control for a strictly convex, quadratic function and an extension of the convergence rate theory for Evolution Strategies are presented and discussed with respect to their implications on Evolutionary Programming.

1 Introduction

Developed independently from each other, three main streams of so-called *Evolutionary Algorithms*, i.e. algorithms based on the model of natural evolution as an optimization process, can nowadays be identified: *Evolutionary Programming* (EP), developed by L. J. Fogel et al. in the U.S. [10], *Genetic Algorithms* (GAs), developed by J. Holland also in the U.S. [15], and *Evolution Strategies* (ESs), developed in Germany by I. Rechenberg [23] and H.-P. Schwefel [29].

These algorithms are based on an arbitrarily initialized population of search points, which by means of randomized processes of *selection*, *mutation*, and (sometimes) *recombination* evolves towards better and better regions in the search space. *Fitness* of individuals is measured by means of an objective function to be optimized, and several applications of these algorithms to real-world problems have clearly demonstrated their capability to yield good approximate solutions even in case of complicated multimodal topological surfaces of the fitness landscape (for overviews of applications, the reader is referred to the conference proceedings [11, 12, 26, 4, 9, 31, 17] or to the annotated bibliography [3]).

Until recently, development of these main streams was completely independent from each other. Since 1990, however, contact between the GA-community and the ES-community has been established, confirmed by collaborations and scientific exchange during regularly alternating conferences in the U.S. (International Conference on Genetic Algorithms and their Applications, ICGA, since 1985) and in Europe (International Conference on Parallel Problem Solving from Nature, PPSN, since 1990). Contact between the EP-community and the ES-community, however, has been established for the first time just in 1992. For algorithms bearing so much similarities as ESs and EP do, this is a surprising fact.

Similar to a paper comparing ES and GA approaches [14], the aim of this article is to give an introduction to ESs and EP and to look for similarities and differences between both approaches. A brief overview of the historical development of ESs as well as an explanation of the basic algorithm are given in section 2. In section 3, the basic EP-algorithm is presented. Section 4 then serves to discuss theoretical results from ESs and their possible relations to the behaviour of an EP-algorithm. Section 5 presents a practical comparison of both algorithms, using a few objective functions with different topological shapes. Finally, an overview of similarities and differences of both approaches is summarized in section 6.

*baeck@ls11.informatik.uni-dortmund.de
†rudolph@ls11.informatik.uni-dortmund.de
‡schwefel@ls11.informatik.uni-dortmund.de

2 Evolution Strategies

Similar to EP, ESs are also based on real-valued object variables and normally distributed random modifications with expectation zero. According to Rechenberg [23], first experimental applications to parameter optimization, performed during the middle of the sixties at the Technical University of Berlin, dealt with hydrodynamical problems like shape optimization of a bended pipe and a flashing nozzle. The algorithm used was a simple mutation-selection scheme working on one individual, which created one offspring by means of mutation. The better of parent and offspring is selected deterministically to survive to the next generation, a selection mechanism which characterizes this *two membered* or (1+1)–ES. Assuming inequality constraints $g_j : \mathbb{R}^n \to \mathbb{R}$ $(j \in \{1, \ldots, v\})$ of the search space, an objective function $f : M \subseteq \mathbb{R}^n \to \mathbb{R}$ (where the feasible region M is defined by the inequality constraints g_j), a minimization task, and individual vectors $x(t) \in \mathbb{R}^n$, where t denotes the generation counter, a (1+1)–ES is defined by the following algorithm:

Algorithm 1 ((1+1)–ES)

$t := 0$;
initalize $P(0) := \{x(0)\}$;
  such that $\forall j : g_j(x_P(0)) \geq 0$;
while *termination criterion not fulfilled* **do**
  mutate $P(t)\ :\ x'(t) := x(t) + \sigma \cdot z(t)$
    with probability density
    $p(z_i(t)) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(z_i(t))^2\right)$;
  evaluate $P(t)\ :\ f(x(t)), f(x'(t))$;
  select $P(t+1)$ *from* $P(t)$:
    if $f(x'(t)) \leq f(x(t))$ *and* $\forall j : g_j(x'(t)) \geq 0$
    then $x(t+1) := x'(t)$
    else $x(t+1) := x(t)$;
  $t := t + 1$;
od

To each component of the vector $x(t)$ the same standard deviation σ is applied during mutation. The variation of σ, i.e. the step-size control of the algorithm, is done according to a theoretically supported rule which is due to Rechenberg [23]. For the objective functions f_1, a *linear corridor* of width b:

$$\begin{aligned} f_1(x) &= \mathrm{F}(x_1) = c_0 + c_1 x_1 \\ &\quad \forall i \in \{2, \ldots, n\} : -b/2 \leq x_i \leq b/2 \end{aligned}$$

and the *sphere model* f_2:

$$f_2(x) = c_0 + c_1 \cdot \sum_{i=1}^{n} (x_i - x_i^*)^2 = c_0 + c_1 \cdot r^2 \quad , \qquad (1)$$

he calculated the optimal expected convergence rates, from which the corresponding optimal success probabilities $p_{opt} \approx 0.184$ and $p_{opt} \approx 0.270$ can be derived for f_1 and f_2, respectively. This forms the basis of Rechenberg's $1/5$ *success rule* [23]:

> *The ratio of successful mutations to all mutations should be $1/5$. If it is greater, increase; if it is less, decrease the standard deviation σ.*

For this algorithm, Schwefel [30] suggested to measure the success probability p by observing this ratio during the search and to adjust $\sigma = \sigma(t)$ according to

$$\sigma(t) = \begin{cases} \sigma(t-1) \cdot c & , \text{ if } p > 1/5 \\ \sigma(t-1)/c & , \text{ if } p < 1/5 \\ \sigma(t) & , \text{ if } p = 1/5 \end{cases} \qquad (2)$$

For the constant c, he proposed to use $c = 0.85^{1/n}$. Doing so, yields convergence rates of linear order in both model cases.

The *multimembered* ES introduces the concepts population, recombination, and self-adaptation of strategy parameters into the algorithm. According to the selection mechanism, the $(\mu+\lambda)$–ES and (μ,λ)–ES are distinguished, the first case indicating that μ parents create $\lambda \geq 1$ offspring individuals by means of recombination and mutation. The μ best individuals out of parents *and* offspring are selected to form the next population. For a (μ,λ)–ES with $\lambda > \mu$ the μ best individuals are selected from the λ offspring only. Each individual is characterized not only by a vector x of object variables, but also by an additional vector of strategy variables. The latter may include up to n different variances $c_{ii} = \sigma_i^2$ $(i \in \{1, \ldots, n\})$ as well as up to $n \cdot (n-1)/2$ covariances c_{ij} $(i \in \{1, \ldots, n-1\},\ j \in \{i+1, \ldots, n\})$ of the generalized n-dimensional normal distribution having a probability density function

$$p(z) = \sqrt{\frac{\det A}{(2\pi)^n}} \exp\left(-\frac{1}{2} z^T A z\right) \quad . \qquad (3)$$

Altogether, up to $w = n \cdot (n+1)/2$ strategy parameters can be varied during the optimum search by means of a selection-mutation-recombination mechanism. To assure positive-definiteness of the covariance matrix A^{-1}, the algorithm uses the equivalent rotation angles α_j $(0 \leq \alpha_j \leq 2\pi)$ instead of the coefficients c_{ij}. The resulting algorithm reads as follows:

ALGORITHM 2 $((\mu,\lambda)$–ES, $(\mu+\lambda)$–ES)

$t := 0$;
initialize $P(0) := \{a_1(0), \ldots, a_\mu(0)\} \in I^\mu$
  where $I = \mathbb{R}^w$;
  $a_k = (x_i, c_{ij} = c_{ji}\ \forall i, j \in \{1, \ldots, n\}\ j \geq i)$;
evaluate $P(0)$;
while *termination criterion not fulfilled* **do**
  recombine $a'_k(t) := r(P(t))\ \forall k \in \{1, \ldots, \lambda\}$;
  mutate $a''_k(t) := m(a'_k(t))$;
  evaluate $P'(t) := \{a''_1(t), \ldots, a''_\lambda(t)\}$;
    $(\{f(x''_1(t)), \ldots, f(x''_\lambda(t))\})$;
  select $P(t+1) :=$ *if* (μ,λ)–ES
    then $s(P'(t))$;
    else $s(P'(t) \cup P(t))$;
  $t := t + 1$;
od

Then, the mutation operator must be extended according to (dropping time counters t):

$$m(a_k) \quad = \quad a'_k = (x', \sigma', \alpha') \in I \quad , \tag{4}$$

performing component-wise operations as follows:

$$\begin{aligned} \sigma'_i &= \sigma_i \cdot \exp(\tau_0 \cdot \Delta\sigma_0) \cdot \exp(\tau \cdot \Delta\sigma_i) \\ \alpha'_j &= \alpha_j + \beta \cdot \Delta\alpha_j \\ x'_i &= x_i + z_i(\sigma', \alpha') \quad . \end{aligned} \tag{5}$$

This way, mutations of object variables are correlated according to the values of the vector α, and σ provides a scaling of the (linear) metrics. Alterations $\Delta\sigma$ and $\Delta\alpha$ are again normally distributed with expectation zero and variance one, and the constants $\tau_0 \propto 1/(\sqrt{2\sqrt{n}})$, $\tau \propto 1/\sqrt{2n}$, and $\beta \approx 0.0873$ (5°) are rather robust exogenous parameters. $\Delta\sigma_0$, scaled by τ_0, is a global factor (identical for all $i \in \{1, \ldots, n\}$), whereas $\Delta\sigma_i$ is an individual factor (sampled anew for all $i \in \{1, \ldots, n\}$) allowing of individual changes of "mean step sizes" σ_i.

Concerning recombination, different mechanisms can be used within ESs, where in addition to the usual recombination of two parents global mechanisms allow for taking into account up to all individuals of a population during creation of one single offspring individual. The recombination rules for an operator creating an individual $a' = (x', \sigma', \alpha') \in I$ are given here representatively for the object variables:

$$x'_i \quad = \quad \begin{cases} x_{S,i} & (1) \\ x_{S,i} \text{ or } x_{T,i} & (2) \\ x_{S,i} + u \cdot (x_{T,i} - x_{S,i}) & (3) \\ x_{S_i,i} \text{ or } x_{T_i,i} & (4) \\ x_{S_i,i} + u_i \cdot (x_{T_i,i} - x_{S_i,i}) & (5) \end{cases} \tag{6}$$

Indices S and T denote two arbitrarily selected parent individuals, and u is a uniform random variable on the interval $[0, 1]$. Besides completely missing recombination (1), the different variants indicated are *discrete recombination* (2), *intermediate recombination* (3) and the *global* versions (4), (5) of the latter two, respectively. Empirically, discrete recombination on object variables and intermediate recombination on strategy parameters have been observed to give best results.

3 Evolutionary Programming

Following the description of an EP algorithm as given by Fogel [7] and using the notational style from the previous section, an EP algorithm is formulated as follows:

ALGORITHM 3 (EP)

$t := 0$;
initialize $P(0) := \{x_1(0), \ldots, x_\mu(0)\} \in I^\mu$
  where $I = \mathbb{R}^n$;
evaluate $P(0)$: $\mathcal{F}(x_k(0)) = G(f(x_k(0)), \nu_k)$;
while *termination criterion not fulfilled* **do**
  mutate $x'_k(t) := m(x_k(t))\quad \forall k \in \{1, \ldots, \mu\}$;
  evaluate $P'(t) := \{x'_1(t), \ldots, x'_\mu(t)\}$;
    $(\{\mathcal{F}(x'_1(t)), \ldots, \mathcal{F}(x'_\mu(t))\})$;
  select $P(t+1) := s(P(t) \cup P'(t))$;
  $t := t + 1$;
od

Besides a missing recombination operator, fitness evaluation, mutation, and selection are different from corresponding operators in ESs. Fitness values $\mathcal{F}(x_i)$ are obtained from objective function values by scaling them to positive values (function G) and possibly by imposing some random alteration ν_i. For mutation, the standard deviation for each individual's mutation is calculated as the square root of a linear transformation of its own fitness value, i.e. for mutation $m(x) = x'$ ($\forall i \in \{1, \ldots, n\}$):

$$\begin{aligned} x'_i &= x_i + \sigma_i \cdot z \\ \sigma_i &= \sqrt{\beta_i \cdot \mathcal{F}(x) + \gamma_i} \quad . \end{aligned} \tag{7}$$

Again, the random variable z has probability density $p(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$. This way, for each component of the vector x a different scaling of mutations can be achieved by tuning the parameters β_i and γ_i (which, however, are often set to one and zero, respectively).

The selection mechanism $s : I^{2\mu} \rightarrow I^\mu$ reduces the set of parents and offspring individuals to a set of μ parents by performing a kind of q-tournament selection ($1 \leq q$). In principle, for each individual x_k from $P(t) \cup P'(t)$ q individuals are selected at random from $P(t) \cup P'(t)$ and compared to x_k with respect to their fitness values $\mathcal{F}_j$ ($j \in \{1, \ldots, q\}$). Then, it is counted for each of the q selected individuals whether x_j outperforms the individual, resulting in a score w_j between 0 and q. When this is finished for all 2μ individuals, the individuals are ranked in descending order of the rank values w_j and the μ individuals having highest ranks w_j are selected to form the next population. Using a more formal notation, rank values w_j are obtained as follows:

$$w_j = \sum_{i=1}^{q} 1_{\mathbb{R}_0^+}(\mathcal{F}(x_{u_i}) - \mathcal{F}(x_j)) \quad . \tag{8}$$

u_i denotes a uniform integer random variable on the range of indices $\{1, \ldots, 2\mu\}$ which is sampled anew for each comparison, the indicator function $1_A(x)$ is one if $x \in A$ and zero otherwise, and $\mathbb{R}_0^+ = \{x \in \mathbb{R} \mid x \geq 0\}$. Intuitively, the selection mechanism implements a kind of probabilistic $(\mu+\mu)$-selection which becomes more and more "deterministic" as the external parameter q is increased, i.e. the probability that the selected set of individuals is the same as in a $(\mu+\mu)$-selection scheme tends to unity as q increases. The selection scheme guarantees survival of the best individual, since this is assigned a guaranteed maximum fitness score of q.

In general, the standard EP algorithm imposes some parameter tuning difficulties to the user concerning the problem of finding useful values for β_i and γ_i in case of arbitrary, high-dimensional objective functions. To overcome these difficulties, D. B. Fogel developed an extension called *meta*-EP that self-adapts n variances $c_1, \ldots, c_n$ per individual quite similar to ESs (see [8], p. 157). Then, mutation $m(a) = a'$ applied to an individual $a = (x, c)$ produces $a' = (x', c')$ according to $(\forall i \in \{1, \ldots, n\})$:

$$\begin{aligned} x_i' &= x_i + \sqrt{c_i} \cdot z &&= x_i + \sigma_i \cdot z \\ c_i' &= c_i + \sqrt{\zeta c_i} \cdot z &&= \sigma_i^2 + \sqrt{\zeta}\sigma_i \cdot z \quad , \end{aligned} \tag{9}$$

where the second identities hold because of $c_i = \sigma_i^2$, and ζ denotes an exogenous parameter. To prevent variances from becoming negative, Fogel proposes to set $c_i' := \varepsilon_c > 0$ whenever by means of the modification rule (9) negative variances would occur. However, while the log-normally distributed alterations of standard deviations in ESs automatically guarantee positivity of σ_i, the mechanism used in meta-EP is expected to cause variances set to ε_c rather often, thus essentially leading to a reduction of the dimension of the search space when ε_c is small. It is surely interesting to perform an experimental investigation on strengths and weaknesses of both self-adaptation mechanisms.

In addition to standard deviations, the *Rmeta*-EP algorithm as proposed by D. B. Fogel (see [8], pp. 287–289) also incorporates the complete vector of $n \cdot (n-1)/2$ correlation coefficients $\rho_{ij} = c_{ij}/\sqrt{\sigma_i \sigma_j} \in [-1, 1]$ $(i \in \{1, \ldots, n-1\}, j \in \{i+1, \ldots, n\})$, representing the covariance matrix A^{-1}, into the genotype for self-adaptation quite similar to correlated mutations in ESs. This algorithm, however, was implemented and tested by Fogel only for $n = 2$, and currently the extension to $n > 2$ is not obvious since positive definiteness and symmetry of A must be guaranteed on the one hand while on the other hand it is necessary to find an implementation capable of producing any valid correlation matrix. For correlated mutations in ESs, the feasibility of the correlation procedure has recently been shown by Rudolph [25].

4 Theoretical Properties of Evolution Strategies

4.1 Problem statement and method formulation

Before summarizing convergence results of ES–type optimization methods some basic definitions and assumptions are to be made.

DEFINITION 1
An optimization problem of the form

$$f^* := f(x^*) := \min\{f(x) \mid x \in M \subseteq \mathbb{R}^n\} \tag{10}$$

is called a *regular global optimization problem* iff

(A1) $f^* > -\infty$,
(A2) $x^* \in \text{int}(M) \neq \emptyset$ and
(A3) $\mu(L_{f^*+\epsilon}) > 0$

Here, f is called the *objective function*, M the *feasible region*, f^* the *global minimum*, x^* the *global minimizer* or *solution* and $L_a := \{x \in M \mid f(x) < a\}$ the *lower level set* of f. □

The first assumption makes the problem meaningful whereas the second assumption is made to facilitate the analysis. The last assumption skips those problems where optimization is a hopeless task (see [33, p. 7]). Furthermore, let us rewrite the ES–type algorithm given in a previous section in a more compact form:

$$X_{t+1} = Y_{t+1} \cdot 1_{L_{f(x_t)}}(Y_{t+1}) + x_t \cdot 1_{L^c_{f(x_t)}}(Y_{t+1}) \quad , \tag{11}$$

where the indicator function $1_A(x)$ is one if $x \in A$ and zero otherwise and where Y_{t+1} is a random vector with some probability density $p_{Y_{t+1}}(y) = p_{Z_t}(y - x_t)$. Usually p_Z is chosen as a *spherical* or *elliptical distribution*:

DEFINITION 2
A random vector z of dimension n is said to possess an *elliptical distribution* iff it has stochastic representation $z \stackrel{d}{=} r\,Q'\,u$, where random vector u is uniformly distributed on a hypersphere surface of dimension n stochastically independent to a nonnegative random variable r and where matrix $Q : k \times n$ with

rank$(Q'Q) = k$. If $Q : n \times n$ and $Q = I$ then z is said to possess a *spherical (symmetric) distribution.* □

From the above definition it is easy to see that an ES-type algorithm using a spherical or elliptical distribution is equivalent to a random direction method with some step size distribution. In fact, if z is multinormally distributed with zero mean and covariance matrix $C = \sigma^2 I$, then the step size r has a $\chi_n(\sigma)$–distribution (see Fang et al. [6] for more details). If matrix Q and the variance of r are fixed during the optimization we shall say that algorithm (11) has a *stationary* step size distribution, otherwise the distribution is called *adapted.*

4.2 Global convergence property

The proof of global convergence has been given by several authors independently. Although the conditions are slightly different among the papers each proof is based on the so–called Borel–Cantelli Lemma which application makes a convergence proof for an elitist GA trivial.

Theorem 1 ([5][2][32][19][20][37])
Let $p_t := P\{X_t \in L_{f^*+\epsilon}\}$ be the probability to hit the level set $L_{f^*+\epsilon}$, $\epsilon > 0$, at step t. If

$$\sum_{t=1}^{\infty} p_t = \infty \tag{12}$$

then $f(X_t) - f^* \to 0$ a.s. for $t \to \infty$ or equivalently

$$P\{\lim_{t\to\infty}(f(X_t) - f^*) = 0\} = 1$$

for any starting point $x_0 \in M$. □

Lemma 1
Let C be the support of stationary p_Z. Then holds: $M \subseteq C$ and M bounded $\Rightarrow L_a$ bounded $\forall\, a \le f(x_0) \Rightarrow p_t \ge p_{min} > 0$ for all $t > 0 \Rightarrow \liminf_{t\to\infty} p_t > 0 \Rightarrow$ (12) □

Of course, theorem 1 is only of academic interest because we have no unlimited time to wait. However, if condition (12) is not fulfilled then one may conclude that the probability to obtain the global minimum for any starting point $x_0 \in M$ with increasing t is zero as pointed out by Pinter [19].

4.3 Convergence rates

The attempt to determine the convergence rate of algorithms of type (11) was initiated by Rastrigin [22] and continued by Schumer and Steiglitz [28] and Rechenberg [23]. Each of them calculated the expected progress w.r.t. the objective value or distance to the minimizer of special convex functions. Since the latter measure is not well–defined in general we shall use the following definition:

Definition 3
The value $\delta_t := \mathrm{E}[f(X_t) - f^*]$ is said to be the *expected error* at step t. An algorithm has a *sublinear convergence rate* iff $\delta_t = O(t^{-b})$ with $b \in (0, 1]$ and a *geometrical convergence rate*, iff $\delta_t = O(r^t)$ with $r \in (0, 1)$. □

Whereas the proof of global convergence can be given for a broad class of problems the situation changes for proofs concerning convergence rates. Seemingly, the only chance is to restrict the analysis to a smaller class of problems that possesses some special properties. This has been done by Rappl and it generalizes the results on convex problems mentioned above:

Theorem 2 ([20][21])
Let f be a (l, Q)–strongly convex function, i.e. f is continuously differentiable and with $l > 0, Q \ge 1$ there holds for all $x, y \in M$

$$l\,||x - y||^2 \le (\nabla f(x) - \nabla f(y))'(x - y) \le Ql\,||x - y||^2$$

and $Z \overset{d}{=} RU$, where R has nonvoid support $(0, a) \subseteq \mathbb{R}$. Then the expected error of algorithm (11) decreases for any starting point $x_0 \in M$ with the following rates:

$$\mathrm{E}[f(X_t) - f^*] \le \begin{cases} O(t^{-2/n}) & , p_Z \text{ stationary} \\ O(\beta^t) & , p_Z \text{ adapted} \end{cases}$$

with $\beta \in (0, 1)$. □

In order to adapt the step sizes one may choose $R_{t+1} = ||\nabla f(x_t)||\, R$. Moreover, Rappl [20, p. 102–143] has shown that the step size can be adapted via a success/failure control similar to the proposal of [23]. The idea is to decrease the step size with a factor $\gamma_1 \in (0, 1)$ if there was a failure and to increase the step size by a factor $\gamma_2 > 1$ if there was a success. Then geometrical convergence follows for $\gamma_1\,\gamma_2 > 1$.

It should be noted that even with geometrical convergence the expected number of steps to achieve a certain accuracy may differ immensely, e.g. consider the number of steps needed if $\beta = 0.9$ or if $\beta = 1 - 10^{-10}$. Therefore, the search for optimal step size schedules should not be neglected.

Example 1
Let $f(x) = ||x||^2$ with $M = \mathbb{R}^n$ be the problem. Clearly, f is $(2, 1)$–strongly convex:

$$(\nabla f(x) - \nabla f(y))'(x - y) = 2\,||x - y||^2 .$$

The mutation vector z in algorithm (11) is chosen to be multinormally distributed with zero mean and covariance matrix $C = \sigma^2 I$. Consequently, for the distribution of the objective function values we have $f(x_t + Z_t) \sim \sigma^2 \chi_n^2(\kappa)$, where $\chi_n^2(\kappa)$ denotes a noncentral

χ^2–distribution with n degrees of freedom and noncentrality parameter $\kappa = ||x_t||^2/\sigma^2$. Using the fact that [16, p. 135]

$$\frac{\chi_n^2(\kappa) - (n+\kappa)}{\sqrt{2(n+2\kappa)}} \to N \sim N(0,1)$$

for $n \to \infty$ the limiting distribution of the normalized variation of objective function values V becomes

$$\begin{aligned}
V &:= \frac{f(x_t) - f(X_{t+1})}{f(x_t)} \\
&= 1 - \frac{\sigma^2}{||\, x_t \,||^2} \chi_n^2(\kappa) \\
&\to 1 - \frac{\sigma^2}{||x_t||^2} (n + \kappa + \sqrt{2n + 4\kappa}\, N) \\
&= -\frac{s^2}{n} - \frac{s^2}{n} \sqrt{\frac{2}{n} + \frac{4}{s^2}}\, N \\
&\asymp -\frac{s^2}{n} - \frac{2s}{n} N
\end{aligned}$$

with $\sigma = s\,||\, x_t \,||/n$. Since algorithm (11) accepts only improvements, we are interested in the expectation of the random variable $V^+ = \max\{0, V\}$:

$$\begin{aligned}
\mathrm{E}[V^+] &= \int_0^\infty \frac{nu}{2s\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{nu+s^2}{2s}\right)^2\right] du \\
&= \frac{1}{n}\left\{ s\sqrt{\frac{2}{\pi}} \exp\left(-\frac{s^2}{8}\right) - s^2\left[1 - \Phi\left(\frac{s}{2}\right)\right]\right\}
\end{aligned}$$

where $\Phi(.)$ denotes the c.d.f. of a unit normal random variable. The expectation becomes maximal for $s^* = 1.224$ (see fig. 1) such that $\mathrm{E}[V^+] = 0.404/n$ and

$$\sigma^* = \frac{1.224}{n}\, ||x|| \tag{13}$$

$$= \frac{1.224}{n} \sqrt{f(x)} \tag{14}$$

$$= \frac{0.612}{n}\, ||\nabla f(x)|| \;, \tag{15}$$

where (13) is the value also given by Rechenberg [23] which is converted to (14) and (15) in the notation of Fogel [7] and Rappl [20], respectively. Obviously, if we modify f to $f_a(x) = ||x||^2 + 1$ then control (14) will fail to provide geometrical convergence. One has to subtract some constant from (14) which depends on the value of the (unknown) global minimum. Similarly, optimizing $f_b(x) = ||x - 1||^2$ control (13) will fail whereas control (15) will succeed in all cases due to its invariance w.r.t. the location of the unknown global minimizer and the value of the unknown global minimum. In addition, the dependence on the problem dimension n is of importance: Omitting this factor geometrical convergence can still be guaranteed but it will be very slow compared to the optimal setting (see fig. 1). □

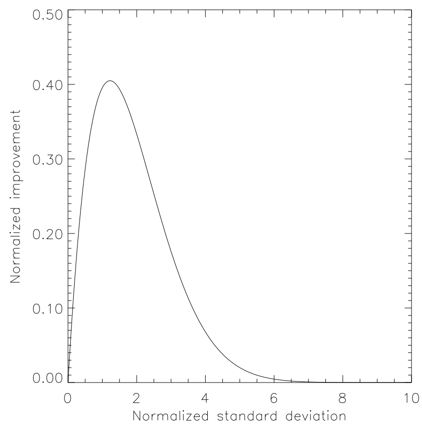

Figure 1: Normalized expected improvement $n\,\mathrm{E}[V^+]$ versus normalized standard deviation $s = n\,\sigma / f(x_t)^{1/2}$

4.4 $(\mu + \lambda)$–Evolution Strategies

Obviously, theorems 1 and 2 can be applied to this type of algorithms. However, as pointed out by Schwefel [29][30] there are some differences concerning the convergence rates: The larger the number of offspring λ the better the convergence rate. We shall discuss the relationship in the next subsection. Wardi [35] proposed a $(1 + \lambda)$–ES where λ is a random variable depending on the amount of improvement, i.e. new offspring are generated as long as the improvement is below a certain decreasing limit which is used to adjust the step sizes as well.

4.5 (μ, λ)–Evolution Strategies

For this type of algorithms theorem 1 cannot be applied. Although it is possible to show that the level set $L_{f^*+\epsilon}$ will be hit with some probability there is no guarantee of convergence. For example, a $(1,1)$–ES is simply a random walk. A possible way to avoid nonconvergence may be achieved by restricting the probability of accepting a worse point. In fact, if this probability is decreasing with a certain rate over time global convergence can be assured under some conditions (see Haario and Saksman [13]). With this additional feature a (μ, λ)–ES (without recombination) is equivalent to special variants of so–called *Simulated Annealing* algorithms that are designed for optimizing in $\mathbb{R}^n$. This relationship is investigated in Rudolph (preprint 92).

Despite the lack of a theoretical guarantee of global convergence it is possible to calculate the convergence rates for some special problems. This has been done by Schwefel [29][30] and Scheel [27] for the same problem as in the previous example using a $(1, \lambda)$–ES:

EXAMPLE 2
Similarly to example 1 for large n the optimal setting of σ can be calculated

$$\sigma^* = \frac{c_{1,\lambda}}{2n} \, || \nabla f(x) || \, .$$

Then, the expected improvement is $\mathrm{E}[V] = c_{1,\lambda}^2/n$ with

$$c_{1,\lambda} = \frac{\lambda}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u \exp\left(-\frac{u^2}{2}\right) \Phi(u)^{\lambda-1} \, du \qquad (16)$$

The value of $c_{1,\lambda}$ has been tabularized in [27]. However, a closer look at (16) reveals that this expression is equivalent with the expected value of the maximum of λ i.i.d. unit normal random variables:

Let $X_i \sim N(0,1)$ for $i = 1, \ldots, \lambda$. Then $M_\lambda := \max\{X_1, \ldots, X_\lambda\}$ is a random variable with c.d.f. $P\{M_\lambda \leq x\} = F^\lambda(x) = \Phi^\lambda(x)$. Consequently, $c_{1,\lambda} = \mathrm{E}[M_\lambda]$. According to Resnick [24, pp. 71–72] we have:

$$\begin{aligned} P\{M_\lambda \leq a_\lambda x + b_\lambda\} &= F^\lambda(a_\lambda x + b_\lambda) \\ &\rightarrow G(x) = \exp(-e^{-x}) \end{aligned}$$

for $\lambda \rightarrow \infty$ with

$$\begin{aligned} a_\lambda &= (2 \log \lambda)^{-\frac{1}{2}} \\ b_\lambda &= (2 \log \lambda)^{\frac{1}{2}} - \frac{\log \log \lambda + \log(4\pi)}{2\sqrt{2 \log \lambda}} \end{aligned}$$

Let Y have c.d.f. $G(x)$, then

$$\frac{M_\lambda - b_\lambda}{a_\lambda} \approx Y \Leftrightarrow M_\lambda \approx a_\lambda Y + b_\lambda$$

and due to the linearity of the expectation operator

$$\begin{aligned} \mathrm{E}[M_\lambda] &\approx a_\lambda \mathrm{E}[Y] + b_\lambda = a_\lambda \gamma + b_\lambda \\ &= \sqrt{2 \log \lambda} + \frac{2\gamma - \log \log \lambda - \log(4\pi)}{2\sqrt{2 \log \lambda}} \\ &\approx \sqrt{2 \log \lambda} \end{aligned}$$

where $\gamma = 0.57721\ldots$ denotes Euler's constant. Now it is possible to derive an asymptotic expression for the speedup assuming that the evaluation of λ trial points are performed in parallel. Let $\mathrm{E}[t_1]$ and $\mathrm{E}[t_\lambda]$ denote the expected number of steps to achieve a given accuracy ϵ for a $(1+\lambda)$–ES and $(1, \lambda)$–ES, respectively. Then for the expected speedup holds

$$\begin{aligned} \mathrm{E}[S_\lambda] &= \frac{\mathrm{E}[t_1]}{\mathrm{E}[t_\lambda]} \\ &= \frac{\log(\epsilon/\delta_0)}{\log(1 - 0.404/n)} \cdot \frac{\log(1 - c_{1,\lambda}^2/n)}{\log(\epsilon/\delta_0)} \\ &\approx \frac{\log(1 - 2\log(\lambda)/n)}{\log(1 - 0.404/n)} \\ &\asymp \frac{2\log(\lambda)/n}{0.404/n} \\ &= O(\log \lambda) \ . \end{aligned}$$

Thus, the speedup is only logarithmic. It can be shown that this asymptotic bound cannot be improved by a $(1+\lambda)$–ES. □

4.6 Open questions

Theorem 2 provides convergence rate results and step size adjustment rules for strongly convex problems. Rappl [20] has shown that the conditions of theorem 2 can be weaken such that the objective function is required to be only almost everywhere differentiable and that the level sets possess a "bounded asphericity", especially close to the minimizer.

The algorithm of Patel et al. [18] called *pure adaptive search* requires only convexity to assure geometrical convergence. However, this algorithm uses a uniform distribution over the lower level sets for sampling a new point. Naturally, this distribution is unknown in general. Recently Zabinsky and Smith [36] have given the impressive result that this algorithm converges geometrically even for lipshitz–continuous functions with several local minima. This rises hope to design an Evolutionary Algorithm that converges to the global optimum of certain nonconvex problem classes with a reasonable rate.

Obviously, convergence rates are closely connected to the adaptation of the sampling distribution. Incorporating distribution parameters within the evolutionary process may be a possible solution. This technique can be subsumed under the term *self–adaptation*. First attempts to analyse this technique have been done by Vogelsang [34].

In this context *recombination* may play an important role because it can be seen as an operation that connects the more or less local mutation distributions of single individuals to a more global mutation distribution of the whole population. However, nothing is known theoretically about recombination up to now.

5 Experimental Comparison

Just to achieve a first assessment of the behaviour of both algorithms, experiments were run on the sphere model $f_2(x) = ||x||^2$ and the generalized variant of a multimodal function by Ackley (see [1], pp. 13–14):

$$
\begin{aligned}
f_9(x) &= -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) \\
&\quad - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e \ ,
\end{aligned}
\tag{17}
$$

test functions that represent the class of strictly convex, unimodal as well as highly multimodal topologies, respectively. The global optimum of f_9 is located at the origin with a function value of zero. Three-dimensional topology plots of both functions are presented in figure 2.

A comparison was performed for a moderate dimension $n = 30$ and $-30.0 \leq x_i \leq 30.0$ defining the feasible region for initialization. The parameterizations of both algorithms were set as follows:

- Evolution Strategy: (30,200)-ES with self-adaptation of 30 standard deviations, no correlated mutations, discrete recombination on object variables and global intermediate recombination on standard deviations.
- Evolutionary Programming: Meta-EP with self-adaptation of 30 variances, population size $\mu = 200$, tournament size $q = 10$ for selection, $\zeta = 6$ (see [8], p. 168).

All results were obtained by running ten independent experiments per algorithm and averaging the resulting data, and 40000 functions evaluations were performed for each run on the sphere model in contrast to 200000 function evaluations on Ackley's function. The resulting curves of the actually best objective function value plotted over the number of function evaluations are shown in figure 3.

The clear difference of convergence velocity on f_2 indicates that the combination of relatively strong selective pressure, recombination, and self-adaptation as used in ESs is helpful on that topology (performance of the ES can still be improved by reducing the amount of strategy parameters to just one standard deviation). Convergence reliability on Ackley's function turns out to be rather good, since both strategies locate the global optimum in each of the ten runs with average final best objective function values of $1.39 \cdot 10^{-2}$ (EP) and $7.48 \cdot 10^{-8}$ (ES), respectively. This behaviour is due to the additional degree of freedom provided by self-adaptation of $n = 30$ strategy parameters.

6 Conclusions

As it turned out in the preceding sections, Evolutionary Programming and Evolution Strategies share many common features, i.e. the real-valued representation of search points, emphasis on the utilization of normally distributed random mutations as main search operator, and, most importantly, the concept of self-adaptation of strategy parameters on-line during the search. There are, however, some striking differences, most notably the missing recombination operator in EP and the softer, probabilistic selection mechanism used in EP. The combination of these properties seems to have some negative impact on the performance of EP, as indicated by the experimental results presented in section 5.

Further investigations of the role of selection and recombination as well as the different self-adaptation methods are surely worthwhile, just as a further extension of theoretical investigations of these algorithms. As demonstrated in section 4, some theory available from research on Evolution Strategies can well be transferred to Evolutionary Programming and is helpful for assessing strengths and weaknesses of the latter approach.

References

[1] D.H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Kluwer Academic Publishers, Boston, 1987.

[2] N. Baba. Convergence of random optimization methods for constrained optimization methods. *Journal of Optimization Theory and Aplications*, 33:451–461, 1981.

[3] Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel. Applications of evolutionary algorithms. Report of the Systems Analysis Research Group (LS XI) SYS–2/92, University of Dortmund, Department of Computer Science, 1992.

[4] Richard K. Belew and Lashon B. Booker, editors. *Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications*, University of California, San Diego, USA, 1991. Morgan Kaufmann Publishers.

[5] Joachim Born. *Evolutionsstrategien zur numerischen Lösung von Adaptationsaufgaben*. Dissertation A, Humboldt-Universität, Berlin, 1978.

[6] K.-T. Fang, S. Kotz, and K.-W. Ng. *Symmetric Multivariate and Related Distributions*, volume 36 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London and New York, 1990.

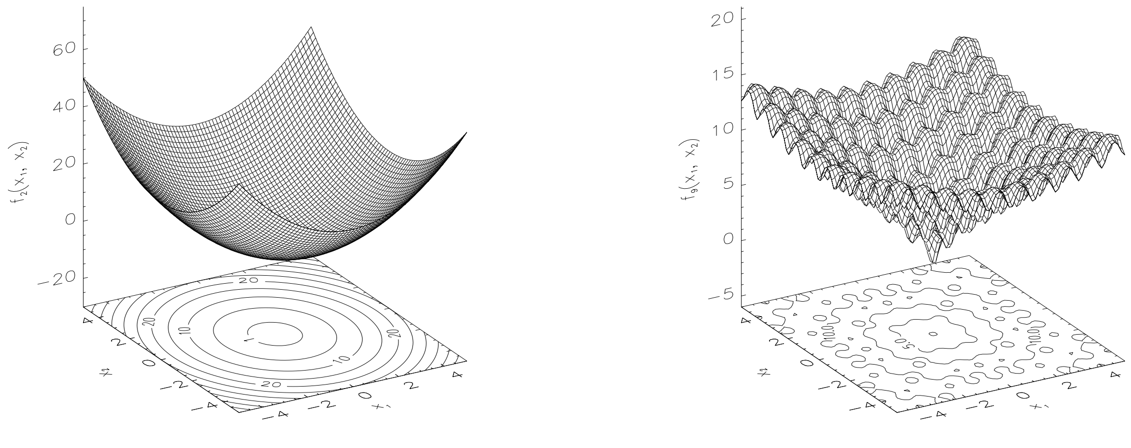

Figure 2: Three-dimensional plots of the sphere model (left) and Ackley's function (right).

[7] David B. Fogel. An analysis of evolutionary programming. In Fogel and Atmar [9], pages 43–51.

[8] David B. Fogel. *Evolving Artificial Intelligence*. PhD thesis, University of California, San Diego, 1992.

[9] David B. Fogel and Wirt Atmar, editors. *Proceedings of the First Annual Conference on Evolutionary Programming*, La Jolla, CA, February 21–22, 1992. Evolutionary Programming Society.

[10] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966.

[11] J. J. Grefenstette, editor. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, Hillsdale, New Jersey, 1985. Lawrence Erlbaum Associates.

[12] J. J. Grefenstette, editor. *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, Hillsdale, New Jersey, 1987. Lawrence Erlbaum Associates.

[13] H. Haario and E. Saksman. Simulated annealing process in general state space. *Adv. Appl. Prob.*, 23:866–893, 1991.

[14] Frank Hoffmeister and Thomas Bäck. Genetic algorithms and evolution strategies: Similarities and differences. In Schwefel and Männer [31], pages 455–470.

[15] John H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.

[16] N.L. Johnson and S. Kotz. *Distributions in Statistics: Continuous Distributions - 2*. Houghton Mifflin, Boston, 1970.

[17] Reinhard Männer and Bernard Manderick, editors. *Parallel Problem Solving from Nature, 2*. Elsevier Science Publishers, Amsterdam, 1992.

[18] N.R. Patel, R.L. Smith, and Z.B. Zabinsky. Pure adaptive search in monte carlo optimization. *Mathematical Programming*, 43(3):317–328, 1988.

[19] J. Pinter. Convergence properties of stochastic optimization procedures. *Math. Operat. Stat. , Ser. Optimization*, 15:405–427, 1984.

[20] G. Rappl. *Konvergenzraten von Random Search Verfahren zur globalen Optimierung*. Dissertation, HSBw München, Germany, 1984.

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

[22] L.A. Rastrigin. The convergence of the random search method in the extremal control of a many-parameter system. *Automation and Remote Corntrol*, 24:1337–1342, 1963.

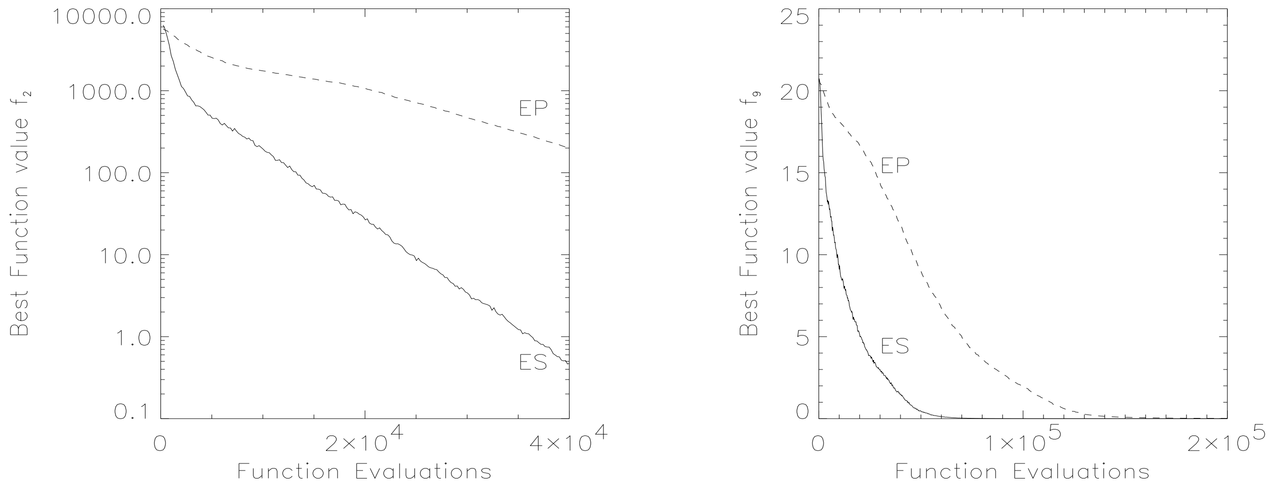

Figure 3: Experimental runs of an Evolution Strategy and Evolutionary Programming on f_2 (left) and f_9 (right).

[23] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann–Holzboog Verlag, Stuttgart, 1973.

[24] S.L. Resnick. *Extreme values, regular variation, and point processes*, volume 4 of *Applied Probability.* Springer, New York, 1987.

[25] Günter Rudolph. On correlated mutations in evolution strategies. In Männer and Manderick [17], pages 105–114.

[26] J. David Schaffer, editor. *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, California, June 1989. Morgan Kaufmann Publishers.

[27] A. Scheel. *Beitrag zur Theorie der Evolutionsstrategie.* Dissertation, TU Berlin, Berlin, 1985.

[28] M.A. Schumer and K. Steiglitz. Adaptive step size random search. *IEEE Transactions on Automatic Control*, 13:270–276, 1968.

[29] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, volume 26 of *Interdisciplinary systems research.* Birkhäuser, Basel, 1977.

[30] Hans-Paul Schwefel. *Numerical Optimization of Computer Models.* Wiley, Chichester, 1981.

[31] Hans-Paul Schwefel and Reinhard Männer, editors. *Parallel Problem Solving from Nature — Proc. 1st Workshop PPSN I*, volume 496 of *Lecture Notes in Computer Science.* Springer, Berlin, 1991.

[32] F.J. Solis and R.J.-B. Wets. Minimization by random search techniques. *Math. Operations Research*, 6:19–30, 1981.

[33] A. Törn and A. Zilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science.* Springer, Berlin and Heidelberg, 1989.

[34] J. Vogelsang. Theoretische Betrachtungen zur Schrittweitensteuerungen in Evolutionsstrategien. Diploma thesis, University of Dortmund, Chair of System Analysis, July 1992.

[35] Y. Wardi. Random search algorithms with sufficient descent for minimization of functions. *Mathematics of Operations Research*, 14(2):343–354, 1989.

[36] Z.B. Zabinsky and R.L. Smith. Pure adaptive search in global optimization. *Mathematical Programming*, 53:323–338, 1992.

[37] A.A. Zhigljavsky. *Theory of global random search*, volume 65 of *Mathematics and its applications (Soviet Series).* Kluwer, AA Dordrecht, The Netherlands, 1991.