Thomas Bäck, Frank Hoffmeister

University of Dortmund, Department of Computer Science, Chair XI,
P.O. Box 50 05 00, W–4600 Dortmund 50

Global Optimization by Means of Evolutionary Algorithms

1 Introduction

Evolution Strategies (ESs) [Rec73, Sch81a] and *Genetic Algorithms* (GAs) [Hol75, Gol89] are two types of *Evolutionary Algorithms*. This term denotes algorithms which rely upon the imitation of the collective learning paradigm of natural populations, including the features of *selection*, *mutation*, and *recombination*. Both algorithms are powerful methods for global optimization, not needing any more information about the objective function than the actual value of the objective function (i.e. they use no predefined internal model of the objective function). Due to their complexity, both algorithms lack the existence of mathematical proofs of general convergence and efficiency (with the exception of two simple, but with respect to their topological surfaces important objective functions in case of a simplified ES [Sch81a]). In this paper a uniform formal description of these algorithms is presented, and it is emphasized on their essential features. It will become obvious, that they have many important features in common, and the most distinguishing property is the self–learning mechanism used by ESs to adapt their strategy parameters during the optimum search (*adaptive search*). A more detailed comparison (also in an experimental way) can be found in [HB90].

2 Evolution Strategies

Neglecting the earlier steps in the development of ESs ((1+1)–ES [Rec73]) it is concentrated here on the $(\mu+\lambda)$–ES and the (μ,λ)–ES as developed by Schwefel [Sch77, Sch81a]. They provide the capability of *self–adaptation* of the strategy parameters. Since their only difference consists of the limited life time of individuals in the (μ,λ)–ES, only a formal description of the latter variant is presented:

$$(\mu,\lambda)\text{–ES} \quad = \quad (P^0, \mu, \lambda, r, m, s, \Delta\sigma, f, g, t) \tag{1}$$

where

P^0	$=$	$(a_1^0, \ldots, a_\mu^0) \in I^\mu$	$I = \mathbf{R}^n \times \mathbf{R}^n$	population
μ	$\in$	$\mathbf{N}$		number of parents
λ	$\in$	$\mathbf{N}$	$\lambda \geq \mu$	number of offspring
r	$:$	$I^\mu \to I$		recombination operator
m	$:$	$I \to I$		mutation operator
s	$:$	$I^\lambda \to I^\mu$		selection operator
$\Delta\sigma$	$\in$	$\mathbf{R}$		step-size meta-control
f	$:$	$\mathbf{R}^n \to \mathbf{R}$		objective function
g_j	$:$	$\mathbf{R}^n \to \mathbf{R}$	$j \in \{1, \ldots, q\}$	constraint functions
t	$:$	$I^\mu \to \{0, 1\}$		termination criterion

Each individual $a_i^t = (x^t, \sigma^t) \in P^t$ within a population consists of two vectors, namely, the set of *object variables* $x^t \in \mathbf{R}^n$ and the set of *step sizes* $\sigma^t \in \mathbf{R}^n$. σ^t is a vector of internal strategy parameters which control the mutation of x^t. The term *population* denotes the set of μ individuals which participate in the generation of the λ offspring individuals. These offspring makes up the *intermediate population* P'^t, which is generated by means of the recombination operator r and the mutation operator m:

$$\begin{aligned} P'^t &= (a_1'^t, \ldots, a_\lambda'^t) \\ a_i'^t &= m(r(P^t)) \qquad ; \forall\, i \in \{1, \ldots, \lambda\} \end{aligned} \tag{2}$$

In case of a $(\mu+\lambda)$–ES the intermediate population P'^t also consists of the μ parents and the selection operator s is modified to $s : I^{\mu+\lambda} \to I^\mu$. By means of selection P'^t is reduced to μ individuals, which become the set of parents P^{t+1} of the next generation.

$$P^{t+1} = s(P'^t) \quad ; \forall\, a_i^{t+1} = (x, \sigma) \; \nexists a_j'^t = (x', \sigma') : f(x') < f(x) \tag{3}$$

According to the rule 'survival of the fittest individuals' the selection operator s removes the least fit individuals from the intermediate population. The iteration $P^t \to P^{t+1}$ stops when the termination criterion $t(P^t) = 1$ holds.

The individuals of P'^t are obtained by applying recombination and mutation, where prior to its mutation an offspring is produced by recombining two parents $a = (x_a, \sigma_a)$ and $b = (x_b, \sigma_b) \in I$ [Sch81b]:

$$\begin{aligned} r(P^t) &= a' = (x', \sigma') \in I \\ v' &= x', \sigma' \\ v_i' &= \begin{cases} v_{a,i} & \text{(A) no recombination} \\ v_{a,i} \text{ or } v_{b,i} & \text{(B) discrete} \\ \frac{1}{2}(v_{a,i} + v_{b,i}) & \text{(C) intermediate} \\ v_{a_i,i} \text{ or } v_{b_i,i} & \text{(D) global, discrete} \\ \frac{1}{2}(v_{a_i,i} + v_{b_i,i}) & \text{(E) global, intermediate} \end{cases} \end{aligned} \tag{4}$$

where $a, b, a_i, b_i \in I$ are parents internally chosen by r, $v_{a,i}$ represents the i-th component of some vector of parent a, and v_i' refers to the i-th component of some vector of the offspring.

By convention all parents in a population have the same mating probabilities. In case of discrete recombination the vector components are chosen from either of the two parents, while with intermediate recombination the offspring values are the arithmetic mean of the parent values. In case of global recombination, for each vector component the parents a_i and b_i are chosen anew, resulting in a higher mixing of the genetic information than in the discrete case.

An offspring obtained by recombination is mutated before it is evaluated and placed into the intermediate population P'^t. For a ES with *simple mutations* the mutation operator works as follows:

$$\begin{aligned} m(a') &= a'' = (x'', \sigma'') \in I \quad ; x'' \in \mathbf{R}^n, \sigma'' \in \mathbf{R}^n \\ \sigma'' &= \sigma' \cdot \exp \mathbf{N_0}(\Delta\sigma) \quad ; \forall\, i \in \{1, \ldots, n\} \\ x'' &= x' + \mathbf{N_0}(\sigma'') \end{aligned} \tag{5}$$

where $\mathbf{N_0}(\Delta\sigma)$ refers to a vector of independent normally distributed random numbers with expectation zero and where the operation $\cdot$ denotes the dot-product.

Note, that the standard deviations σ_i are mutated themselves before they are applied to control the mutation of the object variables x_i. Since the combination of mutation and selection implements some kind of hillclimbing search procedure, the σ_i are often referred to as *step sizes*.

This kind of *self–adaptation* of the strategy parameters is the most important feature of ESs, and in a more extended version of ESs this is also used to allow a topology–depending rotation of the mutation–hyperellipsoids of equal probability density (*correlated mutations*).

Schwefel theoretically investigated the restricted case of a $(1,\lambda)$–ES with simple mutation and non-adaptive, constant σ^t. For the two model functions of a linear corridor and the sphere model, a maximum rate of convergence is achieved for the values $\lambda_1 \approx 6.0$ and $\lambda_2 \approx 4.7$. Thus, ESs with $\mu/\lambda \approx 1/6$ are often tuned for a maximum rate of convergence. With a growing ratio of μ/λ the bias towards local search is shifted towards global search, which in the limit of a (λ,λ)–ES is nothing but random walk.

3 Genetic Algorithms

Abstracting from the work of Holland [Hol75], a GA can be formulated as an 8–tuple:

$$\mathrm{GA} = (P^0, \lambda, l, s, \rho, \Omega, f, t) \tag{6}$$

where

P^0	$=$	$(a_1^0, \ldots, a_\lambda^0) \in I^\lambda$	$I = \{0,1\}^l$	initial population
λ	$\in$	$\mathbf{N}$		population size
l	$\in$	$\mathbf{N}$		length of individuals' representation
s	$:$	$I^\lambda \to I^\lambda$		selection operator
ρ	$:$	$I \to \Omega$		operator determination function
Ω	$\subseteq$	$\{\omega : I \times I^\lambda \to \mathcal{P} \to I\}^1$		genetic operator set
f	$:$	$I \to \mathbf{R}$		fitness function
t	$:$	$I^\lambda \to \{0,1\}$		termination criterion

The GA is working on a *genotype-level* of binary encoded individuals, a choice, which is founded by the argument of maximizing the number of schemata (hyperplanes of varying dimension in the l–dimensional space) available with respect to a given code [Hol75, Gol89].

P^0 is the randomly generated initial population, and the parameters λ and l describe the number of individuals representing one generation and the length of the 'genetic' representation of each individual, respectively.

The selection operator s produces an intermediate population P'^t from the population P^t by the generation of copies of elements from P^t: $P'^t = s(P^t)$. This is done by taking λ subsequent samples from $P^t = (a_1^t, \ldots, a_\lambda^t)$ according to the probability distribution $\mathcal{P} \ni p_s : I \to [0,1]$, where $p_s(a_i^t) = f(a_i^t) \;/\; \sum_{j=1}^{\lambda} f(a_j^t)$ denotes the probability of individual a_i^t to be sampled. This selection scheme is called *proportional selection* [Hol75]. It leads to the expectation of individual a_i^t to occur $\eta_i^t = \lambda p_s(a_i^t)$ times in generation $t+1$ (generally η_i^t is called the *expected value* of a_i^t).

After the selection phase has taken place genetic operators are to be applied. One may think of a mapping $\rho : I \to \Omega$ which determines an operator $\omega_i^t \in \Omega$ for each individual $a_i'^t \in P'^t$ which will be applied to this individual: $\rho(a_i'^t) = \omega_i^t \; \forall i \in \{1, \ldots, \lambda\}$.

The genetic operator set $\Omega \subseteq \{\omega : I \times I^\lambda \to \mathcal{P} \to I\}$ includes genetic operators like *crossover* and *mutation* [Hol75]. The stochastic elements of these operators (application probabilities, e.g. $p_m \approx 0.001$ and $p_c \approx 0.6$, selection of loci) are included in a somehow abstract manner in the probability distribution $p \in \mathcal{P}$. According to an individual $a_i'^t \in P'^t$ the operator selects mating partners out of P^t if needed (e.g. for crossover) and determines the new individual. Operators leading to more than one offspring at first (e.g. crossover) select one of the offspring at random to be the result, while the other offspring is discarded.

Implementations often allow more than one operator to be applied sequentially to one individual, e.g. mutation is often applied to the offspring created by crossover. This might be included in the scheme described above by allowing the function obtained from ρ to be composites of operators, i.e. $\omega_i^t = \omega_{i_1}^t \circ \omega_{i_2}^t \circ \ldots \circ \omega_{i_m}^t$, e.g.: $\Omega = \{\omega_m \circ \omega_{c_1}, \omega_m\}$. Now the transition from generation t to generation $t+1$ may be described as follows:

$$\begin{aligned} s(P^t) &= P'^t = (a_1'^t, \ldots, a_\lambda'^t) & \\ \rho(a_i'^t) &= \omega_i^t & \forall i \in \{1, \ldots, \lambda\} \\ a_i^{t+1} &= \omega_i^t(a_i'^t, P^t) & \forall i \in \{1, \ldots, \lambda\} \\ P^{t+1} &= (a_1^{t+1}, \ldots, a_\lambda^{t+1}) & \end{aligned} \tag{7}$$

The fitness values are obtained by the fitness function $f : I \to \mathbf{R}$, which is usually seen as a black box containing components like decoding of individuals, constraint handling, scaling and the objective function itself. As in ESs, $t : I^\lambda \to \{0,1\}$ denotes the termination criterion.

[1] $\mathcal{P} = \{p : I \to [0,1]\}$ denotes the set of probability distributions on I.

The basic GA does not lead to a high probability of finding the global optimum of a multimodal objective function, since it often converges prematurely instead of further explorating the search space. In such cases a contradiction between a high security to find the global optimum and a high speed of convergence towards a solution becomes apparent. This can also be formulated as a *contradiction between exploration of the search space and the amount of exploitation*.

Some remarkable countermeasures, which try to solve these problems, are e.g. *scaling* of fitness values [Jon75, Gol89, GB89], *crowding* [Jon75] and *generation gap* [Gre86], *sharing* [DG89, GR87], *ranking* [Bak85], and a *multiple–point crossover* [CES89, ECS89, SCED89].

4 Summary

In the course of a closer comparison of the general working scheme it turns out that GAs and ESs are nearly identical. Both are direct, global search methods based on populations of test points which are produced by means of recombination and mutation of the better ones determined by selection. However, besides some less important differences, ESs benefit from their capability of learning on two different levels, i.e. the level of the population and the level of the strategy parameters. A experimental comparison clarifies the power of this problem–depending adaptation capability of the algorithm [HB90]. The most important differences are summarized in table 1.

GAs	ESs
Genotype level of individuals (binary coding)	Phenotype level of individuals (real–valued representation)
No knowledge about the properties of the objective function	Knowledge of the dimension of the objective function (i.e. number of variables)
Parameter space restrictions for decoding purposes	No parameter restrictions apart from machine–dependencies
Proportional selection, Ranking	(μ,λ)–selection, $(\mu+\lambda)$–selection
Recombination serves as the main search operator	Mutation serves as the main search operator
Secondary rôle of mutation	Several recombination schemes
No collective self–learning of parameter settings	**Collective self–learning of strategy parameters**

Table 1: Differences between GAs and ESs

As a concluding remark, the variety of possibilities of a parallelization of these Evolutionary Algorithms has to be mentioned. In general, this topic has been discussed elsewhere [HS90], but we would like to emphasize at least on the possibility of parallelism on different levels (parallel individuals, parallel populations), synchronously or asynchronously.

References

[Bak85] James Edward Baker. Adaptive selection methods for genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the first international conference on genetic algorithms and their applications*, pages 101–111, Hillsdale, New Jersey, 1985. Lawrence Erlbaum Associates.

[CES89] Richard A. Caruna, Larry J. Eshelman, and J. David Schaffer. Representation and hidden bias II: Eliminating defining length bias in genetic search via shuffle crossover. In N. S. Sridharan, editor, *Eleventh international joint conference on artificial intelligence*, pages 750–755. Morgan Kaufmann Publishers, August 1989.

[DG89] Kalyanmoy Deb and David E. Goldberg. An investigation of niche and species formation in genetic function optimization. In J. David Schaffer, editor, *Proceedings of the third international conference on genetic algorithms and their applications*, pages 42–50. Morgan Kaufmann Publishers, 1989.

[ECS89] Larry L. Eshelman, Richard A. Caruna, and J. David Schaffer. Biases in the crossover landscape. In J. David Schaffer, editor, *Proceedings of the third international conference on genetic algorithms and their applications*, pages 10–19. Morgan Kaufmann Publishers, 1989.

[GB89] John J. Grefenstette and James E. Baker. How genetic algorithms work: A critical look at implicit parallelism. In J. David Schaffer, editor, *Proceedings of the third international conference on genetic algorithms and their applications*, pages 20–27. Morgan Kaufmann Publishers, 1989.

[Gol89] David E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.

[GR87] David E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Genetic Algorithms and their Applications*, pages 41–49, Hillsdale, New Jersey, 1987. Lawrence Erlbaum Associates.

[Gre86] John J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, SMC–16(1):122–128, 1986.

[HB90] Frank Hoffmeister and Thomas Bäck. Genetic algorithms and evolution strategies: Similarities and differences. Technical Report "Grüne Reihe" No. 365, Department of Computer Science, University of Dortmund, November 1990.

[Hol75] John H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, 1975.

[HS90] Frank Hoffmeister and Hans-Paul Schwefel. A taxonomy of parallel evolutionary algorithms. In G. Wolf, T. Legendi, and U. Scchendel, editors, *ParCella 90*, volume 2 of *Research in Informatics*, pages 97–107, East-Berlin, GDR, 1990. Akademie Verlag.

[Jon75] Kenneth De Jong. *An analysis of the behaviour of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975. Diss. Abstr. Int. 36(10), 5140B, University Microfilms No. 76–9381.

[Rec73] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann–Holzboog Verlag, Stuttgart, 1973.

[SCED89] J. David Schaffer, Richard A. Caruna, Larry J. Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In J. David Schaffer, editor, *Proceedings of the third international conference on genetic algorithms and their applications*, pages 51–60. Morgan Kaufmann Publishers, 1989.

[Sch77] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie.* Interdisciplinary systems research; 26. Birkhäuser, Basel, FRG, 1977.

[Sch81a] Hans-Paul Schwefel. *Numerical Optimization of Computer Models.* Wiley, Chichester, 1981.

[Sch81b] Hans-Paul Schwefel. Optimum seeking methods: Subroutines for the minimization of non–linear functions of several variables by means of direct (derivative–free) methods. Interner Bericht KFA-STE-IB-7/81, Kernforschungsanlage Jülich GmbH, Jülich, FRG, October 1981.