Adaptive Search by Evolutionary Algorithms

Thomas Bäck* Frank Hoffmeister*

baeck@unido.uucp frank@unido.uucp

December 1990

Abstract

Evolution Strategies (ESs) and Genetic Algorithms (GAs) are compared in a formal way. It is shown, that both algorithms are identical with respect to their major working scheme, but nevertheless they exhibit significant differences with respect to the details of the selection scheme, the amount of the genetic representation and, especially, the self-adaptation of strategy parameters.

1 Introduction

Genetic Algorithms (GAs) and *Evolution Strategies* (ESs) are two strata of consciously pursued attempts to imitate principles of organic evolution processes as rules for optimum seeking procedures. Both rely upon the collective learning paradigm gleaned from natural evolution and implement the principles 'population', 'mutation', 'recombination' and 'selection'. In addition, ESs try to use a collective self–learning mechanism to adapt its strategy parameters during the optimum search (*adaptive search*).

The aim of this paper is to give a formal description of both algorithms and to show their similarities as well as differences. 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 (μ+λ)–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:

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

where

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

*University of Dortmund, Dept. of Computer Science XI, P.O. Box 500500, D–4600 Dortmund 50

Each individual $a_i^t = (x^t, \sigma^t, \theta^t) \in P^t$ within a population consists of three vectors, namely, the set of *object variables* $x^t \in \boldsymbol{R}^n$, the set of *step sizes* $\sigma^t \in \boldsymbol{R}^n$, and the set of *inclination angles* $\theta^t \in \boldsymbol{R}^w$. σ^t and θ^t are internal strategy parameters which control the mutation of x^t.

According to the traditional point of view a population P^t consists of μ parents which produce an intermediate population P'^t made up from λ offspring 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 $s : I^{\mu+\lambda} \rightarrow I^\mu$. By means of selection P'^t is reduced to μ individuals, which become the set of parents P^{t+1} of the next generation.

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

According to the rule 'survival of the fittest individuals' the selection operator s removes the least fit individuals from the intermediate population. Figure 3 illustrates the ES working scheme. The iteration $P^t \rightarrow P^{t+1}$ 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 $a = (x_a, \sigma_a, \theta_a)$ and $b = (x_b, \sigma_b, \theta_b) \in I$ [Sch81b]:

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

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. Two kinds of mutations are distinguished. First, *simple mutations* are introduced:

$$\begin{aligned} m(a') &= a'' = (x'', \sigma'') \in I \quad ; x'' \in \boldsymbol{R}^n, \sigma'' \in \boldsymbol{R}^n \\ \sigma'' &= \sigma' \cdot \exp \mathbf{N}_0(\Delta\sigma) \qquad ; \forall i \in \{1, \ldots, n\} \\ x'' &= x' + \mathbf{N}_0(\sigma'') \end{aligned} \tag{5}$$

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*. For simple mutations the strategy parameter θ is not required to be part of the genetic information.

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 $\lambda_2 \approx 4.7$. These values are pretty close to Rechenberg's 1/5 success rule for his deterministic control of the step size [Rec73]. 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.

Like in a (1+1)–ES the maximum rate of convergence of a (μ+λ)–ES and (μ,λ)–ES is inversely proportional to n, the number of object variables [Sch77, Sch81a]. Due to this fact ESs have the tendency to reduce the search space by adjusting some step sizes towards zero in order to achieve a higher rate of convergence. This might result in a restricted search along some coordinate axes, only, eventually yielding a poor rate of convergence, since flipping some step sizes is a rare event. With the introduction of *correlated mutations* [Sch87] these effects are reduced if not avoided:

$$\begin{aligned} m(a') &= a'' = (x'', \sigma'', \theta'') \in I \quad I = \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R}^w \\ \sigma'' &= \sigma' \cdot \exp \mathbf{N}_0(\Delta\sigma) \\ \theta'' &= \theta' + \mathbf{N}_0(\Delta\theta) \\ x'' &= x' + \mathbf{C}_0(\sigma'', \theta'') \end{aligned} \tag{6}$$

where $\mathbf{N}_0$ denotes a vector of independent Gaussian random numbers with expectation zero and standard deviations $\Delta\sigma_i$ and $\Delta\theta_i$, respectively. $\mathbf{C}_0(\cdot,\cdot)$ refers to a normally distributed random vector with expectation zero and probability density

$$\begin{aligned} \Delta x &= \mathbf{N}_0(\sigma'') \\ p\left(\mathbf{C}_0(\cdot,\cdot)\right) &= \sqrt{\frac{\det \mathbf{A}}{(2\pi)^{\mathrm{n}}}}\, e^{-\frac{1}{2}\Delta x^{\mathrm{T}} \mathbf{A}\, \Delta x} \end{aligned} \tag{7}$$

The diagonal elements of the covariance matrix $\mathbf{A^{-1}}$ are the independent variances $\sigma_i''^2$ (squares of the mutation step sizes) of the object variables x_i^t, while the off-diagonal elements represent the covariances $c_{i,j}$ of the mutations. Schwefel restricts the areas of equal probability density to n-dimensional hyperellipsoids, which are realized by a set of inclination angles $\theta'' \in \boldsymbol{R}^w$ of the main axes of the hyperellipsoid, $w = 1/2\, n(n-1)$. The standard deviations σ_i'' serve as a kind of mean step size along those axes.

The new strategy parameter θ^t is also incorporated into the genetic representation of each individual. This way the ES may adapt to any preferred direction of search by means of self-learning on the long run. The self-adaptation of strategy parameters works, since individuals with better parameter settings are expected to perform better than the rest. Hence, selection automatically favours better adapted settings, eventually speeding up the search. The step sizes σ_i of an individual make up an ellipsoid of equal probability to place an offspring if these

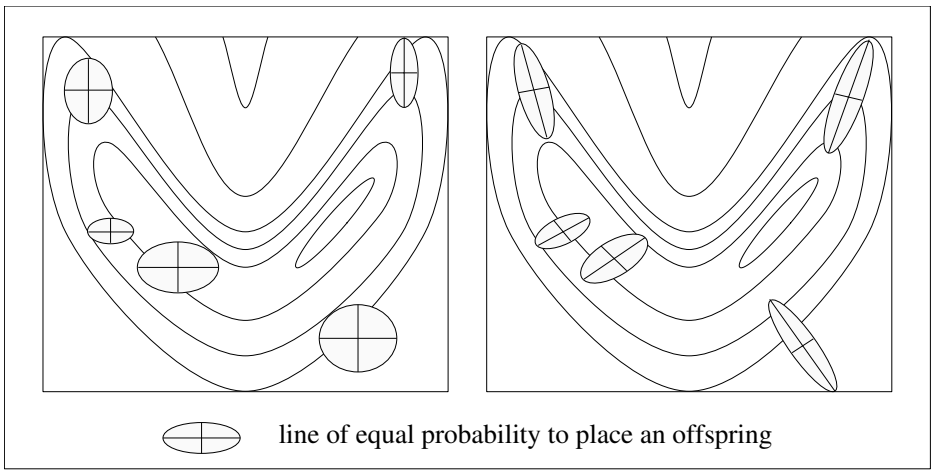

Figure 1: Searching with simple and correlated mutations

step sizes are applied to x^t of the individual itself. The left illustration in figure 1 shows some individuals with their corresponding ellipsoids if the step sizes are *not* correlated (simple mutations), while the right illustration shows the same individuals with correlated mutations.

3 Genetic Algorithms

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

$$\mathrm{GA} = (P^0, \lambda, l, s, \rho, \Omega, f, t) \tag{8}$$

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	$:$	$I \to \boldsymbol{R}$		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 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:

$$\begin{array}{lcll} s(P^t) & = & P'^t = (a_1'^t, \ldots, a_\lambda'^t) & \\ \rho(a_i'^t) & = & \omega_i^t & \forall i \in \{1, \ldots, \lambda\} \\ a_i^{t+1} & = & \omega_i^t(a_i'^t, P^t) & \forall i \in \{1, \ldots, \lambda\} \\ P^{t+1} & = & (a_1^{t+1}, \ldots, a_\lambda^{t+1}) & \end{array} \tag{9}$$

The schematic representation shown in figure 2 should help to remember the general scheme. Here the typical shape of a probability distribution of the selection probabilities p_s is given too, where the individuals are sorted by decreasing fitness values.

These fitness values are obtained by the fitness function $f : I \to \boldsymbol{R}$, 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.

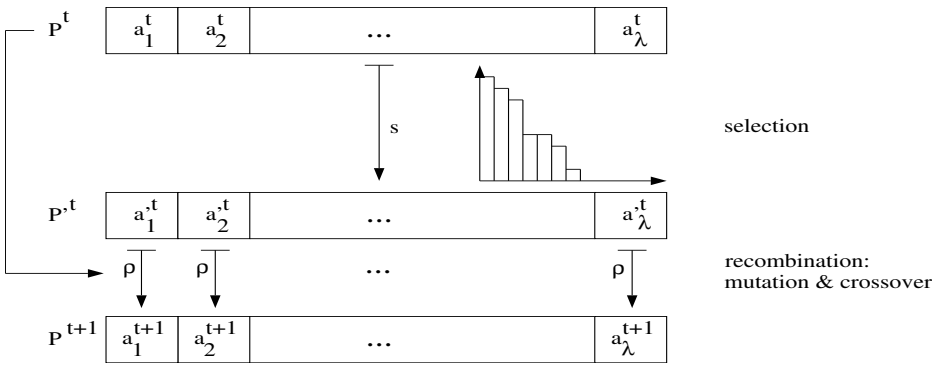

Figure 2: Schema of generation transformation in a genetic algorithm

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 Comparison of the Algorithms

Many differences between ESs and GAs directly or indirectly stem from a substantial difference in the amount of 'genetic' information and its representation. While GAs operate on fixed-sized bitstrings, ESs work on a phenotypic level of real-valued vectors and can utilize much more knowledge about the application domain. The additional vectors σ and θ besides the object variables x_i are used to control *individually* the mutation of x_i, thus controlling the hillclimbing imposed by the combination of mutation and selection. The working scheme of a (μ,λ)–ES is

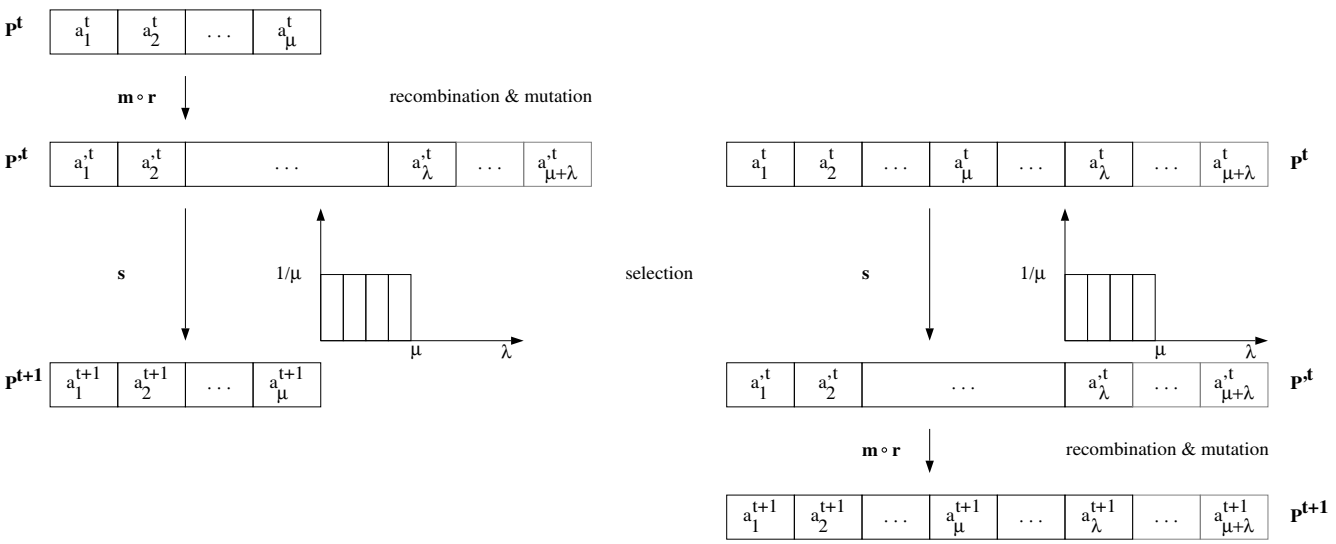

Figure 3: Original (left) and modified (right) working scheme of ESs

outlined in figure 3. The traditional point of view is presented in the left illustration, while a modified working scheme is given by the right illustration in order to allow for an easy comparison of ESs and GAs. In the modified scheme the term 'population' refers to the set of offspring in case of (μ,λ)–ES or to the set of parents plus offspring

in case of (μ+λ)–ES. Then, the working scheme of ESs may be interpreted as follows: Every population P^t consists of λ or $\mu + \lambda$ individuals which are the (parents plus) offspring of the former generation. By means of selection the μ best are chosen to build up an intermediate population P'^t, where worse individuals are replaced by copies of the μ best ones. Then, from P'^t successively parents are selected with uniform probability to produce an offspring by means of recombination and mutation until a new population has been completely produced.

If the details of the genetic operators are neglected this working scheme directly corresponds to the working scheme of GAs (figure 2). For proportional selection in GAs the reproduction rates of individuals are dynamically assigned with respect to their relative fitness, which possibly leads to significantly differing reproduction rates and the danger of premature convergence. In ESs, reproduction rates are assigned statically to the μ best individuals with *no* regard to their relative fitness.

Furthermore, the rôle of mutation and recombination is different in ESs and GAs. While in GAs mutation only serves to recover from lost alleles, in ESs mutation implements some kind of hill-climbing search procedure with self-adapting step sizes σ^t and inclination angles θ^t, which considerably speed up the optimum search. While the standard 1-point crossover in GAs shows no resemblance to recombination in ESs, *uniform crossover* [Sys89] is identical to discrete recombination in ESs if the cardinality of the alphabets is ignored. Intermediate recombination, which is available for non-binary alphabets, might allow an ES to escape from local optima.

5 Summary

In the course of section 4 it turned out that with respect to the major working scheme 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 (figure 3). Nevertheless, the differences imposed by the different amount of 'genetic' information and its representation are significant as the experimental results clearly indicate. 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, (μ+λ)–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

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 Schaffer [Sch89], pages 42–50.

[ECS89] Larry L. Eshelman, Richard A. Caruna, and J. David Schaffer. Biases in the crossover landscape. In Schaffer [Sch89], pages 10–19.

[GB89] John J. Grefenstette and James E. Baker. How genetic algorithms work: A critical look at implicit parallelism. In Schaffer [Sch89], pages 20–27.

[Gol89] David E. Goldberg. Sizing populations for serial and parallel genetic algorithms. In Schaffer [Sch89], pages 70–79.

[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.

[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 Schaffer [Sch89], pages 51–60.

[Sch77] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, volume 26 of *Interdisciplinary systems research.* Birkhäuser, Basel, 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, Germany, October 1981.

[Sch87] Hans-Paul Schwefel. Collective phenomena in evolutionary systems. In *Preprints of the 31st Annual Meeting of the International Society for General System Research, Budapest*, volume 2, pages 1025–1033, June 1987.

[Sch89] J. David Schaffer, editor. *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, California, June 1989. Morgan Kaufmann Publishers.

[Sys89] Gilbert Syswerda. Uniform crossover in genetic algorithms. In Schaffer [Sch89], pages 2–9.