Evolution Strategies and Genetic Algorithms

© FH

UNI DO
SYS

Abstract

Evolution Strategies (ESs) and Genetic Algorithms (GAs) are powerful, independently developed tools for solving global optimization problems. Both methods rely upon imitating the collective learning paradigm of evolution and population genetics and use the principles of *population*, *mutation*, *recombination*, and *selection*. With respect to the major working scheme both algorithms are identical, but the details of their implementations establish

!!

Contents

...

4 Experimental Comparison of the Algorithms 43
4.1 §1 — Sphere Model 46
4.2 §2 — Schwefel's Problem 1.5 25
4.3 §3 — Step Function 49
4.4 §4 — Schwefel's Problem 2.21 51
4.5 §5 — Rosenbrock's Function 57
4.6 §6 — Generalized Rastrigin's Function 24

List of Figures

5.1 Sketch of selection techniques 62
5.2 Right-skewed and left-skewed selection situations 63
5.3 Best performance of GAs on f_1 and f_7 with respect to the degree of extinctiveness 65
5.4 Comparison of different mutation rates for a GA with (10,50)-proportional selection and (50,50)-proportional selection on f_1 (left) and f_7 (right). . . . 66
5.5 Best performance of ESs on f_1 with simple (left) and correlated (right) mutations 67

List of Tables

1

[3]

[3]

It is the major aim of this work to show the similarities as well as the differences of both approaches. As a first step the basic algorithms will be described using a minimum notation (as far as possible) to make the comparison easier. The comparison will demonstrate that the basic CA as well as the basic ES implement the same principles of behavior. Starting with the power of nature's problem-solving mechanism. With respect to the major advantages resulting from these principles and algorithms the almost identical, but more complex mechanisms of the ES and CA are also strengthened by a common discussion of the new results.

It will be shown that both algorithms can be applied to some significant differences.

1.1 Parameter Optimization

Until now the major domain of application of Genetic Algorithms and Evolution Strategies is parameter optimization. As a result there are some basic terms from optimization that will be used frequently throughout the text. In order to provide a self-contained source of information, the required terms are introduced below.

The overall goal of a *parameter optimization problem*

$$f : M \subseteq M_1 \times \ldots \times M_n \to \mathbb{R}\,,\ M \neq \emptyset \tag{1.1}$$

2.1 Genetic Algorithms

Holland's work [Hol75] provides some slightly different definitions of genetic algorithms as a reproductive plan. His descriptions use a semi-formal model for the explanation of the basic algorithms. With the aim to simplify a comparison of GAs to ESs we will try to adopt a more abstract, general description which may help to see differences and similarities between both approaches. In this description a GA is formulated as an 8–tuple:

$$GA = (P^{\mathcal{P}}, \lambda, \mathcal{L}, s, \mathcal{Q}, \mathcal{L}, t) \tag{2.1}$$

generation although only λ individuals are worked on. This effect is denoted by the term *implicit parallelism* [Goldberg, Holland].

λ_0 is the randomly generated initial population, and the parameters λ and μ describe the number of individuals representing one generation and the family of the representation of each individual, respectively.

Willy Hollands, algorithms give a sedimentil, time-by-time generation of offspring to form the next population. the steps of selecting individuals to generate copies and the application of operators will be considered in the basic c.f. to form λ individuals during one step. This is mutation loss of genetility and allows comparison to the formal framework

schemata. This remains true, even when the loss coming from the introduction of genetic operators is taken into account, too, and is designated as the schema theorem. It simply states that the GA is an optimum procedure for the allocation of trials to the operators present schemata. If crossover and mutation (the most commonly used genetic operators) are neglected, two further disadvantages have to be introduced: The order $o(H)$ of a schemata $H \in \{0,1,*\}^\ell$ is given by the number of fixed positions of the schema, i.e.

$$\begin{aligned} o(H) &= |\{i \mid \exists j \in \{0,1\} : h_i = j\}| \\ o &: \{0,1,*\}^\ell \to \{1,\ldots,\ell\} \end{aligned} \tag{5.6}$$

(1) Random selection of a starting parent $a_S = (a_{S,1} \dots a_{S,N})$ out of N_I.

(2) Random selection of a crossover-position $\chi \in \{1, \dots, N-1\}$.

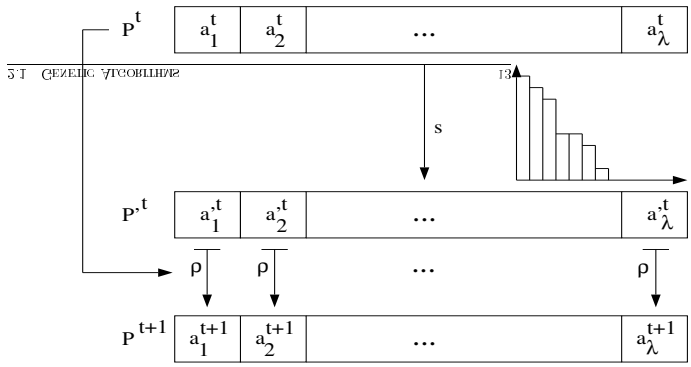
(3) Forming of two new individuals $q_I = (a_{I,1} \dots a_{I,\chi} a_{S,\chi+1} \dots a_{S,N})$ and $q_S = (a_{S,1} \dots a_{S,\chi} a_{I,\chi+1} \dots a_{I,N})$.

(4) Random selection of $\alpha \in \{q_I, q_S\}$.

(e.g. $\Omega = \{\omega_w \circ \omega_{cl}, \omega_w\}^4$ for the basic CA). Now the transition from generation t to generation $t+1$ may be described as follows:

$$\begin{aligned}
s(\lambda_t) &= \lambda_{st} = (a_1^t, \ldots, a_\lambda^t) \\
b(a_k^t) &= \omega_k^t \qquad \forall k \in \{1, \ldots, \lambda\} \\
a_{k+1}^t &= \omega_k^t(a_k^t, \lambda_t) \qquad \forall k \in \{1, \ldots, \lambda\} \\
\lambda_{t+1} &= (a_{1+1}^t, \ldots, a_{\lambda+1}^t)
\end{aligned} \tag{5.11}$$

Although the λ-tuple notation used here to describe a computation seems to imply a deterministic order to individuals, the algorithm called by the selection mechanism


P^t
a_1^t
a_2^t
...
a_λ^t
s
selection
P'^t
a'_1^t
a'_2^t
...
a'_λ^t
ρ
ρ
...
ρ
recombination:
mutation & crossover
P^t+1
a_1^t+1
a_2^t+1
...
a_λ^t+1

this will be chosen to be a characteristic identity of a point GA instead of the primary code
as explained above. This is due to the very early operation of Hoeffding [Hoff], that is, a
GF(2) code construction increases the performance of a GA. For GF(2) codes the decoding
function Γ can be expressed by

$$\Gamma^{a_1 a_2 \ldots a_n}(\sigma_1 \ldots \sigma_n) = a + \frac{b-a}{k-1} \cdot \left(\sum_{i=1}^{k} \left(\bigoplus_{i=1}^{n} \sigma_i \right) - 1 \right) \tag{5.17}$$

where $\oplus$ denotes summation modulo 2 (i.e. $1 \oplus 1 = 0$, logical XOR function).

Combining both qualities, we have the ability to construct a desired Γ

known to be the optimum. Nowadays, the initial uncertainty is usually implemented with a constant population size, by copying the best individuals and filling up to size λ based on proportional selection.

5.5 Evolution Strategies

According to Kozelemsky [Koz73], the first attempts to automate an evolution strategy took place in 1964 at the Technical University of Berlin (TUB). The idea to mimic mutation, recombination and selection in nature was to find optimal solutions to complex technical problems. Ingo Rechenberg and Hans-Paul Schwefel were two students of this university who started to combine their efforts and resources. Schwefel had built a device which was supposed to find the optimal configuration of a flexible pipe bend by means of random changes applied to a prototype. Rechenberg suggested to replace the random changes by small, Gaussian mutations and a population of more than one individual. Thus, the idea to mimic a primordial

The mutation operator is applied to all components of the optcd parameter $\mathbf{x}_k$. Accordingly to the biological observation that only a minority of genes undergo mutation is realized by normally distributed changes occur more often than large ones. mutation is realized by normally distributed random numbers:

$$\mathbf{x}'_k = \mathbf{x}_k + \mathcal{N}(\mathbf{0}, \mathbf{c}_k) \tag{5.5}$$

where $\mathcal{N}(\mathbf{0}, \mathbf{c}_k)$ denotes a vector of independent Gaussian random numbers with mean and standard deviations $\sigma_k^{(i)}$ $(i = 1, \ldots, m)$. The selection operator then determines the fitter individuals to become the parent of the next "generation":

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}'_k & \text{if } f(\mathbf{x}'_k) < f(\mathbf{x}_k) \wedge \forall i \in \{1, \ldots, \mu\} : g_i(\mathbf{x}'_k) > 0 \\ \mathbf{x}_k & \text{if } f(\mathbf{x}'_k) \geq f(\mathbf{x}_k) \vee \exists i \in \{1, \ldots, \mu\} : g_i(\mathbf{x}'_k) \leq 0 \end{cases} \tag{5.6}$$

The longest proof is omitted here; it can be found in [Bot72]. The basic idea is to use the monotone sequence $\gamma(x_0) \leq \gamma(x_1) \leq \ldots \leq \gamma(x_i) \leq \ldots$ of objective function values generated by the process. Then, for the limit value $\gamma^* \geq \gamma = \lim_{i\to\infty}\gamma(x_i)$ under the assumption $\gamma > \gamma^*$ a contradiction emerges, such that $\gamma = \gamma^*$ must hold.

As it is well known from similar theorems for Gradient Methods [AK82] and Genetic Algorithms (section 5.1 and [EAH91]), such results are not of much practical relevance that do the unlimited time consumption. However, they are primarily arguments which show the mathematical rationality of the method. In fact, we are interested in the exploration behavior rate Φ, which is given by the quotient of the distance covered towards the optimum and the number of steps (generations) necessary for it. Unfortunately, this rate can only be determined for very simple objective functions and algorithms, so that practical experiments are unavoidable.

The optimum standard deviations can be computed with the probabilities for a successful run:

$$\begin{aligned} h_i^1 &= \frac{1}{2}\left(1-\sqrt{1-\left(\frac{\sigma_i}{\pi}\right)^{n-1}}\right) && \text{for } n \gg 1 \\ h_i^2 &= \frac{1}{2}\left(1-\operatorname{erf}\left(\frac{n\sigma_i}{\sqrt{2}^n}\right)\right) && \text{for } n \gg 1 \end{aligned} \tag{5.52}$$

For optimum step sizes the probabilities that to the results $h_{\text{opt}}^1 = 1/e \approx 0.81$ and

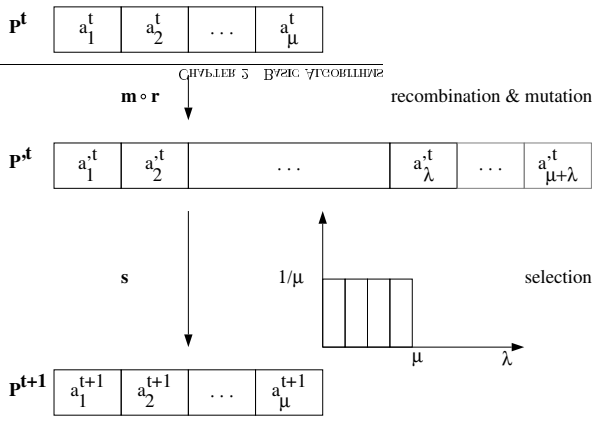
λ_0	$=$	$(a_0^1, \ldots, a_0^n) \in \lambda_n : \lambda = \mathbb{F}_{2^n} \times \mathbb{F}_{2^n}$	bobmalation
λ_4	$>$	λ	multiply by bracket
τ	$:$	$\lambda_n \to \lambda$	tracomplimation obgrator
$\pi\pi$	$:$	$\lambda \to \lambda$	multiation obgrator
σ	$:$	$\lambda_{n+1} \to \lambda_n$	selection obgrator
c_4, c_6	$\in$	$\mathbb{F}_2$	step-size control
λ	$:$	$\mathbb{F}_{2^n} \to \mathbb{F}_2$	oplycrypy function

5.5.3 (N+Y)–ES and (N,Y)–ES

The motivation to extend the (N+1)–ES to a (N+Y)–ES and (N,Y)–ES has been twofold [Schw77, Schw81]: first, to make use of the first time (initial) battle computers, and secondly, to enable self-adaptation of strategic parameters like the (very important) mutation variances in a correlated mutation. Instead of changing the σ_i in an exogenous heuristic in a deterministic manner, Schwefel completely removed σ_i as a part of the genetic information and mutation as well. Those individuals with better adapted strategy parameters are expected to reproduce more often and therefore individuals with poorer reproduction will ignore their accordingly and soonest or later their genetic information is lost. Thus, selection will favour those individuals with better strategy parameters.

τ	:	$\mathbb{N}_n \to \mathbb{N}$	recombination operator
m	:	$\mathbb{N} \to \mathbb{N}$	mutation operator
s	:	$\mathbb{N}_\gamma \to \mathbb{N}_n$	selection operator
Δ^σ	$\in$	$\mathbb{R}$	single-step-size control
ζ	:	$\mathbb{R}_n \to \mathbb{R}$	objective function
g_i	:	$\mathbb{R}_n \to \mathbb{R}$ $\quad : \nu \in \{1,\ldots,q\}$	constraint functions
λ	:	$\mathbb{N}_n \to \{0,1\}$	termination criterion

In case of a $(\mu+\lambda)$-ES the selection operator s is modified to $s : \mathbb{N}_{\mu+\lambda} \to \mathbb{N}_\mu$. The


$\mathbf{p}^t$
a_1^t a_2^t ... a_μ^t
$\mathbf{m} \circ \mathbf{r}$
recombination & mutation
$\mathbf{p'}^t$
a'^t_1 a'^t_2 ... a'^t_λ ... $a'^t_{\mu+\lambda}$
$\mathbf{s}$
$1/\mu$
selection
μ λ
$\mathbf{p}^{t+1}$
a_1^{t+1} a_2^{t+1} ... a_μ^{t+1}

5.3 Comparison of the Basic Algorithms

$\mathbf{P^{t+1}}$ a^{t+1}_1 a^{t+1}_2 ... a^{t+1}_μ

Major differences between ES's and GA's efficacy for multimodal search arise from a supplantial difference in the underlying "genetic" representation used by the algorithms. While in GA's operators do not exist which map arbitrary bit-sized phi strings to the genotypes, ES's operate on fixed-sized phi strings which are simultaneously mapped to the genotypes in the real-valued "arithmetic" which is a second straightforward to the operators arbitrary application of the operators. ES's work on real-valued vectors. This result not be confirmed by the fact that ES's are operating combinatorial on a dynamic level, namely that ES's cannot handle multimodal search spaces efficiently. Instead, ES's are operating on a dynamic level, namely that ES's cannot handle multimodal search spaces efficiently. Instead, ES's are operating combinatorial

optimal of the former generation. By means of selection the λ best are chosen to build up an intermediate population λ_{μ}, where worse individuals are replaced by copies of the λ best ones. Then, from λ_{μ} successively two 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 derivatives of the fitness operators are neglected this notation schema effectively corresponds to the working schema of GAs (figure 5.1). The initially introduced information λ_{μ}^{μ} is of no harm, as the dependencies of the λ best individuals are maintained but it points out the difficult handling of reproduction rates in ES and GAs.

onto individuals are similar, although their topic is different in both algorithms.

In basic GAs recombination is realized by 1-point crossover (§ 4) which provides only a limited mixing of information (building blocks) and which does not take into account the internal substructure imposed by the decoding function. Recombination as realized in ES (§ 5.2) achieves a much better mixing of the genetic information within the individuals, since the population performs operations on the vectors' values.

From a conceptional point of view ES and GAs are nearly identical. The operators differ mainly in their implementation and in the sequence they are applied — of course — influences the search behavior.

Chapter 3

algorithm. i.e. a CA with maximum rule complexity to the random maximally unstable one.

Optimization: One way to speed the basic CA is to store two intermediate steps of the search space directly, but it also compresses the query of converging fragments. Here, the contradiction pathway increases the complexity of finding a global optimum and the intersect of the speed of convergence towards a solution becomes apparent. It may also be formulated as a contradiction pathway exploration of the search space and the minute of exploration to do so, but is stressed that as the most important basic principle for the use of dual optimization techniques. In the next section some of the computational strategies will be discussed.

positive. Hence, the translation of scaling fractions was an implication of Hollands definition of reproductive biases for objective functions which may yield negative values.

Various proposals a scaling technique that not improve results on the initial minimodal antisce of DE. Goldbergs real function λ (Sharkey's toxplots) [Goldb]. His search for a solution led DE. Goldberg to the investigation of a combination of a crossbased scheme with a generation gap technique. In a GA with generation gap G ($0 \geq G \geq 1$) only a part of a generation N are selected as potore according to proportional selection, and $N \cdot (1 - G)$ individuals in N are selected at random (i.e. with probability $1/N$) [Gro93]. This is formalized here as follows:

where $\Lambda^i := (a_1^i, \ldots, a_{\lambda_i}^i)$ $\forall i = 1, \ldots, \nu+1$. Here, each intermediate population Λ^{i+1} is optimized from Λ^i ($1 \geq i \geq \nu$) by a kind of exchange operation.

$$\xi : \Lambda \times \Lambda_f \to \Lambda_f \quad , \quad \xi(a_x, (a_1^1, \ldots, \overbrace{a_i^x, \ldots, a_i^y}^{\Lambda^i}, \ldots, a_{\lambda_f}^f)) = (a_1^1, \ldots, \overbrace{a_x, \ldots, a_y}^{\Lambda^{i+1}}, \ldots, a_{\lambda_f}^f) \tag{3.7}$$

where $\Omega_i \in \bar{C}_i = \{a^1, \ldots, a^{\lambda_i}\}$ is chosen to be the intermediate multi recombinant of the population Λ^i according to a set of all elements from the set $C_i \cdot C_i$ is optimized by planning exploitative similarity to a_0

A genetic scheme applies the population into subpopulations of individuals, where the number of subpopulations ideally should be the number of the multimodal function[3]. The extension of a basic GA by a genetic scheme simply consists the fitness function, which now includes a description of the individuals, fitness according to the presence of other individuals in its neighborhood. Here, λ is extended to

$$\lambda' = \lambda \circ \phi \circ \gamma \tag{3.6}$$

(again scaling may be necessary to prevent negative fitness values), where

$$\lambda' : \mathbb{R} \to \mathbb{R}$$

in finite graphs $\Xi(\alpha_k^i)$ denotes the index of α_k^i in $\mathcal{H}(\alpha_k^i,\ldots,\alpha_\gamma^i)$ is sorted in increasing order by $\gamma(\alpha_k^i)$ (remember the convention that the optimization problem 3.1.5 is given in minimization form!).

$$\forall i\in\{1,\ldots,\gamma\}:\Xi(\alpha_k^i)=\kappa \iff \forall i\in\{1,\ldots,\gamma-1\}:\gamma(\alpha_k^i)\le\gamma(\alpha_{k+1}^i) \quad (3.1.8)$$

Here, $w^{max}=5-w^{worst}$ and $1\le w^{worst}\le 5$ are properly for the maximum and minimum expected lossy values, according to the constraints $\forall i\in\{1,\ldots,\gamma\}:\mu_k(\alpha_k^i)\le 0$ and $\sum_{j=1}^{\gamma}\mu_k(\alpha_k^i)=1$. Again, suppose κ is a value of $w^{worst}=1.1$ for the maximum expected value and w^{max} that for the corresponding minimum and maximum selection [Ha02]. λ^{max} and λ^{worst} that for the corresponding minimum and maximum selection

proportional selection is slower. while the other three seem to perform equally well. the difference between tournament selection and proportional selection is not significant for this problem.

optimal pressure is given by the minimum scaling factor that still allows the best individual to reproduce once on average. if the pressure is too high, premature convergence occurs.

premature and flood variants to proportional selection only fill in gaps—they never occupy the population.

3.5 Tournament Selection

In order for selection to occur, there must be differences between the fitnesses of individuals in the population. Tournament selection, unlike proportional selection, does not require explicit knowledge of the fitness of individuals. Instead, it requires only that individuals can be compared directly with one another to see which is better. Tournament selection is therefore applicable to problems where fitness is defined implicitly by a set of rules or by direct competition between individuals, as in evolving game-playing strategies. To run a GA with tournament selection, a tournament size N is chosen. To select an individual, N individuals are chosen at random from the population and they compete in a tournament, with the winner being selected for reproduction. Normally, the tournament size is 2, a binary tournament. Binary tournaments can be run deterministically so that the fitter individual in every pair is selected for reproduction, or probabilistically so that the fitter individual is selected with a probability p, which is a parameter of the selection method. Binary tournaments can be generalized to tournaments of arbitrary size N, with the fittest individual in the group of N being selected. Tournament size can be deterministic or probabilistic in the same manner as binary tournaments.

selection with is possible (random selection). Thus, it can be expected that genotypic
variety is maintained for a longer time than with proportional selection only, and that
the convergence of a possible elimination of premature convergence slows down.

3.1.3 Recombination Schemes in GAs

In the description of the basic algorithms the operator set of a basic GA included only
one-point crossover χ^{cp}, and mutation χ^{m}, respectively that combination, i.e. $\Omega = \{\chi^{\text{m}}$ or
$\chi^{\text{cp}}, \chi^{\text{m}}\}$. As already explained, this operator combination is not effective enough to

With respect to the genotypic level of GAs multipoint crossover is analogous to bitwise direct two-point recombination in E2s. On the other hand, with respect to the phenotypic level of E2s a crossover–operator for GAs would first to know the positions of the boundaries of the decoded parameter values in the genotype to test a full analogy to the basic recombination in E2s. If boundaries the operator would switch randomly between the parameters to determine the one which transfers the next parameter encoding bit sequence to the offspring. As far as we know no such crossover has been implemented to test yet, but multipoint crossover could easily be extended into new pictures the interpretation into a superlines (genes) and using a crossover mask of length *m*. Furthermore, it should be mentioned here that exponences with an adjacent of bigger cardinality (increased of

Another attempt would be to include the parameter settings of a CA into the representation of individuals as it has been done for ES's. Until now, there has been little experience and investigation of such approaches. As suggested in 1991 by De Jong [Jon92], strategy parameters can be used for mapping from the space of individuals to the space of solutions, which changes dynamically based upon internal measurements over the population [Bäc97].

As mentioned above, the adaptively changing CA-complications, the functionality crossover operator has to be mentioned [ZM97]. This operator attaches to each individual a number encoded description of how to perform crossover on this individual. This operator must

In order to describe the mechanism of this operator, an example adopted from [Sch87] is given here in a slightly modified notation:

$$a_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$$
$$a_2 = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

Here, the crossmutation marks indicate crossover points, i.e. the individuals would be decomposed in the crossed version as

$$a_1 = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$
$$a_2 = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \mid 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$$

Punctuated crossover realizes in

3.3.1 Recombination Types

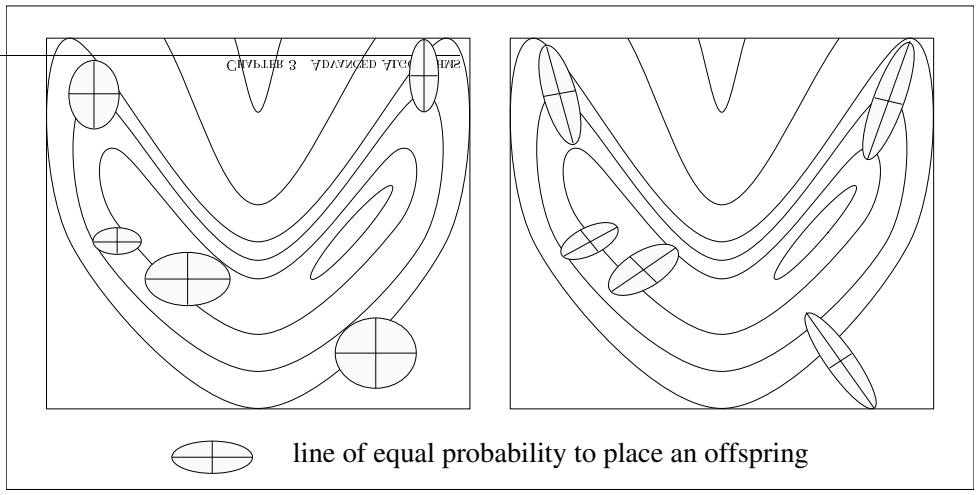
The introduction of intermediate recombination is motivated by the following deficiencies of standard crossover. With a population structure of μ individuals and λ offspring, the individuals will rarely occupy adjacent positions in $\mathbb{N}^n$ for $n > 1$. Therefore, the crossover operator will usually produce offspring that are located in a hypercube that is relatively far away from the parents. This fact slows down the search. Intermediate recombination can be defined as follows. Each component of the offspring vector is the arithmetical mean of the i-th components of two parents that are randomly drawn from the population. Formally, this is done for each $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, \lambda\}$:

$$\begin{aligned} r(x_k) &= a'_k = (a'_{k,1}, \ldots, a'_{k,n}) \in I \\ a'_{k,i} &= \frac{a_{m,i} + a_{f,i}}{2} \quad \text{with } m, f \in \{1, \ldots, \mu\} \end{aligned} \tag{3.9}$$

m and f are randomly selected integers between 1 and μ which are generated anew for each k and i. Thus, in contrast to standard crossover, the two parents are not identical for all components. Hence, the offspring will usually not be located on a parent hypercube corner. Instead, it will be located somewhere near the center of the hypercube spanned by the parents. This operator is called discrete recombination. It is the standard recombination operator in evolution strategies.

3.2.2 Correlated Mutations

In ES mutation realizes a kind of hill–climbing search procedure (2.26), when it is considered in combination with selection alone. With dedicated σ_i for each object variable x_i; preferred directions of search can be established only along the axes of the coordinate system. In general, the best search direction (the gradient) is not aligned on those axes. Thus, an optimum rate of progress is achieved only by chance when suitable mutations coincide, i.e. when they are correlated. Otherwise, the trajectory of the population through the search space is zigzagging along the gradient. In order to avoid this reduction of the rate of progress, Schwefel [Sch81] extended the mutation operator to handle *correlated*


line of equal probability to place an offspring

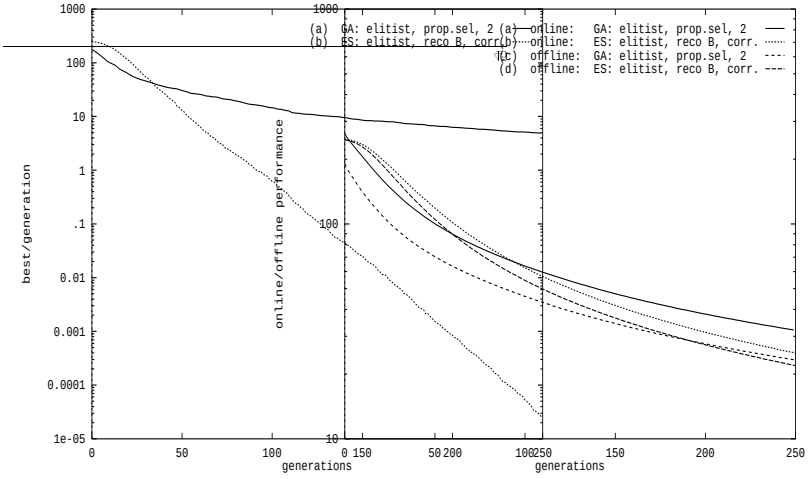
optimization [Kut91], and parameter optimization in general [BBKS91, Bot97, BBT09].

Furthermore, the scalable parallelism of such an Evolutionary Algorithm [Hol92] as proposed in the Quarterly of Optimund, especially on a small Transputer network with 96 transputers [Bor89, Lud91].

CA's	E2's
Genotype level of multicellular (plant vs. animal)	Phenotype level of multicellular (tegl-lsyned representation)
No knowledge about the objective function's properties	Knowledge of the dimension of the objective function (i.e. number of variables)
Parameter space restrictions for decoding purposes	No parameter restrictions are made from machine-dependencies

Table 4.2: Test of local functions

number	description	duration
λ_1	Depth Method	30
λ_2	Rosenbrock's Function	5
λ_3	step Function	2
λ_4	Shekel's Foxholes	5
λ_5	Schwefel's Problem 1.5	50
λ_6	Generalized Rastrigin's Function	50
λ_7	Depth Method	30

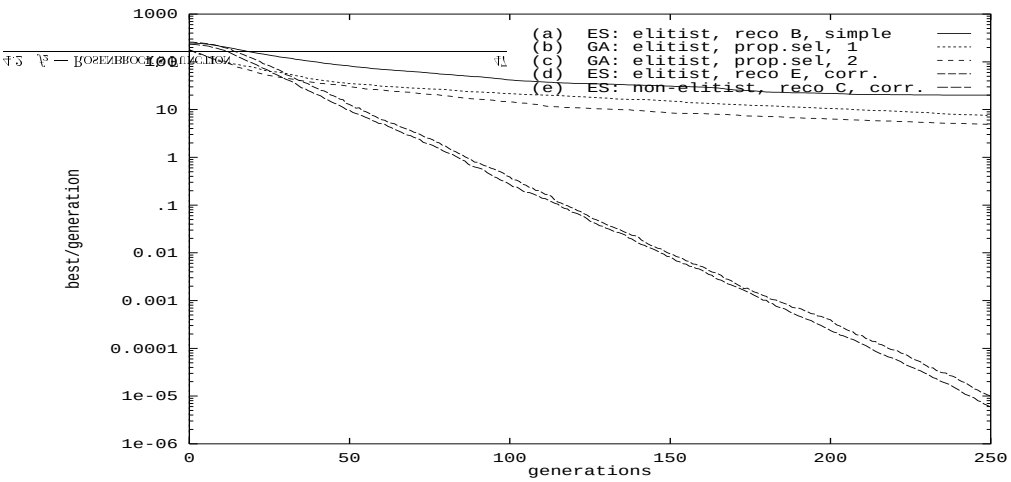

(a) GA: elitist, prop.sel, 2
(b) ES: elitist, reco B, corr.
(a) online: GA: elitist, prop.sel, 2
(b) online: ES: elitist, reco B, corr.
(c) offline: GA: elitist, prop.sel, 2
(d) offline: ES: elitist, reco B, corr.
best/generation
online/offline performance
generations
generations
1000
100
10
1
.1
0.01
0.001
0.0001
1e-05
0
50
100
150
200
250

4.1 f1 — Sphere Model

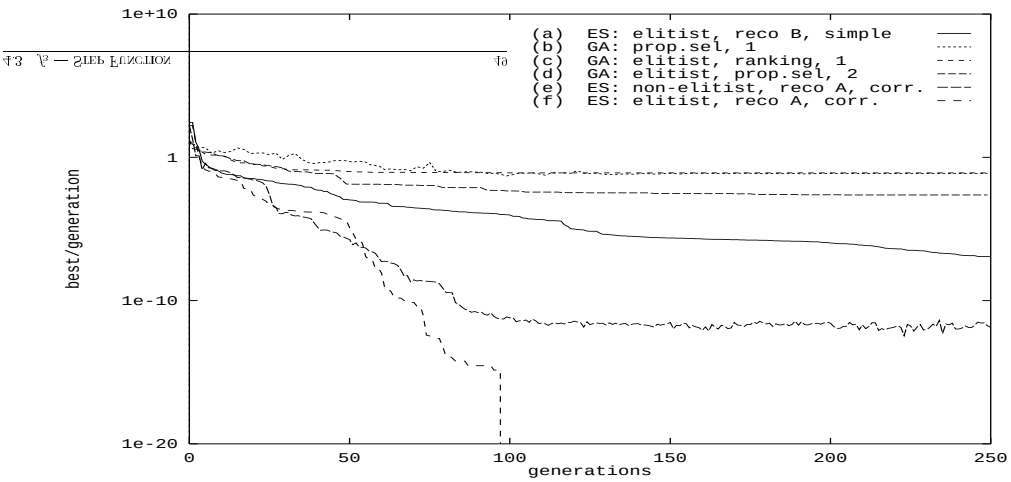
The Sphere Model is a continuous, strictly convex, unimodal function, which comprises the simplest distractive function. Due to its simplicity it is often used as a test case as it was also done in the first experiments on CMA and ES's in parameter optimization cp. [Jon12] and [Sch77].

f1 — Sphere Model

x(3)=0


(a) ES: elitist, reco B, simple
(b) GA: elitist, prop.sel, 1
(c) GA: elitist, prop.sel, 2
(d) ES: elitist, reco E, corr.
(e) ES: non-elitist, reco C, corr.
best/generation
generations
1000
100
10
1
.1
0.01
0.001
0.0001
1e-05
1e-06
0
50
100
150
200
250

bottom, since most optimum seeking methods directly locate the valley and search its floor at the beginning of the search. That is, due to a relatively small change in a variable early in the search, a large change in the objective function will usually result. The progress of many algorithms is very slow and the search direction may change permanently to reach the optimum.


1e+10
1
1e-10
1e-20
0
50
100
150
200
250
best/generation
generations
(a) ES: elitist, reco B, simple
(b) GA: prop.sel, 1
(c) GA: elitist, ranking, 1
(d) GA: elitist, prop.sel, 2
(e) ES: non-elitist, reco A, corr.
(f) ES: elitist, reco A, corr.

f3_Step_Function
generations
x(3)=0 x(4)=0 x(5)=0
f(x)
x(2)
x(1)

on the type of crossover used. multi-objective scheme is supplied for GCE. Although the

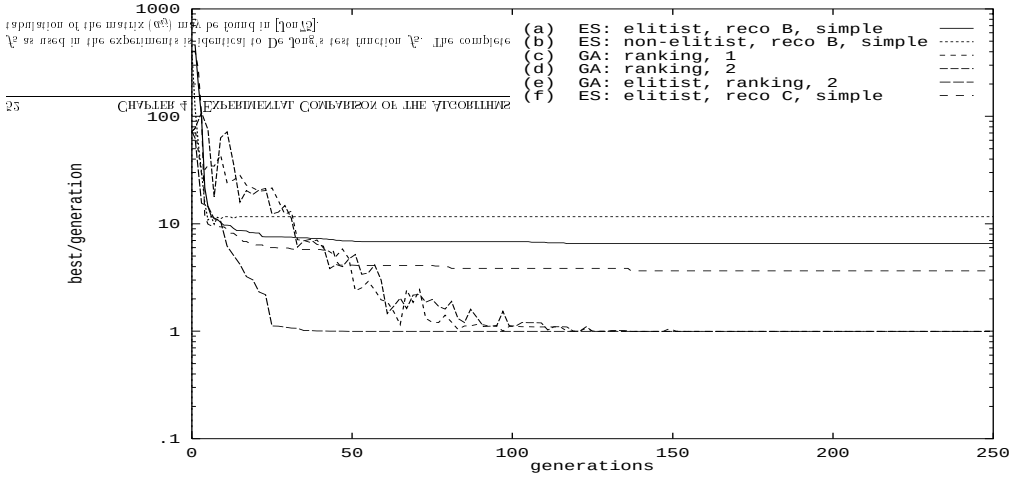
non-elitist EP-variants like one would simply multations and elapsed. instantaneously incompli-

nation (d) performs poor. It has a slightly better performance than the GA's. The poor

elitist EP-variant with corresponding $\eta(\epsilon)$ is the very-ill poor algorithm on Ω pur

it does not find the global optimum multi variants.

4.4 Ω — Shekel's Foxholes



(a) ES: elitist, reco B, simple
(b) ES: non-elitist, reco B, simple
(c) GA: ranking, 1
(d) GA: ranking, 2
(e) GA: elitist, ranking, 2
(f) ES: elitist, reco C, simple
best/generation
generations
1000
100
10
1
.1
0
50
100
150
200
250

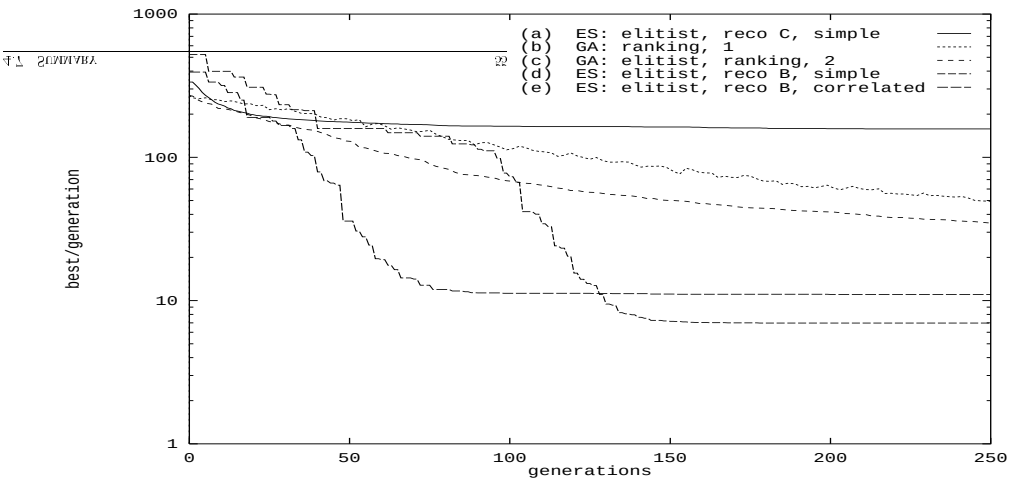
f6 — Schwefel's Problem 1.2
best/generation
generations
10
100
1000
0
50
100
150
200
250

f(x)
x(1)
x(2)
21125
16900
12675
8450
4225
0
-4225
-8450
-12675
-16900
-21125
-65
-52
-39
-26
-13
0
13
26
39
52
65

Although λ_2 is a multimodal multimodal non-elitist EA-variant, because of the good elitist ES-variant (1+1) with the ratio of 1/4 success rule, in very few generations.

4.6 λ_2 — Generalized Rastrigin's Function

λ_2 computes a scalable, continuous, multimodal real function which is made from λ_1 by multiplying it with $\cos(\pi x_i)$. It was first proposed by Rastrigin as a 2-dimensional


1000
100
10
1
0
50
100
150
200
250
best/generation
generations
(a) ES: elitist, reco C, simple
(b) GA: ranking, 1
(c) GA: elitist, ranking, 2
(d) ES: elitist, reco B, simple
(e) ES: elitist, reco B, correlated

used. In general, a GA has no notion about test-variable dependencies (Li). Instead, it simply operates on bit strings and there is no feed-back from evaluated offspring to the integral strategy parameters which influence the search of the next generations. Furthermore, selection are applied in GAs to be mirror image to that in (02,8)−ES. It should be noted, that the properties of (µ,λ)−selection used in the (02,8)−ES. If σ is small enough to exploit the search space until then ES's. Thus the σ is usually to find the global optimum or find it only by chance. some GA-variants found parsed towards global search. While on multimodal functions like λ and $\frac{\lambda}{\mu}$ the (02,8)−

Table 4.3: Hardware/software for efficient computation of multimodal and multifocal topologies

multimodal objective function	multimodal objective function
convergence criterion	convergence criterion
"flat" region selection	"low" region selection
small gradient strength	large gradient strength
best attempt	worse attempt
exhaustive search	exhaustive search

[22]

(1) *Dynamic versus static selection:* The selection probabilities can depend on the actual fitness values (proportional selection). Hence, they change permanently during the simulation, or they can depend on the rank of the fitness values only (linear ranking, (μ,λ)-selection) which remains invariant. This can easily be formalized as follows:

DEFINITION 5 (Dynamic selection)

A selection scheme is called dynamic:

$$\Longleftrightarrow \exists i \in \{1,\ldots,\gamma\}\ \forall t \geq 0 : \mathrm{p}_i(\mathrm{q}_i^t) = c_i \text{ where the } c_i \text{ are constants.}$$

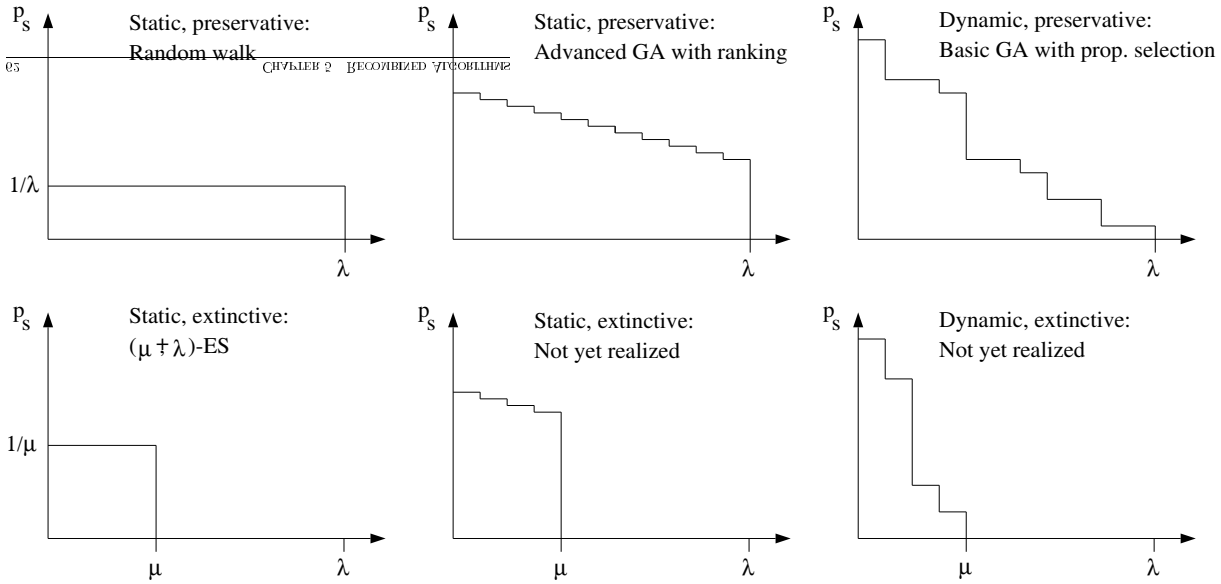
DEFINITION 7 (Recall: Extinctive selection)

A selection scheme is called weakly extinctive:

$$\forall \lambda \geq 0 \, \forall \alpha = (\alpha_1^I, \ldots, \alpha_\lambda^I) \, \exists \lambda \in \{s, \ldots, \lambda\} : \hat{\lambda} \leq \lambda \implies \mathrm{h}^s(\alpha_{\hat{\lambda}}^x) = 0 \iff$$

Of course, in any concrete condition $\sum_{\hat{\lambda}=1}^{\lambda} \mathrm{h}^s(\alpha_{\hat{\lambda}}^x) = 1$ must be satisfied. Apart from slightly assignment of reproduction rates there are other characteristics of selection:

(i) Anxiety versus brave selection: Normally, bacteria are allowed to reproduce in one generation only. Then, they die and the reproduced by some offspring. A selection


p_s
Static, preservative:
Random walk
1/λ
λ
p_s
Static, preservative:
Advanced GA with ranking
λ
p_s
Dynamic, preservative:
Basic GA with prop. selection
λ
p_s
Static, extinctive:
(μ ∓ λ)-ES
1/μ
μ
λ
p_s
Static, extinctive:
Not yet realized
μ
λ
p_s
Dynamic, extinctive:
Not yet realized
μ
λ

evolvable and adaptive performance in case of a multimodal objective function that has local peaks and valleys. Additionally, it could be of interest to integrate a static extinction CA-variant (section 2.1.1) or even progressive ES-variants (section 2.1.5).

Plan classifying a selection scheme as "plus" or "comma" and "fixed" or "variable" selection (parameters) are important, but also the actual fitness distribution within the population — the selection situation — has a major impact on the resulting effects of selection.

The above discussion indicates, that the selective pressure towards the pattern individuals is not only controlled by the selfness of the applied selection mechanism, but is also influenced by physical selection situation for a given problem.

From the reflections discussed in the previous section the idea emerges to transfer selection mechanisms from ES's to GA's and vice versa and to investigate the existing variants of proportional selection and tandem, respectively. Thus, the following definitions for (μ,λ)-proportional selection and (μ,λ)-tandem variants can be formulated very naturally: A selection probability $p_s(a_i,Q_t)$ is assumed that individuals are sorted by fitness and that the index i of an individual a_i^t is identical to its rank. Hence, we identify them and write i instead of $\Xi(a_i^t)$.

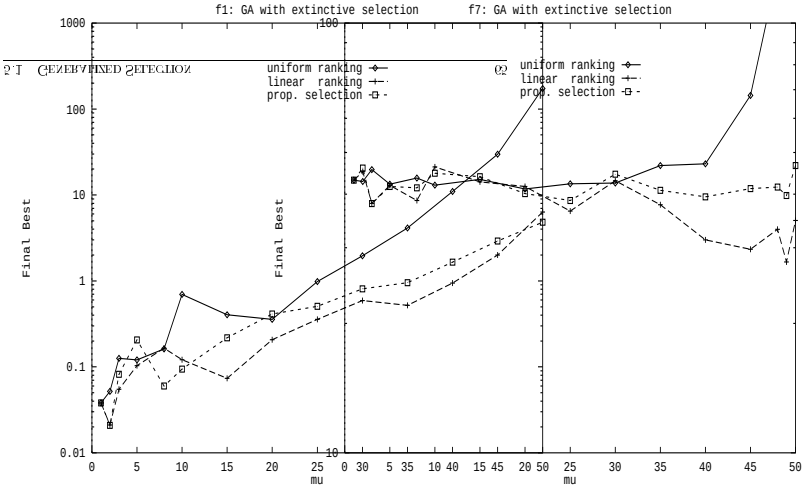

f1: GA with extinctive selection
f7: GA with extinctive selection
uniform ranking
linear ranking
prop. selection
Final Best
mu
1000
100
10
1
0.1
0.01
0
5
10
15
20
25
30
35
40
45
50

Figure 2.19: Progress rates for $(10,50)$- and $(50,50)$-proportional selection on the sphere model for $\varphi = 0$ and $\varphi = 1.0$.

proportional selection can be noticed for small fitness differences. The behaviour is similar to truncation selection with a small truncation parameter. The progress rate is maximized for $\varphi = 0$ and $\varphi = 1.0$ for N^{max}, and $(1,\lambda)$-ES is globally superior to multi-recombination ES for $N^{\text{max}} \in [1.5]$. However, as before all excellent progress rates origin from a small region of σ values, and multi-recombination ES is preferable if external parameters are disregarded. Figure 2.19 shows the progress rates for $(10,50)$- and $(50,50)$-proportional selection on the sphere model for different values of

2.1.3 Generalized Selection in ES^2

From all presented selection schemes in section 2.1 only (λ,λ)-selection and its elitist variant $(\mu+\lambda)$-selection, as well, base on the principle of *proportional selection*. In contrast to the (μ,λ)-linear ranking and (μ,λ)-tournament selection, as well, the probability of selecting an individual is directly proportional to its fitness. Therefore, we call (λ,λ)- and $(\mu+\lambda)$-selection *proportional selection schemes*. This fact leads to the idea to investigate the following *generalized proportional selection scheme*:

All λ individuals have the same probability $\frac{1}{\lambda}$ to be selected as parents.

The proportional selection schemes in section 2.1 only (λ,λ)-selection and its elitist variant $(\mu+\lambda)$-selection base on the principle of *proportional selection*. In contrast to the linear ranking and tournament selection, as well, the probability of selecting an individual is directly proportional to its fitness. Therefore, we call (λ,λ)- and $(\mu+\lambda)$-selection *proportional selection schemes*. This fact leads to the idea to investigate the following *generalized proportional selection scheme*:

All λ individuals have the same probability $\frac{1}{\lambda}$ to be selected as parents.

selection in scheme in ES2: the best final value, i.e. maximum progress is achieved for a $(10,20)-ES$, which needs exactly the results from theory [Sch77, Sch81]. Furthermore there seems to be a lower bound on the number of parents with regard to self-adaptation that is a superlinear loss of performance, which results from a non-optimal setting of the strategy parameters, i.e. the mutation rates of the object variables. For $\lambda < 10$ the strategy parameters cannot adapt sufficiently and the performance drops also enormously. For multimembered ES ($\mu > 1$) the mutation rates of the object variables are averaged. This leads to better progress, because information about successful search steps is passed to the next generation. Thus, the strategy parameters can be changed in a more directed way. Recombination of object variables leads to genetic repair effects [Rec94]. Multiparent recombination, i.e. $\rho > 2$, is essential for self-adaptation of mutation rates [Sch77, Sch81]. Multirecombination with intermediate recombination of object parameters and geometric recombination of strategy parameters is the best choice [Rec94]. Here, $\rho = \mu$ is used. Mutation of the strategy parameters is performed by adding a normally distributed random value with expectation zero and a given standard deviation. Usually, one first changes the step sizes and then uses the changed values to modify the object parameters. This ensures the production of offspring even if the mutation strength has become very small. To introduce a bias towards small mutation rates, which are necessary near the optimum, all mutations are carried out with the reciprocal value of the standard deviation first. Then, the resulting individuals undergo selection. Two different selection methods are available: plus and comma selection. Plus selection ($\mu + \lambda$) means that μ parents create λ offspring by recombination and mutation and the best μ individuals out of both parents and offspring survive. Comma selection (μ, λ) means that μ parents create λ offspring ($\lambda > \mu$) by recombination and mutation and the best μ individuals out of the offspring survive. Normally, comma selection is preferred to plus selection, because it is able to leave local optima and it can adapt to changing environments. Plus selection preserves good solutions over generations paying the price of stagnation due to outdated strategy parameters. Comma selection with $\lambda \approx 7\mu$ is recommended [Sch81].

log final optimum
mu
uniform ranking
linear ranking
prop. selection
log final optimum
mu
uniform ranking
linear ranking
prop. selection

variants of the selection schemes as performed. Three results are presented in the right box of figure 2.6, which firmly confirms the perfect performance of "for" selection it the result is not dampened by lost information. Again some degree of exhaustiveness is tedmitted for maximum performance with all selection schemes.

The very first results with proportional selection and linear ranking in E_{22} show a minor superiority of poll combined to minor ranking in case of minimodal optimization. within the initial selection in the minimodal case ($\mathcal{O}(k)$-minor ranking applies to pe superiority. Furthermore, the classification of proportional selection to pe "market" dan linear ranking which again is "market" dan minor ranking is multiplied by the presented result.

[0.0.0.0.0.0] [2GED20]. The result of Schaffer et al. has also been formulated as the empirical expression

$$\mu_m \approx \frac{1.75}{N\sqrt{\chi}} \tag{2.5}$$

where χ denotes the population size and N is the length of an individuals' genotype. Hesser and Männer [0.0.0.0.0.0] [2GED20] derived a theoretically optimal dynamic mutation rate schedule

$$\mu_m(\lambda) = \sqrt{\frac{\alpha}{\lambda}} \cdot \frac{\left(\frac{k}{2}\right)^{\frac{k}{2}} \cdot e^{-\frac{k}{2}}}{\Gamma\left(\frac{k}{2}+1\right)} \tag{2.6}$$

genotype, which is now described as:

$$\begin{aligned} I' &= \left(I \times \left(\{0,1\}^{l_m}\right)^n\right) \quad \text{or} \\ I' &= \left(I \times \{0,1\}^{l_m}\right) \end{aligned} \tag{5.5}$$

Here, l_m denotes the length of the representation of a mutation rate. For $l = n \cdot l_\pi$ an individual $a \in I'$ can be denoted as follows:

$$a = (\alpha_1 \cdots \alpha_{n_1} \cdots \alpha_{n_2} \cdots \alpha_{n_{n_2}} \cdots \beta_{1_1} \cdots \beta_{1_{n_1}} \cdots \beta_{n_1} \cdots \beta_{n_{n_1}}) \tag{5.6}$$

Figure 2.5: Schematic working mechanism of adaptive mutation

1.) Mutation of mutation rates

2.) Mutation of object variables

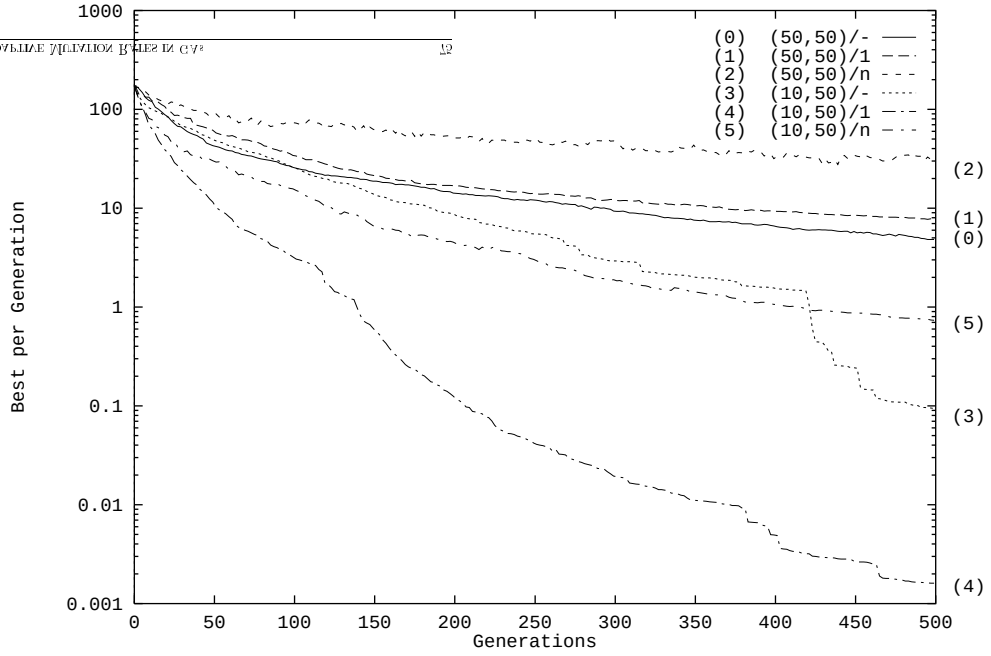
Hence, $\Lambda_{a\to(a,a)}^{-}(w)$ increases monotonically over time. Since $\sum_{i=0}^{k-1}\Lambda_{a\to\mathrm{pred}(i)}^{-}(w)=1$ and $\Lambda_{a\to(a,a)}^{-}(w)\geq 1$ for

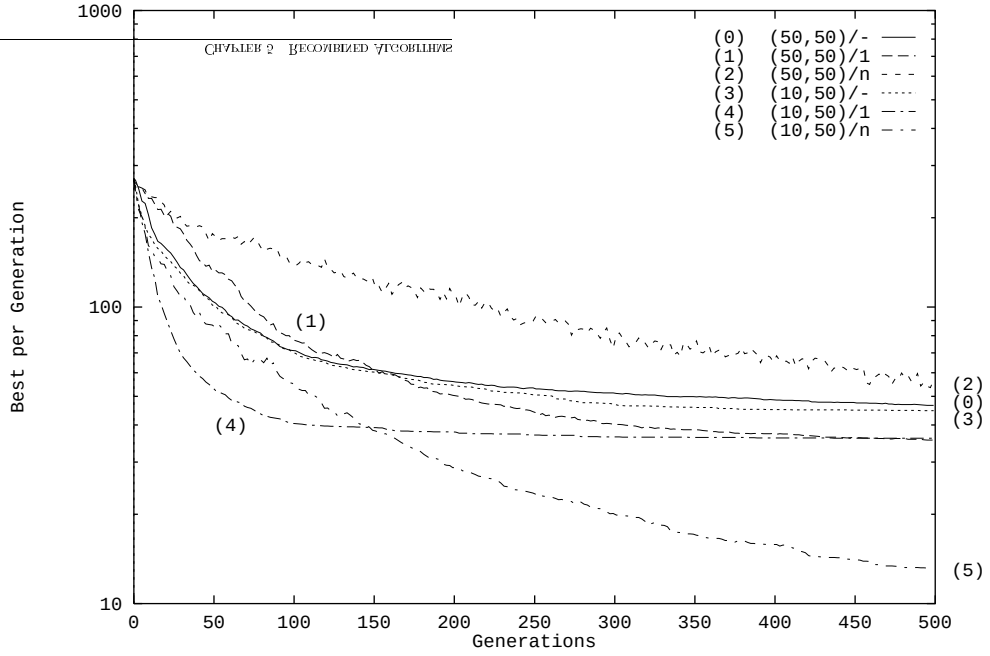
$$0<\sum_{i=1}^{k-1}\Lambda_{a\to\mathrm{pred}(i)}^{-}(w-1)\,\Lambda_{\mathrm{pred}(i)\to(a,a)}^{-}(1)$$

so

$$\Lambda_{a\to(a,a)}^{-}(w-1)=1$$

until the initial $\Lambda_{\mathrm{pred}(i)\to(a,a)}^{-}(1)=\Lambda_{\mathrm{pred}(i)\to(a,a)}^{-}(0)\neq 0$ for $i\neq 0$. Since only finitely many $\Lambda_{a\to(a,a)}^{-}(0)$ are initialized, $\Lambda_{a\to(a,a)}^{-}(w)<\Lambda_{a\to(a,a)}^{-}(w-1)$ for all w after a finite number of steps. Therefore, $\Lambda_{a\to(a,a)}^{-}(w)$ converges to 1 as $w\to\infty$.


(0) (50,50)/-
(1) (50,50)/1
(2) (50,50)/n
(3) (10,50)/-
(4) (10,50)/1
(5) (10,50)/n
Best per Generation
Generations
1000
100
10
1
0.1
0.01
0.001
0
50
100
150
200
250
300
350
400
450
500
(2)
(1)
(0)
(5)
(3)
(4)


1000
100
10
0
50
100
150
200
250
300
350
400
450
500
Best per Generation
Generations
(0) (50,50)/-
(1) (50,50)/1
(2) (50,50)/n
(3) (10,50)/-
(4) (10,50)/1
(5) (10,50)/n
(1)
(4)
(2)
(0)
(3)
(5)

Concerning the A.A this psychopathic is a daring one. Holland has exhibited the traits of murderation as a "psychopathic obsession", people is a medicalmism of intimidance that is also related, but not as a secret operator ([Holland, pp. 111–01]). On the other hand, the traits bolow the threshold are more difficult to diagnose, because many of the traits below the threshold are also found in people who are not psychopathic. Although the investigations of Eigen [Eigen] do not provide information into account, they the theoretically well-founded and provide a more reliable instrument for diagnosing the topic of murderation in the field of psychopathic people, as well as the development of new diagnostic methods.

subsequent sections. There is no further distinguishment between them for the discussion of various parallel implementations.

6.1 Scalable Parallelism

In the context of MIMD computers distributed or parallel programs are often measured with respect to the degree of parallelism they utilize. In general, two major cases are distinguished: *coarse-grain* and *fine-grain* parallelism. In the former case a parallel pro-

(1) non-Venemian computers (SISD)
(5) vector processors (SIMD)
(3) LAN-based multi-computers (MIMD)
(4) coarse-grain battled computers (MIMD)
(2) fine-grain battled computers (MIMD)

control of an architecture. Experimentally, Algorithm 1 on an optimization method by Zedek and Nash [Nash82]. Unfortunately, Schwegler proved that the population of such a $(\mu+1)-ES$ cannot be both irreducible and p-ary (multiplication takes o to the topology of the optimization function [Schi78]. Under these conditions, stability of stationary batteries is the "mean" size as parameterized by the well-known phenomenon of two-stable oscillation.

In the experiment case architectures Evolutionary Algorithms show no major advantage, but rather restrictions selection schemes are a major protection to key bit of a centralized control in battalion DIAL implementations which limits the amount of efficiently adapted battalion.

M
S_1
S_2
S_n
evaluation
offspring generation
idle time
communication time

total computation time. Only if the CPU-time work-weight is the additional time required for communication between the master and a slave a decrease of the total computation time is achievable.

Suppose f denotes the total computational performance of a possibly parallel program consisting of slave tasks λ_{sw} and λ_{ms} total to the total time required for the master. λ_{sw} and λ_{ms} total to the total time required for distributed by the sequential and parallel part. Then the minimum time $\lambda(n)$ to compute the optimize function on the slave side must total to the time for a single communication λ_c as follows:

(5) Time-train parallelism by assigning a set of individuals, each of which interacts only with some of the other individuals in its vicinity.

Both approaches promise to increase the throughput dramatically, because the selection operation is applied only locally, i.e. selection is "often" than in conventional GAs and ESs and it requires communication. One might initiate biological hypermutation like simulation or speciation very much artificially by deleting a limit of genetic distances permitting individuals within a sub-population.

Such modified strategies do not directly compete to the conventional, i.e. sequential algorithm but they are very successful in many applications.

6.2 VALID CONVERTERS

sub-population (deme)
migration path

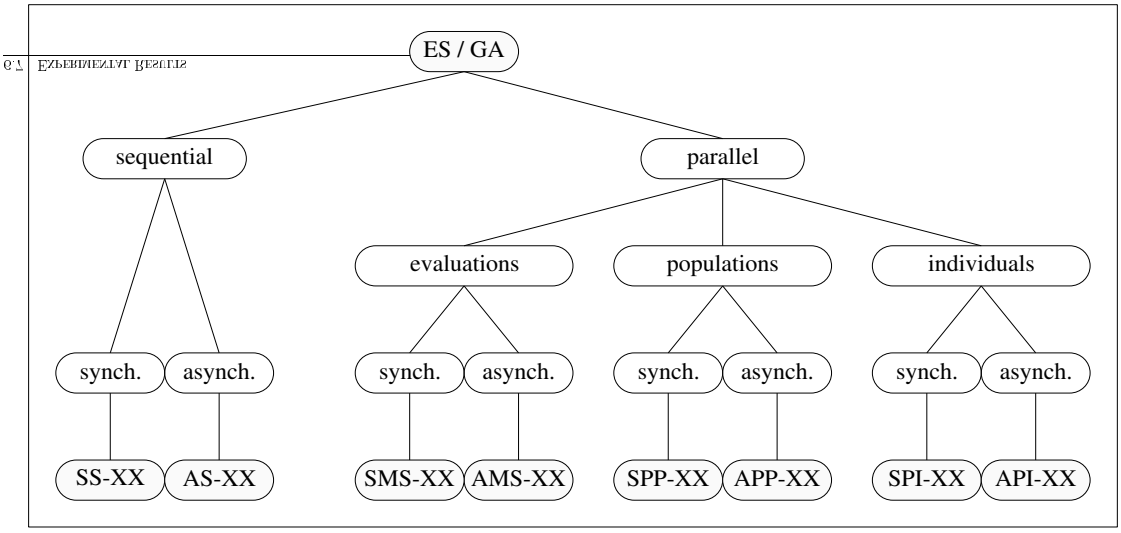
neighborhood of central element

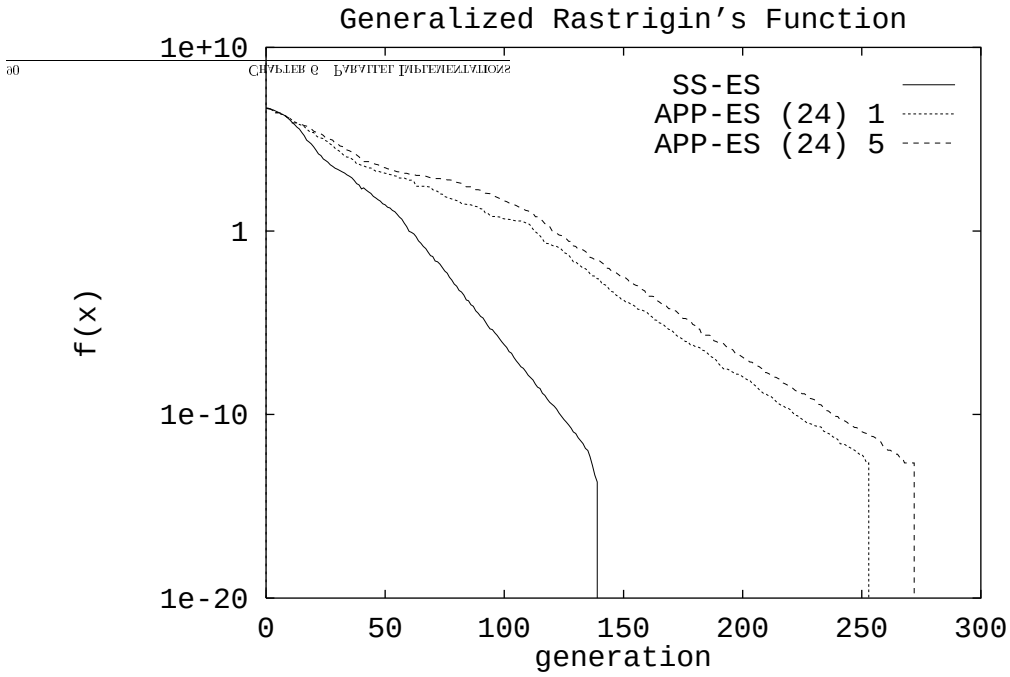
(5) limited, local interactions of the process

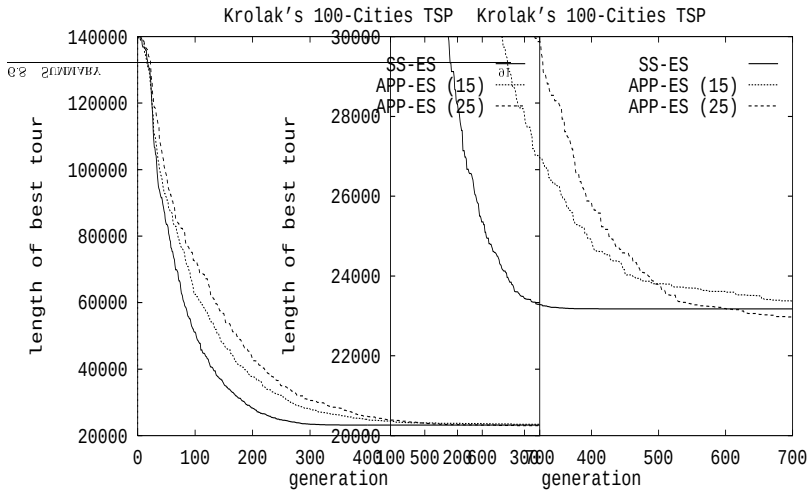
(3) independence of the process

(4) self-organization

The size of the neighborhood and the diameter of the environment (integer number of neighborhoods between individuals) has a major impact as it influences the amount of reproduction in the population. In general, an optimum between exploration and exploitation is a compromise between exploration and exploitation. The former is achieved with


ES / GA
sequential
parallel
evaluations
populations
individuals
synch.
asynch.
synch.
asynch.
synch.
asynch.
synch.
asynch.
SS-XX
AS-XX
SMS-XX
AMS-XX
SPP-XX
APP-XX
SPI-XX
API-XX


Generalized Rastrigin's Function
SS-ES
APP-ES (24) 1
APP-ES (24) 5
1e+10
1
1e-10
1e-20
f(x)
0
50
100
150
200
250
300
generation


Krolak's 100-Cities TSP
length of best tour
generation
SS-ES
APP-ES (15)
APP-ES (25)
Krolak's 100-Cities TSP
length of best tour
generation
SS-ES
APP-ES (15)
APP-ES (25)

Patient CA to ES." 20 Jan. 1979, pp. 61–62. JAMA, doi:10.1001/jama.1979.03290030061029.

ES. "Very often such improvements also lead to a better understanding of cytoplasmic processes.

Bibliography

[DG89] Kalyanmoy Deb and David E. Goldberg. An investigation of niche and species formation in genetic function optimization. In Schaffer [Sch89], pages 42–50.

[EAH91] A. E. Eiben, E. H. L. Aarts, and K. M. Van Hee. Global convergence of genetic algorithms: an infinite markov chain analysis. In Schwefel and Männer [SM91], pages 4–12.

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

[Eig76] Manfred Eigen. Wie entsteht Information? Prinzipien der Selbstorganisation in der

[CS90] Martina Gorges-Schleuter. *Genetic Algorithms and Population Structures — A Massively Parallel Algorithm*. PhD thesis, University of Dortmund, August 1990.

[Har74] Dietrich Hartmann. *Optimierung balkenartiger Zylinderschalen aus Stahlbeton mit elastischem und plastischem Werkstoffverhalten*. PhD thesis, University of Dortmund, July 1974.

[Har90] Richard F. Hartl. A global convergence proof for a class of genetic algorithms. Technische Universität Wien, 1990.

[HB90] Frank Hoffmeister and Thomas Bäck. Genetic algorithms and evolution strategies: Similarities and differences. Technical Report "Ceres Reihe", No. 365, Department

[Kur91] Frank Kursawe. A variant of Evolution Strategies for vector optimization. In Schwefel and Männer [SM91], pages 193–197.

[Loh91] Reinhard Lohmann. Application of evolution strategy in parallel populations. In Schwefel and Männer [SM91], pages 198–208.

[NM65] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

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

[Sch81] Hans-Paul Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, 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.

[Sch88] Hans-Paul Schwefel. Evolutionary learning optimum-seeking on parallel computer architectures. In A. Sydow, S. G. Tzafestas, and R. Vichnevetsky, editors, *Proceedings of the International Symposium on Systems Analysis and Simulation 1988, I: Theory and Foundations*, pages 217–225. Akademie der Wissenschaften der DDR,