The Interaction of Mutation Rate, Selection, and Self-Adaptation Within a Genetic Algorithm

Thomas Bäck[a]

[a]University of Dortmund, Department of Computer Science, P.O. Box 50 05 00, 4600 Dortmund 50, Germany

Abstract

A simplified Genetic Algorithm, essentially a mutation-selection scheme, is analyzed theoretically with respect to convergence rates and success probabilities for mutations on a simple bit-counting objective function. The theoretical results are then confirmed by experimental runs and provide a basis for the explanation of the self-adaptation mechanism of mutation rates, which turns out to enable a near-optimal schedule for the mutation rate by means of a self-organizing process. This way, a first step towards a theoretical foundation for the usefulness of self-adaptation within Genetic Algorithms is presented.

1. INTRODUCTION

For *Genetic Algorithms* (GAs) [1], a large number of experimental investigations towards discovering optimal settings of the exogenous parameters, mainly the *mutation rate* p_m, the *crossover probability* p_c, and the *population size* λ, are reported in literature. Some resulting settings are $p_m = 0.001$, $p_c = 0.6$, $50 \leq \lambda \leq 100$ [2], $p_m \in [0.005, 0.01]$, $p_c \in [0.75, 0.95]$, $20 \leq \lambda \leq 30$ [3], $p_m = 0.01$, $p_c = 0.95$, $\lambda = 30$ [4]. The settings of Grefenstette were obtained by using a meta-level GA which optimizes the parameter settings of a GA. These investigations, which serve as starting points to most applications of GAs, are all based on the idea to find a setting which maximizes the performance over a set of test functions of very different characteristics and is constant for all these different functions. Although this is in accordance with Holland's definition of *robust* adaptive plans (see [1], page 27), a recent result of Hart and Belew shows that no single nondeterministic algorithm exists which is able to approach the optimal function values of arbitrary functions $f : \{0,1\}^l \rightarrow \mathbb{Z}$ to a certain accuracy in an efficient way, i.e. in a time polynomial w.r.t. l [5]. This result is extremely important since it essentially means that given an Evolutionary Algorithm the most efficient parameter setting must depend on the fitness function, at least. It is quite evident that strategy parameters should vary during a run for finding an optimum in order to achieve even higher efficiencies.

In the following, ideas from *Evolution Strategies* (ESs) [6, 7] on the *self-adaptation* of strategy parameters are combined with a simple GA in order to find a way for the theoretical investigation of a hybrid algorithm. Furthermore, some new arguments concerning the importance of mutation in genetic search are presented. In section 2 a theoretical argument for introducing a time-dependent schedule of the mutation probability is presented for a simple objective function. Based upon this, in section 3 the convergence velocity is calculated for a mutation-selection mechanism, and the result is confirmed by experiments. These results are used in section 4 to demonstrate that the self-adaptation mechanism for mutation rates in GAs (first introduced to GAs in [8]) provides a self-organizing near-optimal schedule for mutation rates.

2. SEARCH BY MUTATION

Holland has introduced mutation to Genetic Algorithms as a "background operator" ([1], page 111), which assures the principal possibility to recover from lost alleles, i.e. alleles which are converged within the population. In contrast to ESs, which use mutation as the main search operator, mutations in GAs are not mainly intended to serve as a mechanism for generating new structures. Furthermore, GAs have in most cases been analyzed in terms of *schemata* (hyperplanes in l-dimensional bitspace) and the disruptive effect of mutation and crossover to schemata, an approach which is absolutely independent of any particular objective function.

In contrast to schema analysis approaches this work analyzes a particular objective function defined on bitstrings which is analogous to the corridor model in ESs. Additionally, in the following mutation is used as the only search operator in a GA, and the efficiency of this search by a mutation-selection algorithm is demonstrated. We focus on the simple bit-counting function

$$f : \{0,1\}^l \to \{0,\ldots,l\}\ ,\ f(\alpha_1,\ldots,\alpha_l) = \sum_{i=1}^{l} \alpha_i \tag{1}$$

which is unimodal and an easy problem for GAs. For this function, it is possible to calculate the *success probability* for mutations, i.e. the probability that mutation of an individual increases the fitness of the individual. More formally, let $\omega_m : \{0,1\}^l \to \{0,1\}^l$ denote the mutation operator and $a \in \{0,1\}^l$ an individual. Then, we define the following probabilities:

$$p_a^+ = \mathcal{P}(f(\omega_m(a)) > f(a)) \quad \text{success probability} \tag{2}$$

$$p_a^- = \mathcal{P}(f(\omega_m(a)) < f(a)) \quad \text{failure probability} \tag{3}$$

$$p_a^0 = \mathcal{P}(f(\omega_m(a)) = f(a)) \quad \text{stagnation probability} \tag{4}$$

These are the probabilities of improvement, worsening, or constancy, respectively, of the individuals' fitness subject to mutation. For the bit-counting function (1), these probabilities are given in theorem 1.

Theorem 1
Let $a \in \{0,1\}^l$ be a bitstring of length l, p the bit-mutation probability, and $f_a := f(a)$ denotes the number of ones within a, i.e. its function value under objective function (1). Then:

$$p_a^+ = \sum_{i=0}^{f_a} \binom{f_a}{i} p^i (1-p)^{f_a - i} \sum_{j=i+1}^{l-f_a} \binom{l-f_a}{j} p^j (1-p)^{l-f_a-j} \tag{5}$$

$$p_a^- = \sum_{i=0}^{l-f_a} \binom{l-f_a}{i} p^i (1-p)^{l-f_a-i} \sum_{j=i+1}^{f_a} \binom{f_a}{j} p^j (1-p)^{f_a-j} \tag{6}$$

$$p_a^0 = \sum_{i=0}^{f_a} \binom{f_a}{i} \binom{l-f_a}{i} p^{2i} (1-p)^{l-2i} \tag{7}$$

Proof: To give a proof for the expression for p_a^+, two cases can be distinguished:

(1) None of the f_a one-bits in a mutates. Then, at least one zero-bit up to all $l - f_a$ zero-bits must undergo mutation in order to increase the number of ones, i.e. we have between $i = 1$ and $i = l - f_a$ mutations of zero-bits. The choice, which of the bits is mutated, is arbitrary, such that there exist $\binom{l-f_a}{i}$ choices to mutate i bits (probability p^i) and to leave $l-i$ bits unmutated (probability $(1-p)^{l-i}$). Combining this, the first case results in a success probability

$$p_1^+ = \sum_{i=1}^{l-f_a} \binom{l-f_a}{i} p^i (1-p)^{l-i} \tag{8}$$

(2) Between $i = 1$ and $i = f_a$ ones are mutated to zero-bits. Then, in order to increase the number of ones in spite of this initial loss, at least $j = i + 1$ up to $j = l - f_a$ zero-bits must be mutated to ones. From the first case of this proof the expression for the mutation of the zero-bits can be taken, which has to be multiplied by the probability of mutating i ones, summing up over all values of i:

$$p_2^+ = \sum_{i=1}^{f_a} \left[\binom{f_a}{i} p^i (1-p)^{f_a-i} \cdot \left(\sum_{j=i+1}^{l-f_a} \binom{l-f_a}{j} p^j (1-p)^{l-f_a-j} \right) \right] \tag{9}$$

Summing up the probabilities p_1^+ and p_2^+ and using the fact that p_1^+ is a special case of the outer sums' argument in p_2^+ (for $i = 0$), one obtains $p_a^+ = p_1^+ + p_2^+$. By an analogous calculation, the expression for p_a^- is derived. Furthermore, stagnation can be incorporated into both expressions for p_a^+ and p_a^- by starting the second sum at $j = i$ instead $j = i + 1$, meaning that at least the same number of zeroes and ones must be mutated. Then, by taking only the summand for $j = i$ in the inner sum, the expression for p_a^0 results. Of course, the identity $p_a^+ + p_a^- + p_a^0 = 1$ holds for these probabilities. However, the proof of this identity involves a longish calculation and is therefore omitted here. Q.E.D.

From this theorem, expressions can easily be derived for the probability $p_a^+(k) = \mathcal{P}(f(\omega_m(a)) = f(a) + k)$ for improving the bitstring a by adding exactly k ones ($0 \leq k \leq l - f(a)$) and $p_a^-(k) = \mathcal{P}(f(\omega_m(a)) = f(a) - k)$ for worsening the bitstring a by subtracting exactly k ones ($0 \leq k \leq f(a)$). The expressions result from taking into account only the summand for $j = i + k$ in the inner sum of the expressions presented in theorem 1. It should explicitly be noted that the expressions for $p_a^+(k)$ and $p_a^-(k)$ now include the stagnation probability, i.e. $p_a^+(0) = p_a^-(0) = p_a^0$:

Corollary 1

$$0 \leq k \leq l - f_a \quad : \quad p_a^+(k) = \sum_{i=0}^{f_a} \binom{f_a}{i} \binom{l-f_a}{i+k} p^{2i+k}(1-p)^{l-2i-k} \tag{10}$$

$$0 \leq k \leq f_a \quad : \quad p_a^-(k) = \sum_{i=0}^{l-f_a} \binom{l-f_a}{i} \binom{f_a}{i+k} p^{2i+k}(1-p)^{l-2i-k} \tag{11}$$

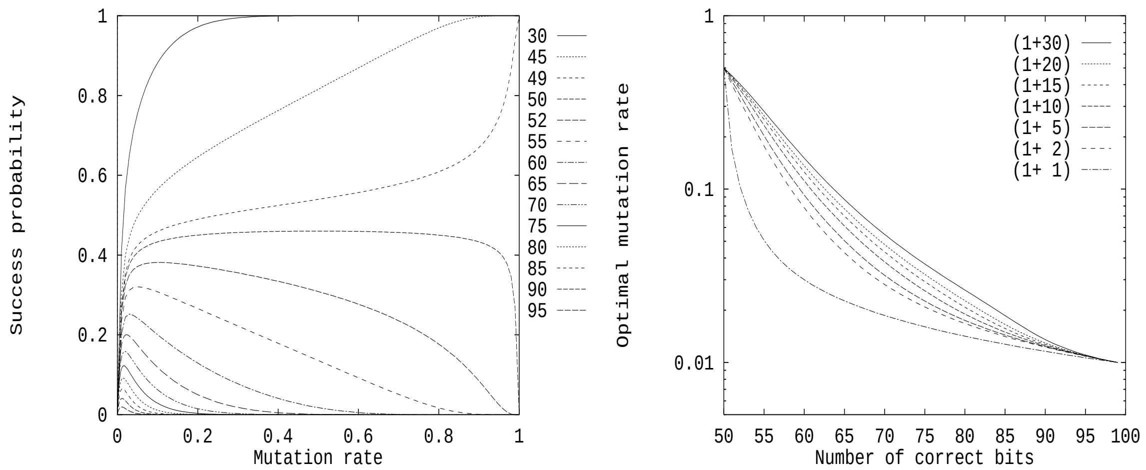

Figure 1: Success probabilities with $l = 100$, $f_a \in \{30, \ldots, 95\}$ (left) and optimal mutation rate for different (1+λ)-settings, $\lambda \in \{1, 2, 5, 10, 15, 20, 30\}$ (right).

It is extremely instructive to plot, given the total string length l as well as the number of bits f_a which are already correct in the bitstring a, the dependence of the success probability p_a^+ on the mutation rate p. For $l = 100$, the resulting curves are shown in the left part of figure 1 for different values of correct bits in the string (i.e. the curves are plotted for $f_a \in \{30, 45, 49, 50, 52, 55, 60, 65, 70, 75, 80, 85, 90, 95\}$ and $l = 100$). In both figures the legend describes the plots in the order from top to bottom. Clearly, when less than half of the bits in a string are correct, the optimal mutation probability is one (inverting all bits guarantees improvement). When half of the bits are correct, the optimal probability is 0.5 and decreases quickly for even small increases of the number of correct bits.

Although it is easy to calculate the expression for the derivative of p_a^+ with respect to p, the resulting equation when setting $\frac{dp_a^+}{dp} = 0$ can not be solved for p in a closed analytical form. Calculating the zeroes by means of numerical methods yields the lowest curve plotted in the right part of figure 1, where the dependence of the optimal mutation rate on the number of correct bits is shown. The remaining curves will be discussed and explained in section 3. The curve gives a clear demonstration of the strong dependence of the optimal mutation rate setting on the current Hamming distance to the optimal solution. In other words, the mutation rate should not be constant but should decrease over time during the search.

3. SELECTION AND PROGRESS RATES

So far, nothing but a single individual subject to mutation was taken into account. In order to combine the result on mutation rates with the concepts of a population and of selection, we focus here on the $(\mu+\lambda)$-selection and the (μ,λ)-selection. In the latter case, the μ best individuals out of the λ offspring are selected to become parents of the next generation, while in the former case the μ best individuals are selected from the set of μ parents *and* λ offspring individuals. This way, the $(\mu+\lambda)$-scheme is elitist since it will only accept improvements, a behaviour advantageous for convergence velocity but disadvantageous when convergence reliability and self-adaptation of mutation rates are desired. Both selection mechanisms were originally introduced by Schwefel in ESs [7], later on the (μ,λ)-selection was tested successfully also in the context of GAs [9].

Following the theory developed for ESs by Schwefel [7] and Rechenberg [6], the convergence velocity φ, i.e. the expectation of the improvement per generation, can be derived for a $(1,\lambda)$-GA and a $(1+\lambda)$-GA on objective function f as defined in equation (1). To do so, the probability density function $w_1(k)$ is needed, which describes the case that the best individual among the λ offspring constitute an improvement by k:

$$w_1(k) = \sum_{i=1}^{\lambda} \binom{\lambda}{i} \cdot w_{k'=k}^{i} \cdot w_{k'<k}^{\lambda-i} \tag{12}$$

This expression results from the fact that due to the discrete nature of the objective function any number of individuals between $i = 1$ and $i = \lambda$ covering the distance k can be the best after mutation. The probability for i individuals to cover the distance k at the same time is just $w_{k'=k}^{i}$, the remaining $\lambda - i$ individuals cover a smaller distance with probability $w_{k'<k}^{\lambda-i}$. For each choice of i best individuals out of λ there exist $\binom{\lambda}{i}$ possibilities, and the sum must be calculated for all possible values of i ($1 \le i \le \lambda$). For a $(1,\lambda)$-GA k may take positive as well as negative values, while for a $(1+\lambda)$-GA k may only be positive or zero (only improvements or stagnation are accepted). From the calculations in section 2, the probabilities $w_{k'=k}$, $w_{k'<k}$ and $w_{k'>k}$ (for an improvement by more than k) can be given ($-f_a \le k \le l - f_a$):

$$w_{k'=k} = \begin{cases} p_a^{+}(k) & , \ k \ge 0 \\ p_a^{-}(-k) & , \ k < 0 \end{cases} \tag{13}$$

$$w_{k'>k} = \begin{cases} \displaystyle\sum_{i=k+1}^{l-f_a} p_a^{+}(i) & , \ k \ge 0 \\ \displaystyle\sum_{i=k+1}^{-1} p_a^{-}(-i) + \sum_{i=0}^{l-f_a} p_a^{+}(i) & , \ k < 0 \end{cases} \tag{14}$$

$$w_{k'<k} = 1 - w_{k'=k} - w_{k'>k} \tag{15}$$

Altogether, an expression for φ can be formulated, which is given in theorem 2.

THEOREM 2

Let $w_1(k)$ be the probability density function as defined in (12) and (13). For f as in equation (1), the expectation of the convergence velocity φ for a $(1+\lambda)$-GA or $(1,\lambda)$-GA is

$$\varphi(l, f_a, p, \lambda) = \sum_{k=k_{min}}^{l-f_a} k \cdot w_1(k) = \sum_{k=k_{min}}^{l-f_a} k \cdot \sum_{i=1}^{\lambda} \binom{\lambda}{i} \cdot w_{k'=k}^{i} \cdot w_{k'<k}^{\lambda-i} \tag{16}$$

where $k_{min} = -f_a$ for a (1,λ)-GA and $k_{min} = 1$ for a (1+λ)-GA (strictly speaking, $k_{min} = 0$, but this summand does not contribute anything to the sum in equation (16)). In the following, we will write $\varphi_{(1,\lambda)}$ in the former case and $\varphi_{(1+\lambda)}$ in the latter case.

Proof: The expression for φ is clear by definition of the expectation value and the remarks on the range allowed for k as given before. Q.E.D.

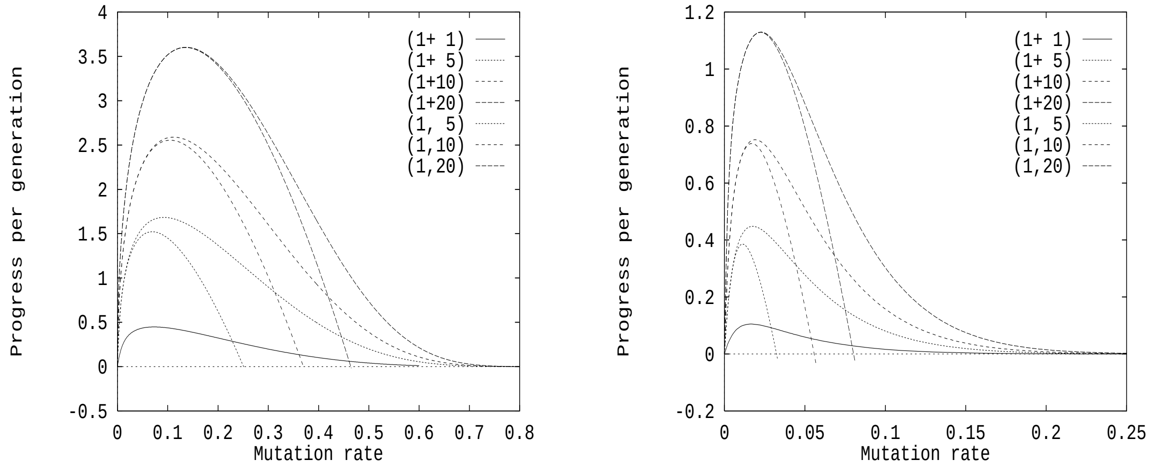

Figure 2: Progress plots for $f_a = 60$ (left) and $f_a = 80$ (right).

The convergence rates are always absolute, local values, i.e. they depend on the current quality f_a of the best individual and they give the improvement w.r.t. the quality function value which is expected for the best individual. In figure 2 the resulting plots of a (1+λ)-GA and a (1,λ)-GA are shown for different values of $\lambda \in \{1, 5, 10, 20\}$, again for strings of length $l = 100$ and assuming actual function values of $f_a = 60$ (left plot) and $f_a = 80$ (right plot).

It is obvious and intuitively clear that, when the number of offspring increases, the expected progress increases for both the (1+λ)-GA and the (1,λ)-GA. While there is no difference between the (1+λ)-selection scheme and the (1,λ)-selection scheme for small mutation rates (and values of λ not too small), progress decreases very rapidly for the (1,λ)-strategies above the optimal mutation rate, becoming even negative as soon as the mutation rate is too large. The (1+λ)-strategy always yields a positive progress approaching zero only from above for growing mutation rate. Furthermore, the progress curves are narrower for larger f_a than for smaller f_a, i.e. the range of mutation rates for which progress is achieved becomes smaller and smaller as the optimum is approached.

Once again, it is interesting to have a look at the dependence of the optimal mutation rate on the number of correct bits, which can be calculated by numerical methods. This is done for a (1+λ)-GA with $\lambda \in \{1, 2, 5, 10, 15, 20, 30\}$ in the right part of figure 1. The plot for the (1+1)-case is of course identical to the result obtained by maximizing p_a^+ in section 2, since $p_a^+ = \sum_{k=1}^{l-f_a} p_a^+(k)$ and $\varphi_{(1+1)} = \sum_{k=1}^{l-f_a} k \cdot p_a^+(k)$. While the values of $p^*(l/2) = 0.5$ and $p^*(l-1) = 1/l$ remain constant when increasing λ, the optimal mutation rate in between these boundaries grows slowlier and slowlier when λ is increased. Most remarkable is the strong increase of p^* as λ changes from 1 to 2, i.e. when the "redundancy" of offspring allows for the risk of producing garbage, which is at the same time a chance of producing larger improvements.

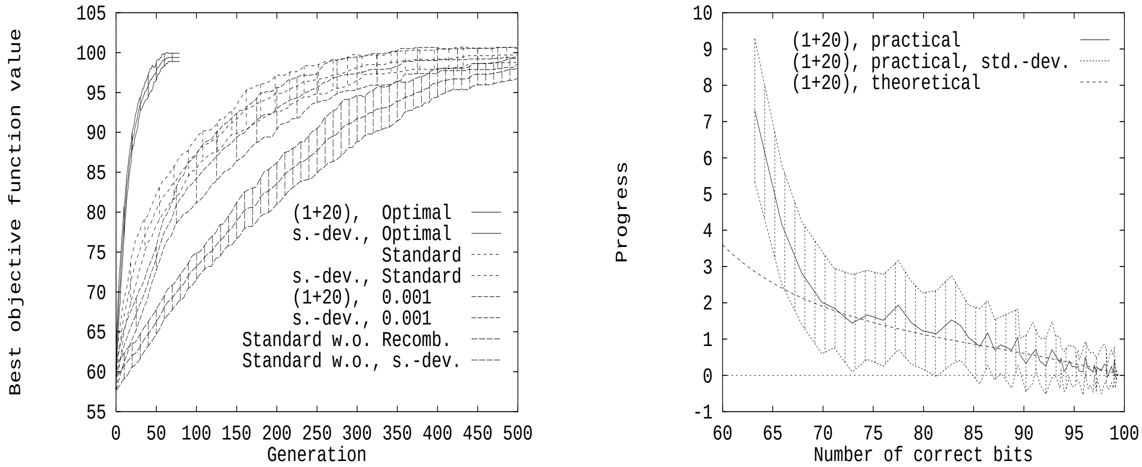

Figure 3: Comparison of the convergence behaviour of different variants (left) and comparison of theoretical and experimental progress results for an optimally adjusted mutation rate (right)

In order to perform a first experimental test of the theory derived so far, tests were run for the case of a (1+20)-GA and $l = 100$. It should be noted that for $\mu = 1$ recombination has no effect, since recombining identical individuals always yields the parent individual. The best performance values of the following algorithmic variants are compared in the left plot of figure 3:

- A (1+20)-GA which adjusts its mutation rate to the optimal value with respect to maximization of the expected progress $\varphi_{(1+20)}(l = 100, f_a, p)$ according to the measured performance value f_a of the best individual in each generation (this variant is labeled "(1+20), Optimal" in figure 3).
- A standard GA with $\lambda = 20$, $p_m = 0.001$, $p_c = 0.6$, one-point crossover, and proportional selection (labeled "Standard").
- A (1+20)-GA with a constant mutation rate of $p_m = 0.001$ (labeled "(1+20), 0.001").
- A GA with $\lambda = 20$, $p_m = 0.001$, $p_c = 0$, and proportional selection, i.e. a standard GA without recombination (labeled "Standard w.o. Recomb.").

These curves are plotted together with their standard deviations (results are averaged for 10 runs, each) from left to right in the order given above in figure 3 (left part). The resulting plot for the optimally adjusted mutation rate has a very small standard deviation and reaches the one percent vicinity of the optimum a factor larger than six times faster than both the standard GA and the (1+20)-GA with $p_m = 0.001$.

Additionally, for the optimally adjusted mutation rate it is possible to check the dependence of the local progress $\varphi_{(1+20)}(l = 100, f_a)$ on the number f_a of correct bits. The experimental progress values are calculated from the data by extracting the improvement of the best value between generations and plotting the improvement $f_a^*(t+1) - f_a^*(t)$ (i.e. the progress per generation) as a function of the actual best performance $f_a^*(t)$. The result and its standard deviation is plotted in the right part of figure 3, together with the theoretically optimal progress $\varphi^*_{(1+20)}(f_a) = \max\{\varphi_{(1+20)}(l = 100, f_a, p) \mid 0 \leq p \leq 1\}$. A discrepancy between theoretical and experimental values at the beginning of the run can be explained by the fact that the GA run starts with a population of 20 individuals initialized at random instead of one as assumed in this theory. As a consequence, the initial experimental progress is much larger than the theoretical one and the experiment starts at a best solution of quality 63. However, after only two generations the experimental progress curve approaches the theoretical one rather well, indicating that the theory developed here provides a reasonably valid mathematical approach.

4. TOWARDS UNDERSTANDING SELF-ADAPTATION

As demonstrated in the previous sections, the optimal schedule for the mutation rate depends on several quantities, i.e. the length l of the bitstrings, the current objective function value f_a, the population size λ, and, of course, the actual objective function f. Some authors have either empirically observed that a time-dependent decrease of the mutation rate is advantageous (see [10], who does not present an exact expression for the time-schedule used) or derived a similar result by a theoretical approach (see [11]). The expression $p_m(t) = \sqrt{\alpha/\beta} \cdot \frac{\exp(-\gamma t/2)}{\lambda\sqrt{l}}$ (α, β, γ are constants) by Hesser and Männer introduces a time-dependence resulting in curves over time which are similar to that one presented here. The problem is, however, that a predefined deterministic time schedule can not take into account the particularities of different fitness functions but is again fixed by some new exogenous parameters such that the schedule must be tuned according to the fitness function, resembling the temperature schedule tuning problem in Simulated Annealing. *Self-adaptation* of mutation rates is a different approach, intending a self-organizing schedule of mutation rates during the course of evolution. Basically, the idea stems from Evolution Strategies [7]. For GAs, a first approach towards self-adaptation of the mutation rates was presented recently, demonstrating the advantages of the approach experimentally [8]. The basic working mechanism of self-adaptive mutation within GAs can be described briefly as follows:

- Each individual is extended by additional $\hat{l}$ bits ($\hat{l} = 20$ turned out to be a reasonable choice) which are initialized at random. These bits are interpreted as the encoding of a real number between 0 and 0.5, the individuals' private mutation rate, which determines the mutation probability for the individual. On average, half of the bits can expected to be correct, such that the upper bound of 0.5 on possible values of the mutation rate makes sense.

- Mutation of the individual i works by decoding the mutation rate of the individual, resulting in mutation probability $\hat{p}_i$, then mutating with mutation rate $\hat{p}_i$ the part of the individual which encodes $\hat{p}_i$, again decoding the (now mutated) mutation rate, which yields mutation rate p_i, and finally mutating the object variable part of the individual by using mutation rate p_i.

This way, private mutation rates linked to the individuals are subject to mutation and therefore change over time probabilistically. However, the process is directed due to the selection process, which favours good object variable information as well as, indirectly, appropriate strategy parameter settings. Experimentally, a ratio of $\mu/\lambda \approx 1/6$ turned out to be a reasonable setting for the selection pressure, balancing between maintaining sufficient diversity within the strategy parameters on the one hand and driving the search towards better regions of the search space on the other hand [8].

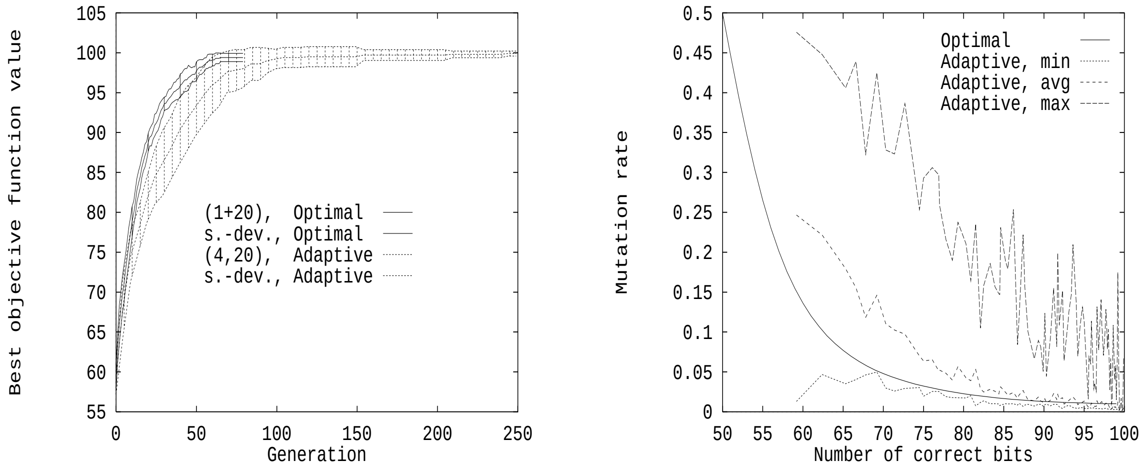

Figure 4: Course of the optimization (left) and schedule of the mutation rate (right) for an optimally adjusted mutation rate and self-adaptive mutation rates

In analogy to figure 3, in the left part of figure 4 the dependence of the best objective function value over the number of generations is plotted, again for a (1+20)-GA with a mutation rate adjusted optimally according to the theoretical schedule. Only slightly worse behaves the (4,20)-GA with self-adaptive mutation rates, which is also shown in the same graph. This provides a strong hint to the assumption that the self-adaptation mechanism enables the algorithm to encounter a near-optimal schedule of the mutation rates by its own. This assumption is confirmed by the graph shown in the right part of figure 4 where the mutation rates over the number of correct bits are shown for the theoretical case of a (1+20)-GA as well as for the average, minimum, and maximum mutation rates found in the genetic material of self-adaptive mutation rates for the (4,20)-GA. It it natural here to ask, why a (1+20)-GA and a (4,20)-GA are compared with one another. The theory developed in section 3 can be extended in an approximative way to cover the general (μ,λ)-case as well as the (μ+λ)-case. However, the theoretical plots for the optimal mutation rate for a (1+20)-GA and a (4,20)-GA are very similar, such that the (1+20)-GA is a good representative. The further elaboration of the theory is not presented here due to space limitations. The graph demonstrates clearly that the optimal mutation rates are always available in the population during the course of evolution, such that by means of selection the algorithm can make use of the most appropriate mutation rates. Furthermore, a large diversity of mutation rates is available such that also at the beginning of the search appropriate mutation rates can be utilized. This leads to the conclusion that the self-adaptation mechanism works by an interaction of sufficient genetic diversity within the strategy parameters, an implicit link between advantageous settings of strategy parameters and good objective function values, and hence a preference of useful strategy parameters by means of a well chosen selective pressure, strong enough to eliminate garbage and soft enough to maintain diversity of strategy parameters.

1 J. H. Holland. *Adaptation in natural and artificial systems.* (The University of Michigan Press, Ann Arbor, 1975).

2 K. De Jong. *An analysis of the behaviour of a class of genetic adaptive systems.* PhD thesis, (University of Michigan, 1975).

3 J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In Schaffer [13], pages 51–60.

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

5 W. E. Hart and R. K. Belew. Optimizing an arbitrary function is hard for the genetic algorithm. In Belew and Booker [12], pages 190–195.

6 I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* (Frommann–Holzboog Verlag, Stuttgart, 1973).

7 H.-P. Schwefel. *Numerical Optimization of Computer Models.* (Wiley, Chichester, 1981).

8 T. Bäck. Self-adaptation in genetic algorithms. In *Proceedings of the First European Conference on Artificial Life, December 11-13*, Paris, France, 1991. (MIT Press, 1992).

9 T. Bäck and F. Hoffmeister. Extended selection mechanisms in genetic algorithms. In Belew and Booker [12], pages 92–99.

10 T. C. Fogarty. Varying the probability of mutation in the genetic algorithm. In Schaffer [13], pages 104–109.

11 J. Hesser and R. Männer. Towards an optimal mutation probability in genetic algorithms. In H.-P. Schwefel and R. Männer, eds., *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 23–32. (Springer, 1991).

12 R. K. Belew and L. B. Booker, eds. *Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications*, University of California, San Diego, USA, 1991. (Morgan Kaufmann Publishers, 1991).

13 J. D. Schaffer, ed. *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, California, June 1989. (Morgan Kaufmann Publishers, 1989).