Optimization of binary mutations for Evolutionary Algorithms

Josselin Garnier
Centre de Mathématiques Appliquées
Ecole Polytechnique
91128 Palaiseau Cedex, France
Josselin.Garnier@polytechnique.fr

Leila Kallel
Centre de Mathématiques Appliquées
Ecole Polytechnique
91128 Palaiseau Cedex, France
Leila.Kallel@polytechnique.fr

Abstract

This paper deals with theoretical parameter design for Evolutionary Algorithms, in the case of the mutation parameter. First, we prove the explicit relationship between the convergence time of an Evolutionary Algorithm and its mutation operator, in the case of the Onemax problem and for all possible memoryless binary mutations. Second, this relationship is used for deriving the optimal mutation strategy that minimizes the convergence time of the algorithm towards the optimum. We address both the static and time dependent mutation cases, and we show that the optimal mutation strategy does not depend on the string size.

Keywords

Evolutionary algorithms, stochastic analysis, binary mutations, Markov chains, hitting times.

1 Introduction

Evolutionary Algorithms (abbreviate EAs) are a class of stochastic function optimization techniques that have been given an increasing interest in the last decades. For instance, EAs provide numerical solutions to difficult problems for which the deterministic optimization methods cannot be applied.

However, the convergence times of EAs depend on the choices of their parameters in a manner that is problem dependent and rather vague up to nowadays.

This paper investigates the theoretical relationship between the convergence time of an Evolutionary Algorithm and its mutation operator. In order to make this analysis theoretically amenable, we consider a simple function and algorithm, namely the Onemax function [2] to be optimized with a stochastic hill-climber algorithm with population size one.

Previous work concerning the setting of EA parameters is often restricted to empirical comparisons of EA performance when using different parameters [7, 6, 9], sometimes leading to heuristic schemes for problem-dependent settings of the parameters [13]. However, such empirical comparisons present two drawbacks :

- They use elementary statistical tools to interpret the simulation data (for example statistical tests or confidence range of the average performances of EAs).
- They can miss the optimal parameter. In fact, such a trial and error method cannot extrapolate the influence of non-experimented parameters.

To overcome these drawbacks, a sophisticated statistical method has been proposed, recently, allowing a more reliable interpretation of the simulation data [10]. More precisely, it allows one to model the relationship between EA's performance and its parameter values, thanks to the simulation data with different parameters. The advantage of this method is that one can extrapolate the obtained model in order to find optimal parameter settings. Furthermore, this method can be applied with any fitness function, and allows to model the influence of any parameter or the joint effect of a set of parameters.

However, this method remains time consuming and of limited generalization properties, in the sense that the simulation data is necessarily obtained after a limited number of –user defined– time steps. Hence, the obtained model might depend on this arbitrary choice, as discussed by the authors in [10].

On the other hand, theoretical parameter design is a far from easy task. Nevertheless, some results have been obtained considering severe simplifications of fitness functions and algorithms. Rechenberg's 1/5-rule [18] or Back's results [5] are famous examples. More generally, some theoretical investigation of EA's performance with particular parameter settings, is available for problems where the fitness writes as a function of the distance to solution [23, 20, 19, 21, 8] (see [1] for a review) or for unimodal fitness functions (only 1 local optimum) as the Needle in a haystack [12], the *onemax* [16, 17, 12, 3, 15, 4, 22] the *long k-path* functions [21, 8, 11] and *controlled-path* functions [14].

In particular, the Onemax function have been given much interest in the EA literature: the order of the convergence time ($l \ln l$) has been derived since the very early days of EAs [3, 15], for algorithms with 1-sized populations using standard 1-bit-mutation (flip exactly one bit uniformly chosen) and c/l-mutations (flip each bit with probability c/l). Recent studies derived exact expressions proving that the 1-bit mutation implies a smaller expected convergence time ($l \ln l$) than the c/l-mutation ($1/(1-e^{-c})\, l \ln l$) [12]. However, the influence of population size on the convergence time, and the optimal mutation strategy, are still unknown up to know for the Onemax problem.

This paper provides some answers to these issues. Section 2 formalizes the general class of memoryless mutations and introduces some notations. Section 3 calculates the convergence time as a function of the mutation strategy. Section 4 investigates the link between the instantaneous velocity (or convergence rate towards the optimum) and the mutation strategy. This leads to deriving optimal time-dependent (one should say fitness-dependent) mutation strategy that maximizes the instantaneous velocity of the algorithm, and so minimizes the convergence time on the Onemax problem. Section 5 comments on the relation between the convergence rate of EAs, the slope of the convergence curve and standard quantities used for computing the "speed" of a deterministic or stochastic process. Finally, some numerical experiments validate this theory (asymptotic in string size l), and also investigate the influence of population size.

2 Notations

State space. In this paper l stands for an integer, and we shall develop our theory in the asymptotic framework $l \gg 1$. The state space of the processes that will be studied is $E = \{0,1\}^l$, whose cardinality is 2^l. We shall consider the Hamming distance d defined between two strings $X, X' \in E$ by

$$d(X, X') = \sum_{i=1}^{l} \mathbf{1}_{X_i \neq X'_i}.$$

The algorithm. The Evolutionary Algorithm we consider can be described as a Markov Chain $X(t)$, as follows,

$$\mathbf{X(t)} \overset{Mutation}{\longrightarrow} Y(t) \overset{Selection}{\longrightarrow} \mathbf{X(t+1)} = argmax\{f(X(t)), f(Y(t))\}, \tag{1}$$

where $X(0)$ is chosen uniformly in E, and f is the fitness function to be maximized. In the following we consider the Onemax function,

$$\forall X \in E,\ f(X) = \sum_{i=1}^{l} X_i.$$

The class of memoryless mutations. We construct a general class of mutations that includes the common 1-bit-mutation (flip exactly one bit uniformly chosen), the c/l-mutation (flip each bit with probability c/l), and any other memoryless stochastic or deterministic mutation.

Definition 2.1
The mutated string of $X(t)$ is obtained by the following way.
At each time t,

1. *Choose an integer k according to the a priori distribution $\mathbf{p}(X(t)) = (p_k(X(t)))_{k \in \mathbf{N}}$.*
2. *Flip exactly k bits of $X(t)$ uniformly among all possible choices of k bits.*
3. *Then build the string $Y(t)$ by inverting the k chosen bits of $X(t)$.*

Examples of mutations: With the notations of this definition, note that,

- The 1-bit-mutation is obtained for the deterministic choice $\mathbf{p} = \delta_1$ independent of X (i.e. $p_1 = 1$ and $p_k = 0$, $k \neq 1$).
- The c/l-mutation is obtained for a Binomial distribution $\mathbf{p} = \mathcal{B}(l, \frac{c}{l})$ independent of X (i.e. $p_k = C_l^k \left(\frac{c}{l}\right)^k \left(1 - \frac{c}{l}\right)^{l-k}$). Note that this distribution converges as $l \to \infty$ to a Poisson distribution with parameter c (i.e. $p_k = e^{-c}\frac{c^k}{k!}$).

This class **includes any mutation** one can think of, regarded that it is memoryless. A memory here means that one would decide of the mutation at time t given the results obtained at times 0,...,$t-1$. Note that a dependence of the distribution $\mathbf{p}$ at time t with respect to $X(t)$ is permitted. If so the mutation is said to be inhomogeneous. If the distribution $\mathbf{p}$ does not depend on $X(t)$ then the mutation is said to be homogeneous.

Goal of the study. Find the expected first hitting times T of the optimum $X^f \in E$ (say for instance $X^f = (00\ldots0)$) by the Markov chain $X(t)$ (equation 1) as a function of the mutation strategy (choice of the distribution $\mathbf{p}$ in definition 2.1).

3 Homogeneous mutation

The process $X(t)$ is a Markov chain whose stationary transition matrix of size $2^l \times 2^l$ defined by:

$$P_{X,X'} := \mathbb{P}(X(1) = X' | X(0) = X),$$

is given by:

$$P_{X,X'} = p_{d(X,X')} \left(C_l^{d(X,X')} \right)^{-1}$$

Let us introduce the auxiliary process $Z(t)$ with state space $\{0, ..., l\}$ defined by:

$$Z(t) = d(X(t), X^f). \tag{2}$$

Z is a Markov chain starting from $Z(0) = d(X^0, X^f)$ with stationary transition matrix Q.

Lemma 3.1: *The coefficients of the transition matrix Q of the chain Z are given by:*

$$\begin{aligned} Q_{j,j-2j'} &= \sum_{k=1}^{\infty} p_{2k} \frac{C_j^{j'+k} C_{l-j}^{k-j'}}{C_l^{2k}} \ , \ 2 \leq 2j' \leq j, & (3) \\ Q_{j,j-2j'-1} &= \sum_{k=0}^{\infty} p_{2k+1} \frac{C_j^{j'+k+1} C_{l-j}^{k-j'}}{C_l^{2k+1}} \ , \ 1 \leq 2j'+1 \leq j, & (4) \\ Q_{j,j-j'} &= 0 \ , \ \textit{if } j' \leq 0. & (5) \end{aligned}$$

with $Q_{j,j} = 1 - \sum_{j'=1}^{j} Q_{j,j-j'}$.

Proof: From the expression of the transition matrix of the chain X:

$$\begin{aligned} Q_{j,j-j'} &= \sum_{k=1}^{\infty} p_k \left(C_l^k\right)^{-1} \mathcal{C}_{j,j',k} \ , \text{ if } 1 \leq j' \leq j, & (6) \\ Q_{j,j-j'} &= 0 \ , \text{ if } j' \leq 0, & (7) \end{aligned}$$

with $Q_{j,j} = 1 - \sum_{j'=1}^{j} Q_{j,j-j'}$. Here $\mathcal{C}_{j,j',k}$ is a combinatorial factor that is defined by:

$$\mathcal{C}_{j,j',k} = \text{card}\left\{X',\, d(X', X^f) = j - j',\, d(X', X) = k\right\} \text{ where } d(X, X^f) = j.$$

In order to compute the combinatorial factor one may assume by symmetry arguments that $X^f = (0, ..., 0)$ and $X = (1, .., 1, 0, .., 0)$ (with the first j bits equal to 1). As a result $\mathcal{C}_{j,j',k}$ is equal to:

$$\mathcal{C}_{j,j',k} = \text{card}\left\{X',\, \sum_{i=1}^{l} X_i' = j - j',\, \sum_{i=1}^{j}(1 - X_i') + \sum_{i=j+1}^{l} X_i' = k\right\}$$

which also reads:

$$\mathcal{C}_{j,j',k} = \text{card}\left\{X',\, \sum_{i=1}^{j} X_i' = j - \frac{k+j'}{2},\, \sum_{i=j+1}^{l} X_i' = \frac{k-j'}{2}\right\}$$

The final result depends on the parity of the difference $k - j'$:

$$\mathcal{C}_{j,j',k} = \begin{cases} 0 & \text{if } k = j' + 1 \bmod 2 \\ C_j^{\frac{k+j'}{2}} C_{l-j}^{\frac{k-j'}{2}} & \text{if } k = j' \bmod 2 \end{cases}$$

□

Corollary 3.2: *In the asymptotic framework $l \gg 1$, the expression of the transition matrix Q can be simplified into:*

$$\begin{aligned} Q_{j,j-2j'} &= \sum_{k=1}^{\infty} p_{2k} C_{2k}^{j'+k} \left(1 - \frac{j}{l}\right)^{k-j'} \left(\frac{j}{l}\right)^{k+j'} \quad , \ 2 \le 2j' \le j, & (8) \\ Q_{j,j-2j'-1} &= \sum_{k=0}^{\infty} p_{2k+1} C_{2k+1}^{j'+k+1} \left(1 - \frac{j}{l}\right)^{k-j'} \left(\frac{j}{l}\right)^{k+j'+1} \quad , \ 1 \le 2j'+1 \le j, & (9) \\ Q_{j,j-j'} &= 0 \ , \ \textit{if } j' \le 0. & (10) \end{aligned}$$

with $Q_{j,j} = 1 - \sum_{j'=1}^{j} Q_{j,j-j'}$.

Proof: These are straightforward expansions of the expressions derived in Proposition 3.1. They are based on the limit behaviors of the combinatorial factors:

$$C_n^p \simeq \frac{n^p}{p!} \text{ as } n \to \infty$$

□

Let us assume that the chain Z arrives at point j at time t. We then denote by τ_j the waiting time for the next jump, and by N_j the length of this jump. By the strong Markov property, the random variables τ_j and N_j are independent and their respective statistical distributions are given by:

$$\begin{aligned} \mathbb{P}(\tau_j = t) &= (1 - Q_{j,j}) Q_{j,j}^{t-1} \text{ for } t \in \mathbb{N}^*, & (11) \\ \mathbb{P}(N_j = n) &= \frac{Q_{j,j-n}}{1 - Q_{j,j}} \text{ for } n = 1, ..., j. & (12) \end{aligned}$$

The statistical distribution of the first hitting time T can be expressed in terms of the variables N_j and τ_j. In particular, conditioning with respect to the first jump yields:

$$\mathbb{P}_j (T = t) = \sum_{t'=1}^{t} \sum_{n=1}^{j} \mathbb{P}_j (T = t | \tau_j = t', N_j = n) \, q_j(t') \mathbb{P}(N_j = n),$$

where $q_j(t) = \mathbb{P}(\tau_j = t)$. Since the weak Markov property insures that:

$$\mathbb{P}_j (T = t | \tau_j = t', N_j = n) = \mathbb{P}_{j-n} (T = t - t'),$$

we get the inductive relation:

$$\mathbb{P}_j (T = t) = \sum_{t'=1}^{t} \sum_{n=1}^{j} \mathbb{P}_{j-n} (T = t - t') \, \mathbb{P}(\tau_j = t') \mathbb{P}(N_j = n). \tag{13}$$

Iterating this formula establishes the following closed-form expression which holds true for any k and l,

$$\mathbb{P}_k(T=t) = \sum_{J=1}^{k} \sum_{\substack{n_1+\ldots+n_J=k \\ n_j \geq 1}} \prod_{i=1}^{J} \mathbb{P}(N_{k-n_1-\ldots-n_{i-1}} = n_i) q_k \star q_{k-n_1} \star \ldots \star q_{k-n_1-\ldots n_{J-1}}(t), \tag{14}$$

where $\star$ stands for the discrete convolution,

$$q \star r(t) = \sum_{t'=1}^{t-1} q(t') r(t-t').$$

If we assume that the starting point X^0 is chosen uniformly over the state space $\{0,1\}^l$, then the chain Z starts from the random point which obeys the distribution $\mu(k) = 2^{-l} C_l^k$ so that:

$$\mathbb{P}_\mu(T=t) = \sum_{k=0}^{l} 2^{-l} C_l^k \mathbb{P}_k(T=t). \tag{15}$$

We now turn to the study of the expectation of T in order to exhibit its behavior for large values of l. Multiplying (13) by t and summing over t yields the inductive relation:

$$\mathbb{E}_j[T] = \mathbb{E}[\tau_j] + \sum_{n=1}^{j} \mathbb{E}_{j-n}[T] \mathbb{P}(N_j = n). \tag{16}$$

Iterating this formula establishes the following closed-form expression which holds true for any k and l:

$$\mathbb{E}_k[T] = \sum_{J=1}^{k} \sum_{n_1+\ldots+n_J=k} \left(\sum_{i=0}^{J-1} \mathbb{E}[\tau_{k-n_1-\ldots-n_i}] \right) \prod_{i=1}^{J} \mathbb{P}(N_{k-n_1-\ldots-n_{i-1}} = n_i). \tag{17}$$

In the asymptotic framework $l \gg 1$, this complicated expression gets much simpler.

Proposition 3.3: *In the asymptotic framework $l \gg 1$, the expectation of the first hitting time can be expanded as:*

$$\mathbb{E}_\mu[T] = l\left(p_1^{-1}\ln(l) + R(\mathbf{p})\right)$$

where $R(\mathbf{p}) = -p_1^{-1}\ln 2 + p_1^{-1} C + G(1/2, \mathbf{p})$, $C \simeq 0.58$ is the Euler's constant, and the function G that depends only on the distribution $(p_k)_{k\in\mathbb{N}}$ is defined in Equation (23).

According to this proposition, the best choice for the a priori distribution $(p_k)_{k\in\mathbb{N}}$ is $\mathbf{p} = \delta_1$ (i.e. $p_1 = 1$ and $p_k = 0$, $k \neq 1$). Then $R(\mathbf{p}) = C - \ln(2)$.

Proof of Proposition 3.3.

Step 1. There exist functions f_τ and f_N which do not depend on l such that, for $l \gg 1$: $\mathbb{E}[\tau_j] = f_\tau(j/l)$ and $\mathbb{E}[N_j] = f_N(j/l)$. The so-called velocity

$$v(a) := \frac{\mathbb{E}[N_j]}{\mathbb{E}[\tau_j]} = \frac{f_N(a)}{f_\tau(a)} \tag{18}$$

is of the form:

$$v(a,\mathbf{p}) = \sum_{k=0}^{\infty} p_k v_k(a) \tag{19}$$

with:

$$v_{2k}(a) = \sum_{i=1}^{k} 2iC_{2k}^{k-i}(1-a)^{k-i}a^{k+i} \tag{20}$$

$$v_{2k+1}(a) = \sum_{i=0}^{k} (1+2i)C_{2k+1}^{k+1+i}(1-a)^{k-i}a^{k+i+1} \tag{21}$$

This step results from straightforward calculations of the expectations of N_j and τ_j whose statistical distributions are given by (12). In the asymptotic framework, the transition matrix Q gets the simple form given in Corollary 3.2, so that one easily establishes the general expressions of the v_k's. In particular the explicit forms of the first v_k's are:

$$\begin{aligned}
v_1(a) &= a\\
v_2(a) &= 2a^2\\
v_3(a) &= 3a^2\\
v_4(a) &= 4a^3(2-a)\\
v_5(a) &= 5a^3(2-a)\\
v_6(a) &= 6a^4(2a^2-6a+5)\\
v_7(a) &= 7a^4(2a^2-6a+5)
\end{aligned}$$

More generally, $v_{2k}(a) = \frac{2k}{2k+1}v_{2k+1}(a)$.

Step 2. If $a \in (0,1]$, then we have for $l \gg 1$[1]*:*

$$\mathbb{E}_{[al]}[T] = \sum_{j=1}^{[al]} v^{-1}(j/l,\mathbf{p}) \tag{22}$$

Let $0 < \delta \ll 1$, such that $1 \ll \delta l \ll l$. By Step 1, we get that over the interval $j \in \{[al], ..., [al]-[\delta l]\}$, the statistical distributions of N_j and τ_j are almost identical to the ones of $N_{[al]}$ and $\tau_{[al]}$. Using the inductive relation (16):

$$\mathbb{E}_{[al]}[T] \simeq \sum_{J=1}^{\infty} \sum_{\substack{n_1+...+n_{J-1}\leq[\delta l]<n_1+...+n_J\\ n_j\geq 1}} \prod_{j=1}^{J} \mathbb{P}(N_{[al]}=n_j)\left(J\mathbb{E}[\tau_{[al]}] + \mathbb{E}_{[al]-n_1+...+n_J}[T]\right).$$

On the one hand, by the renewal theory, if $\tilde{N}_j$ are independent random variables with the same distribution as $N_{[al]}$, then:

$$\frac{1}{[\delta l]}\sum_{J=1}^{\infty} J\mathbb{P}\left(\tilde{N}_1+...+\tilde{N}_{J-1} \leq [\delta l] < \tilde{N}_1+...+\tilde{N}_J\right) \overset{l\to\infty}{\simeq} \frac{1}{\mathbb{E}[N_{[al]}]} = \frac{1}{f_N(a)},$$

[1] $[x]$ stands for the integral part of x

On the other hand, we have

$$\sum_{J=1}^{\infty} \mathbb{P}\left(\tilde{N}_1 + \ldots + \tilde{N}_{J-1} \le [\delta l] < \tilde{N}_1 + \ldots + \tilde{N}_J\right) = 1.$$

We thus get that:

$$\mathbb{E}_{[al]}[T] \simeq \frac{f_\tau(a)}{f_N(a)}[\delta l] + \mathbb{E}_{[al]-[\delta l]}[T].$$

Iterating this process completes the proof of Step 2.

Step 3. Computation of $\mathbb{E}_\mu[T]$ *for* $l \gg 1$.
The function $a \mapsto v^{-1}(a, \mathbf{p})$ has a singularity near the point $a = 0$. Nevertheless

$$g := a \mapsto v^{-1}(a, \mathbf{p}) - p_1 a^{-1}$$

is a smooth function. As a consequence:

$$\mathbb{E}_{[al]}[T] = \sum_{j=1}^{[al]} p_1^{-1} \frac{l}{j} + \sum_{j=1}^{[al]} g(j/l, \mathbf{p})$$

which yields:

$$\mathbb{E}_{[al]}[T] = l\left(p_1^{-1} \ln l + p_1^{-1} \ln a + p_1^{-1} C + G(a, \mathbf{p})\right)$$

where G is the integral of g:

$$G(a, \mathbf{p}) = \int_0^a g(s, \mathbf{p}) ds. \tag{23}$$

If the chain X starts from a point X^0 which is chosen uniformly over the state space $\{0,1\}^l$, then the chain Z starts from the point k_0 which obeys the distribution $\mu(k) = 2^{-l} C_l^k$ so that:

$$\mathbb{E}_\mu[T] = \sum_{k=0}^{l} \mu(k) \mathbb{E}_k[T].$$

Using Stirling's formula (Abramowitz and Stegun, 1965) to expand $\mu(k)$:

$$\mathbb{E}_\mu[T] = l\left(p_1^{-1} \ln l + p_1^{-1} C + \sqrt{l} \int_0^1 e^{-l f(a)} \frac{\ln a + G(a, \mathbf{p})}{\sqrt{2\pi a(1-a)}} da\right)$$

where $f(a) = \ln 2 + a \ln(a) + (1-a)\ln(1-a)$. Using Laplace technique to compute the integral then establishes the result by choosing $R(\mathbf{p}) = -p_1^{-1} \ln 2 + p_1^{-1} C + G(1/2, \mathbf{p})$. □

Conclusion: The key result of this section, given by proposition 3.3 shows that the optimal homogeneous memoryless mutation is deterministic in terms of the number of flipped bits: it flips 1-bit uniformly chosen. However, it does not significantly improves the convergence time of the EA under study, which remains of the order of $l \ln l$.

4 Inhomogeneous chain

The process $X(t)$ is a process starting from some point X^0, whose evolution is governed by the following way.
At each time t,

1. choose an integer k according to the distribution $(p_k(X(t)))_{k\in\mathbb{N}}$. p_k depends on $X(t)$ only through $d(X(t), X^f)$.
2. build the string $Y(t)$ by choosing uniformly k bits of $X(t)$ and inverting them.

The process $X(t)$ is a Markov chain whose non-stationary transition matrix of size $2^l \times 2^l$ is given by:

$$P_{X,X'} = p_{d(X,X')}(d(X,X^f)) \left(C_l^{d(X,X')}\right)^{-1}$$

The analysis of this system is the very same as its homogeneous counterpart. One establishes the expression of the expectation of the first hitting time (the same as that of equation 22) recalled in the following proposition.

Proposition 4.1: *Starting from a string at distance* $[al]$ *(*$a \in (0,1)$*) from optimum, the expectation of the first hitting time is,*

$$\mathbb{E}_{[al]}[T] = \sum_{j=1}^{[al]} v^{-1}(j/l, \mathbf{p}(j)), \tag{24}$$

where

$$v(b, \mathbf{p}) = \sum_{k=0}^{\infty} p_k v_k(b) \tag{25}$$

and

$$v_{2k}(b) = \sum_{i=1}^{k} 2iC_{2k}^{k-i}(1-b)^{k-i}b^{k+i} \tag{26}$$

$$v_{2k+1}(b) = \sum_{i=0}^{k} (1+2i)C_{2k+1}^{k+1+i}(1-b)^{k-i}b^{k+i+1} \tag{27}$$

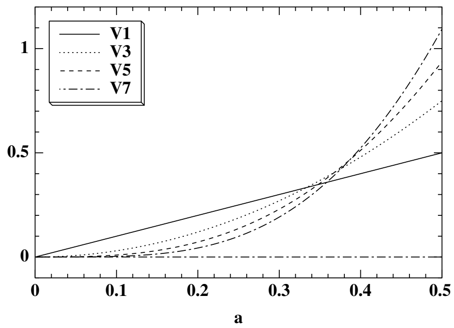
The identity (24) justifies the term "instantaneous velocity" for the quantity v (see section 5 for more details). The best choice for the site-dependent distribution $\mathbf{p}$ is the one that maximizes the velocity v at any instant (this is due to the fact that $\mathbb{E}[T]$ is decreasing with respect to the velocities $v(b, \mathbf{p})$ whatever the state b). From the expression (25) of v, one can derive the optimal instantaneous mutation.

Proposition 4.2:
Suppose the current string is at normalized Hamming distance a *from optimum. Then we have that,*

1. *The optimal mutation distribution flips exactly* $(2k(a)+1)$ *bits uniformly chosen:*

   $$\mathbf{p}([al]) = \delta_{2k(a)+1}$$

   where $k(a)$ *(explicitly given below) is the value of* k *that maximizes the velocity* $v_{2k+1}(a)$ *(27).*



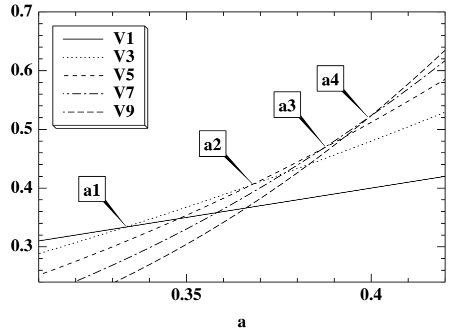


Figure 1: Theoretical instantaneous velocity $v_k(a)$ for different k. The velocity corresponding to $k = 1$ is the largest for $a < 1/3$.

2. *The integer-valued function* $k(a)$, $a \in (0, 1)$ *is stepwise constant and monotonously increasing,*

$$k(a) = s \quad \textit{if} \quad a \in [a_{s-1}, a_s),$$

where a_{s-1} *and* a_s *are the end-points that delimit the interval in which* $v_s(a)$ *is the largest of the* $v_k(a)$.

The value of a_s can be computed easily from the above expressions of the v_k's (27). We have found that:
- $a_0 = 0$,
- $a_1 = 1/3 \simeq 0.333$,
- $a_2 = 1 - \sqrt{2/5} \simeq 0.368$,
- $a_3 = 1 - \frac{1}{42}\sqrt[3]{5292 + 84\sqrt{3885}} - 2/\sqrt[3]{5292 + 84\sqrt{3885}} \simeq 0.387$, ...
- $a_s \uparrow 1/2$ as $s \to \infty$.

Remark: We can exhibit several remarkable features.

• At any instant, the optimal distribution $\mathbf{p}$ is a Dirac function, i.e. the optimal number of flipped bits is deterministic. It may seem quite obvious that there exists an optimal deterministic strategy. Indeed the reduced state space has $l + 1$ states. For each state, there are $l + 1$ different deterministic strategies (how many bits should be flipped). Hence the number of deterministic strategies is finite. A randomized strategy is a probability distribution over this set. Its expected runtime is the mean of the expected runtimes of the deterministic strategies. Hence it is trivial that there exists an optimal deterministic strategy. Note however that these heuristic arguments hold true for finite l, but fail in the limit $l \to \infty$ as the number of deterministic strategies also tends to infinity. The remarkable features that are proved here above are that there still exists an optimal deterministic strategy in the asymptotic framewrok $l \gg 1$, and that this optimal strategy can be described with high precision.

• As soon as the chain is at distance smaller than 1/3 from the final point, the optimal distribution is $\mathbf{p} = \delta_1$, i.e. $p_1 = 1$ and $p_k = 0$, $k \neq 1$ (flip 1 bit uniformly chosen). It is not so surprising that close to the optimum one bit mutations are optimal. Indeed flipping more bits has a small chance for greater improvements, but a large chance of flipping more suitable bits leading to worse strings. These heuristic arguments are proved rigourously in this paper, and we also provide the exact distance 1/3 from the optimum below which one bit mutations are optimal.

• Whatever the position, the optimal number of bit-inversions is an odd number. This number diverges as $j \to l/2$. This is in some sense expected. Indeed, if $d(X(t), X^f) > l/2$, then the operation which consists in inverting all bits immediately leads to a configuration for which $d(X(t), X^f) < l/2$.

Conclusion: The key result of this section is that the optimal inhomogeneous mutation flips a deterministic number of bits uniformly chosen, and this number is increasing (and step-wise constant) w.r.t. the distance to optimum. Again, as in the homogeneous case, the corresponding convergence time remains of the order of $l \ln l$.

5 The mean velocity and the convergence rate

Last section shows that maximizing the velocity as defined in (18) is equivalent to minimizing the convergence time (see Eq. (24)). However, up to this stage of the paper, we still don't know the graphical or physical interpretation of the proposed expression for the velocity.

As a consequence, we need clarifying two key issues, before turning to some experimental validation.

- What is the precise interpretation of the – so-called – velocity (18) defined in previous section ?
- What is the relation between the velocity and the *convergence rate* (definition 5.2) of an Evolutionary Algorithm, i.e. the slope of the convergence curve in the 'fitness versus time' space ?

In order to address these issues, we start by considering natural velocity definitions for deterministic processes, then for general stochastic processes, and finally for the stochastic process $X(t)$ corresponding to the Evolutionary Algorithm addressed in this paper.

5.1 Velocity of a deterministic process

Consider first the case of deterministic real valued process $x(t)$, where $t \in \mathbb{R}^+$ denotes the time parameter. To avoid unnecessary complications in the discussion let us assume that $x(t)$ is strictly monotonous. A natural definition for the *local velocity* at x of this deterministic process is:

$$v(x) = \lim_{\delta t \to 0} \frac{x(t+\delta t) - x(t)}{\delta t}.$$

This latter expression characterizes the change in $x(t)$ per time-unit, given the fact that $x(t) = x$. But one can also be interested in the change in time per a unit increase in $x(t)$:

$$w(x) = \lim_{\delta x \to 0} \frac{T_{x \to x+\delta x}}{\delta x},$$

where $T_{a \to b}$ denotes the time necessary for $x(t)$ to move from a to b. These quantities are equivalent in the sense that one is the inverse of the other:

$$w(x) = \frac{1}{v(x)}.$$

Furthermore, we have the remarkable relation between the first hitting time of b and the instantaneous local velocities v:

$$T_{a\to b} = \int_a^b \frac{1}{v(x)} dx$$

However, the two latter equalities are not directly generalizable to the case of stochastic processes, in the sense that it does no longer hold if we replace the instantaneous velocities by their expectations.

5.2 Mean velocity of a stochastic process

Consider now a stochastic process $X(t)$, $t \in \mathbb{R}^+$. The velocity of $X(t)$ can be defined analogously to the deterministic case, so that its expectation is:

$$\bar{V}(a) := \mathbb{E}[V(a)] = \lim_{\delta t\to 0} \frac{\mathbb{E}[X(t+\delta t) - X(t)|X(t) = a]}{\delta t}. \qquad (28)$$

On the other hand one can also be interested in the expected time necessary for a unitary increase in $X(t)$,

$$\bar{W}(a) := \mathbb{E}[W(a)] = \lim_{\delta a\to 0} \frac{\mathbb{E}[T_{a\to a+\delta a}]}{\delta a},$$

where $T_{a\to b}$ is a random variable that denotes the time necessary for $X(t)$ to move from a to b.

The expected values $\bar{V}(a)$ and $\bar{W}(a)$ are no longer the inverse one of the other as in the deterministic case, but they are related in a complex way depending on the process $X(t)$. Indeed the random variables satisfy $W(a) = 1/V(a)$, but $\mathbb{E}[W(a)] = \mathbb{E}[1/V(a)] \neq 1/\mathbb{E}[V(a)]$. We shall explicit the relationship between $\bar{V}(a)$ and $\bar{W}(a)$, as well as the link between velocity and *convergence rate* in the case of the discrete time stochastic process $X(t)$, $t \in \mathbb{N}$, studied in this paper.

5.3 Mean velocity of the Evolutionary Algorithm

For Evolutionary Algorithms, one is usually interested in the *convergence rate* towards the optimum, which can be seen as the slope of the convergence curve "fitness vs. time", i.e. the expected increase in fitness per time unit. In order to formalize this *convergence rate* in terms of random variables, we start by introducing some notations.

Notations : Let us assume that the chain $X(t)$ arrives at a point at distance j from the optimum, at time t. We denote

- τ_j the waiting time for the next jump (jump to a fitter point closer to the optimum),
- N_j the length of this jump in terms of the fitness, i.e. $(f(X(t+\tau_j)) - f(X(t)))$.
- N_j^s the cumulative length of the jumps (if any) after s time steps, i.e. $(f(X(t+s)) - f(X(t)))$,

where f denotes the Onemax function or equivalently the Hamming distance to X^f.

The velocity of an Evolutionary Algorithm can be defined as above (Eq. 28). However, since we are interested in the fitness increase, we can consider the process $f(X(t))$ rather than $X(t)$. Then, one defines the velocity of the Evolutionary Algorithm as follows.

Definition 5.1
The mean velocity of an Evolutionary Algorithm in terms of fitness improvement is,

$$\bar{V}(j) = \mathbb{E}[f(X(t+1)) - f(X(t)) | f(X(t)) = j] = \mathbb{E}[N_j^1].$$

Note that this definition joins the commonly used *convergence rate* corresponding to the slope of the convergence curve. that is defined as follows.

Definition 5.2
Consider a time scale δt such that $1 \leq \delta t \ll l$. The expected convergence rate (slope of the convergence curve) can be defined as

$$\frac{\mathbb{E}[N_j^{\delta t}]}{\delta t}$$

Note also that this definition seems to be dependent on the scale δt, but we shall see it is not whenever $\delta t \ll l$.

It is essential here to note that the convergence rate or the mean velocity defined here above are *a priori* different from the expression we called velocity (18) in the last section (given by $\mathbb{E}[N_j]/\mathbb{E}[\tau_j]$): The convergence rate (resp. mean velocity) gives the expected fitness-improvement in δt (resp. 1) time steps, whereas the second gives the ratio of expected first-fitness-improvement and its expected occurrence time. Nevertheless, the following proposition establishes that these three quantities are equal, which justifies the term 'velocity' used in sections 3 and 4.

Proposition 5.3: *Assuming the process $X(t)$ is at distance j from the optimum at time t, we have,*

$$\frac{\mathbb{E}[N_j^{\delta t}]}{\delta t} = \frac{\mathbb{E}[N_j]}{\mathbb{E}[\tau_j]} \tag{29}$$

This key proposition shows that the quantity $\frac{\mathbb{E}[N_j]}{\mathbb{E}[\tau_j]}$ (the quotient of the expected first-increase of $f(X(t))$ and the expected time for the occurrence of this increase) precisely reflects the convergence rate or the velocity of the Evolutionary Algorithm $X(t)$.

Proof: By summing over the possible number of jumps, we get that for δt smaller than l (so that the statistical distributions of the jumps that depend on the parameter j/l are locally stationary):

$$\mathbb{E}[N_j^{\delta t}] = \sum_{k=1}^{\delta t} \underbrace{\mathbb{P}(k \text{ jumps happen within } \delta t \text{ time steps})}_{p_k} k\mathbb{E}[N_j]$$

If we denote by p the probability that a jump happens at one time step ($p = \mathbb{P}(\tau_j = 1)$), we obviously see that p_k is the expression of a Binomial probability distribution $\mathcal{B}(\delta t, p)$ with mean $p\delta t$:

$$p_k = C_{\delta t}^k \, p^k \, (1-p)^k$$

Replacing the above sum by the expectation of the Binomial, we get that,

$$\mathbb{E}[N_j^{\delta t}] = \delta t \, p \, \mathbb{E}[N_j]$$

Since the random variable τ_j (waiting time for a jump) follows a geometric law (13), we have that $\mathbb{E}[\tau_j] = 1/\mathbb{P}(\tau_j = 1)$, hence, $p = 1/\mathbb{E}[\tau_j]$, which establishes (29). □

Proposition 5.3 holds true for any mutation scheme as soon as it is memoryless (i.e. the choice of the mutated bits at some time does not depend on the previous choices). The absence of memory ensures that the waiting time for the first jump τ_j obeys a geometric law, which is the key argument in the demonstration of Eq. (29).

A noticeable feature here is that,

$$\frac{\mathbb{E}[N_j]}{\mathbb{E}[\tau_j]} \neq \mathbb{E}[\frac{N_j}{\tau_j}]$$

Hence the slope one can observe on the convergence curve, in the 'fitness versus time' space, is not given by $\mathbb{E}[N_j/\tau_j]$ as one could suppose intuitively. To be convinced with this fact, and also to determine the amount of difference between these two expressions, we compute $\mathbb{E}[N_j/\tau_j]$ in the following lemma.

Lemma 5.4: *Suppose $X(t)$ is at distance j from the optimum, at time step t. The convergence rate at time t, given by $\mathbb{E}[N_j]/\mathbb{E}[\tau_j]$ is related to the quantity $\mathbb{E}[N_j/\tau_j]$ as follows,*

$$\mathbb{E}[\frac{N_j}{\tau_j}] = -\frac{\ln(p)}{1-p}\ \frac{\mathbb{E}[N_j]}{\mathbb{E}[\tau_j]},$$

where p is the probability that a jump happens for at one time step.

Proof: The random variables N_j and τ_j are independent, so that:

$$\mathbb{E}[\frac{N_j}{\tau_j}] = \mathbb{E}[N_j]\mathbb{E}[1/\tau_j]$$

Since τ_j follows a geometric law with parameter $p = (1 - Q_{j,j})$ (13), we have that,

$$\begin{aligned} \mathbb{E}[1/\tau_j] &= \sum_{k=1}^{\infty} \frac{1}{k}\mathbb{P}(\tau_j = k) \\ &= \sum_{k=1}^{\infty} \frac{1}{k} p(1-p)^{k-1} \\ &= \frac{-p\ln(p)}{1-p} \end{aligned}$$

The last equality follows from the series expansion of $\ln(1-x)$. Recalling that $\mathbb{E}[\tau_j] = 1/p$ yields the desired result. □

Lemma 5.4 shows that $\mathbb{E}[N_j]/\mathbb{E}[\tau_j] < \mathbb{E}[N_j/\tau_j]$ for any $p \in (0,1)$. However, for values of p close to 1, we have $\frac{-\ln(p)}{1-p} \simeq 1$. Lemma 5.4 then implies: $\mathbb{E}[N_j]/\mathbb{E}[\tau_j] \simeq \mathbb{E}[N_j/\tau_j]$. This is in some sense expected, since p is the probability of improvement in one step so that the limit configuration $p \to 1$ is close to a deterministic case.

Conclusions: We end this section with the following conclusions that answer the questions stated in the beginning of this section.

- One can use indifferently the term velocity (Definition 5.1) or convergence rate (Definition 5.2) for the Evolutionary Algorithm $X(t)$.
- One can use the expression $\mathbb{E}[N_j]/\mathbb{E}[\tau_j]$ to compute the velocity (defined as $\mathbb{E}[N_j^{\delta t}]/\delta t$) for small δt values, provided that a memoryless mutation is used.

This latter fact, in its turn, justifies the term velocity used in sections 3 and 4, and allows to use the slope of the convergence curves in the next section to compare the empirical mutation velocities.

6 Experimental Validation

6.1 The (1+1) Evolutionary Algorithm

In the previous section, we proved that the optimal number of bit-flips writes as an increasing step function of the normalized distance to the optimum. In the following, we propose some numerical validation of the asymptotic results of last section, in the case of finite string length l.
We plot the distance to optimum versus time, during the run of the (1+1) Evolutionary Algorithm (1), for l taking values 100 and 1000. Note that the velocity (or convergence rate towards the optimum) are directly reflected by the slopes of the so plotted convergence curves.
Figure 2 shows that the optimal mutation strategy only depends on the normalized distance to optimum, and is the same for different values of l : the largest velocity (and steepest slope) is obtained with the 1-bit-flip mutation for a distance to optimum under 1/3; with the 3-bit-flip mutation for a distance to optimum in the range $[0.368, 1/3]$ and with the 5-bit-flip mutation for a distance to optimum in the range $[0.378, 0.368]$.

The slopes of the convergence curves (Figure 2) are not easy to precisely compare "graphically". For this reason, we also compute the velocities at different distances from the optimum, shown at figure 3.

We compute the velocity using the expression of equation (18). Note however that this cannot be achieved during the run of the algorithm, because one cannot guarantee that the evolutionary process visits strings at every distance to the optimum. Hence, we repeat 10000 times the following scheme, in order to calculate the value of $\mathbb{E}[N_j]/\mathbb{E}[\tau_j]$ by sampling 10000 values for N_j and τ_j.

1. Pick a point $x \in E$ at random at distance j from the optimum,
2. Iterate mutating x, until a fitter point (closer to the optimum here) is found for the first time (the mutation flips a number of bits uniformly chosen of string x, which number is sampled according to the distribution $\mathbf{p}$).
3. Compute the values of N_j (fitness increase) and τ_j (the number of mutation iterations for this fitness-increase).

6.2 Extension to the case of populations

In this section we address the following question : is the optimal mutation derived for 1-sized populations, still optimal for multi-membered populations ?
The rigorous answer to this question would probably require a precise investigation of the whole distribution of the instantaneous velocity, and not only its expectation derived in this paper. Nevertheless some numerical experiments have been conducted with an elitist generational GA, with mutation rate 1 and no crossover.

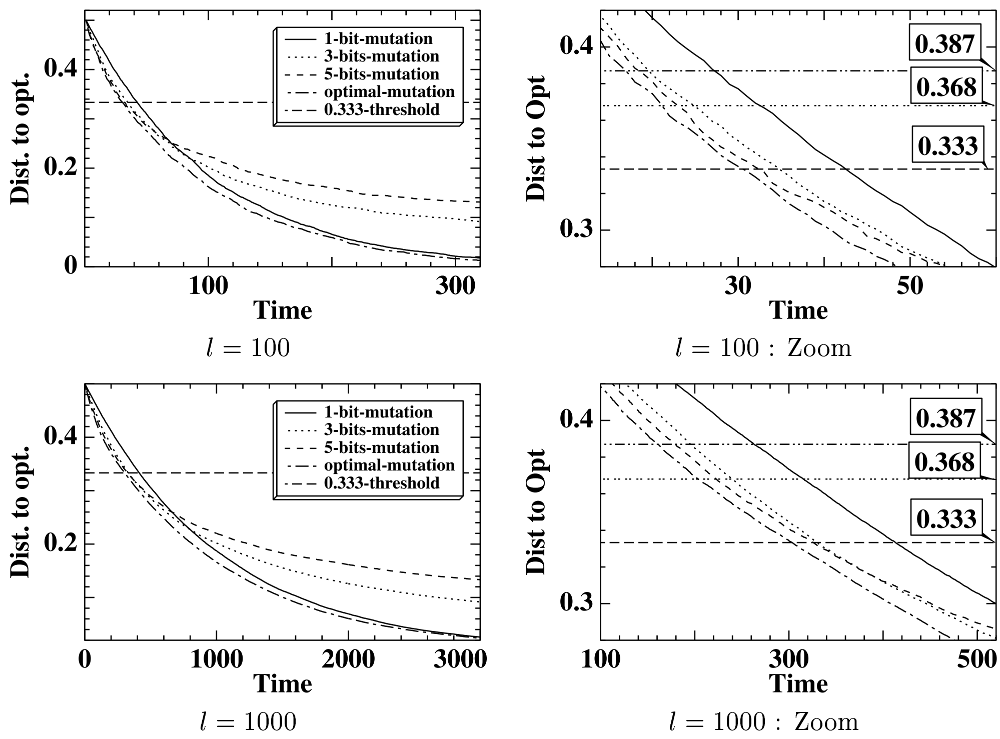


Figure 2: Convergence curve of the **process** $X(t)$ **(mutation only (1+1)-ES)**, on the Onemax Problem, averaged over 50 runs. The instantaneous velocity can be compared via the slopes of the convergence curves: As expected, for both l values, the largest velocity is obtained with the 1-bit-flip mutation for a distance to optimum under 1/3, with the 3-bit-flip mutation for a distance to optimum in the range $[0.368, 1/3]$, and with the 5-bit-flip mutation for a distance to optimum in the range $[0.378, 0.368]$.

Figure 4 shows that for finite l values, the threshold of 1/3 is not valid anymore. In fact the slope of the 5-bit-flip mutation curve is better (smaller) than that of the 1-bit-flip for a range of distances below 1/3 from the optimum. Hence the optimal mutation strategy for the Onemax Problem depends on the size of the population.

7 Discussion

We investigated the class of memoryless binary mutations. These mutations are compared by deriving the first hitting time of the optimum for a simple (1+1)-Evolutionary Algorithm on the Onemax Problem, for large l values. This study leads to the following results.

None of the mutations investigated in this paper significantly improves the convergence speed. The expected convergence time remains of the order of $l \ln l$. However, the terms of lower order in l can be decreased, leading to the following result.

The most efficient homogeneous mutation is to flip a deterministic number of bits: exactly one bit, uniformly chosen per bitstring.

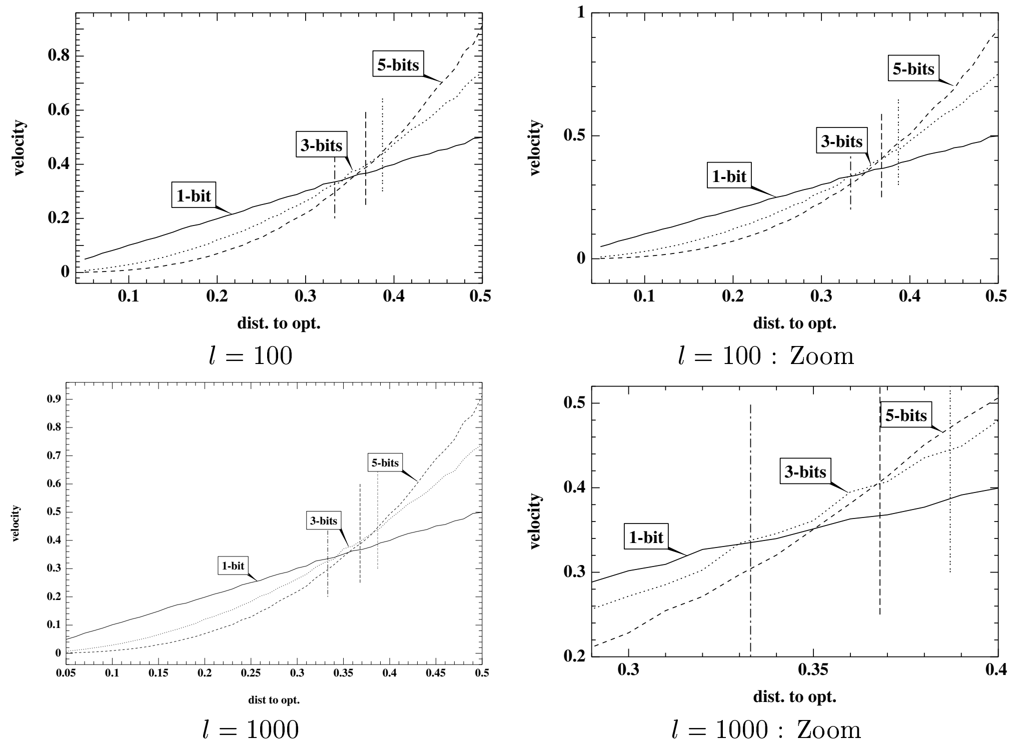


Figure 3: Empirical velocity averaged over 10000 simulations, for 3 types of mutations, (flip 1,3 or 5 bits per bitstring, uniformly chosen). The vertical lines indicate the theoretical distances where the optimal number of bit-flips changes.

In the inhomogeneous case, we show that the optimal mutation strategy is again deterministic in the number of bit-flips. The optimal number of bit-flips is an odd number $2k(a)+1$ depending on a, the normalized distance to optimum: the closer to optimum, the smaller the optimal number of bit-flips is. More precisely, $k(a)$ is a stepwise constant monotonously increasing function of a.

A noticeable feature is that $2k(a)+1=1$ if $a<1/3$, which means that the optimal inhomogeneous mutation corresponds to 1-bit-flips most of the time (for all strings at distances below $1/3$ from the optimum).

Experiments confirm these asymptotic results with population size 1. But, these results are not directly generalizable to population based algorithms. In fact the simulations show that the $1/3$ threshold is not valid any more with 100 or 1000 population sizes. We conclude that optimal mutation strategies for the Onemax problem depend on the size of the population.

Extending the analysis to population based EAs (or even to parallel processes with 1-sized populations) require more information than the expected instantaneous velocity derived in this paper. Deriving the extreme values of the velocity-distribution is then mandatory.

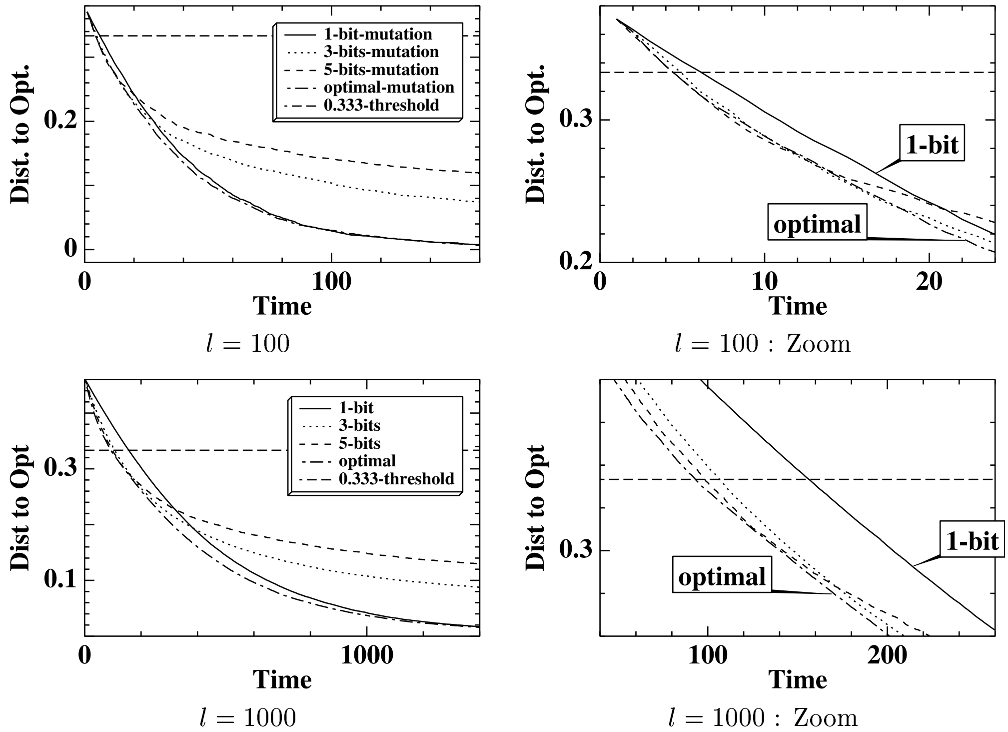


Figure 4: Convergence curve of a **mutation-only generational GA** (population size 100) on the Onemax Problem, averaged over 50 runs. The instantaneous velocity can be compared via the slopes of the convergence curves. Note that the 1/3 threshold does not seem to hold any more, as the slope of the 1-bit-mutation curve is does not becomes the best for distances below 1/3 from the optimum.

References

[1] Prugel-Bennet, A. and Rogers, A. (1999). *Modelling GA dynamics.* Springer. to appear in the *proceedings of the second Evonet Suumerschool on theoretical aspects of Evolutionary Computing.*

[2] Ackley, D. H. (1987). *A connectionnist Machine for Genetic Hill-climbing.* Kluwer, Boston.

[3] Bäck, T. (1992). The interaction of mutation rate, selection, and self-adaptation in genetic algorithm. In R. Manner and B. Manderick, editors, *Proceedings of the* 2^{nd} *Conference on Parallel Problems Solving from Nature*, pages 85–94. Morgan Kaufmann.

[4] Bäck, T. (1993). Optimal Mutation Rate in Genetic Search. In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann.

[5] Bäck, T. and Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23.

[6] Davis, L. (1991). *Handbook of Genetic Algorithms.* Van Nostram Reinhold, New York.

[7] DeJong, K. A. (1975). *The Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, Ann Harbor. *Dissertation Abstract International*, 36(10), 5140B. (University Microfilms No 76-9381).

[8] Droste, S., Jensen, T., and Wegener, I. (1998). A rigorous comlexity analysis of the (1+1) evolutionary algorithm for linear functions with boolean inputs. In *Proceedings of the Fifth IEEE International Conference on Evolutionary Computation.* IEEE Press.

[9] Fogel, D. B. (1995). *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence.* IEEE Press, Piscataway, NJ.

[10] François, O. and Lavergne, C. (2000). Design of evolutionary algorithms: A statistical perspective. *tp appear in IEEE Transactions on Evolutionary Computation. Preprint available at http://www-lmc.imag.fr/lmc-sms/Olivier.Francois/.*

[11] Garnier, J. and Kallel, L. (1999). Statistical distribution of the convergence time for long k-path problems. *IEEE Transactions on Evolutionary Computation.*.

[12] Garnier, J., Kallel, L., and Schoenauer, M. (1999). Rigourous results of the first hitting times for binary mutations. *Evolutionary Computation.*, 7(2):173–203.

[13] Kallel, L. (2000). *Convergence of Evolutionary Algorithms in time and space.* PhD thesis, Available at http://www.cmap.polytechnique.fr/~kallel/, Ecole Polytechnique – Palaiseau – France.

[14] Kallel, L. and Naudts, B. (1999). Longpaths for the genetic algorithm. In W. Banzhaf and C. Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 27–43. Morgan Kaufmann.

[15] Mühlenbein, H. (1992). How genetic algorithms really work: I. mutation and hill-climbing. In R. Manner and B. Manderick, editors, *Proceedings of the* 2^{nd} *Conference on Parallel Problems Solving from Nature*, pages 15–25. Morgan Kaufmann.

[16] Prügel-Bennet, A. and Shapiro, J. L. (1994). An analysis of genetic algorithms using statistical mechanics. *Physics Review Letters*, 72(9):1305–1309.

[17] Prügel-Bennett, A. and Shapiro, J. L. (1997). The dynamics of a genetic algorithm for simple Ising systems. *Physica D*, 104:75–114.

[18] Rechenberg, I. (1973). *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution.* Fromman-Holzboog Verlag, Stuttgart.

[19] Rogers, A. and Prügel-Bennet, A. (1999). Modelling the dynamics of a steady state genetic algorithm. In W. Banzhaf and C. Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 57–68.

[20] Rogers, A. and Prügel-Bennet, A. (1999). A solvable model of a hard optimization problem. to appear in the *proceedings of the second Evonet Suumerschool on theoretical aspects of Evolutionary Computing*.

[21] Rudolph, G. (1996). How mutation and selection solve long path problems in polynomial expected time. *Evolutionary Computation*, 4(2):195–205.

[22] Rudolph, G. (1997). *Convergence Properties of Evolutionary Algorithms*. Kovac, Hamburg.

[23] Shapiro, J. L. and Prügel-Bennet, A. (1997). Genetic algorithm dynamics in a two well potential. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 101–115. Morgan Kaufmann.