Self-adaptation of Mutation Rates in Non-elitist Populations

Duc-Cuong Dang and Per Kristian Lehre(✉)

School of Computer Science, University of Nottingham, Nottingham, UK
{duc-cuong.dang,PerKristian.Lehre}@nottingham.ac.uk

Abstract. The runtime of evolutionary algorithms (EAs) depends critically on their parameter settings, which are often problem-specific. Automated schemes for parameter tuning have been developed to alleviate the high costs of manual parameter tuning. Experimental results indicate that self-adaptation, where parameter settings are encoded in the genomes of individuals, can be effective in continuous optimisation. However, results in discrete optimisation have been less conclusive. Furthermore, a rigorous runtime analysis that explains how self-adaptation can lead to asymptotic speedups has been missing. This paper provides the first such analysis for discrete, population-based EAs. We apply level-based analysis to show how a self-adaptive EA is capable of fine-tuning its mutation rate, leading to exponential speedups over EAs using fixed mutation rates.

1 Introduction

An obstacle when applying Evolutionary Algorithms (EAs) is that their efficiency depends crucially, and sometimes unpredictably, on their parameter settings, such as selective pressure and mutation rates [12]. *Parameter tuning* [7], where the parameters are fixed before running the algorithm, is the most common way of choosing the parameters. A weakness with parameter tuning is that optimal parameter settings may depend on the current state of the search process. In contrast, *parameter control* allows the parameters to change during the execution of the algorithm, e. g. according to a fixed schedule as in simulated annealing, through feedback from the search, or via self-adaptation [7]. Adaptive parameters can be essential and advantageous (e. g. covariance-matrix adaptation [9]) in continuous search spaces. In discrete spaces, it has been shown that changing the mutation rate as a function of the current fitness [2] can improve the runtime, and the 1/5-rule has been used to adapt the population size [5].

While previous studies have shown the benefit of adaptive parameters, only global parameters were analysed. Instead, we look at so-called "evolution of evolution" or true self-adaptation [7], in which the parameter is encoded in the genome of individual solutions. The existing studies on this topic from the EC literature is mostly experimental [1,7,13], or about proving the convergence of the population model at their limit [1], i. e. infinite population.

We study evolution of mutation rates in *non-elitist* populations, where the mutation rates of individuals are encoded in their own genomes. The mutation

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 803–813, 2016.
DOI: 10.1007/978-3-319-45823-6_75

rate of the mutation rate is a *strategy parameter* p, which in *endogenous* control is itself evolved [1,14]. We consider *exogenous* control, where the parameter p is fixed globally. Our contribution is twofold: using a benchmark function, we provide necessary and sufficient conditions for self-adaptation to be effective; we show that self-adaptation is necessary in optimising a variant of this function. More precisely, an EA with a fixed or uniform mixing of mutation rates requires exponential time, while self-adaptation is efficient. As a by-product, we also prove that a non-elitist EA can outperform the elitist $(\mu + \lambda)$ EA.

2 Preliminaries

For any $n \in \mathbb{N}$, define $[n] := \{1, \dots, n\}$. The natural logarithm is denoted by $\ln(\cdot)$, and the logarithm to the base 2 is denoted by $\log(\cdot)$. For $x \in \{0,1\}^n$, we write $x(i)$ for the i-th bit value. The Hamming distance is denoted by $\mathrm{H}(\cdot,\cdot)$ and the Iverson bracket by $[\cdot]$. Given a partition of a search space $\mathcal{X}$ into m ordered "levels" $(A_1, \dots, A_m)$, we define $A_{\geq j} := \cup_{i=j}^{m} A_i$. A *population* is a vector $P \in \mathcal{X}^\lambda$, where the i-th element $P(i)$ is called the i-th *individual*. Given $A \subseteq \mathcal{X}$, we let $|P \cap A| := |\{i \mid P(i) \in A\}|$ be the number of individuals in population P that belong to the subset A.

All algorithms considered here are of the form of Algorithm 1 [4]. A new population P_{t+1} is generated by independently sampling λ individuals from an existing population P_t according to a selection mechanism p_{sel}, and perturbing each of the selected individuals by a variation operator p_{mut}. A fitness function $g : \mathcal{Y} \to \mathbb{R}$ is implicitly embedded in the selection mechanism p_{sel}.

Algorithm 1. [4]

Require: Finite search space $\mathcal{Y}$ with an initial population $P_0 \in \mathcal{Y}^\lambda$.
1: **for** $t = 0, 1, 2, \dots$ until a termination condition is met **do**
2: **for** $i = 1$ to λ **do**
3: Sample $I_t(i) \in [\lambda]$ according to $p_{\mathrm{sel}}(P_t)$, and set $x := P_t(I_t(i))$.
4: Sample $x' \in \mathcal{Y}$ according to $p_{\mathrm{mut}}(x)$, and set $P_{t+1}(i) := x'$.

We consider the standard bitwise mutation operator, where for any pair of bitstrings $x, x' \in \{0,1\}^n$ and any *mutation rate* $\chi \in (0, n]$, the probability of obtaining x' from x is $\Pr(x' = \mathrm{mut}(x,\chi)) = (\chi/n)^{H(x,x')}(1-\chi/n)^{n-H(x,x')}$. To model the parameter control problem, we assume that Algorithm 1 must choose the mutation rate parameter χ from a predefined set $\mathcal{M}$.

Uniform mixing, denoted $p_{\mathrm{mut}}^{\mathrm{mix}}$, chooses the mutation rate χ uniformly at random from the set $\mathcal{M}$ every time an individual is mutated, $p_{\mathrm{mut}}^{\mathrm{mix}}(x) := \mathrm{mut}(x,\chi)$, where $\chi \sim \mathrm{Unif}(\mathcal{M})$. The special case of $|\mathcal{M}| = 1$, i.e. a fixed mutation rate, has been studied extensively [4,12]. Here, we focus on $|\mathcal{M}| > 1$. It is known that such mixing of mutation operators can be beneficial [6,11].

Self-adaptation uses an extended search space $\mathcal{Y} = \mathcal{X} \times \mathcal{M}$, where each element (x, χ) consists both of a search point $x \in \mathcal{X}$ and a mutation rate $\chi \in \mathcal{M}$. A fitness function $g : \mathcal{Y} \to \mathbb{R}$ is defined by $g((x, \chi)) := f(x)$ for all $(x, \chi) \in \mathcal{Y}$. The mutation operator p_{mut} is written as $p_{\text{mut}}^{\text{adapt}}$ and it is parameterised by a globally fixed parameter $p \in (0, 1/2]$ such that $p_{\text{mut}}^{\text{adapt}}((x, \chi)) := (x', \chi')$ where $\chi' = \chi$ with probability $1 - p$, and $\chi' \sim \text{Unif}(\mathcal{M} \setminus \{\chi\})$ otherwise, and $x' = \text{mut}(x, \chi')$.

We analyse the runtime of Algorithm 1 using the level-based theorem [3]. This theorem applies to any population-based process where the individuals in P_{t+1} are sampled independently from the same distribution $D(P_t)$, where D maps populations to distributions over the search space $\mathcal{X}$. In Algorithm 1, the map is $D = p_{\text{mut}} \circ p_{\text{sel}}$, i.e., composition of selection and mutation.

Theorem 1 ([3]). *Given a partition $(A_1, \ldots, A_{m+1})$ of $\mathcal{X}$, define $T := \min\{t\lambda \mid |P_t \cap A_{m+1}| > 0\}$ to be the first point in time that elements of A_{m+1} appear in P_t of Algorithm 1. If there exist parameters $z_1, \ldots, z_m, z_* \in (0, 1]$, $\delta > 0$, a constant $\gamma_0 \in (0, 1)$ and a function $z_0 : (0, \gamma_0) \to \mathbb{R}$ such that for all $j \in [m]$, $P \in \mathcal{X}^\lambda$, $y \sim D(P)$ and $\gamma \in (0, \gamma_0]$ we have*

(G1) $\Pr(y \in A_{\geq j} \mid |P \cap A_{\geq j-1}| \geq \gamma_0\lambda) \geq z_j \geq z_*$
(G2) $\Pr(y \in A_{\geq j} \mid |P \cap A_{\geq j-1}| \geq \gamma_0\lambda, |P \cap A_{\geq j}| \geq \gamma\lambda) \geq z_0(\gamma) \geq (1+\delta)\gamma$
(G3) $\lambda \geq \frac{2}{a} \ln\left(\frac{16m}{ac\varepsilon z_*}\right)$ *with* $a = \frac{\delta^2\gamma_0}{2(1+\delta)}$, $\varepsilon = \min\{\delta/2, 1/2\}$ *and* $c = \varepsilon^4/24$

then $\mathbf{E}[T] \leq (2/c\varepsilon)(m\lambda(1 + \ln(1 + c\lambda)) + \sum_{j=1}^{m} 1/z_j)$.

We apply the *negative drift theorem for populations* [10] to obtain tail bounds on the runtime of Algorithm 1. For any individual $P_t(i)$, where $t \in \mathbb{N}$ and $i \in [\lambda]$, define $R_t(i) := |\{j \in [\lambda] \mid I_t(j) = i\}|$, i.e., the number of times the individual was selected. We define the *reproductive rate* of the individual $P_t(i)$ to be $\mathbf{E}[R_t(i) \mid P_t]$, i.e., the expected number of offspring from individual $P_t(i)$. Informally, the theorem states that if all individuals close to a given search point $x^* \in \mathcal{X}$ have reproductive rate below a certain threshold α_0, then the algorithm needs exponential time to reach x^*. The threshold depends on the mutation rate. Here, we derive a variant of this theorem for algorithms that use multiple mutation rates. In particular, we assume that the algorithm uses m mutation rates, where mutation rate χ_i/n for $i \in [m]$ is chosen with probability q_i. The proof of this theorem is similar to that of Theorem 4 in [10], and thus omitted.

Theorem 2. *For any $x^* \in \{0,1\}^n$, define $T := \min\{t \mid x^* \in P_t\}$, where P_t is the population of Algorithm 1 at time $t \in \mathbb{N}$. If there exist constants $\alpha_0, c, c', \delta > 0$ such that with probability $1 - e^{-\Omega(n)}$*

- *the initial population satisfies $H(P_0, x^*) \geq c'n$*
- *for all $t \leq e^{cn}$ and $i \in [\lambda]$, if $H(P_t(i), x^*) \leq c'n$, then the reproductive rate of individual $P_t(i)$ is no more than α_0,*
- *$\sum_{j=1}^{m} q_j e^{-\chi_j} \leq (1-\delta)/\alpha_0$, and $\max_j \chi_j \leq \chi_{\max}$ for a constant $\chi_{\max}$,*

then $\Pr\left(T \leq e^{c''n}\right) = e^{-\Omega(n)}$ for a constant $c'' > 0$.

3 General Negative Results

Using Theorem 2, we can now show general negative results for uniform mixing and self-adaptation of two mutation rates for any function with a unique global optimum x^*, assuming that the initial population is positioned sufficiently far away from x^*. The following theorem is a special case of Theorem 2 for $|\mathcal{M}| = 1$.

Theorem 3. *The runtime of Algorithm 1 with reproductive rate α_0 and mutation rate $\chi_{high}/n \geq (\ln(\alpha_0) + \delta)/n$ for some constant $\delta > 0$ satisfies $\Pr(T \leq e^{cn}) = e^{-\Omega(n)}$ on any function with a unique global optimum x^* assuming that $\mathrm{H}(P_0, x^*) \geq c'n$ for two constants $c > 0$ and $c' \in (0, 1)$.*

For binary tournament and (μ, λ)-selection, α_0 is bounded from above by 2 and λ/μ respectively. Hence, any mutation rate above $\ln(2)$ for 2-tournament selection and $\ln(\lambda/\mu)$ for (μ, λ)-selection by a constant renders the EA inefficient.

For $|\mathcal{M}| = 2$, we have the following general result, again due to Theorem 2.

Theorem 4. *Consider Algorithm 1 with reproductive rate α_0 and mutation rates χ_{low}/n and χ_{high}/n. If there exist constants $\delta_1, \delta_2, \varepsilon > 0$ such that*

- *$\chi_{low} \geq \ln(\alpha_0) - \ln(1 + \delta_1)$ and $\chi_{high} \geq \ln(\alpha_0) - \ln(1 - \delta_2)$,*
- *the EA chooses mutation rate χ_{high} with probability at least $\frac{\delta_1(1+\varepsilon)}{\delta_1+\delta_2}$,*

then $\Pr(T \leq e^{cn}) = e^{-\Omega(n)}$ on any function with a unique optimum x^* given that $\mathrm{H}(P_0, x^*) \geq c'n$ for some constants $c', c > 0$.

Uniform mixing selects the mutation rate χ_{high}/n with probability 1/2. Thus, if $\delta_1/(\delta_1 + \delta_2)$ is below 1/2 by a constant then the EA is inefficient. For example, in binary tournament, the setting $\chi_{\text{low}} \geq \ln(3/2) - \ln(100/99)$ and $\chi_{\text{high}} \geq \ln 3 + \ln(33/32)$ satisfies the conditions for $\delta_1 = 103/297$, $\delta_2 = 105/297$ and $\delta_1/(\delta_1+\delta_2) = 103/208 < 1/2$. In contrast, Theorem 8 shows that self-adaptation is efficient in this setting. In self-adaptation, χ_{high}/n is selected with at least probability p, thus self-adaptation becomes inefficient if $p > \delta_1/(\delta_1 + \delta_2)$.

4 Robustness of Self-adaptation

The previous section showed how critically non-elitist EAs depend on having appropriate mutation rates. A slightly too high mutation rate χ_{high} can lead to an exponential increase in runtime. Uniform mixing of mutation rates can fail if the set of allowed mutation rates $\mathcal{M}$ contains one mutation rate which is too high, even though the set also contains an appropriate mutation rate χ_{low}.

Self-adaptation has a similar problem if the strategy parameter p is chosen too high. However, we will prove for a simple, unimodal fitness function that for a sufficiently small strategy parameter p, self-adaptation becomes highly robust, and is capable of fine-tuning the mutation rate. For the rest of this section, we consider a set of two mutation rates $\mathcal{M} = \{\chi_{\text{low}}, \chi_{\text{high}}\}$ which for arbitrary parameters $\ell \in [n]$ and $\varepsilon > 0$ are defined by $\left(1 - \frac{\chi_{\text{high}}}{n}\right)^{\ell} < \frac{\mu}{\lambda} \leq$

$\left(1-\frac{\chi_{\text{high}}}{n}\right)^{\ell-1}$ and $\frac{\mu}{\lambda}(1+\varepsilon) \leq \left(1-\frac{\chi_{\text{low}}}{n}\right)^{n}$. By the previous section, if ℓ is chosen sufficiently small, and hence χ_{high} sufficiently high, then uniform mixing will fail on any problem with a unique optimum. In contrast, using a Chernoff and a union bound, the following lemma shows that individuals that have chosen χ_{high} will quickly vanish from a self-adapting population, and the population will be dominated by individuals choosing the appropriate mutation parameter χ_{low}.

Lemma 1. *Let $Y_t := |P_t \cap A_{-1}|$ where P_t is the population of Algorithm 1 at time $t \in \mathbb{N}$ with (μ,λ)-selection on* LeadingOnes *and the set A_{-1} is as defined in Eq. (1). Then $\Pr\left(Y_t \geq \max((3/4)\mu, (1-p/3)^t Y_0)\right) \leq t \cdot e^{-\Omega(\lambda)}$ for all $t \in \mathbb{N}$.*

Theorem 5. *Algorithm 1 with (μ,λ)-selection where $\lambda \geq c\ln(n)$ for a sufficiently large constant $c > 0$, and self-adaptation from the set $\mathcal{M} = \{\chi_{low}, \chi_{high}\}$ using a sufficiently small constant strategy parameter p satisfying $(1+\varepsilon)(1-p) \geq 1 + p\varepsilon$ has expected runtime $O(n\lambda\log(\lambda) + n^2)$ on* LeadingOnes.

Proof. We partition the search space into the following $n+2$ levels

$$A_j := \begin{cases} \{(x,\chi_{\text{high}}) \mid \textsc{Lo}(y) \geq \ell\} & \text{if } j = -1 \\ \{(x,\chi_{\text{low}}),(x,\chi_{\text{high}}) \mid \textsc{Lo}(x) = j\} & \text{if } 0 \leq j \leq \ell-1 \\ \{(x,\chi_{\text{low}}) \mid \textsc{Lo}(x) = j\} & \text{if } \ell \leq j \leq n. \end{cases} \tag{1}$$

The special level A_{-1} contains search points with too high mutation rate. We first estimate the expected runtime assuming that there are never more than $(3/4)\mu$ individuals in level A_{-1}. In the end, we will account for the generations where this assumption does not hold.

We now show that conditions (G1) and (G2) of the level-based theorem hold for the parameters $\gamma_0 := (1/8)(\mu/\lambda)$, $\delta := p\varepsilon$, and $z_j = \Omega(1/n)$. Assume that the current population has at least $\gamma_0\lambda = \mu/8$ individuals in $A_{\geq j-1}$ and $\gamma\lambda < \gamma_0\lambda$ individuals in $A_{\geq j}$, for $0 \leq j \leq n$ and $\gamma \in [0,\gamma_0)$. If $0 \leq j \leq \ell-1$, then an individual can be produced in levels $A_{\geq j}$ if one of the $\gamma\lambda$ individuals in these levels is selected, and none of the first j bits are mutated. Assuming in the worst case that the selected individual has chosen the high mutation rate, the probability of this event is at least $(\frac{\gamma\lambda}{\mu})\left(\left(1-\frac{\chi_{\text{high}}}{n}\right)^{j}(1-p) + \left(1-\frac{\chi_{\text{low}}}{n}\right)^{j} p\right) > (\frac{\gamma\lambda}{\mu})\left(\left(1-\frac{\chi_{\text{high}}}{n}\right)^{\ell-1}(1-p) + \left(1-\frac{\chi_{\text{low}}}{n}\right)^{n} p\right) \geq \gamma(1+p\varepsilon)$. All individuals in levels $j \geq \ell$ use the low mutation rate. Hence, an individual in levels $A_{\geq j}$ can be produced by selecting one of the $\gamma\lambda$ individuals in this level, not change the mutation rate, and not flip any of the first $j \leq n$ leading 1-bits. The probability of this event is at least $\frac{\gamma\lambda}{\mu}\left(1-\frac{\chi_{\text{low}}}{n}\right)^{j}(1-p) > \frac{\gamma\lambda}{\mu}\left(\frac{\mu}{\lambda}(1+\varepsilon)(1-p)\right) \geq \gamma(1+\delta)$. Condition (G2) is therefore satisfied for all levels. For condition (G1), assume that the population does not contain any individuals in $A_{\geq j}$. Then in the worst case, it suffices to select one of the at least $\gamma_0\lambda$ individuals in level A_j, switch the mutation rate, and only flip the first 0-bit and no other bits. The probability of this event is higher than $\frac{\gamma_0\lambda}{\mu}\left(\frac{\chi_{\text{low}}}{n}\right)\left(1-\frac{\chi_{\text{high}}}{n}\right)^{n-1} p = \Omega(1/n)$.

Condition (G3) holds for any population size $\lambda \geq c\ln(n)$ and a sufficiently large constant c, because γ_0 and δ are constants. It follows that the expected

number of generations until the optimum is found is $t_1(n) = O(n\log(\lambda)+n^2/\lambda)$. By Markov's inequality, the probability that the algorithm has not found the optimum after $2t_1(n)$ generations is less than $1/2$.

Finally, we account for the generations with more than $(3/4)\mu$ individuals in level A_{-1}. We call a phase *good* if after $t_0(n) = O(\log(\lambda))$ generations and for the next $2t_1(n)$ generations, there are fewer than $(3/4)\mu$ individuals in level A_{-1}. By Lemma 1, a phase is good with probability $1-(t_0(n)+2t_1(n))\cdot e^{-\Omega(\lambda)} = \Omega(1)$, for $\lambda \geq c\ln(n)$ and c a sufficiently large constant. By the level-based analysis, the optimum is found with probability at least $1/2$ during a good phase. Hence, the expected number of phases required to find the optimum is $O(1)$. The theorem now follows by keeping in mind that each generation costs λ evaluations. □

We have shown that the EA can self-adapt to choose the low mutation parameter χ_{low} when required. Nevertheless, uniform mixing of mutation rates with a sufficiently small χ_{low} could achieve the same asymptotic performance. Furthermore, naively picking a mutation rate from the beginning also has a constant probability of optimising the function in polynomial time. Our aim is therefore to show that there exists a setting for which all the above approaches, except self-adaptation, fail. To prove this, we have identified a problem f_m where a high mutation rate is required in one part of the search space, and a low mutation rate is required in another part.

$$f_m(x) := \begin{cases} m & \text{if } x = 0^n, \text{ and} \\ \textsc{LeadingOnes}(x) & \text{otherwise.} \end{cases}$$

We call the local optimum 0^n the *peak*, and assume that all individuals in the initial population are peak individuals. The elitist $(\mu+\lambda)$ EA without any diversity mechanism will only accept a search point if it has at least m leading 1-bits.

Theorem 6. *Starting at 0^n, the $(\mu+\lambda)$ EA has expected runtime $n^{\Omega(m)}$ on f_m.*

To reach the optimal search point more efficiently, it is necessary to accept worse individuals into the population, e.g. a non-elitist selection scheme should be investigated. Since f_m has a unique global optimum, either using only a too high mutation rate or uniformly mixing a correct mutation rate with a too high one can lead to exponential runtime as discussed above. Analogously to the $(\mu+\lambda)$ EA, we also prove that using a too low mutation rate fails because the population is trapped on the peak (e. g. due to Theorem 2, individuals fell off the peak have too low reproductive rate to optimise m leading 1-bits). Subsequent proofs use the two functions $q(i) := (1-\chi_{\text{low}}/n)^i$ and $r(i) := (1-\chi_{\text{high}}/n)^i$, which are the probabilities of not flipping the first $i \in [n]$ bits using mutation rate χ_{low}/n and χ_{high}/n respectively. Clearly, $q(i)$ and $r(i)$ are monotonically decreasing in i. We also use the function $\beta(\gamma) := 2\gamma(1-\gamma/2)$, which is the probability that binary tournament selection chooses one of the $\gamma\lambda$ fittest individuals.

Theorem 7. *The runtime of Algorithm 1 on f_m with tournament size 2, initialised with the population at 0^n and fixed mutation rate $\chi \leq \ln(3/2)-\varepsilon$ for any constant $\varepsilon \in (0,\ln(3/2))$ satisfies $\Pr(T \leq e^{cn}) = e^{-\Omega(\lambda)}$ for a constant $c > 0$.*

Theorem 8. *If* $\mathcal{M} = \{\chi_{low}, \chi_{high}\}$ *where* $\chi_{low} := \ln(\frac{3}{2}) - \varepsilon$ *for any constant* $\varepsilon \in (0, \ln(\frac{100}{99}))$, *and* $\ln(3) \leq \chi_{high} = O(1)$, *then there exists an* $m \in \Theta(n)$ *such that Algorithm 1 starting with the population at* 0^n, *with tournament size* 2, *population size* $\lambda \geq c \ln n$ *for some constant* $c > 0$ *and self-adaptation of* $\mathcal{M}$ *with* $p = 1/20$ *has expected runtime* $O(n\lambda \log(\lambda) + n^2)$ *on* f_m.

Recall that uniform mixing is inefficient in this setting. Our intuition is that with sufficiently high mutation rate, some individuals fall off the peak and form a sub-population which starts optimising the LeadingOnes part of the problem. If the selective pressure is not too high, the sub-population should escape the local optimum, adapt the mutation rate, and reach the optimal search point 1^n. We used the level-based technique to infer constraints on the mutation rates and the strategy parameter p that allow this to happen. We use Lemma 2 to show that there are few individuals on the peak, or with "incorrect" mutation rates.

Lemma 2. *Given any subset* $A \subset \mathcal{X}$, *let* $Y_t := |P_t \cap A|$ *be the number of individuals in generation* $t \in \mathbb{N}$ *of Algorithm 1 with tournament size* 2, *that belong to subset* A. *If there exist three parameters* $\rho, \sigma, \varepsilon \in (0,1)$ *such that* $\Pr(p_{\text{mut}}(y) \in A) \leq \rho$ *for all* $y \in A$ *and* $\Pr(p_{\text{mut}}(y) \in A) \leq \sigma\gamma_* - \varepsilon$ *for all* $y \notin A$, *where* $\gamma_* := 2 - (1-\sigma)/\rho$, *then* $\Pr(Y_t \geq \max(\gamma_*\lambda, (1-\varepsilon/2)^t Y_0)) \leq t \cdot e^{-\Omega(\lambda)}$.

Proof (of Theorem 8). We apply the level-based theorem with respect to a partitioning of the search space $\mathcal{X} = \{0,1\}^n \times \mathcal{M}$ into the following $n+2$ levels

$$A_j := \begin{cases} \{(0^n, \chi_{\text{low}}), (0^n, \chi_{\text{high}})\} & \text{if } j = -1, \\ \{(x, \chi_{\text{low}}), (x, \chi_{\text{high}}) \mid \textsc{Lo}(x) = 0 \wedge x \neq 0^n\} & \text{if } j = 0, \\ \{(x, \chi_{\text{low}}), (x, \chi_{\text{high}}) \mid \textsc{Lo}(x) = j\} & \text{if } 1 \leq j \leq \ell - 2, \\ \{(x, \chi_{\text{low}}), (y, \chi_{\text{high}}) \mid \textsc{Lo}(x) = \ell - 1, \textsc{Lo}(y) \geq \ell - 1\} & \text{if } j = \ell - 1, \\ \{(x, \chi_{\text{low}}) \mid \textsc{Lo}(x) = j\} & \text{if } \ell \leq j \leq n. \end{cases}$$

where $\ell \in [n]$ is the unique integer such that $\left(1 - \frac{\chi_{\text{high}}}{n}\right)^\ell < \frac{85}{171} \leq \left(1 - \frac{\chi_{\text{high}}}{n}\right)^{\ell-1}$. Note that as long as $m \leq \ln(171/85)(n-1)/\chi_{\text{high}}$, we have $\left(1 - \frac{\chi_{\text{high}}}{n}\right)^m \geq (e^{-\chi_{\text{high}}})^{\frac{m}{n-1}} \geq \frac{85}{171} > \left(1 - \frac{\chi_{\text{high}}}{n}\right)^\ell$, hence $\ell > m$.

We first estimate the expected runtime assuming that every population contains less than $\psi\lambda$ individuals in A_{-1}, and less than $\xi\lambda$ individuals in the set $B := \{(y, \chi_{\text{high}}) \mid \textsc{Lo}(y) \geq \ell\}$, where $\psi := 123/250$ and $\xi := 1/5$. In the end, we will account for the generations where these assumptions do not hold. We begin by showing that condition (G2) of the level-based theorem hold for all levels.

Levels $0 \leq j \leq m$: Assume that the population contains $\gamma\lambda$ individuals in $A_{\geq j}$ for any $\gamma \in (0, \gamma_0)$. An individual in $A_{\geq j}$ will be selected if the tournament contains at least one individual in $A_{\geq j}$, and no individuals in A_{-1}. The probability of this event is $\beta(\gamma) \geq 2\gamma(1 - \gamma_0/2 - \psi)$. The mutated offspring of the selected individual will belong to levels $A_{\geq j}$ if none of the first $j \leq m$ bits are flipped, which occurs with probability at least $r(m)$. Hence, condition (G2) is satisfied if there exists a $\gamma_0 \in (0,1)$ and a constant $\delta > 0$ such that for all

$\gamma \in (0, \gamma_0]$, it holds $\beta(\gamma) r(m) \geq \gamma(1+\delta)$, i.e., it is sufficient to choose $m \in \mathbb{N}$ sufficiently small such that $r(m) = \left(1 - \frac{\chi_{\text{high}}}{n}\right)^m \geq \frac{1+\delta}{2(1-\gamma_0/2-\psi)}$. Note that such an $m = \Theta(n)$ exists, because $2(1-\gamma_0/2-\psi) = \frac{127}{125} - \gamma_0 > 1+\delta$ when γ_0 and δ are sufficiently small.

Levels $m+1 \leq j < \ell$: The probability of mutating an individual from $A_{\geq j}$ into $A_{\geq j}$, pessimistically assuming that the selected individual uses the high mutation rate χ_{high}, is at least $r(\ell-1)(1-p)+q(\ell-1)p > r(\ell-1)(1-p)+q(n)p > (85/171)(1-p) + (2/3)p = 1/2 + 1/180$. Hence, assuming that the current population has $\gamma\lambda$ individuals in $A_{\geq j}$ where $\gamma \in (0, \gamma_0)$, the probability of selecting one of these individuals and mutating them into $A_{\geq j}$ is at least $\beta(\gamma)(r(\ell-1)(1-p) + q(\ell-1)p) > 2\gamma(1-\gamma_0/2)(1/2+1/180) = \gamma(1-\gamma_0/2)(1+1/90) > \gamma(1+\delta')$ for some $\delta' > 0$ given that γ_0 is a sufficiently small constant. Note that the lower bound on $\beta(\gamma)$ here does not depend on ψ, and nor on ξ because in this setting the peak individuals have lower fitness than the individuals in A_j, and $B \subset A_{\geq j}$.

Levels $\ell \leq j \leq n$: By the level-partitioning, any individual in these levels uses the low mutation rate χ_{low}, and other individuals with at least ℓ leading 1-bits belong to the set B. Assume that the current population contains $\gamma \in (0, \gamma_0)$ individuals in $A_{\geq j}$. An individual in $A_{\geq j}$ can be produced by having a binary tournament with at least one individual from $A_{\geq j}$ and none of the at most $\xi\lambda$ individuals in B, not mutating any of the bits, and not changing the mutation rate. The probability of this event is at least $2\gamma(1-\gamma_0/2-\xi)q(n)(1-p) \geq \gamma(4/5-\gamma_0/2)(19/15) = \gamma(1+1/75-(19/30)\gamma_0) > \gamma(1+\delta')$ for some constant $\delta' > 0$, assuming that γ_0 is sufficiently small.

We now show that condition (G1) is satisfied for a parameter $z = \Omega(1/n)$ in any level j. Assume that there are at least $\gamma_0\lambda$ individuals in $A_{\geq j}$. Then, to create an individual in $A_{\geq j+1}$, it is sufficient to create a tournament of two individuals from $A_{\geq j}$, flip at most one bit, and either keep or switch the mutation rate. The probability of such an event is at least $\gamma_0^2(\chi_{\text{low}}/n)(1-\chi_{\text{high}}/n)^{n-1}p = \Omega(1/n)$.

To complete the application of the level-based theorem, we note that since δ and γ_0 are constants, condition (G3) is satisfied when $\lambda \geq c \ln n$ for some constant c. Hence, under the assumptions on the number of individuals in level A_{-1} and B described above, the level-based theorem implies that the algorithm obtains the optimum in expected $t_1(n) = O(n\log(\lambda) + n^2/\lambda)$ generations. Furthermore, by Markov's inequality, the probability that the optimum has not been found within $2t_1(n)$ generations is less than $1/2$.

To complete the proof, we justify the assumption that less than $\psi\lambda$ individuals belong to level A_{-1}, and less than $\xi\lambda$ individuals belong to B. We will show using Lemma 2 that starting with any population, these assumptions hold after an initial phase of $t_0(n) = O(\log(\lambda))$ generations. We call a phase *good* if the assumptions hold for the next $t_1(n) < e^{c\lambda}$ generations.

To apply Lemma 2 with respect to level A_{-1}, we note that the probability of obtaining an individual in A_{-1} by mutating an individual in A_{-1} is bounded from above by $q(n)(1-p) + r(n)p \leq (2/3)e^{\varepsilon}(1-p) + p/3 \leq 65/99$. Furthermore, to mutate an individual from $\mathcal{X} \setminus A_{-1}$ into A_{-1}, it is necessary to flip at least one specific bit-position, i.e., with probability $O(1/n)$. Therefore, by Lemma 2

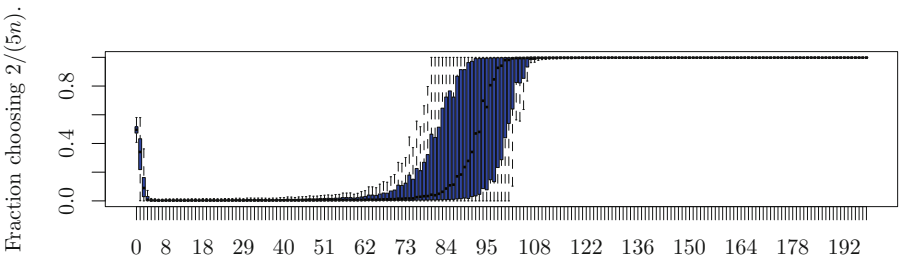
with $\sigma = 49/4950$ and $\rho = 65/99$, it holds for all t where $t_0(n) < t < e^{cn}$ and $t_0(n) = O(\log(\lambda))$ that $\Pr(|P_t \cap A_{-1}| \geq \psi\lambda) = e^{-\Omega(\lambda)}$ where $\psi := 123/250$.

Similarly, the probability of not destroying a B-individual with mutation is by definition of ℓ at most $\left(1 - \frac{\chi_{\text{high}}}{n}\right)^{\ell}(1-p) \leq \left(\frac{85}{171}\right)\left(\frac{19}{20}\right) = \frac{17}{36} =: \rho$. To create a B-individual from $\mathcal{X} \setminus B$, it is in the best case necessary to change the mutation rate from χ_{low} to χ_{high} and not mutate the first ℓ bit-positions. The probability of this event is $\left(1 - \frac{\chi_{\text{high}}}{n}\right)^{\ell} p \leq \left(\frac{85}{171}\right)\left(\frac{1}{20}\right) = \frac{17}{684}$. Therefore, by Lemma 2 wrt $\sigma := 3/20$ and the above value of ρ, for every generation t where $t_0(n) < t < e^{c\lambda}$ and $t_0(n) = O(\log(\lambda))$ it holds $\Pr(|P_t \cap B| \geq \xi\lambda) = e^{-\Omega(\lambda)}$, where $\xi := 1/5$.

To summarise, starting from any configuration of the population, a phase of length $t_0(n) + 2t_1(n) = O(n\log(\lambda) + n^2/\lambda)$ generations is *good* with probability $1 - e^{-\Omega(\lambda)}$. If a phase is good, then the optimum will be found by the end of that phase with probability at least $1/2$. Hence, the expected number of phases required to find the optimum is $O(1)$, and the theorem follows, keeping in mind that each generation costs λ function evaluations. □

5 Experiments

Below are results from 1000 experiments with the self-adaptive EA on the LeadingOnes function for $n = 200$, $p = 1/1000$ using (μ, λ)-selection for $\mu = 500$, $\lambda = 4\mu$, and mutation parameters $\mathcal{M} = \{2/5, 2\}$. For each $j \in [n]$, the figure contains a box-plot describing the distribution of the fraction of the population choosing χ_{low} over all generations where the (1/10)-ranked individual in the population has j leading one-bits.



The initial population, including mutation rates, are sampled uniformly at random. Hence the (1/10)-ranked individual will have fitness close to 1 in the first generations. For $j \leq 5$, i. e. early in the run, approximately half of the population chooses the low mutation. However, the population quickly switches to the higher mutation χ_{high} until the (1/10)-ranked individual in the population reaches a value approximately $j \geq 60$ where the population switches to the lower mutation χ_{low}. Almost all individuals choose χ_{low} for $j \geq 108$. These experimental results confirm that the population adapts the mutation rate according to the region of the fitness landscape currently searched.

6 Conclusion

In this first runtime analysis of self-adaptation, we have shown that self-adaptation with a sufficiently low strategy parameter can robustly control mutation-rates in non-elitist EAs, and that this automated control can lead to exponential speedups compared to EAs that use fixed mutation rates, or uniform mixing of mutation rates. The results were obtained via level-based analysis, further demonstrating the strength of this technique in handling complex population dynamics.

Acknowledgements. This work received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 618091 (SAGE).

References

1. Bäck, T.: Self-adaptation in genetic algorithms. In: Proceedings of ECAL 1992, pp. 263–271 (1992)
2. Böttcher, S., Doerr, B., Neumann, F.: Optimal fixed and adaptive mutation rates for the leadingones problem. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 1–10. Springer, Heidelberg (2010)
3. Corus, D., Dang, D.-C., Eremeev, A.V., Lehre, P.K.: Level-based analysis of genetic algorithms and other search processes. In: Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J. (eds.) PPSN 2014. LNCS, vol. 8672, pp. 912–921. Springer, Heidelberg (2014)
4. Dang, D.-C., Lehre, P.K.: Refined upper bounds on the expected runtime of non-elitist populations from fitness-levels. In: Proceedings of GECCO 2014, pp. 1367–1374 (2014)
5. Doerr, B., Doerr, C.: Optimal parameter choices through self-adjustment: applying the 1/5-th rule in discrete settings. In: Proceedings of GECCO 2015, pp. 1335–1342 (2015)
6. Doerr, B., Doerr, C., Kötzing, T.: Solving problems with unknown solution length at (almost) no extra cost. In: Proceedings of GECCO 2015, pp. 831–838 (2015)
7. Eiben, A.E., Michalewicz, Z., Schoenauer, M., Smith, J.E.: Parameter control in evolutionary algorithms. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms. SCI, vol. 54, pp. 19–46. Springer, Heidelberg (2007)
8. Gerrish, P.J., Colato, A., Perelson, A.S., Sniegowski, P.D.: Complete genetic linkage can subvert natural selection. PNAS **104**(15), 6266–6271 (2007)
9. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001)
10. Lehre, P.K.: Negative drift in populations. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 244–253. Springer, Heidelberg (2010)
11. Lehre, P.K., Özcan, E.: A runtime analysis of simple hyper-heuristics: to mix or not to mixoperators. In: Proceedings of FOGA 2013, pp. 97–104 (2013)
12. Lehre, P.K., Yao, X.: On the impact of mutation-selection balance on the runtime of evolutionary algorithms. IEEE Trans. Evol. Comput. **16**(2), 225–241 (2012)

13. van Rijn, S., Emmerich, M.T.M., Reehuis, E., Bäck, T.: Optimizing highly constrained truck loadings using a self-adaptive genetic algorithm. In: Proceedings of CEC 2015, pp. 227–234 (2015)
14. Xue, J.Z., Kaznatcheev, A., Costopoulos, A., Guichard, F.: Fidelity drive: a mechanism for chaperone proteins to maintain stable mutation rates in prokaryotes over evolutionary time. J. Theor. Biol. **364**, 162–167 (2015)