# Analysis of the (1+1) EA for a Noisy OneMax

Stefan Droste⋆

LS Informatik 2, Universität Dortmund, 44221 Dortmund, Germany
stefan.droste@udo.edu

**Abstract.** In practical applications evaluating a fitness function is frequently subject to noise, i.e., the "true fitness" is disturbed by some random variations. Evolutionary algorithms (EAs) are often successfully applied to noisy problems, where they have turned out to be particularly robust. Theoretical results on the behavior of EAs for noisy functions are comparatively very rare, especially for discrete search spaces. Here we present an analysis of the (1+1) EA for a noisy variant of OneMax and compute the maximal noise strength allowing the (1+1) EA a polynomial runtime asymptotically exactly. The methods used in the proofs are presented in a general form with clearly stated conditions in order to simplify further applications.

## 1   Introduction

When trying to optimize problems in practice, it is rarely possible to determine the fitness value of a search point exactly. Most often noise changes the fitness value by some (typically small) amount because of interferences during the experiment resulting in the fitness value. Evolutionary algorithms (EAs) are often good at coping with noise due to the use of a population of search points. But there are only few theoretical results about the effects of noise on the performance of EAs for discrete search spaces, while continuous search spaces found more attention (see [AB03] for an overview).

In the last years the rigorous runtime analysis of EAs (e.g. see [DJW02], [JW99] or [WW03]) has proven to be a useful approach for extending our knowledge about EAs, besides other well-known approaches like schema theory or macroscopic, statistical, fitness landscape resp. local analysis. Here we extend this approach to the optimization of noisy functions by EAs. Since this is the first rigorous runtime analysis for noisy functions, we start with a simple EA and fitness function: the (1+1) EA for a noisy variant of OneMax. The (1+1) EA uses only one individual, changed by mutation only, and has been analyzed for a number of different fitness functions (e.g. see [Rud97] or [DJW02]), greatly extending the knowledge about the random processes underlying the (1+1) EA and appropriate proof techniques. Note that all these fitness functions are static and noise-free (just recently, dynamic variants of OneMax have been analyzed, see [Dro02] and [Dro03]), which is not very common in practical applications.

---

In this paper we analyze the influence of the noise strength on the runtime of the (1+1) EA until the optimum of ONEMAX is found. We show that noise, which changes one uniformly chosen bit before evaluation with probability $p$, makes an efficient optimization of the (1+1) EA impossible if and only if $p$ grows asymptotically faster than $\log(n)/n$ (where $n$ is the dimension of the search space). This is especially interesting since $\log(n)/n$ is also the critical value for the (1+1) EA on a dynamic ONEMAX where in each step with probability $p$ one uniformly chosen bit of the target bit string changes. Although the two processes have no obvious interpretation in terms of each other, we can generalize the methods used to analyze the dynamic ONEMAX (see [Dro02]) to noisy ONEMAX.

In the next section, we formally define the (1+1) EA, our noise model, and the runtime. Section 3 presents the methods used in Section 4 to analyze the runtime of the (1+1) EA for noisy ONEMAX. This separation should increase comprehensibility, because the methods in Section 3 clearly state the conditions necessary for applying them, while Section 4 shows that the (1+1) EA on noisy ONEMAX fulfills these conditions. We finish with some conclusions.

## 2    The (1+1) EA and Noisy ONEMAX

In this section, we formally define the (1+1) EA and the noise model investigated. The (1+1) EA is the most basic EA since it uses only one individual changed by mutation only. This simplicity makes it an ideal starting point for theoretical analyzes (see [DJW02]). Furthermore, the (1+1) EA has proven to be surprisingly efficient for some functions compared to more involved EAs (see [FM92]), making its analysis worthwhile in itself.

The (1+1) EA for noisy functions differs from the (1+1) EA for noise-free evaluation only in one point: since the fitness-value of the parental search point evaluated in a generation before is not guaranteed to be the correct one, *both* the parent and the child are evaluated in each generation. Without this resampling a noisy evaluation can only be corrected if a copy of the search point is evaluated (e. g. if the mutation has no effect). Let $f_N : \{0,1\}^n \to \mathbb{R}$ denote the noisy function. Then the (1+1) EA looks as follows:

**Definition 1 ((1+1) EA for maximization of noisy functions).**

1. Set $t := 0$ and choose $x_t \in \{0,1\}^n$ randomly uniformly.
2. Set $x' := x_t$ and independently flip each bit of $x'$ with prob. $1/n$.
3. If $f_N(x') \geq f_N(x_t)$, set $x_{t+1} := x'$, else $x_{t+1} := x_t$.
4. Set $t := t + 1$ and go to step 2.

In the following we will look at the case that $f_N$ results from the *true fitness function* $f : \{0,1\}^n \to \{0,1\}$ by randomly changing the argument, i. e. $f_N(x) = f(N(x))$, where the random function $N : \{0,1\}^n \to \{0,1\}^n$ represents the noise. Hence, our noise model considers noise which takes effect before evaluation. We assume that the noisy evaluations are independent, i. e., the result of previous evaluations does not influence the actual evaluation.

We are interested in the number of generations the (1+1) EA needs to evaluate a maximum of the true fitness function $f$ for the first time. Since there are two fitness evaluations per generation, the number of generations uniquely determines the number of fitness evaluations, which is often the most costly operation in practice. Considering an optimum of the true fitness function as an optimum of the noisy function seems to be a more natural choice than investigating the number of generations until a search point with maximal noisy fitness value is evaluated, since such a point can be "far away" from a true maximum. Hence, we analyze the number $T_{f,N}$ of generations the (1+1) EA needs to find a maximum of the underlying true fitness function $f$ under noise $N$:

**Definition 2 (Runtime of the (1+1) EA for noisy functions).** *The run-time $T_{f,N}$ of the (1+1) EA for a noisy function $f_N$ is the number of generations until a maximum of the true fitness function $f$ is found:*

$$T_{f,N} := \min\{t \in \mathbb{N}_0 \mid f(x_t) = \max\{f(x) \mid x \in \{0,1\}^n\}\}.$$

We analyze the noise model $N_p^1$ changing with probability $p$ exactly one uniformly chosen bit of $x$, while no bit is changed with probability $1 - p$. Such a model can be appropriate for applications where the genotype-phenotype mapping is error-prone, but the evaluation of the phenotype exact.

**Definition 3 (One-bit noise $N_p^1$).** *Let $p \in [0,1]$. The random noise function $N_p^1 : \{0,1\}^n \to \{0,1\}^n$ is defined as follows:*

$$\forall x,y \in \{0,1\}^n : P(N_p^1(x) = y) = \begin{cases} 1 - p & \text{if } x = y, \\ p/n & \text{if } H(x,y) = 1, \\ 0 & \text{if } H(x,y) > 2. \end{cases}$$

We say that noisy evaluation increases resp. decreases the fitness of a point $x \in \{0,1\}^n$ if $f_N(x) > f(x)$ resp. $f_N(x) < f(x)$. For $f = \text{ONEMAX}$ (with $\text{ONEMAX}(x_1,\ldots,x_n) := x_1 + \cdots + x_n$) the noise function $N_p^1$ can only change the fitness of a point by one, and the probability of $N_p^1$ increasing resp. decreasing the fitness of $x$ is directly proportional to $n - \text{ONEMAX}(x)$ resp. $\text{ONEMAX}(x)$.

Our goal is to determine the critical noise strength $p = p(n)$ such that the runtime of the (1+1) EA is polynomial in $n$ with high probability if the noise is at most $p$, but super-polynomial with high probability if the noise is asymptotically larger than $p$. Although only one bit of the search point to be evaluated is changed by $N_p^1$, this can make the (1+1) EA accept a child $x'$ whose true fitness $\text{ONEMAX}(x')$ is by two smaller than the true fitness $\text{ONEMAX}(x_t)$ of the parent $x_t$ (see Section 4). This differs from the (1+1) EA for a dynamically changing ONEMAX, where the target bit string is changed by the operator $N_p^1$ and the Hamming distance to the target is to be minimized (see [Dro02] for an analysis that the expected runtime is polynomial if and only if $p = O(\log(n)/n)$). Although in the latter process the distance to the optimum can only increase by one in one step, our techniques can also cope with the first process and show that in this case $p = O(\log(n)/n)$ is the critical noise strength, too.

To distinguish more clearly between the key properties of the (1+1) EA on noisy ONEMAX resulting in polynomial resp. super-polynomial runtime and the necessary technical proofs, we show some general methods for proving polynomial upper and super-polynomial lower bounds in the next section.

# 3   Techniques for Bounding Markov Chains

In this section, we present some general results that can be applied to analyze the (1+1) EA for the noisy ONEMAX presented in the last section. They are presented in a general form with clearly stated conditions in order to make them easily applicable to other problems. Furthermore, this should make their proofs more comprehensible, since they need not to take care of the specific details of the EA analyzed as long as the EA fulfills the general conditions.

The (1+1) EA on noisy ONEMAX is a Markov process on the state space $\{0, \ldots, n\}$, where state $i$ represents that the actual individual $x_t$ has true fitness $\text{ONEMAX}(x_t) = i$. Hence, we identify the process by $(n+1)^2$ transition probabilities $p_{\cdot,\cdot} = (p_{i,j})_{i,j\in\{0,\ldots,n\}}$, where $p_{i,j} \in [0,1]$ is the probability of moving from state $i$ to state $j$ in one step. Let $T_{i,j}$ be the number of steps to come from state $i$ to $j$ for the first time, i.e. $T_{i,n}$ is the runtime when starting in state $i$.

## 3.1   Upper Bound Techniques

Our first result, implicitly already used in [Dro02], upper bounds the runtime of a process $p_{\cdot,\cdot}$ that can only move from $i$ to $i-1$, $i$ or $i+1$ in one step (a so-called $\{-1, 0, +1\}$-process) by a polynomial:

**Lemma 1.** *Let $p_{\cdot,\cdot}$ be a $\{-1, 0, +1\}$-process on $\{0, \ldots, n\}$. If there are two constants $c^+, c^- \in \mathbb{R}^+$ such that*

$$\forall i \in \{0, \ldots, n\}:\ p_{i,i+1} \geq c^+ \cdot \frac{n-i}{n} \ \text{ and } \ p_{i,i-1} \leq c^- \cdot \frac{\log(n)}{n} \cdot \frac{i}{n},$$

*then for all $i \in \{0, \ldots, n-1\}$*

$$E(T_{i,i+1}) \leq \frac{n^{1+c^-/(c^+\cdot\ln(2))}}{c^+ \cdot (n-i)} \ \text{ and } \ E(T_{0,n}) = O\left(n^{1+c^-/(c^+\cdot\ln(2))} \cdot \log(n)\right).$$

*Proof.* It is well known (e.g. see [DJW00]) that $E(T_{i,i+1})$ for $\{-1, 0, +1\}$-processes is $\sum_{k=0}^{i} \frac{1}{p_{k,k+1}} \cdot \prod_{l=k+1}^{i} \frac{p_{l,l-1}}{p_{l,l+1}}$. Utilizing the bounds for $p_{i,i-1}$ and $p_{i,i+1}$, we get:

$$E(T_{i,i+1}) \leq \sum_{k=0}^{i} \frac{n}{c^+(n-k)} \cdot \prod_{l=k+1}^{i} \frac{n}{c^+(n-l)} \cdot \frac{c^- \cdot \log(n) \cdot l}{n^2}$$

$$= \sum_{k=0}^{i} \frac{n}{c^+(n-k)} \cdot \left(\frac{c^-}{c^+} \frac{\log(n)}{n}\right)^{i-k} \cdot \frac{i!}{k!} \cdot \frac{(n-i-1)!}{(n-k-1)!}$$

$$= \frac{n}{c^+} \cdot \sum_{k=0}^{i} \left(\frac{c^-}{c^+} \frac{\log(n)}{n}\right)^{i-k} \frac{\binom{i}{k}}{\binom{n-k}{i-k}(n-i)} \leq \frac{n}{c^+(n-i)}\left(1 + \frac{c^-}{c^+}\frac{\log(n)}{n}\right)^i$$

$$\leq \frac{n}{c^+(n-i)} \cdot \exp\left(\frac{c^-}{c^+}\frac{\log(n)}{n} \cdot i\right) \leq \frac{n^{1+c^-/(c^+\cdot\ln(2))}}{c^+(n-i)}.$$

Summing up these values for all $i \in \{0, \ldots, n-1\}$ gives the desired upper bound on $E(T_{0,n})$ because $\sum_{i=1}^{n} 1/i$ is $O(\log(n))$. □

To upper bound a Markov process $p_{.,.}$, which is no $\{-1, 0, +1\}$-process (i. e. replacing it by a process whose finite runtime to come from $i$ to $n$ stochastically dominates the runtime $T_{i,n}$ of the old process for all $i < n$) , we can proceed as follows: first, we "delete" all improvements by more than one, i. e., we set all probabilities $p_{i,i+d}$ for $d \geq 2$ to zero and increase $p_{i,i}$ by $\sum_{d=2}^{n-i} p_{i,i+d}$. This leads to a process whose runtime stochastically dominates the runtime of the old process (see [Dro03]), where a random variable $X$ stochastically dominates a random variable $Y$ if for all values $d$ of $X$ and $Y$: $P(X \geq d) \geq P(Y \geq d)$.

Afterwards, we have a process that can move from state $i$ only to states $0, \ldots, i+1$, a so-called $\leq 1$-process. To upper bound such a $\leq 1$-process $p_{.,.}$ by a $\{-1, 0, 1\}$-process $\tilde{p}_{.,.}$, we can use the following lemma proven in [Dro03]:

**Lemma 2.** *Let $(p_{.,.})$ be a $\leq 1$-process and $(\tilde{p}_{.,.})$ be a $\{-1, 0, 1\}$-process on the state space $\{0, \ldots, n\}$. If the following conditions hold*

1. $\forall i \in \{1, \ldots, n-1\} : p_{i,0} \leq \prod_{k=1}^{i} \tilde{p}_{k,k-1}$,
2. $\forall i \in \{1, \ldots, n-1\} : \forall j \in \{1, \ldots, i-1\} : p_{i,j} \leq \tilde{p}_{j,j} \prod_{k=j+1}^{i} \tilde{p}_{k,k-1}$,
3. $\forall i \in \{0, \ldots, n-1\} : p_{i,i+1} \geq \tilde{p}_{i,i+1}$,

*then $E(T_{i,n}) \leq E(\tilde{T}_{i,n})$ for all $i \in \{0, \ldots, n\}$.*

Now we can use Lemma 1 to upper bound the expected runtime of this slower $\{-1, 0, +1\}$-process $\tilde{p}_{.,.}$ by a polynomial if the transition probabilities fulfill the conditions. We will see in Section 4 that the process resulting from the (1+1) EA on noisy ONEMAX for small noise strength $p = O(\log(n)/n)$ has this form.

## 3.2   Lower Bound Techniques

If the probabilities $p_{i,i-1}$ of the process $p_{.,.}$ are only a little bit larger than necessary for Lemma 1, i. e. by a non-constant factor $\alpha(n)$, the expected runtime of the process is super-polynomially. To be more exact, the runtime is polynomial only with super-polynomially small probability $o(1/poly(n))$, i. e. smaller than $1/q(n)$ for any polynomial $q$ (see also [Dro02] and [Dro03]):

**Lemma 3.** *Let $p_{.,.}$ be a Markov process on $\{0, \ldots, n\}$. If for a function $\alpha \colon \mathbb{N}^{+} \to \mathbb{R}$ with $\alpha(n) \overset{n \to \infty}{\longrightarrow} \infty$ and $\alpha(n) \leq n/\log(n)$ the conditions*

1. $\forall i \geq 0, \ d = \omega(\log(n)) \colon p_{i,i+d} = o(1/poly(n))$,
2. $\forall i \geq n - \alpha(n)\log(n), \ d = \omega(1) \colon p_{i,i+d} = o(1/poly(n))$, *and*
3. $\exists c^{+}, c^{-} \in \mathbb{R}^{+} \colon \forall i \geq n - \alpha(n)\log(n) :$

$$\sum_{j=1}^{n-i} p_{i,i+j} \leq c^{+} \cdot \frac{n-i}{n} \quad and \quad p_{i,i-1} \geq c^{-} \cdot \alpha(n) \cdot \frac{\log(n)}{n} \cdot \frac{i}{n}$$

*hold then for all $i \leq n - \alpha(n)\log(n)$ the probability that $T_{i,n}$ is polynomial is super-polynomially small, i. e. $o(1/poly(n))$.*

*Proof.* Let $I_t \in \{0, \ldots, n\}$ be the state of the process at time step $t$. Since for all $i$ the probability of a direct step from $i$ to $i + d$ with $d = \omega(\log(n))$ is super-polynomially small, we assume that only improving steps by $O(\log(n))$ happen. Hence, regardless of the initial state $i \leq n - \alpha(n)\log(n)$, the process reaches state $n$ only via a state between $n - \alpha(n)\log(n)$ and $n - \alpha(n)\log(n)/2$. Since a constant factor does not matter for our definition of $\alpha(n)$, for an arbitrary constant $a \in \,]0, 1[$ there have to be time steps $t_1 < t_2$, such that $I_t \in \{n - \alpha(n)^a\log(n), \ldots, n-1\}$ for all $t \in \{t_1, \ldots, t_2-1\}$, and $I_{t_2} = n$ (a similar technique is used in [RRS95]). We show that it is super-polynomially unlikely to "bridge the gap" from state $n - \alpha(n)^a\log(n)$ to $n$ in polynomially many steps. Hence, we can assume that $i \geq n - \alpha(n)^a\log(n)$ and that no improvements by more than $\alpha(n)^d$ for a constant $d \in \,]0, 1[$ do happen.

Since $(n - \alpha(n)^a\log(n))/n$ converges to 1, we bound $p_{i,i-1}$ and $\sum_{d=1}^{n-i} p_{i,i+d}$ for all $n$ larger than a constant $n_0$, i.e. $n$ *large enough*:

$$p_{i,i-1} \geq \frac{c^-}{2} \cdot \alpha(n)\frac{\log(n)}{n} \text{ and } \sum_{d=1}^{n-i} p_{i,i+d} \leq c^+ \cdot \alpha(n)^a \frac{\log(n)}{n}.$$

Therefore, the probability of an improving step under the condition that the step changes the state (i.e. the step is *effective*) is at most

$$\frac{c^+\alpha(n)^a\log(n)/n}{c^+\alpha(n)^a\log(n)/n + c^-\alpha(n)\log(n)/(2n)} \leq \frac{2c^+}{c^-} \cdot \frac{1}{\alpha(n)^{1-a}}.$$

Hence, the expected number of improving steps during $t$ effective steps is at most $(2c^+/c^-) \cdot t/\alpha(n)^{1-a}$. However, to bridge the gap of size $\alpha(n)^a\log(n)$ in $t$ effective steps consisting of $t^+$ improving and $t^-$ decreasing steps, we must have

$$\alpha(n)^d \cdot t^+ - t^- \geq \alpha(n)^a \cdot \log(n) \iff t^+ \geq \frac{\alpha(n)^a\log(n) + t}{\alpha(n)^d + 1}.$$

The last term is at least $t/\alpha(n)^{c+d}$ for every constant $c > 0$ and $n$ large enough. If we choose $a$, $c$, and $d$ with $c + d < 1 - a$, the number of improving steps necessary to reach state $n$ is by a non-constant factor larger than its expected number. For $n$ large enough, this factor is at least 2.

By Chernoff bounds (see [MR95]) the probability that the sum of binary random variables is by a constant factor $1 + \delta$ larger than its expected value $\mu$ is upper bounded by (we use $\delta = 1$ and $\mu = (2c^+/c^-)\alpha(n)^{a-1}t$)

$$\left(\frac{\exp(\delta)}{(1+\delta)^{1+\delta}}\right)^\mu = \left(\frac{\exp(1)}{4}\right)^{(2c^+/c^-)\alpha(n)^{a-1}t}.$$

If $t$ is small, this bound is not super-polynomially small. But since the process must bridge a gap of size $\alpha(n)^a\log(n)$ and any improvement by more than $\alpha(n)^d$ is excluded, the number $t$ of effective steps must be at least $\alpha(n)^{a-d}\log(n)$. Hence, the above bound is at most $(\exp(1)/4)^{(2c^+/c^-)\alpha(n)^{2a-d-1}\log(n)}$, which is super-polynomially small if $2a - d - 1 > 0$. Since $a = 2/3$, $c = 1/7$, and $d = 1/7$ fulfill $2a - d - 1 > 0$ and $c + d < 1 - a$, the probability of every phase of polynomial length reaching the optimum is super-polynomially small. $\square$

If the noise strength is too large, this result cannot be applied to lower bound the (1+1) EA on noisy ONEMAX. In this case a step decreasing the state is by a constant factor more likely than a step increasing it if the process is close to the optimal state $n$. If furthermore an improvement by $d$ is by a non-constant factor more likely than an improvement by $d + 1$, we can apply the following result:

**Lemma 4.** *Let $p_{.,.}$ be a Markov process on $\{0, \ldots, n\}$ such that*

1. $\forall i \geq 0, \ d = \omega(\log(n))\colon p_{i,i+d} = o(1/poly(n))$,
2. $\exists \delta, \varepsilon, \gamma > 0\colon \forall i \geq n - n^\delta, d \in \{1, \ldots, n - i\}\colon$

$$\sum_{j=1}^{i} p_{i,i-j} \geq (1 + \varepsilon) \cdot \sum_{j=1}^{n-i} p_{i,i+j} \quad and \quad \frac{p_{i,i+d}}{p_{i,i+d+1}} \geq n^\gamma.$$

*Then for all $i \leq \lfloor n - n^\delta \rfloor$ the probability of $T_{i,n}$ being polynomially is super-polynomially small.*

*Proof.* Analogously to the proof of Lemma 3, we can assume that to reach the target state $n$, there has to be a phase ranging from time steps $t_1$ to $t_2$, such that $I_{t_2} = n$ and for all $t \in \{t_1, \ldots, t_2 - 1\}$ we have $I_t \in \{\lfloor n - n^\delta \rfloor, \ldots, n-1\}$ . We show that every phase of polynomial length is super-polynomially unlikely.

Let $X_t \in \{-(\lfloor n - n^\delta \rfloor), \ldots, \lceil n^\delta \rceil\}$ be the change of state in time step $t$, i. e. $P(X_t = d) = p_{i,i+d}$ if the process is in state $i$ at time step $t$. The phase can only be successful if $X_{t_1} + \cdots + X_{t_2} = n^\delta$, which will be shown to be exponentially unlikely. Since $X_t < 0$ is at least by a factor $1 + \varepsilon$ more likely than $X_t > 0$, we make the process only faster if we assume that $P(X_t < 0) = (1 + \varepsilon) \cdot P(X_t > 0)$ and that every step decreasing the number of ones decreases it by exactly one.

Since $p_{i,i+d}/p_{i,i+d+1}$ is at least $n^\gamma$ for $d > 0$, we replace the step size $d$ of an improving step by the value of a geometrically distributed random variable $Y$ with success probability $1 - n^{-\gamma}$: this leads to a stochastically dominating process because $P(Y = d)/P(Y = d+1)$ is exactly $n^\gamma$. All in all, we have replaced every step of $p_{.,.}$ in the considered phase by the value of a $\{-1, \ldots, n^\gamma\}$-valued random variable $X'$ defined by

$$P(X' = d) = \begin{cases} \frac{1+\varepsilon}{2+\varepsilon} & \text{if } d = -1 \\ \frac{1}{2+\varepsilon} \cdot (n^{-\gamma})^{d-1} \cdot (1 - n^{-\gamma}) & \text{if } d \geq 1 \end{cases}$$

(Note that $X' < 0$ is by a factor $1 + \varepsilon$ more likely than $X' > 0$.) Hence, the random variable $S_{t(n)} = X'_1 + \cdots + X'_{t(n)}$ stochastically dominates the improvement of the process $p_{.,.}$ during a phase of length $t(n)$ starting from a state at least $\lfloor n - n^\delta \rfloor$. As the expected value of $Y$ is $1/(1 - n^{-\gamma})$, for $n$ large enough we have

$$E(X') = -\frac{1+\varepsilon}{2+\varepsilon} + \frac{1}{2+\varepsilon} \cdot \frac{1}{1 - n^{-\gamma}} = -\frac{1+\varepsilon}{2+\varepsilon} + \frac{1}{2+\varepsilon} \cdot \frac{n^\gamma}{n^\gamma - 1} \leq -\frac{\varepsilon/2}{2+\varepsilon}.$$

Hence, the expected value of $S_{t(n)}$ is for $n$ large enough and a constant $\varepsilon' > 0$ at most $-t(n)\varepsilon'$. In order for a phase of length $t(n)$ to be successful $S_{t(n)}$ has to be at least $n^\gamma$. We show that even $P(S_{t(n)} > 0)$ is super-polynomially unlikely

for all possible phase lengths $t(n)$. To estimate $P(S_{t(n)} > 0)$ we cannot use Chernoff bounds, since $S_{t(n)}$ is no sum of $\{0, 1\}$-valued, but $\{-1, \ldots, n^\gamma\}$-valued random variables. In this situation we use Hoeffding's inequality ([Hoe63]), a generalization of Chernoff bounds, stating in our notation

$$P\big(S_{t(n)} - E(S_{t(n)})\big) \geq t(n)\varepsilon\big) \leq \exp\left(\frac{-2t(n)\varepsilon^2}{z^2}\right)$$

where $z$ is the number of different values of $X_i'$. Because of Condition 1 we can assume that no improvements by more than $\log(n)^2 - 2$ happen implying $z = \log(n)^2$ and $t(n) \geq n^\gamma / \log(n)^2$. Therefore, Hoeffding's inequality gives us an exponentially small upper bound on the probability, that a phase of polynomial length is successful. $\qquad\square$

## 4    Application of the Techniques for the (1+1) EA on Noisy ONEMAX

Now we apply the techniques presented in the previous section to determine the maximal noise strength $p$ under which the (1+1) EA is still able to optimize ONEMAX with single-bit noise in expected polynomial time. Hence, let $Mut_i \in \{-i, \ldots, n-i\}$ be the random variable denoting the change of the number of ones after a mutation of a search point $x$ with $\text{ONEMAX}(x) = i$. Depending on the value $d \neq 0$ of $Mut_i$ we determine if and when this mutation leads to a change of the number of ones of the actual individual by $d$:

- If $Mut_i = d$ with $d \geq 2$, the mutation will be accepted regardless of the noisy evaluations. Hence, for all $i \in \{0, \ldots, n-2\}$ and $d \in \{2, \ldots, n-i\}$:

$$p_{i,i+d} = P(Mut_i = d).$$

- If $Mut_i = 1$, the mutation will be accepted except in case that the noisy evaluation of the parent increases its value while the noisy evaluation of the child decreases its value. Hence, for all $i \in \{0, \ldots, n-1\}$:

$$p_{i,i+1} = P(Mut_i = 1) \cdot \left(1 - p\frac{n-i}{n} \cdot p\frac{i+1}{n}\right). \tag{1}$$

- If $Mut_i = -1$, the mutation will be accepted if and only if one of the three following cases happens:
  - the noisy evaluation of the parent decreases its value and the evaluation of the child is not disturbed by noise,
  - the noisy evaluation of the parent decreases its value and the noisy evaluation of the child increases it value, or
  - the evaluation of the parent is not changed by noise and the noisy evaluation of the child increases its value.

  Hence, for all $i \in \{1, \ldots, n\}$:

$$p_{i,i-1} = P(Mut_i = -1) \cdot \left(p\frac{i}{n}\left(1 - p + p\frac{n-i+1}{n}\right) + (1-p)p\frac{n-i+1}{n}\right). \tag{2}$$

– If $Mut_i = -2$, the mutation will be accepted only if the noisy evaluation of the parent decreases its value and the noisy evaluation of the child increases its value. Hence, for all $i \in \{2, \ldots, n\}$:

$$p_{i,i-2} = P(Mut_i = -2) \cdot \left(p\frac{i}{n}p\frac{n-i+2}{n}\right). \tag{3}$$

As every mutation decreasing the number of ones by at least two is not accepted, $p_{i,i-d} = 0$ for all $d \geq 2$.

It is obvious that the probability $P(Mut_i = d)$ is essential when analyzing the (1+1) EA on noisy ONEMAX. We can easily show, that this probability is $O(((n-i)/n)^d)$ for $d > 0$ and $O((i/n)^d)$ for $d < 0$:

$$d > 0 : P(Mut_i = d) = \sum_{k=d}^{n-i} \binom{n-i}{k} \cdot \binom{i}{k-d} \cdot \left(\frac{1}{n}\right)^{2k-d} \left(1 - \frac{1}{n}\right)^{n-2k+d}$$

$$\leq \left(\frac{n-i}{n}\right)^d \left(1 - \frac{1}{n}\right)^{n-d} \frac{1}{d!} + \sum_{k=d+1}^{n-i} \left(\frac{n-i}{n}\right)^k \frac{1}{k!}. \tag{4}$$

$$d < 0 : P(Mut_i = d) = \sum_{k=d}^{i} \binom{i}{k} \cdot \binom{n-i}{k-d} \cdot \left(\frac{1}{n}\right)^{2k-d} \left(1 - \frac{1}{n}\right)^{n-2k+d}$$

$$\leq \left(\frac{i}{n}\right)^d \cdot \left(1 - \frac{1}{n}\right)^{n-d} \cdot \frac{1}{d!} + \sum_{k=d+1}^{i} \left(\frac{i}{n}\right)^k \cdot \frac{1}{k!}. \tag{5}$$

These bounds are asymptotically tight as long as $d$ is a constant:

$$d > 0 : P(Mut_i = d) = \sum_{k=d}^{n-i} \binom{n-i}{k} \cdot \binom{i}{k-d} \cdot \left(\frac{1}{n}\right)^{2k-d} \left(1 - \frac{1}{n}\right)^{n-2k+d}$$

$$\geq \binom{n-i}{d} \left(\frac{1}{n}\right)^d \cdot \exp(-1) \geq \left(\frac{n-i}{n}\right)^d \cdot \frac{\exp(-1)}{d^d}. \tag{6}$$

$$d < 0 : P(Mut_i = d) = \sum_{k=d}^{i} \binom{i}{k} \cdot \binom{n-i}{k-d} \cdot \left(\frac{1}{n}\right)^{2k-d} \left(1 - \frac{1}{n}\right)^{n-2k+d}$$

$$\geq \binom{i}{d} \left(\frac{1}{n}\right)^d \cdot \exp(-1) \geq \left(\frac{i}{n}\right)^d \cdot \frac{\exp(-1)}{d^d}. \tag{7}$$

### 4.1    A Polynomial Upper Bound for $p = O(\log(n)/n)$

First, we want to upper bound the expected runtime of the (1+1) EA on noisy ONEMAX for $p = O(\log(n)/n)$. Therefore, we "delete" all transitions going from $i$ to $i+d$ with $d \geq 2$ (making the process only slower) and bound the remaining transition probabilities in the following way for $n$ large enough:

$$p_{i,i+1} \overset{(1),(6)}{\geq} \frac{n-i}{n} \cdot \exp(-1) \cdot \left(1 - p^2 \frac{(n-i)(i+1)}{n^2}\right) \geq \frac{n-i}{n} \cdot \frac{3\exp(-1)}{4},$$

$$p_{i,i-1} \overset{(2)}{\leq} \frac{i}{n} \cdot 2p \text{ and } p_{i,i-2} \overset{(3)}{\leq} \binom{i}{2}\left(\frac{1}{n}\right)^2 p^2 = \frac{i(i-1)}{n^2} \cdot \frac{p^2}{2}.$$

To upper bound $p_{.,.}$, we replace it by the following $\{-1, 0, +1\}$-process $\tilde{p}_{.,.}$:

$$\tilde{p}_{i,i+1} = \frac{n-i}{n} \cdot \frac{3\exp(-1)}{4} \text{ and } \tilde{p}_{i,i-1} := \frac{i}{n} \cdot 4p.$$

It is obvious that $\tilde{p}_{i,i+1} \geq p_{i,i+1}$. Since $3\exp(-1)/4 < 1/2$ and $4p$ converges to 0, $\tilde{p}_{i,i}$ is at least $1/2$ for $n$ large enough. Hence, $\tilde{p}_{i,i-1} \cdot \tilde{p}_{i-1,i-1} \geq p_{i,i-1}$ and $\tilde{p}_{i,i-1} \cdot \tilde{p}_{i-1,i-2} \cdot \tilde{p}_{i-2,i-2} \geq p_{i,i-2}$ hold, implying that Lemma 2 guarantees that $\tilde{p}_{.,.}$ upper bounds $p_{.,.}$. Finally, applying Lemma 1 to $\tilde{p}_{.,.}$ gives us:

**Theorem 1.** *The expected runtime $T_{\mathrm{ONEMAX},N_p^1}$ of the (1+1) EA on ONEMAX with noise function $N_p^1$ is polynomial for all $p = O(\log(n)/n)$.*

### 4.2 A Super-polynomial Lower Bound for $p = \omega(\log(n)/n)$

Let us now look at the case that $p$ grows asymptotically faster than $\log(n)/n$, i.e. $p \geq \gamma(n)\log(n)/n$ for some function $\gamma : \mathbb{N}^+ \to \mathbb{R}$ with $\gamma(n) \overset{n\to\infty}{\longrightarrow} \infty$. In the following, we want to show that the conditions of Lemma 3 hold for $p_{.,.}$ with $\alpha(n) := \min\{\log(n), \gamma(n)\}$, if $p \leq 1 - \alpha(n)\log(n)/n$.

It is obvious that $\alpha(n) \overset{n\to\infty}{\longrightarrow} \infty$. Condition 1 holds, since the probability $P(Mut_i = d)$ is at most $1/d!$, which is super-polynomially small for $d = \omega(\log(n))$. Furthermore, $p_{i,i+d} = o(1/poly(n))$ for all $i \geq n - \log(n)\alpha(n)$ and non-constant $d$ because of the upper bound (4) for $P(Mut_i = d)$. Hence, condition 2 holds. We can upper bound $\sum_{d=1}^{n-i} p_{i,i+d}$ (using (4)) by

$$\sum_{d=1}^{n-i}\sum_{k=d}^{n-i} \left(\frac{n-i}{n}\right)^k \cdot \frac{1}{k!} = \sum_{k=1}^{n-i} \left(\frac{n-i}{n}\right)^k \cdot \frac{1}{(k-1)!} = O\left(\frac{n-i}{n}\right).$$

In turn, we lower bound $p_{i,i-1}$ for $i \geq n - \alpha(n)\log(n)$ and $n$ large enough by:

$$p_{i,i-1} \overset{(2),(7)}{\geq} \frac{i}{n} \cdot \exp(-1) \cdot p \cdot \left(\frac{i}{n}\left(1 - p + p\frac{n-i+1}{n}\right) + (1-p)\frac{n-i+1}{n}\right)$$

$$\geq \frac{i}{n} \cdot \exp(-1) \cdot \frac{1}{4} \cdot \frac{\alpha(n)\log(n)}{n}.$$

The last inequality holds due to the following case inspection: if $p \leq 1/2$, we use $p \geq \alpha(n)\log(n)/n$, $i/n \geq 1/2$, $1 - p \geq 1/2$ and replace all other terms by zero. If $1/2 < p \leq 1 - \alpha(n)\log(n)/n$, we use $p > 1/2$, $i/n \geq 1/2$, $1 - p \geq \alpha(n)\log(n)/n$, and replace all other terms by zero. Hence, all conditions of Lemma 3 hold.

Let us now look at the case $p > 1 - \alpha(n)\log(n)/n$. We cannot lower bound $p_{i,i-1}$ by $\Theta(\alpha(n)(\log(n)/n)(i/n))$ anymore: e. g. for $p = 1$ decreasing the number of ones implies that the evaluation of the parent decreases its fitness, while the evaluation of the child increases its fitness, i.e. $p_{i,i-1} = P(Mut_i = -1) \cdot (i/n) \cdot (n-i+1)/n$, which is smaller than $\alpha(n)(\log(n)/n)(i/n)$ for $i$ close to $n$.

So, we use Lemma 4 to lower bound the runtime for $p > 1 - \alpha(n) \log(n)/n$. First, we show that decreasing the number of ones is by a constant factor more likely than increasing it, if $i \geq n - n^{1/3}$:

$$\sum_{d=1}^{n-i} p_{i,i+d} \leq \sum_{d=1}^{n-i} P(Mut_i = d) \overset{(4)}{\leq} \frac{n-i}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} + \sum_{d=2}^{n-i} O\left(\left(\frac{n-i}{n}\right)^d\right)$$

$$= \frac{n-i}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} + O\left(\frac{1}{n^{4/3}}\right),$$

$$p_{i,i-1} \overset{(2)}{\geq} P(Mut_i = -1) \cdot p^2 \cdot \frac{i}{n} \frac{n-i}{n} \overset{(7)}{\geq} \left(\frac{i}{n} \cdot p\right)^2 \left(1 - \frac{1}{n}\right)^{n-1} \cdot \frac{n-i}{n},$$

$$p_{i,i-2} \overset{(3)}{\geq} P(Mut_i = -2) \cdot p^2 \cdot \frac{i}{n} \frac{n-i}{n} \overset{(7)}{\geq} \frac{i^2(i-1)}{2n^3} \cdot p^2 \cdot \left(1 - \frac{1}{n}\right)^{n-2} \cdot \frac{n-i}{n}.$$

Hence, we have

$$\frac{\sum_{d=1}^{n-i} p_{i,i+d}}{p_{i,i-1} + p_{i,i-2}} \leq \frac{\frac{n-i}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} + O\left(\frac{1}{n^{4/3}}\right)}{\left(\frac{i}{n} \cdot p\right)^2 \left(1 - \frac{1}{n}\right)^{n-1} \cdot \frac{n-i}{n} + \frac{i^2(i-1)}{2n^3} \cdot p^2 \cdot \left(1 - \frac{1}{n}\right)^{n-2} \cdot \frac{n-i}{n}}$$

$$= \frac{1 + O\left(\frac{1}{n^{1/3}}\right)}{\left(\frac{i}{n} \cdot p\right)^2 + \frac{i^2(i-1)}{2n^3} \cdot p^2 \cdot \left(1 - \frac{1}{n}\right)^{-1}}.$$

Since the denominator converges to $3/2$ and the nominator to $1$, the last term is at least $4/3$ for $n$ large enough. Hence, an impairment is by a constant more likely than an improvement for all $i \geq n - n^{1/3}$ and $n$ large enough.

To apply Lemma 4 we must show additionally that $p_{i+d}/p_{i+d+1}$ is at least $n^\gamma$ for all $i \geq n - n^{1/3}$, $d \geq 1$, and a constant $\gamma > 0$:

$$\frac{p_{i+d}}{p_{i+d+1}} \overset{(1),(4),(6)}{\geq} \frac{\Omega\left(\left(\frac{n-i}{n}\right)^d\right)}{O\left(\left(\frac{n-i}{n}\right)^{d+1}\right)} = \Omega\left(\frac{n}{n-i}\right) = \Omega(n^{2/3}).$$

Hence, Lemma 4 tells us that the runtime of the (1+1) EA on noisy ONEMAX for $p \geq 1 - \alpha(n) \log(n)$ is polynomial only with super-polynomially small probability:

**Theorem 2.** *The runtime $T_{\text{ONEMAX},N_p^1}$ of the (1+1) EA on ONEMAX with noise function $N_p^1$ is polynomial with super-polynomially small probability for all $p = \omega(\log(n)/n)$.*

## 5   Conclusions

We have analyzed the runtime of the (1+1) EA on a variant of ONEMAX where every evaluation is subject to noise with probability $p$ and noise changes one uniformly chosen bit. In this case, the (1+1) EA optimizes ONEMAX with high probability in polynomial time if and only if $p$ is $O(\log(n)/n)$. This is the first

rigorous analysis of an EA for a noisy fitness function without any assumptions. Moreover, the paper presents the methods used to prove this result in a general and modular way in order to help understanding the ideas and to simplify application to other EAs and/or noise models. For instance, generalizing the result for a noise model where every bit is changed by noise with some probability $p$ is an obvious goal for future research.

# References

[AB03]     Dirk V. Arnold and H.-G. Beyer. A comparison of evolution strategies with other direct search methods in the presence of noise. *Computational Optimization and Applications*, 24:135 – 159, 2003.

[DJW00]    S. Droste, Th. Jansen, and I. Wegener. Dynamic parameter control in simple evolutionary algorithms. In *Proceedings of FOGA 2000*, pages 275–294, 2000.

[DJW02]    S. Droste, Th. Jansen, and I. Wegener. On the analysis of the (1 + 1) EA. *Theoretical Computer Science*, (276):51–81, 2002.

[Dro02]    S. Droste. Analysis of the (1+1) EA for a dynamically changing OneMax-variant. In *Proceedings of CEC 2002*, pages 55–60, 2002.

[Dro03]    S. Droste. Analysis of the (1+1) EA for a dynamically bit-wise changing OneMax-variant. In *Proceedings of GECCO 2003*, pages 909–921, 2003.

[FM92]     S. Forrest and M. Mitchell. Relative building block fitness and the building block hypothesis. In *Proceedings of FOGA 1992*, pages 198–226, 1992.

[Hoe63]    W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[JW99]     Th. Jansen and I. Wegener. On the analysis of evolutionary algorithms – a proof that crossover really can help. In *Proceedings of ESA 1999*, pages 184–193, 1999.

[MR95]     R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[RRS95]    Y. Rabani, Y. Rabinovich, and A. Sinclair. A computational view of population genetics. In *Proceedings of STOC 1995*, pages 83–92, 1995.

[Rud97]    G. Rudolph. *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kovač, 1997.

[WW03]     I. Wegener and C. Witt. On the optimization of monotone polynomials by the (1+1) EA and randomized local search. In *Proceedings of GECCO 2003*, pages 622–633, 2003.