

---

# Tight Bounds on the Optimization Time of a Randomized Search Heuristic on Linear Functions<sup>†</sup>

---

CARSTEN WITT

DTU Informatics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark  
(e-mail: cawi@imm.dtu.dk)

*Received 12 March 2012; revised 28 November 2012; first published online 9 January 2013*

The analysis of randomized search heuristics on classes of functions is fundamental to the understanding of the underlying stochastic process and the development of suitable proof techniques. Recently, remarkable progress has been made in bounding the expected optimization time of a simple evolutionary algorithm, called (1+1) EA, on the class of linear functions. We improve the previously best known bound in this setting from  $(1.39 + o(1))en \ln n$  to  $en \ln n + O(n)$  in expectation and with high probability, which is tight up to lower-order terms. Moreover, upper and lower bounds for arbitrary mutation probabilities  $p$  are derived, which imply expected polynomial optimization time as long as  $p = O((\ln n)/n)$  and  $p = \Omega(n^{-C})$  for a constant  $C > 0$ , and which are tight if  $p = c/n$  for a constant  $c > 0$ . As a consequence, the standard mutation probability  $p = 1/n$  is optimal for all linear functions, and the (1+1) EA is found to be an optimal mutation-based algorithm. Furthermore, the algorithm turns out to be surprisingly robust since the large neighbourhood explored by the mutation operator does not disrupt the search.

AMS 2010 *Mathematics subject classification*: Primary 68W20  
Secondary 68Q25

## 1. Introduction

Consider the following modified coupon collector process. There are  $n$  bins, initially empty, each one with a positive real weight. At each time step, go through the bins and flip the state (full/empty) of each bin independently with probability  $1/n$ . Then check whether the total weight of the full bins has decreased compared to the previous time step. If so, restore the previous configuration, otherwise keep the new one. How long does it take until all bins are full at the same time?

If all bins have the same weight, then an  $O(n \log n)$  bound on the expected time follows along the lines of the standard analysis of the coupon collector problem. However, if the weights are different, then the analysis becomes much more involved. In fact, this problem has been studied for more than a decade in the analysis of randomized search heuristics (RSH) and is known as the linear function problem there.

<sup>†</sup> An extended abstract of this paper appeared at *STACS '12* [27].

RSHs are general problem solvers that may be used when no problem-specific algorithm is available. Famous examples are simulated annealing, evolutionary computation, tabu search, *etc.* In order to understand the working principles of RSHs, and to give theoretically founded advice on the applicability of certain RSHs, rigorous analyses of the runtime of RSHs have been conducted. This is a growing research area where many results have been obtained in recent years. It started off in the early 1990s [20] with the consideration of very simple evolutionary algorithms such as the well-known (1+1) EA on very simple example functions such as the ONEMAX function. Later on, results regarding the runtime on classes of functions were derived (*e.g.*, [10, 12, 25, 26]) and powerful tools for the analysis were developed. Today the state of the art in the field allows for the analysis of different types of search heuristics on problems from combinatorial optimization [21].

Recently, the analysis of evolutionary algorithms on linear functions has experienced a great renaissance. The first proof that the (1+1) EA optimizes any linear function in expected time  $O(n \log n)$  by Droste, Jansen and Wegener [10] was highly technical since it did not yet explicitly use the analytic framework of drift analysis [11], which allowed for a considerably simplified proof of the  $O(n \log n)$  bound; see He and Yao [13] for the first complete proof using the method.<sup>1</sup> Another major improvement was made by Jägersküpfer [15, 16], who for the first time stated bounds on the constant hidden in the  $O(n \log n)$  term. This constant was finally improved by Doerr, Johannsen and Winzen [7] to the bound  $(1.39 + o(1))en \ln n$  using a clean framework for the analysis of multiplicative drift [8]. The best known lower bound for general linear functions with non-zero weights is  $en \ln n - O(n)$  and was also proved by Doerr, Johannsen and Winzen [7], building upon the ONEMAX function analysed by Doerr, Fouz and Witt [3, 4].

The standard (1+1) EA flips each bit with probability  $p = 1/n$ , but different values for the mutation probability  $p$  have also been studied in the literature. Recently, it has been proved by Doerr and Goldberg [5, 6] that the  $O(n \log n)$  bound on the expected optimization time of the (1+1) EA still holds (also with high probability) if  $p = c/n$  for an arbitrary constant  $c > 0$ . This result uses the multiplicative drift framework mentioned above and a drift function being cleverly tailored towards the particular linear function. However, the analysis is also highly technical and does not yield explicit constants in the  $O$ -term. For  $p = \omega(1/n)$ , no runtime analyses have been known until now.

In this paper, we prove that the (1+1) EA optimizes all linear functions in expected time  $en \ln n + O(n)$ , thereby closing the gap between the upper and the lower bound up to lower-order terms. Moreover, we show a general upper bound depending on the mutation probability  $p$ , which implies that the expected optimization time is polynomial as long as  $p = O((\ln n)/n)$  (and  $p = \Omega(n^{-C})$  for some constant  $C > 0$ ). We will also show that the expected optimization time is superpolynomial for  $p = \omega((\ln n)/n)$ . Together, these results show that there is a transition from polynomial to superpolynomial optimization time in the region  $\Theta((\ln n)/n)$ . If the mutation probability is  $c/n$  for some constant  $c > 0$ , the expected optimization time is proved to be  $(1 \pm o(1))\frac{ec}{c} n \ln n$ . Altogether, we obtain that the standard choice  $p = 1/n$  of the mutation probability is optimal for all linear functions.

<sup>1</sup> Note, however, that not the original (1+1) EA but a variant rejecting offspring of equal fitness is studied in that paper.

In fact, the lower bounds turn out to hold for the large class of so-called mutation-based EAs, in which the (1+1) EA with  $p = 1/n$  is found to be an optimal algorithm.

Our findings are interesting from both a theoretical and a practical perspective. On the theoretical side, it is noteworthy that  $e^c/c$  is basically the expected waiting time for a mutation step that changes only a single bit. Hence, the mutation operator (in conjunction with the acceptance criterion) is surprisingly robust in the sense that steps flipping many bits neither help nor harm. On the practical side, the optimality of  $p = 1/n$  is remarkable since this seems to be the choice that is most often recommended by researchers in evolutionary computation [2]. Furthermore, the fact that the (1+1) EA is an optimal mutation-based algorithm emphasizes that its runtime analysis can be crucial for obtaining results for more complex approaches.

The proofs of the upper bounds use the recent multiplicative drift theorem by Doerr and Goldberg [5, 6] and a drift function adapted towards both the linear function and the mutation probability. As a consequence of our main result, we obtain the results by Doerr and Goldberg with less effort and explicit constants in front of the  $n \ln n$ -term. All these bounds hold also with high probability, which follows from the recent tail bounds added to the multiplicative drift theorem by Doerr and Goldberg. The lower bounds are based on a new multiplicative drift theorem for lower bounds. By deriving exact results, we show that the research area is maturing, and provides very strong and, at the same time, general tools.

This paper is structured as follows. Section 2 sets up definitions, notations and other preliminaries. Section 3 summarizes and explains the main results. In Sections 4 and 5, respectively, we prove an upper bound for general mutation probabilities and a refined result for  $p = 1/n$ . Lower bounds are shown in Section 6. We finish in Section 7 with some conclusions.

## 2. Preliminaries

The (1+1) EA is a basic search heuristic for the optimization of pseudo-boolean functions  $f: \{0, 1\}^n \rightarrow \mathbb{R}$ . It reflects the typical behaviour of more complicated evolutionary algorithms, serves as basis for the study of more complex approaches and is therefore intensively investigated in the theory of randomized search heuristics [1]. For the case of minimization, it is defined as Algorithm 1.

---

### Algorithm 1 (1+1) EA

---

$t := 0$ .

choose an initial bit string  $x_0 \in \{0, 1\}^n$  uniformly at random.

**repeat**

    create  $x'$  by flipping each bit in  $x_t$  independently with probability  $p$  (*mutation*).

$x_{t+1} := x'$  if  $f(x') \leq f(x_t)$ , and  $x_{t+1} := x_t$  otherwise (*selection*).

$t := t + 1$ .

**forever.**

---

The (1+1) EA can be considered as a simple hill-climber where search points are drawn from a stochastic neighbourhood based on the mutation operator. The parameter  $p$ , where  $0 < p < 1$ , is often chosen as  $1/n$ , which then is called *standard mutation probability*. We call a mutation from  $x_t$  to  $x'$  *accepted* if  $f(x') \leq f(x_t)$ , i.e., if the new search point is taken over; otherwise we call it *rejected*. In our theoretical studies, we ignore the fact that the algorithm in practice will be stopped at some time. The *runtime* (synonymously, *optimization time*) of the (1+1) EA is defined as the first point in time  $t$  such that the search point  $x_t$  has optimal, i.e., minimum  $f$ -value. This corresponds to the number of  $f$ -evaluations until it reaches the optimum. In many cases, one aims for results on the expected optimization time. Here, we also prove results that hold *with high probability*, which means probability  $1 - o(1)$ .

The (1+1) EA is also an instantiation of the algorithmic scheme called *mutation-based EA* by Sudholt [23], which is displayed as Algorithm 2. It is a general population-based approach that includes many variants of evolutionary algorithms with parent and offspring populations as well as parallel evolutionary algorithms. Any mechanism for managing the populations, which are multisets, is allowed as long as the mutation operator is the only variation operator and follows the independent bit-flip property with probability  $0 < p \leq 1/2$ . Again the smallest  $t$  such that  $x_t$  is optimal defines the runtime. Sudholt has proved that for  $p = 1/n$  no mutation-based EA can locate a unique optimum faster than the (1+1) EA can optimize ONEMAX. We will see that the (1+1) EA is the best mutation-based EA for a broad class of functions and for different mutation probabilities.

---

**Algorithm 2** Scheme of a mutation-based EA with population size  $\mu$

---

```

 $t := -1.$ 
repeat
   $t := t + 1.$ 
  create  $x_t \in \{0, 1\}^n$  uniformly at random.
until  $t = \mu - 1.$ 
repeat
  select a parent  $x \in \{x_0, \dots, x_t\}$  according to  $t$  and  $f(x_0), \dots, f(x_t).$ 
  create  $x_{t+1}$  by flipping each bit in  $x$  independently with probability  $p \leq 1/2.$ 
   $t := t + 1.$ 
forever.

```

---

Throughout this paper, we are concerned with linear pseudo-boolean functions. A function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  is called *linear* if it can be written as  $f(x_n, \dots, x_1) = w_n x_n + \dots + w_1 x_1 + w_0$ . As is common in the analysis of the (1+1) EA, we assume without loss of generality that  $w_0 = 0$  and  $w_n \geq \dots \geq w_1 > 0$  hold. Search points are read from  $x_n$  down to  $x_1$  such that  $x_n$ , the most significant bit, is said to be on the left-hand side and  $x_1$ , the least significant bit, on the right-hand side. Since it fits the proof techniques more naturally, we also assume without loss of generality that the (1+1) EA (or, more generally, the mutation-based EA at hand) is minimizing  $f$ , implying that the all-zeros string is the optimum. Our assumptions do not restrict generality since we can permute bits and

negate the weights of a linear function without affecting the stochastic behaviour of the (1+1) EA/mutation-based EA.

Probably the most studied linear function is  $\text{ONEMAX}(x_n, \dots, x_1) = x_n + \dots + x_1$ , occasionally also called the *CountingOnes* problem (which would be the more appropriate name here since we will be minimizing the function). In this paper, we will see that  $\text{ONEMAX}$  is the easiest linear function not only because of its definition but also in terms of expected optimization time. On the other hand, the upper bounds obtained for  $\text{ONEMAX}$  hold for every linear function up to lower-order terms. Hence, surprisingly the (1+1) EA is basically as efficient on an arbitrary linear function as it is on  $\text{ONEMAX}$ . This underlines the robustness of the randomized search heuristic and, in retrospect and for the future, is a strong motivation to investigate the behaviour of randomized search heuristics on the  $\text{ONEMAX}$  problem thoroughly.

Our proofs of the upper bounds shown later in this paper use the multiplicative drift theorem in its most recent version (see [8] and [6]). The key idea of multiplicative drift is to identify a relative progress of the algorithm that is expected in every time step. This expected value is called drift.

**Theorem 2.1 (multiplicative drift, upper bound).** *Let  $S \subseteq \mathbb{R}$  be a finite set of positive numbers with minimum 1. Let  $\{X^{(t)}\}_{t \geq 0}$  be a sequence of random variables over  $S \cup \{0\}$ . Let  $T$  be the random variable that gives the first point in time  $t \geq 0$  for which  $X^{(t)} = 0$ .*

*Suppose that there exists a  $\delta > 0$  such that*

$$\mathbb{E}(X^{(t)} - X^{(t+1)} \mid X^{(t)} = s) \geq \delta s$$

*for all  $s \in S$  and all  $t \geq 0$  with  $\mathbb{P}(X^{(t)} = s) > 0$ . Then, for all  $s_0 \in S$  with  $\mathbb{P}(X^{(0)} = s_0) > 0$ ,*

$$\mathbb{E}(T \mid X^{(0)} = s_0) \leq \frac{\ln(s_0) + 1}{\delta}.$$

*Moreover, we have  $\mathbb{P}(T > (\ln(s_0) + r)/\delta) \leq e^{-r}$  for all  $r > 0$ .*

As an easy example application, consider the (1+1) EA on  $\text{ONEMAX}$  and let  $X^{(t)}$  denote the number of one-bits at time  $t$ . As worse search points are not accepted,  $X^{(t)}$  is non-increasing over time. We obtain  $\mathbb{E}(X^{(t)} - X^{(t+1)} \mid X^{(t)} = s) \geq s(1/n)(1 - 1/n)^{n-1} \geq s/(en)$ , in other words a multiplicative drift of at least  $\delta = 1/(en)$ , since there are  $s$  disjoint single-bit flips that decrease the  $X$ -value by 1. Theorem 2.1 applied with  $\delta = 1/(en)$  and  $\ln(X^{(0)}) \leq \ln n$  gives us the upper bound  $en(\ln n + 1)$  on the expected optimization time, which is basically the same as would be yielded by the classical method of fitness-based partitions [24, 23] or coupon collector arguments [19].

On a general linear function, it is not necessarily a good choice to let  $X^{(t)}$  count the current number of one-bits. Consider, for example, the natural and well-studied function

$$\text{BINVAL}(x_n, \dots, x_1) = \sum_{i=1}^n 2^{i-1} x_i.$$

The (1+1) EA might replace the search point  $(1, 0, \dots, 0)$  by the better search point  $(0, 1, \dots, 1)$ , amounting to a loss of  $n - 2$  zero-bits. More generally, replacing  $(1, 0, \dots, 0)$

by a better search point is equivalent to flipping the leftmost one-bit. In such a step, an expected number of  $(n - 1)p$  zero-bits flip, which decreases the expected number of zero-bits by only  $1 - (n - 1)p$ . The latter expectation (the so-called additive drift) is only  $1/n$  for the standard mutation probability  $p = 1/n$  and might be negative for larger  $p$ . Therefore,  $X^{(t)}$  is typically defined as  $X^{(t)} := g(x^{(t)})$ , where  $x^{(t)}$  is the current search point at time  $t$  and  $g(x_n, \dots, x_1)$  is another linear function called *drift function* or *potential function*. Doerr, Johannsen and Winzen [8] use  $x_1 + \dots + x_{n/2} + (5/4)(x_{n/2+1} + \dots + x_n)$  as the potential function in their application of the multiplicative drift theorem in order to bound the expected optimization time of the (1+1) EA on linear functions. This choice leads to a good lower bound on the multiplicative drift on the one hand and a small maximum value of  $X^{(t)}$  on the other hand. In our proofs of upper bounds in the Sections 4 and 5, it is crucial to define appropriate potential functions.

For the lower bounds in Section 6, we need the following variant of the multiplicative drift theorem, whose proof is also given in Section 6.

**Theorem 2.2 (multiplicative drift, lower bound).** *Let  $S \subseteq \mathbb{R}$  be a finite set of positive numbers with minimum 1. Let  $\{X^{(t)}\}_{t \geq 0}$  be a sequence of random variables over  $S$ , where  $X^{(t+1)} \leq X^{(t)}$  for any  $t \geq 0$ , and let  $s_{\min} > 0$ . Let  $T$  be the random variable that gives the first point in time  $t \geq 0$  for which  $X^{(t)} \leq s_{\min}$ . If there exist positive reals  $\beta, \delta \leq 1$  such that, for all  $s > s_{\min}$  and all  $t \geq 0$  with  $\mathbb{P}(X^{(t)} = s) > 0$ ,*

- (1)  $\mathbb{E}(X^{(t)} - X^{(t+1)} \mid X^{(t)} = s) \leq \delta s,$
- (2)  $\mathbb{P}(X^{(t)} - X^{(t+1)} \geq \beta s \mid X^{(t)} = s) \leq \beta \delta / \ln s,$

then for all  $s_0 \in S$  with  $\mathbb{P}(X^{(0)} = s_0) > 0$ ,

$$\mathbb{E}(T \mid X^{(0)} = s_0) \geq \frac{\ln(s_0) - \ln(s_{\min})}{\delta} \cdot \frac{1 - \beta}{1 + \beta}.$$

Compared to the upper bound, the lower-bound version includes a condition on the maximum stepwise progress and requires non-increasing sequences. As a technical detail, the theorem allows for a positive target  $s_{\min}$ , which is required in our applications.

### 3. Summary of main results

We now list the main consequences of the lower bounds and upper bounds that we will prove in the following sections.

**Theorem 3.1.** *On any linear function, the following holds for the expected optimization time  $\mathbb{E}(T_p)$  of the (1+1) EA with mutation probability  $p$ .*

- (1) *If  $p = \omega((\ln n)/n)$  or  $p = o(n^{-C})$  for every constant  $C > 0$ , then  $\mathbb{E}(T_p)$  is superpolynomial.*
- (2) *If  $p = \Omega(n^{-C})$  for some constant  $C > 0$  and  $p = O((\ln n)/n)$ , then  $\mathbb{E}(T_p)$  is polynomial.*
- (3) *If  $p = c/n$  for a constant  $c > 0$ , then  $\mathbb{E}(T_p) = (1 \pm o(1)) \frac{e^c}{c} n \ln n$ .*
- (4)  *$\mathbb{E}(T_p)$  is minimized for mutation probability  $p = 1/n$  (up to lower-order terms).*
- (5) *No mutation-based EA has an expected optimization time smaller than  $\mathbb{E}(T_{1/n})$  (up to lower-order terms).*

In fact, our analyses below are even more precise; in particular, Theorem 3.1 does not state tail bounds. These are presented below in the more general but also more complicated Theorem 4.1.

**Proof of Theorem 3.1.** The first statement of our summarizing Theorem 3.1 follows from Theorems 6.3, 6.4 and 6.5 below. The second statement is proved in Corollary 4.3, which follows from the already mentioned Theorem 4.1. The third statement takes together Corollaries 4.2 and 6.6. Since  $e^c/c$  is minimized for  $c = 1$ , the fourth statement follows from the third one in conjunction with Corollary 6.6. The fifth statement is also contained in Theorems 6.3 and 6.5.  $\square$

It is worth noting that the optimality of the mutation probability  $p = 1/n$  was apparently unknown even for the case of ONEMAX before this paper appeared in its conference version [27]. However, very recently Sudholt [23] showed the optimality of  $p = 1/n$  for ONEMAX using a different approach. Prior to these two publications, tight upper and lower bounds on the expected optimization time of the (1+1) EA on ONEMAX were only available when the mutation probability was fixed to the standard  $p = 1/n$  [22, 4]. For the general case of linear functions, the strongest previous result said that  $p = \Theta(1/n)$  is optimal [10]. Our result on the optimality of the mutation probability  $1/n$  is interesting since this is the choice commonly recommended by practitioners.

#### 4. Upper bounds

In this section we show a general upper bound that applies to any non-trivial mutation probability.

**Theorem 4.1.** *On any linear function on  $n$  variables, the optimization time of the (1+1) EA with mutation probability  $0 < p < 1$  is at most*

$$(1-p)^{1-n} \left( \frac{n\alpha^2(1-p)^{1-n}}{\alpha-1} + \frac{\alpha}{\alpha-1} \frac{\ln(1/p) + (n-1)\ln(1-p) + r}{p} \right) =: b(r),$$

*with probability at least  $1 - e^{-r}$  for any  $r > 0$ , and it is at most  $b(1)$  in expectation, where  $\alpha > 1$  can be chosen arbitrarily (even depending on  $n$ ).*

Before we prove the theorem, we note two important consequences in more readable form. The first one (Corollary 4.2) displays upper bounds for mutation probabilities  $c/n$ . The second one (Corollary 4.3) is used in Theorem 3.1 above, which states a transition from polynomial to superpolynomial expected optimization times in the region  $p = \Theta((\ln n)/n)$ .

**Corollary 4.2.** *On any linear function, the optimization time of the (1+1) EA with mutation probability  $p = c/n$ , where  $c > 0$  is a constant, is bounded from above by  $(1 + o(1))((e^c/c)n \ln n)$  with probability  $1 - o(1)$  and also in expectation.*

**Proof.** Let  $\alpha := \ln \ln n$  or any other sufficiently slowly growing function. Then  $\alpha/(\alpha - 1) = 1 + O(1/\ln \ln n)$  and  $\alpha^2/(\alpha - 1) = O(\ln \ln n)$ . Moreover,  $(1 - c/n)^{1-n} \leq (1 + o(1))e^c$ . The  $b(r)$  in Theorem 4.1 becomes at most

$$e^c \cdot \left( O(n \ln \ln n) + (1 + o(1)) \frac{n(\ln(n) + \ln(1/c) + r)}{c} \right),$$

and the corollary follows by choosing, e.g.,  $r := \ln \ln n$ . □

**Corollary 4.3.** *On any linear function, the optimization time of the (1+1) EA with mutation probability  $p = O((\ln n)/n)$  and  $p = \Omega(n^{-C})$  for some constant  $C > 0$  is polynomial with probability  $1 - o(1)$  and also in expectation.*

**Proof.** Let  $\alpha := 2$ . By making all positive terms at least 1 and multiplying them, we obtain that the upper bound  $b(r)$  from Theorem 4.1 is at most

$$8n(1 - p)^{2-2n} \cdot \frac{\ln(e/p) + r}{p} \leq 8ne^{2pn} \cdot \frac{\ln(e/p) + r}{p}.$$

Assume  $1/p = O(n^C)$  and  $p \leq c(\ln n)/n$  for constants  $c, C > 0$  and sufficiently large  $n$ . Then  $e^{2pn} \leq n^{2c}$  and the whole expression is polynomial for  $r = 1$  (proving the expectation) and also if  $r = \ln n$  (proving the probability  $1 - o(1)$ ). □

The proof of Theorem 4.1 uses an adaptive potential function as in [6]. That is, the random variables  $X^{(t)}$  used in Theorem 2.1 map the current search point of the (1+1) EA via a potential function to some value in a way that depends also on the linear function at hand. As a special case, if the given linear function happens to be ONEMAX,  $X^{(t)}$  just counts the number of one-bits at time  $t$ . The general construction shares some features with the one of Doerr and Goldberg [6], but both construction and proof are less involved.

**Proof of Theorem 4.1.** Let  $f(x) = w_n x_n + \dots + w_1 x_1$  be the linear function at hand. Define

$$\gamma_i := \left( 1 + \frac{\alpha p}{(1 - p)^{n-1}} \right)^{i-1}$$

for  $1 \leq i \leq n$ , and let  $g(x) = g_n x_n + \dots + g_1 x_1$  be the potential function defined by  $g_1 := 1 = \gamma_1$  and

$$g_i := \min \left\{ \gamma_i, g_{i-1} \cdot \frac{w_i}{w_{i-1}} \right\}$$

for  $2 \leq i \leq n$ . Note that the  $g_i$  are non-decreasing with respect to  $i$ . Intuitively, if the ratio of  $w_i$  and  $w_{i-1}$  is too extreme, the minimum function caps it appropriately, otherwise  $g_i$  and  $g_{i-1}$  are in the same ratio. We consider the stochastic process  $X^{(t)} := g(a^{(t)})$ , where  $a^{(t)}$  is the current search point of the (1+1) EA at time  $t$ . Obviously,  $X^{(t)} = 0$  if and only if  $f$  has been optimized.



Let  $\Delta_t := X^{(t)} - X^{(t+1)}$ . We claim that

$$\mathbb{E}(\Delta_t \mid X^{(t)} = s) \geq s \cdot p \cdot (1 - p)^{n-1} \cdot \left(1 - \frac{1}{\alpha}\right) \tag{4.1}$$

and first prove the theorem using (4.1), which is proved afterwards.

The initial value satisfies

$$X^{(0)} \leq g_n + \dots + g_1 \leq \sum_{i=1}^n \gamma^i \leq \frac{\left(1 + \frac{\alpha p}{(1-p)^{n-1}}\right)^n - 1}{\alpha p (1-p)^{1-n}} \leq \frac{e^{n\alpha p(1-p)^{1-n}}}{\alpha p (1-p)^{1-n}},$$

which means

$$\ln(X^{(0)}) \leq n\alpha p(1-p)^{1-n} + \ln(1/p) + \ln((1-p)^{n-1}).$$

The multiplicative drift theorem (Theorem 2.1) yields that the optimization time  $T$  is bounded from above by

$$\frac{\ln(X^{(0)}) + r}{p(1-p)^{n-1}(1-1/\alpha)} \leq \frac{\alpha(n\alpha p(1-p)^{1-n} + \ln(1/p) + \ln((1-p)^{n-1}) + r)}{(\alpha-1)p(1-p)^{n-1}} = b(r)$$

with probability at least  $1 - e^{-r}$ , and  $\mathbb{E}(T) \leq b(1)$ , which proves the theorem.

To show (4.1), we fix an arbitrary current value  $s$  and an arbitrary search point  $a^{(t)}$  satisfying  $g(a^{(t)}) = s$ . In the following, we implicitly assume  $X^{(t)} = s$  but mostly omit this for the sake of readability. We denote by  $I := \{i \mid a_i^{(t)} = 1\}$  the index set of the one-bits in  $a^{(t)}$  and by  $Z := \{1, \dots, n\} \setminus I$  the zero-bits. We assume  $I \neq \emptyset$  since there is nothing to show otherwise. Denote by  $a'$  the random (not necessarily accepted) offspring produced by the (1+1) EA when mutating  $a^{(t)}$  and by  $a^{(t+1)}$  the next search point after selection. Recall that  $a^{(t+1)} = a'$  if and only if  $f(a') \leq f(a^{(t)})$ . In the following, we will use the event  $A$  that  $a^{(t+1)} = a' \neq a^{(t)}$  and note that  $\Delta_t = 0$  if  $A$  does not occur. Let  $I^* := \{i \in I \mid a_i' = 0\}$  be the set of one-bits of  $a^{(t)}$  that are flipped and let  $Z^* := \{i \in Z \mid a_i' = 1\}$  be the set of zero-bits of  $a^{(t)}$  that are flipped (not conditioned on  $A$ ). Note that  $I^* \neq \emptyset$  if  $A$  occurs.

We need further definitions to analyse the drift carefully. For  $i \in I$ , we define  $k(i) := \max\{j \leq i \mid g_j = \gamma_j\}$  as the most significant position to the right of  $i$  (possibly  $i$  itself) where the potential function takes its maximum on a  $\gamma$ -coefficient, i.e.,  $k(i)$  is the most significant position to the right of  $i$  where we cannot be sure that  $g_{k(i)}/g_{k(i)-1} = w_{k(i)}/w_{k(i)-1}$  holds. Note that  $k(i) \geq 1$  since  $g_1 = \gamma_1$ . Let  $L(i) := \{k(i), \dots, n\} \cap Z$  be the set of zero-bits left of (and including)  $k(i)$  and let  $R(i) := \{1, \dots, k(i) - 1\} \cap Z$  be the remaining zero-bits. Both sets may be empty. For event  $A$  to occur, it is necessary that there is some  $i \in I$  such that bit  $i$  flips to zero and

$$\sum_{j \in I^*} w_j - \sum_{j \in Z^* \cap L(i)} w_j \geq 0,$$

since we are taking only zero-bits out of consideration; more precisely, the last inequality is weaker than the condition  $\sum_{j \in I^*} w_j - \sum_{j \in Z^*} w_j \geq 0$  necessary for  $A$ .

Now, for  $i \in I$ , let  $A_i$  be the event that:

- (1)  $i$  is the leftmost flipping one-bit (i.e.,  $i \in I^*$  and  $\{i + 1, \dots, n\} \cap I^* = \emptyset$ ) and
- (2)  $\sum_{j \in I^*} w_j - \sum_{j \in Z^* \cap L(i)} w_j \geq 0$ .

If none of the  $A_i$  occurs, then  $\Delta_t = 0$ . Furthermore, the  $A_i$  are mutually disjoint.

For any  $i \in I$ ,  $\Delta_i$  can be written as the sum of the two terms

$$\Delta_L(i) := \sum_{j \in I^*} g_j - \sum_{j \in Z^* \cap L(i)} g_j \quad \text{and} \quad \Delta_R(i) := - \sum_{j \in Z^* \cap R(i)} g_j.$$

By the law of total probability and the linearity of expectation, we have

$$\mathbb{E}(\Delta_i) = \sum_{i \in I} \mathbb{E}(\Delta_L(i) \mid A_i) \cdot \mathbb{P}(A_i) + \mathbb{E}(\Delta_R(i) \mid A_i) \cdot \mathbb{P}(A_i). \tag{4.2}$$

In the following, the bits in  $R(i)$  are pessimistically assumed to flip from 0 to 1 independently with probability  $p$  each if  $A_i$  happens. This leads to

$$\mathbb{E}(\Delta_R(i) \mid A_i) \geq -p \sum_{j \in R(i)} g_j. \tag{4.3}$$

In order to estimate  $\mathbb{E}(\Delta_L(i))$ , we carefully inspect the relation between the weights of the original function and the potential function. By definition, we obtain  $g_j/g_{k(i)} = w_j/w_{k(i)}$  for  $k(i) \leq j \leq i$  and  $g_j/g_{k(i)} \leq w_j/w_{k(i)}$  for  $j > i$  whereas  $g_j/g_{k(i)} \geq w_j/w_{k(i)}$  for  $j < k(i)$ . Hence, if  $A_i$  occurs then  $g_j \geq g_{k(i)} \cdot \frac{w_j}{w_{k(i)}}$  for  $j \in I^*$  (since  $i$  is the leftmost flipping one-bit) whereas  $g_j \leq g_{k(i)} \cdot \frac{w_j}{w_{k(i)}}$  for  $j \in L(i)$ . Together, we obtain, under the assumption that  $A_i$  has occurred, the non-negativity of the random variable  $\Delta_L(i)$ :

$$\begin{aligned} \text{on } A_i, \quad \Delta_L(i) &= \sum_{j \in I^*} g_j - \sum_{j \in Z^* \cap L(i)} g_j \\ &\geq \sum_{j \in I^*} g_{k(i)} \cdot \frac{w_j}{w_{k(i)}} - \sum_{j \in Z^* \cap L(i)} g_{k(i)} \cdot \frac{w_j}{w_{k(i)}} \geq 0, \end{aligned}$$

using the definition of  $A_i$ .

Now let  $S_i := \{|Z^* \cap L(i)| = 0\}$  be the event that no zero-bit from  $L(i)$  flips. Using the law of total probability, we obtain that

$$\mathbb{E}(\Delta_L(i) \mid A_i) \cdot \mathbb{P}(A_i) = \mathbb{E}(\Delta_L(i) \mid A_i \cap S_i) \cdot \mathbb{P}(A_i \cap S_i) + \mathbb{E}(\Delta_L(i) \mid A_i \cap \bar{S}_i) \cdot \mathbb{P}(A_i \cap \bar{S}_i).$$

Since  $\Delta_L(i) \geq 0$  if  $A_i$  occurs, the conditional expectations are non-negative. We bound the second term on the right-hand side by 0. In conjunction with (4.2), we get

$$\mathbb{E}(\Delta_i) \geq \sum_{i \in I} \mathbb{E}(\Delta_L(i) \mid A_i \cap S_i) \cdot \mathbb{P}(A_i \cap S_i) + \mathbb{E}(\Delta_R(i) \mid A_i) \cdot \mathbb{P}(A_i).$$

Obviously,  $\mathbb{E}(\Delta_L(i) \mid A_i \cap S_i) \geq g_i$ . We estimate  $\mathbb{P}(A_i \cap S_i) \geq p(1-p)^{n-1}$  since it is sufficient to flip only bit  $i$ , and  $\mathbb{P}(A_i) \leq p$  since it is necessary to flip this bit. In (4.3), we have bounded  $\mathbb{E}(\Delta_R(i) \mid A_i)$ . Taking everything together, we get

$$\begin{aligned} \mathbb{E}(\Delta_i) &\geq \sum_{i \in I} \left( p(1-p)^{n-1} g_i - p^2 \sum_{j \in R(i)} g_j \right) \\ &\geq \sum_{i \in I} \left( p(1-p)^{n-1} \frac{g_i}{g_{k(i)}} \gamma_{k(i)} - p^2 \sum_{j=1}^{k(i)-1} \gamma_j \right). \end{aligned}$$

The term for  $i$  equals

$$\begin{aligned}
 & p(1-p)^{n-1} \frac{g_i}{g_{k(i)}} \left(1 + \frac{\alpha p}{(1-p)^{n-1}}\right)^{k(i)-1} - \frac{p^2 \cdot \left(\left(1 + \frac{\alpha p}{(1-p)^{n-1}}\right)^{k(i)-1} - 1\right)}{\left(\frac{\alpha p}{(1-p)^{n-1}}\right)} \\
 & \geq \left(1 - \frac{1}{\alpha}\right) p(1-p)^{n-1} \frac{g_i}{g_{k(i)}} \left(1 + \frac{\alpha p}{(1-p)^{n-1}}\right)^{k(i)-1} = \left(1 - \frac{1}{\alpha}\right) p(1-p)^{n-1} g_i,
 \end{aligned}$$

where the inequality uses  $g_i \geq g_{k(i)}$ . Hence,

$$\mathbb{E}(\Delta_t) \geq \sum_{i \in I} \left(1 - \frac{1}{\alpha}\right) p(1-p)^{n-1} g_i = \left(1 - \frac{1}{\alpha}\right) p(1-p)^{n-1} g(a^{(t)}),$$

which proves (4.1) and therefore the theorem. □

### 5. Refined upper bound for mutation probability $1/n$

In this section we consider the standard mutation probability  $p = 1/n$  and refine the result from Corollary 4.2. More precisely, we obtain that the lower-order terms are  $O(n)$ . The proof will be shorter, and uses a simpler potential function.

**Theorem 5.1.** *On any linear function, the expected optimization time of the (1+1) EA with  $p = 1/n$  is at most  $en \ln n + 2en + O(1)$ , and the probability that the optimization time exceeds  $en \ln n + (1+r)en + O(1)$  is at most  $e^{-r}$  for any  $r > 0$ .*

**Proof.** Let  $f(x) = w_n x_n + \dots + w_1 x_1$  be the linear function at hand and let  $g(x) = g_n x_n + \dots + g_1 x_1$  be the potential function defined by

$$g_i = \left(1 + \frac{1}{n-1}\right)^{\min\{j \mid j \leq i \wedge w_j = w_i\} - 1},$$

hence  $g_i = (1 + 1/(n-1))^{i-1}$  for all  $i$  if and only if the  $w_i$  are mutually distinct. We consider the stochastic process  $X^{(t)} := g(a^{(t)})$ , where  $a^{(t)}$  is the current search point of the (1+1) EA at time  $t$ . Obviously,  $X^{(t)} = 0$  if and only if  $f$  has been optimized.

Let  $\Delta_t := X^{(t)} - X^{(t+1)}$ . In a case analysis (partly inspired by Doerr, Johannsen and Winzen [8]), we will show below for  $n \geq 4$  that  $\mathbb{E}(\Delta_t \mid X^{(t)} = s) \geq s/(en)$ . The initial value satisfies

$$X^{(0)} \leq g_n + \dots + g_1 \leq \sum_{i=0}^{n-1} \left(1 + \frac{1}{n-1}\right)^i \leq n \left(1 + \frac{1}{n-1}\right)^{n-1} \leq en.$$

Hence,  $\ln(X^{(0)}) \leq \ln n + 1$ . Assuming  $n \geq 4$ , Theorem 2.1 yields  $\mathbb{E}(T) \leq en(\ln n + 2)$  and  $\mathbb{P}(T > en(\ln n + r + 1)) \leq e^{-r}$  regardless of the starting point, from which the theorem follows.

The case analysis fixes an arbitrary current search point  $a^{(t)}$ . We use the same notation as in the proof of Theorem 4.1 and denote by  $I := \{i \mid a_i^{(t)} = 1\}$  the index set of its one-bits and by  $Z := \{1, \dots, n\} \setminus I$  its zero-bits. We assume  $I \neq \emptyset$  since there is nothing to show otherwise. Denote by  $a'$  the random (not necessarily accepted) offspring produced by the

(1+1) EA when mutating  $a^{(t)}$  and by  $a^{(t+1)}$  the next search point after selection. Recall that  $a^{(t+1)} = a'$  if and only if  $f(a') \leq f(a^{(t)})$ . In what follows, we will often condition on the event  $A$  that  $a^{(t+1)} = a' \neq a^{(t)}$  holds and note that  $\Delta_t = 0$  if  $A$  does not occur. Let  $I^* := \{i \in I \mid a'_i = 0\}$  be the set of one-bits of  $a^{(t)}$  that are flipped and let  $Z^* := \{i \in Z \mid a'_i = 1\}$  be the set of zero-bits of  $a^{(t)}$  that are flipped (not conditioned on  $A$ ). Note that  $I^* \neq \emptyset$  if  $A$  occurs.

**Case 1.** Event  $S_1 := \{|I^*| \geq 2\} \cap A$  occurs. Under this condition, each zero-bit in  $a^{(t)}$  has been flipped to 1 in  $a^{(t+1)}$  with probability at most  $1/n$ . Since  $g_i \geq 1$  for  $1 \leq i \leq n$ , we have

$$\begin{aligned} \mathbb{E}(\Delta_t \mid S_1) &\geq |I^*| - \frac{1}{n} \sum_{i \notin I} g_i \geq 2 - \frac{1}{n} \sum_{i=1}^n \left(1 + \frac{1}{n-1}\right)^{i-1} \\ &= 2 - \frac{(1 + 1/(n-1))^n - 1}{n/(n-1)} \geq 2 - \left(e - \left(1 - \frac{1}{n}\right)\right) \geq 0 \end{aligned}$$

for  $n \geq 4$ , where we have used  $1 + 1/(n-1) = 1/(1 - 1/n)$ . Hence, it is pessimistic to assume  $\mathbb{E}(\Delta_t \mid S_1) = 0$ .

**Case 2.** Event  $S_2 := \{|I^*| = 1\} \cap A$  occurs. Let  $i^*$  be the single element of  $I^*$  and note that this is a random variable.

**Subcase 2.1.**  $S_{21} := \{|I^*| = 1\} \cap \{Z^* = \emptyset\} \cap A$  occurs. Since  $\{|I^*| = 1\}$  and  $\{Z^* = \emptyset\}$  together imply  $A$ , the index  $i^*$  of the flipped one-bit is uniform over  $I$ . Hence,

$$\mathbb{E}(\Delta_t \mid S_{21}) = \sum_{i \in I} g_i / |I|.$$

Moreover,  $\mathbb{P}(S_{21}) \geq |I|(1/n)(1 - 1/n)^{n-1} \geq |I|/(en)$ , implying

$$\mathbb{E}(\Delta_t \mid S_{21}) \cdot \mathbb{P}(S_{21}) \geq g(a^{(t)})/(en) = X^{(t)}/(en).$$

If we can show that  $\mathbb{E}(\Delta_t \mid \{|I^*| = 1\} \cap \{|Z^*| \geq 1\} \cap A) \geq 0$ , which will be proved in Subcase 2.2 below, then  $\mathbb{E}(\Delta_t \mid X^{(t)} = s) \geq s/(en)$  follows by the law of total probability and the proof is complete.

**Subcase 2.2.**  $S_{22} := \{|I^*| = 1\} \cap \{|Z^*| \geq 1\} \cap A$  occurs. Let  $j^* := \max\{j \mid j \in Z^*\}$  be the index of the leftmost flipping zero-bit, and note that  $j^*$  is also random. Since we work under  $|I^*| = 1$  and the  $w_j$  are monotone increasing with respect to  $j$ , for  $A$  to occur it is necessary that  $w_{j^*} \leq w_{i^*}$  holds.

**Subcase 2.2.1.**  $S_{221} := \{|I^*| = 1\} \cap \{|Z^*| \geq 1\} \cap \{j^* > i^*\} \cap A$  occurs. Then  $w_{j^*} = w_{i^*}$  and  $|Z^*| = 1$  must hold. In this case  $g_{j^*} = g_{i^*}$ , by the definition of  $g$ , and  $\mathbb{E}(\Delta_t \mid S_{221}) = 0$  follows immediately.

**Subcase 2.2.2.**  $S_{222} := \{|I^*| = 1\} \cap \{|Z^*| \geq 1\} \cap \{j^* < i^*\} \cap A$  occurs. If  $w_{j^*} = w_{i^*}$  then  $|Z^*| = 1$  must hold for  $A$  to occur, and zero drift follows as in the previous subcase. Now let us assume  $w_{j^*} < w_{i^*}$  and thus  $g_{j^*} < g_{i^*}$ . For notational convenience, we redefine  $i^* := \min\{i \mid w_i = w_{i^*}\}$ . We consider  $Z_r := Z \cap \{1, \dots, i^* - 1\}$ , the set of potentially flipping zero-bits right of  $i^*$ , denote  $k := |Z_r|$  and note that in the worst case,  $Z_r = \{i^* - 1, \dots, i^* - k\}$  as the  $g_i$  are non-decreasing. By using  $\tilde{p} := \mathbb{P}(Z^* \cap Z_r \neq \emptyset) = 1 - (1 - 1/n)^k$  and the

definition of conditional probabilities, we obtain under  $S_{222}$  that every bit from  $Z_r$  is flipped (not necessarily independently) with probability at most  $(1/n)/\tilde{p} = \frac{1}{n(1-(1/n)^k)}$ . We now assume that all the corresponding  $a'$  are accepted. This is pessimistic for the following reasons. Consider a rejected  $a'$ . If  $|Z^*| = 1$  then our prerequisite  $j^* < i^*$  and the monotonicity of the  $g_i$  imply a negative  $\Delta_i$ -value. If  $|Z^*| > 1$  then the negative  $\Delta_i$ -value is due to the fact that  $g_i < g_{i-1} + g_{i-2}$  for  $3 \leq i \leq n$ . Hence, using the linearity of expectation we get

$$\begin{aligned} \mathbb{E}(\Delta_i \mid S_{222}) &\geq g_{i^*} - \frac{1}{n\tilde{p}} \cdot \sum_{j \in Z_r} g_j \geq g_{i^*} - \sum_{j=1}^k \frac{g_{i^*-j}}{n(1-(1/n)^k)} \\ &= \left(1 + \frac{1}{n-1}\right)^{i^*-1} - \sum_{j=0}^{k-1} \frac{(1 + 1/(n-1))^{i^*-1-j}}{n(1-(1/n)^k)} \\ &= \left(1 + \frac{1}{n-1}\right)^{i^*-k} \left( \left(1 + \frac{1}{n-1}\right)^{k-1} - \frac{((1 + 1/(n-1))^k - 1) \cdot (n-1)}{n(1-(1/n)^k)} \right) = 0, \end{aligned}$$

where the last equality follows since  $1 + 1/(n-1) = (1 - 1/n)^{-1}$  and

$$\frac{((1 + 1/(n-1))^k - 1) \cdot (n-1)}{n(1-(1/n)^k)} = \left(1 - \frac{1}{n}\right) \frac{(1 - 1/n)^{-k} - 1}{1 - (1 - 1/n)^k} = \left(1 - \frac{1}{n}\right)^{1-k}.$$

This completes the proof. □

### 6. Lower bounds

In this section we state lower bounds that prove the results from Theorem 4.1 to be tight up to lower-order terms for a wide range of mutation probabilities. Moreover, we show that the lower bounds hold for the very large class of mutation-based algorithms (Algorithm 2). Recall that a list of the most important consequences is given above in Theorem 3.1. For technical reasons, we split the proof of the lower bounds into two main cases, namely  $p = O(n^{-2/3-\varepsilon})$  and  $p = \Omega(n^{\varepsilon-1})$  for any constant  $\varepsilon > 0$ . Unless  $p > 1/2$ , the proofs go back to ONEMAX as a worst case, as outlined in the following subsection.

#### 6.1. ONEMAX as easiest linear function

Doerr, Johannsen and Winzen [7] show with respect to the (1+1) EA with standard mutation probability  $1/n$  that ONEMAX is the ‘easiest’ function from the class of functions with unique global optimum, which comprises the class of linear functions. More precisely, the expected optimization time on ONEMAX is proved to be smallest within the class.

We will generalize this result to  $p \leq 1/2$  with moderate additional effort. In fact we will relate the behaviour of an arbitrary mutation-based EA on ONEMAX to the (1+1) EA $_{\mu}$  in a similar way to Sudholt [23, Section 7]. The latter algorithm, displayed as Algorithm 3, creates search points uniformly at random from time 0 to time  $\mu - 1$ , and then chooses a best one from these to be the current search point at time  $\mu - 1$ ; afterwards it works as the standard (1+1) EA. Note that we obtain the standard (1+1) EA for  $\mu = 1$ . Moreover, we will only consider the case  $\mu = n^{O(1)}$  in order to bound the running time of the

initialization. This makes sense since even drawing  $2^{\sqrt{n}}$  random search points will with overwhelming probability fail to lead to a unique optimum (such as the all-zeros string in ONEMAX).

---

**Algorithm 3** (1+1) EA $_{\mu}$

---

```

t := -1.
repeat
  t := t + 1.
  create  $x_t \in \{0, 1\}^n$  uniformly at random.
until t =  $\mu - 1$ .
 $x_t := \arg \min\{f(x) \mid x \in \{x_0, \dots, x_t\}\}$  (breaking ties uniformly).
repeat
  create  $x'$  by flipping each bit in  $x_t$  independently with probability  $p$ .
   $x_{t+1} := x'$  if  $f(x') \leq f(x_t)$ , and  $x_{t+1} := x_t$  otherwise.
  t := t + 1.
forever.
```

---

Our analyses need the monotonicity statement from Lemma 6.1 below, which is similar to Lemma 11 of Doerr, Johannsen and Winzen [7] and whose proof is already sketched in Droste, Jansen and Wegener [9, Section 5]. Note, however, that Doerr, Johannsen and Winzen [7] only consider  $p = 1/n$  and have a stronger statement for this case. More precisely (using the notation defined in the lemma), they show  $\mathbb{P}(|\text{mut}(a)|_1 = j) \geq \mathbb{P}(|\text{mut}(b)|_1 = j)$ , which does not hold for large  $p$ . Here and below,  $|x|_1$  denotes the number of ones in a bit string  $x$ .

**Lemma 6.1.** *Let  $a, b \in \{0, 1\}^n$  be two search points satisfying  $|a|_1 < |b|_1$ . Denote by  $\text{mut}(x)$  the random string obtained by flipping each bit of  $x$  independently with probability  $p$ . Let  $0 \leq j \leq n$  be arbitrary. If  $p \leq 1/2$  then*

$$\mathbb{P}(|\text{mut}(a)|_1 \leq j) \geq \mathbb{P}(|\text{mut}(b)|_1 \leq j).$$

**Proof.** We prove the result only for  $|b|_1 = |a|_1 + 1$ . The general statement then follows by induction on  $|b|_1 - |a|_1$ .

By the symmetry of the mutation operator,  $\mathbb{P}(|\text{mut}(x)|_1 \leq j)$  is the same for all  $x$  with  $|x|_1 = |a|_1$ . We therefore assume  $b \geq a$  (i.e.,  $b$  is componentwise not less than  $a$ ). In the following, let  $s^*$  be the unique index where  $b_{s^*} = 1$  and  $a_{s^*} = 0$ . Let  $S(x)$  be the event that bit  $s^*$  flips when  $x$  is mutated. Since bits are flipped independently, it holds that  $\mathbb{P}(S(x)) = p$  for any  $x$ . We write  $a' := \text{mut}(a)$  and  $b' := \text{mut}(b)$ . Assuming  $p \leq 1/2$ , the aim is to show  $\mathbb{P}(|a'|_1 \leq j) \geq \mathbb{P}(|b'|_1 \leq j)$ , which is equivalent to

$$\begin{aligned} & (\mathbb{P}(|a'|_1 \leq j \mid \overline{S(a)}) - \mathbb{P}(|b'|_1 \leq j \mid \overline{S(b)})) \cdot (1 - p) \\ & + (\mathbb{P}(|a'|_1 \leq j \mid S(a)) - \mathbb{P}(|b'|_1 \leq j \mid S(b))) \cdot p \geq 0. \end{aligned} \tag{6.4}$$

Note that the relation  $\mathbb{P}(|a'|_1 \leq j \mid \overline{S(a)}) \geq \mathbb{P}(|b'|_1 \leq j \mid \overline{S(b)})$  follows from a simple coupling argument as  $a' \leq b'$  holds if the mutation operator flips the bits other than  $s^*$  in the

same way with respect to  $a$  and  $b$ . Moreover,

$$\mathbb{P}(|a'|_1 \leq j \mid \overline{S(a)}) - \mathbb{P}(|b'|_1 \leq j \mid \overline{S(b)}) = \mathbb{P}(|b'|_1 \leq j \mid S(b)) - \mathbb{P}(|a'|_1 \leq j \mid S(a))$$

since  $a$  is obtained from  $b$  by flipping bit  $s^*$  and *vice versa*. Hence, (6.4) follows. □

The following theorem is a generalization of Theorem 9 by Doerr, Johannsen and Winzen [7] to the case  $p \leq 1/2$  instead of  $p = 1/n$ . However, we not only generalize to higher mutation probabilities but also consider the more general class of mutation-based algorithms. Finally, we prove stochastic ordering, while Doerr, Johannsen and Winzen [7] inspect only the expected optimization times. Still, many ideas of the original proof can be taken over and combined with the proof of Theorem 5 of Sudholt [23].

**Theorem 6.2.** *Consider a mutation-based EA  $A$  with population size  $\mu$  and mutation probability  $p \leq 1/2$  on any function with a unique global optimum. Then the optimization time of  $A$  is stochastically at least as large as the optimization time of the (1+1)  $EA_\mu$  on ONEMAX.*

**Proof.** Let  $f$  denote the function with unique global optimum, which, without loss of generality, we assume to be the all-zeros string. For any sequence  $\mathcal{X} = (x_0, \dots, x_{\ell-1})$  of search points over  $\{0, 1\}^n$ , let  $q(\mathcal{X})$  be the probability that  $\mathcal{X}$  represents the first  $\ell$  search points  $x_0, \dots, x_{\ell-1}$  created by Algorithm  $A$  on  $f$  (its so-called history up to time  $\ell - 1$ ). For any history  $\mathcal{X}$  with  $q(\mathcal{X}) > 0$ , let  $T_f(\mathcal{X})$  denote the random optimization time of Algorithm  $A$  on  $f$ , given that its history up to time  $\ell$  equals  $\mathcal{X}$ . Let

$$\Xi_\ell := \left\{ \mathcal{X} = (x_0, \dots, x_{\ell-1}) \in \prod_{i=1}^{\ell} \{0, 1\}^n \mid q(\mathcal{X}) > 0 \right\}$$

denote the set of all possible histories of length  $\ell$  with respect to Algorithm  $A$  on  $f$ , and let

$$\Xi := \left\{ \bigcup_{\ell=1}^m \Xi_\ell \mid m \in \mathbb{N} \right\}$$

denote all possible histories of finite length. Finally, for any  $\mathcal{X} \in \Xi$ , let  $L(\mathcal{X})$  denote the length of  $\mathcal{X}$ .

Given any  $\mathcal{X} \in \Xi$ , let (1+1) EA( $\mathcal{X}$ ) be the algorithm that chooses a search point with minimal number of ones from  $\mathcal{X}$  as current search point at time  $L(\mathcal{X}) - 1$  (breaking ties uniformly) and afterwards proceeds as the standard (1+1) EA on ONEMAX. Now, let  $T_{\text{ONEMAX}}(\mathcal{X})$  denote the random optimization time of the (1+1) EA( $\mathcal{X}$ ). We claim that the stochastic ordering

$$\mathbb{P}(T_f(\mathcal{X}) \geq t) \geq \mathbb{P}(T_{\text{ONEMAX}}(\mathcal{X}) \geq t)$$

holds for every  $\mathcal{X} \in \Xi$  satisfying  $L(\mathcal{X}) \geq \mu$  and every  $t \geq 0$ . Note that the random vector of initial search points  $\mathcal{X}^* := (x_0, \dots, x_{\mu-1})$  follows the same distribution in both Algorithm  $A$  and the (1+1)  $EA_\mu$ . In particular, the two algorithms are identical before time  $\mu - 1$ , *i.e.*, before initialization is finished. Furthermore, (1+1) EA( $\mathcal{X}^*$ ) is the (1+1)  $EA_\mu$  initialized with  $\mathcal{X}^*$ . Altogether, the claimed stochastic ordering implies the theorem. Moreover,

regardless of the length  $L(\mathcal{X})$ , the claim is obvious for  $t \leq L(\mathcal{X})$  since the behaviour up to time  $L(\mathcal{X})$  is fixed.

For any  $\mathcal{X} \in \Xi$ , let  $|\mathcal{X}|_1 := \min\{|x|_1 \mid x \in \mathcal{X}\}$  denote the best number of ones in the history, where  $x \in (x_0, \dots, x_{\ell-1})$  means that  $x = x_i$  for some  $i \in \{0, \dots, \ell - 1\}$ . For every  $k \in \{0, \dots, n\}$ , every  $\ell \geq \mu$  and every  $t \geq 0$ , let

$$p_{k,\ell}(t) := \min\{\mathbb{P}(T_{\text{ONEMAX}}(\mathcal{X}) \geq \ell + t \mid \mathcal{X} \in \Xi_\ell, |\mathcal{X}|_1 = k)\}$$

be the minimum probability of the (1+1) EA( $\mathcal{X}$ ) needing at least  $\ell + t$  steps to optimize ONEMAX from a history of length  $\ell$  whose best search point has exactly  $k$  one-bits. Due to the symmetry of the ONEMAX function and the definition of (1+1) EA( $\mathcal{X}$ ), we have  $\mathbb{P}(T_{\text{ONEMAX}}(\mathcal{X}) \geq \ell + t) = p_{k,\ell}(t)$  for every  $\mathcal{X}$  satisfying  $|\mathcal{X}| = \ell$  and  $|\mathcal{X}|_1 = k$ . In other words, the minimum can be omitted from the definition of  $p_{k,\ell}$ .

Furthermore, for every  $k \in \{0, \dots, n\}$ , every  $\ell \geq \mu$  and every  $t \geq 0$ , let

$$\tilde{p}_{k,\ell}(t) := \min\{\mathbb{P}(T_f(\mathcal{X}) \geq \ell + t \mid \mathcal{X} \in \Xi_\ell, |\mathcal{X}|_1 \geq k)\}$$

be the minimum probability of Algorithm  $A$  needing at least  $\ell + t$  steps to optimize  $f$  from a history of length  $\ell \geq \mu$  whose best search point has at least  $k$  one-bits. We will show  $\tilde{p}_{k,\ell}(t) \geq p_{k,\ell}(t)$  for any  $k \in \{0, \dots, n\}$  and  $\ell \geq \mu$  by induction on  $t$ . In particular, by choosing  $\ell := \mu$  and applying the law of total probability with respect to the outcomes of  $|\mathcal{X}^*|_1$ , this will imply the above-mentioned stochastic ordering and therefore the theorem.

If  $k \geq 1$  then  $p_{k,\ell}(0) = \tilde{p}_{k,\ell}(0) = 1$  for any  $\ell \geq \mu$ , since  $k \geq 1$  means that the first  $\ell$  search points do not contain the optimum. Moreover,  $p_{0,\ell}(t) = \tilde{p}_{0,\ell}(t) = 0$  for any  $t \geq 0$  and  $\ell \geq \mu$ , since a history beginning with the all-zeros string corresponds to optimization time 0 and thus minimizes both  $\mathbb{P}(T_f(\mathcal{X}) \geq t + \ell)$  and  $\mathbb{P}(T_{\text{ONEMAX}}(\mathcal{X}) \geq t + \ell)$ . Now let us assume that there is some  $t \geq 0$  such that  $\tilde{p}_{k,\ell}(t') \geq p_{k,\ell}(t')$  holds for all  $0 \leq t' \leq t$ ,  $k \in \{0, \dots, n\}$ , and  $\ell \geq \mu$ . Note that the inequality has already been proved for all  $t$  if  $k = 0$ .

Consider the (1+1) EA( $\mathcal{X}$ ) for an arbitrary  $\mathcal{X}$  satisfying  $L(\mathcal{X}) = \ell \geq \mu$  and  $|\mathcal{X}|_1 = k + 1$  for some  $k \in \{0, \dots, n - 1\}$ . Let some  $x \in \{0, 1\}^n$ , where  $|x|_1 = k + 1$ , be chosen from  $\mathcal{X}$  and let  $y \in \{0, 1\}^n$  be the random search point generated by flipping each bit in  $x$  independently with probability  $p$ . The (1+1) EA( $\mathcal{X}$ ) will accept  $y$  as new search point at time  $\ell + 1 > \mu$  if and only if  $|y|_1 \leq |x|_1 = k + 1$ . Hence,

$$p_{k+1,\ell}(t + 1) = \mathbb{P}(|y|_1 \geq k + 1) \cdot p_{k+1,\ell+1}(t) + \sum_{j=0}^k \mathbb{P}(|y|_1 = j) \cdot p_{j,\ell+1}(t). \tag{6.5}$$

Next, let  $\mathcal{X}$ , where again  $L(\mathcal{X}) = \ell \geq \mu$ , be a history satisfying  $\mathbb{P}(T_f(\mathcal{X}) \geq t + 1) = \tilde{p}_{k+1,\ell}(t + 1)$  and let  $\tilde{x}$  be the (random) search point that is chosen for mutation at time  $\ell$  in order to obtain the equality of the two probabilities. Note that  $|\tilde{x}|_1 \geq k + 1$ . Moreover, let  $\tilde{y} \in \{0, 1\}^n$  be the random search point generated by flipping each bit in  $\tilde{x}$  independently



with probability  $p$ . Let  $\mathcal{X}'$  be the concatenation of  $\mathcal{X}$  and  $\tilde{y}$ . Then

$$\begin{aligned} \tilde{p}_{k+1,\ell}(t+1) &= \mathbb{P}(|\tilde{y}|_1 \geq k+1) \cdot \mathbb{P}(T_f(\mathcal{X}') \geq t \mid |\tilde{y}|_1 \geq k+1) \\ &\quad + \sum_{j=0}^k \mathbb{P}(|\tilde{y}|_1 = j) \cdot \mathbb{P}(T_f(\mathcal{X}') \geq t \mid |\tilde{y}|_1 = j), \end{aligned}$$

which, by definition of the  $\tilde{p}_i(t)$ , gives us the lower bound

$$\tilde{p}_{k+1,\ell}(t+1) \geq \mathbb{P}(|\tilde{y}|_1 \geq k+1) \cdot \tilde{p}_{k+1,\ell+1}(t) + \sum_{j=0}^k \mathbb{P}(|\tilde{y}|_1 = j) \cdot \tilde{p}_{j,\ell+1}(t).$$

To relate the last inequality to (6.5) above, we interpret the right-hand side as a function of  $k+2$  variables. More precisely, let

$$\phi(a_0, \dots, a_{k+1}) := \sum_{j=0}^{k+1} a_j \tilde{p}_{j,\ell+1}(t),$$

and consider the vectors

$$v^{(f)} = (v_0^{(f)}, \dots, v_{k+1}^{(f)}) := (\mathbb{P}(|\tilde{y}|_1 = 0), \dots, \mathbb{P}(|\tilde{y}|_1 = k), \mathbb{P}(|\tilde{y}|_1 \geq k+1))$$

and

$$v^{(O)} = (v_0^{(O)}, \dots, v_{k+1}^{(O)}) := (\mathbb{P}(|y|_1 = 0), \dots, \mathbb{P}(|y|_1 = k), \mathbb{P}(|y|_1 \geq k+1)).$$

If we can show that  $\phi(v^{(f)}) \geq \phi(v^{(O)})$ , then we can conclude

$$\begin{aligned} \tilde{p}_{k+1,\ell}(t+1) &\geq \phi(v^{(f)}) \geq \phi(v^{(O)}) \\ &\geq \mathbb{P}(|y|_1 \geq k+1) \cdot p_{k+1,\ell+1}(t) + \sum_{j=0}^k \mathbb{P}(|y|_1 = j) \cdot p_{j,\ell+1}(t) = p_{k+1,\ell}(t+1), \end{aligned}$$

where the last inequality follows from the induction hypothesis and the equality is from (6.5). This will complete the induction step.

To show the claimed inequality  $\phi(v^{(f)}) \geq \phi(v^{(O)})$ , we use that for  $0 \leq j \leq k$

$$\mathbb{P}(|y|_1 \leq j) \geq \mathbb{P}(|\tilde{y}|_1 \leq j),$$

which follows from Lemma 6.1 since  $|\tilde{x}|_1 \geq |x|_1$  and  $p \leq 1/2$ . In other words,

$$\sum_{i=0}^j v_i^{(O)} \geq \sum_{i=0}^j v_i^{(f)}$$

for  $0 \leq j \leq k$  and

$$\sum_{i=0}^{k+1} v_i^{(O)} = \sum_{i=0}^{k+1} v_i^{(f)},$$

since we are dealing with probability distributions. Altogether, the vector  $v^{(O)}$  majorizes the vector  $v^{(f)}$ . Since they are based on increasingly restrictive conditions, the  $\tilde{p}_j(t)$  are non-decreasing in  $j$ . Hence,  $\phi$  is Schur-concave (see Theorem A.3 in Chapter 3 of [18]), which proves  $\phi(v^{(f)}) \geq \phi(v^{(O)})$  as desired.  $\square$

### 6.2. Large mutation probabilities

It is not too difficult to show that mutation probabilities  $p = \Omega(n^{\varepsilon-1})$ , where  $\varepsilon > 0$  is an arbitrary constant, make the (1+1) EA (and also the (1+1) EA $_{\mu}$ ) flip too many bits for it to optimize linear functions efficiently.

**Theorem 6.3.** *On any linear function, the optimization time of an arbitrary mutation-based EA with  $\mu = n^{O(1)}$  and  $p = \Omega(n^{\varepsilon-1})$  for some constant  $\varepsilon > 0$ , is bounded from below by  $2^{\Omega(n^{\varepsilon})}$  with probability  $1 - 2^{-\Omega(n^{\varepsilon})}$ .*

**Proof.** Due to Theorem 6.2, it suffices to show the result for the (1+1) EA $_{\mu}$  on ONEMAX. The following two statements follow from Chernoff bounds (and a union bound over  $\mu = n^{O(1)}$  search points in the second statement).

- (1) Due to the lower bound on  $p$ , the probability of a single step not flipping at least  $\lfloor pi/2 \rfloor$  bits out of a set of  $i$  bits is at most  $2^{-\Omega(pi)} = 2^{-\Omega(in^{\varepsilon-1})}$ .
- (2) The search point  $x_{\mu-1}$  has at least  $n/3$  and at most  $2n/3$  one-bits with probability  $1 - 2^{-\Omega(n)}$ .

Furthermore, as we consider ONEMAX, the number of one-bits is non-increasing over time. We assume an  $x_{\mu-1}$  being non-optimal and having at most  $2n/3$  one-bits, which contributes a term of only  $2^{-\Omega(n)}$  to the failure probability. The assumption means that all future search points accepted by the (1+1) EA $_{\mu}$  will have at least  $n/3$  zero-bits. In order to reach the optimum, none of these is allowed to flip. As argued above, the probability of this happening is  $2^{-\Omega(n^{\varepsilon})}$ , and by the union bound, the total probability is still  $2^{-\Omega(n^{\varepsilon})}$  in a number of  $2^{cn^{\varepsilon}}$  steps if the constant  $c$  is chosen small enough. □

Mutation-based EAs have only been defined for  $p \leq 1/2$  since flipping bits with higher probability seems to contradict the idea of a mutation. However, for the sake of completeness, we also analyse the (1+1) EA with  $p > 1/2$  and obtain exponential expected optimization times. Note that we do not know whether ONEMAX is the easiest linear function in this case.

**Theorem 6.4.** *On any linear function, the expected optimization time of the (1+1) EA with mutation probability  $p > 1/2$  is bounded from below by  $2^{\Omega(n)}$ .*

**Proof.** We distinguish between two cases.

**Case 1.**  $p \geq 3/4$ . Here we assume that the initial search point has at least  $n/2$  leading zeros and is not optimal, the probability of which is at least  $2^{-n/2-1}$ . Since the  $n/2$  most significant bits are set correctly in this search point, all accepted search points must have at least  $n/2$  zeros as well. To create the optimum, it is necessary that none of them flip. This occurs only with probability at most  $(1/4)^{n/2}$ , hence the expected optimization time under the assumed initialization is at least  $4^{n/2}$ . Altogether, the unconditional expected optimization time is at least  $2^{-n/2-1} \cdot 4^{n/2} = 2^{\Omega(n)}$ .

**Case 2.**  $1/2 < p \leq 3/4$ . Now the aim is to show that all created search points have a number of ones that is in the interval  $I := [n/8, 7n/8]$  with probability  $1 - 2^{-\Omega(n)}$ . This will imply the theorem by the usual waiting time argument.

Let  $x$  be a search point such that  $|x|_1 \in I$ . We consider the event of mutating  $x$  to some  $x'$  where  $|x'|_1 < n/8$ . Since  $p > 1/2$ , this is most likely if  $|x|_1 = 7n/8$  (using the ideas behind Lemma 6.1 for the complement of  $x$ ). Still, using Chernoff bounds and  $p \leq 3/4$ , at least  $(1/5) \cdot (7n/8) > n/8$  one-bits are not flipped with probability  $1 - 2^{-\Omega(n)}$ . By a symmetrical argument, the probability is  $2^{-\Omega(n)}$  that  $|x'|_1 > 7n/8$ .  $\square$

As was to be expected, no polynomial expected optimization times were possible for the range of  $p$  considered in this subsection.

### 6.3. Small mutation probabilities

We now turn to mutation probabilities that are bounded from above by roughly  $1/n^{2/3}$ . Here relatively precise lower bounds can be obtained.

**Theorem 6.5.** *On any linear function, the expected optimization time of an arbitrary mutation-based EA with  $\mu = n^{O(1)}$  and  $p = O(n^{-2/3-\varepsilon})$  is bounded from below by*

$$(1 - o(1))(1 - p)^{-n}(1/p) \min\{\ln n, \ln(1/(p^3 n^2))\}.$$

As a consequence of Theorem 6.5, we obtain that the bound from Theorem 4.1 is tight (up to lower-order terms) for the (1+1) EA as long as  $\ln(1/(p^3 n^2)) = \ln n - o(\ln n)$ . This condition is weaker than  $p = O((\ln n)/n)$ . If  $p = \omega((\ln n)/n)$  or  $p = o(n^{-C})$  for every constant  $C > 0$  then Theorem 6.5 in conjunction with Theorem 6.3 and 6.4 imply superpolynomial expected optimization time. Thus, the bounds are tight for all  $p$  that allow polynomial optimization times.

Before giving the proof, we state another important consequence, implying the statement from Theorem 3.1 that using the (1+1) EA with mutation probability  $1/n$  is optimal for any linear function.

**Corollary 6.6.** *On any linear function, the expected optimization time of a mutation-based EA with  $\mu = n^{O(1)}$  and  $p = c/n$ , where  $c > 0$  is a constant, is bounded from below by  $(1 - o(1))((e^c/c)n \ln n)$ . If  $p = \omega(1/n)$  or  $p = o(1/n)$ , the expected optimization time is  $\omega(n \ln n)$ .*

**Proof.** The first statement follows immediately from Theorem 6.5 using  $(1 - c/n)^{-n} \geq e^c$  and  $\ln(1/(p^3 n^2)) = \ln n - O(\ln c)$ . The second one follows, depending on  $p$ , either from Theorem 6.3 or, in that case assuming  $p = O((\ln n)/n)$ , from Theorem 6.5, noting that

$$(1 - p)^{-n}(1/p) \geq e^{np}/p = \omega(n)$$

if  $p = \omega(1/n)$  or  $p = o(1/n)$ .  $\square$

Recall that by Theorem 6.2, it is enough to prove Theorem 6.5 for the (1+1) EA $_{\mu}$  on ONEMAX. As mentioned above, this is a well-studied function, for which strong upper and

lower bounds are known in the case  $p = 1/n$ . Our result for general  $p$  is inspired by the proof of Theorem 1 of Doerr, Fouz and Witt [3], which uses an implicit multiplicative drift theorem for lower bounds. Therefore, we now need an upper bound on the multiplicative drift, which is given by the following generalization of Lemma 6 in [4].

**Lemma 6.7.** *Consider (1+1) EA with mutation probability  $p$  for the minimization of ONEMAX. Given a current search point with  $i$  one-bits, let  $I'$  denote the number of one-bits in the subsequent search point (after selection). Then we have  $\mathbb{E}(i - I') \leq ip(1 - p + ip^2/(1 - p))^{n-i}$ .*

**Proof.** Note that  $I' \leq i$  since the number of one-bits in the process is non-increasing. Hence, only mutations that flip at least as many one-bits as zero-bits have to be considered. The event that the total number of one-bits is decreased by  $k \geq 0$  can be partitioned into the subevents  $F_{k,j}$  that  $k + j$  one-bits and  $j$  zero-bits flip, for all  $j \in \mathbb{Z}_0^+$ . The probability of an individual event  $F_{k,j}$  equals

$$\binom{i}{k+j} \binom{n-i}{j} p^{k+2j} (1-p)^{n-k-2j},$$

where  $\binom{a}{b} := 0$  for  $b > a$ . Thus, we have

$$\begin{aligned} \mathbb{E}(i - I') &\leq \sum_{k=1}^i k \sum_{j \geq 0} \binom{i}{k+j} \binom{n-i}{j} p^{k+2j} (1-p)^{n-k-2j} \\ &\leq \underbrace{\sum_{k=1}^i k \binom{i}{k} p^k (1-p)^{n-k}}_{=:S_1} \cdot \underbrace{\sum_{j=0}^{n-i} i^j \binom{n-i}{j} \left(\frac{p}{1-p}\right)^{2j}}_{=:S_2}, \end{aligned}$$

where the second inequality uses  $\binom{i}{k+j} \leq i^j \cdot \binom{i}{k}$ . Factoring out  $(1-p)^{n-i}$  of  $S_1$ , we recognize the expected value of a binomial distribution with parameters  $i$  and  $p$ , which means  $S_1 = (1-p)^{n-i} \cdot ip$ . Regarding  $S_2$ , we apply the Binomial Theorem and obtain  $S_2 = (1 + i(p/(1-p))^2)^{n-i}$ . The product of  $S_1$  and  $S_2$  is the upper bound from the lemma. □

The proof of Theorem 6.5 will use the preceding lemma and Theorem 2.2, the lower-bound version of the multiplicative drift theorem. To make this paper self-contained, we fully prove Theorem 2.2 now before completing the proof of Theorem 6.5. Note, however, that a very similar drift theorem for lower bounds was also proved by Lehre and Witt [17, Theorem 5].

The proof of Theorem 2.2 uses the following additive drift theorem.

**Theorem 6.8 (Jägersküpper [14]).** *Let  $X^{(1)}, X^{(2)}, \dots$  be random variables with bounded support and let  $T$  be the stopping time defined by  $T := \min\{t \mid X^{(1)} + \dots + X^{(t)} \geq g\}$  for a given  $g > 0$ . If  $\mathbb{E}(T)$  exists and  $\mathbb{E}(X^{(i)} \mid T \geq i) \leq u$  for  $i \in \mathbb{N}$ , then  $\mathbb{E}(T) \geq g/u$ .*

In addition, we need the following simple fact.

**Fact 6.9.** *Let  $X$  be a random variable with finite expectation, and  $k$  any real number. If  $\mathbb{P}(X < k) > 0$ , then  $\mathbb{E}(X) \geq \mathbb{E}(X \mid X < k)$ .*

**Proof.** [of Theorem 2.2] The proof generalizes the proof of Theorem 1 in Doerr, Fouz and Witt [3]. We fix an arbitrary starting value  $s_0$  and analyse the random variable  $T$  under the condition  $X^{(0)} = s_0$ . Note that  $T$  is non-negative. Hence, its expectation  $\mathbb{E}(T)$  is either positive infinite or positive finite. In the first case, the theorem is trivial. From now on, we consider the case of finite  $\mathbb{E}(T)$ .

We fix an arbitrary  $t \geq 0$  and an arbitrary  $s > s_{\min}$  where  $\mathbb{P}(X^{(t)} = s) > 0$ . In the following, we condition on the joint event  $(T > t \wedge X^{(t)} = s)$ , but we omit stating this event in the expectations for notational convenience. We define the sequence of random variables  $Y^{(t)} := \ln(X^{(t)})$  (note that  $X^{(t)} \geq 1$ ), and apply Theorem 6.8 with respect to the random variables

$$\Delta_{t+1}(s) := (Y^{(t)} - Y^{(t+1)}) = \left( \ln \left( \frac{s}{X^{(t+1)}} \right) \right).$$

We consider the time until  $X^{(t)} \leq s_{\min}$  (still assuming  $X^{(0)} = s_0$ ) and use the parameter  $g := \ln(s_0/s_{\min})$ . The expectation of  $\Delta_{t+1}(s)$  can be expressed as

$$\begin{aligned} &\mathbb{P}(s - X^{(t+1)} \geq \beta s) \cdot \mathbb{E}(\Delta_{t+1}(s) \mid s - X^{(t+1)} \geq \beta s) \\ &+ \mathbb{P}(s - X^{(t+1)} < \beta s) \cdot \mathbb{E}(\Delta_{t+1}(s) \mid s - X^{(t+1)} < \beta s). \end{aligned} \tag{6.6}$$

By applying the second condition from the theorem, the first term in (6.6) can be bounded from above by  $(\beta\delta/\ln s) \ln s = \beta\delta$ . The logarithm function is concave. Hence, by Jensen’s inequality the second term in (6.6) is at most

$$\ln \left( \mathbb{E} \left( \frac{s}{X^{(t+1)}} \mid s - X^{(t+1)} < \beta s \right) \right) = \ln \left( 1 + \mathbb{E} \left( \frac{s - X^{(t+1)}}{X^{(t+1)}} \mid s - X^{(t+1)} < \beta s \right) \right).$$

By using the inequality  $\ln(1 + x) \leq x$  as well as the conditions  $X^{(t+1)} \geq (1 - \beta)s$  and  $X^{(t+1)} \leq X^{(t)}$ , this can be bounded by

$$\mathbb{E} \left( \frac{s - X^{(t+1)}}{X^{(t+1)}} \mid s - X^{(t+1)} < \beta s \right) < \mathbb{E} \left( \frac{s - X^{(t+1)}}{(1 - \beta)s} \mid s - X^{(t+1)} < \beta s \right).$$

By Fact 6.9 and the first condition from the theorem, it follows that the second term in (6.6) is at most

$$\mathbb{E} \left( \frac{s - X^{(t+1)}}{(1 - \beta)s} \right) \leq \frac{\delta}{1 - \beta}.$$

Altogether, we obtain  $\mathbb{E}(\Delta_{t+1}(s)) \leq (\beta + 1/(1 - \beta))\delta \leq ((\beta + 1)/(1 - \beta))\delta$ . From Theorem 6.8, it now follows that

$$\mathbb{E}(T \mid X^{(0)} = s_0) \geq \frac{1}{\delta} \cdot \frac{1 - \beta}{1 + \beta} \cdot \ln \left( \frac{s_0}{s_{\min}} \right). \quad \square$$

Now we are ready to prove the desired lower bound on the expected optimization time.

**Proof of Theorem 6.5.** As already mentioned, we may assume that the linear function is ONEMAX and that the algorithm is the (1+1) EA<sub>μ</sub>. Then the proof mainly applies Theorem 2.2 for a suitable choice of its various parameters. Let  $\tilde{p} := \max\{p, 1/n\}$ . We first observe that the probability of flipping at least  $b := \tilde{p}n \ln n$  bits in a single step is bounded from above by

$$\binom{n}{\tilde{p}n \ln n} \cdot p^{\tilde{p}n \ln n} \leq \left(\frac{e\tilde{p}n}{\tilde{p}n \ln n}\right)^{\tilde{p}n \ln n} = 2^{-\Omega(\tilde{p}n(\ln n)(\ln \ln n))},$$

where we have used  $p \leq \tilde{p}$ . Hence, the probability is superpolynomially small. In the following, we assume that the number of one-bits changes by at most  $b$  in each of a total number of at most  $(1 - p)^{-n} \ln n = 2^{O(\tilde{p}n) + O(\ln \ln n)}$  steps that are considered for the lower bound we want to prove. This event holds with probability  $1 - o(1)$ , which decreases the bound only by a factor of  $1 - o(1)$ .

Let  $X^{(t)}$  denote the number of one-bits at time  $t$  and note that this is non-increasing over time. We choose  $s_{\min} := n\tilde{p} \ln^2 n$  and  $\beta := 1/\ln n$  and introduce  $s_{\max} := 1/(2\tilde{p}^2 n \ln n)$  as an additional upper bound. Note that  $s_{\max} \leq n/(2 \ln n)$  due to  $\tilde{p} \geq 1/n$ . Since the  $\mu$  initial search points are drawn uniformly at random and  $\mu = n^{O(1)}$ , it holds that  $X_\mu \geq s_{\max}$  with probability  $1 - o(1)$ . Again, assuming this occurs, we lose a factor  $1 - o(1)$  in the bound we want to prove. Moreover, due to our assumption  $p = O(n^{-2/3-\epsilon})$  (which means  $\tilde{p} = O(n^{-2/3-\epsilon})$ ), we have  $b = n\tilde{p} \ln n \leq 1/(4\tilde{p}^2 n \ln n) = s_{\max}/2$  for  $n$  large enough. Altogether, it holds that  $s_{\max}/2 \leq X_{t^*} \leq s_{\max}$  at the first point of time  $t^*$  where  $X_{t^*} \leq s_{\max}$ . To simplify issues, we consider the process only from time  $t^*$  on. Skipping the first  $t^*$  steps, we pessimistically assume  $s_0 := s_{\max}/2$  as a starting point and  $X^{(t)} \leq s_{\max}$  for all  $t \geq 0$ . The second condition of the drift theorem is now fulfilled since the bound on  $\tilde{p}$  also implies  $b = \tilde{p}n \ln n \leq 1/(2\tilde{p}^2 n \ln^2 n) = \beta s_{\max}$ , where  $\beta s_{\max}$  is the largest value for  $\beta s$  to be taken into account.

Assembling the factors from the lower bound in Theorem 2.2, we obtain  $\frac{1-\beta}{1+\beta} = 1 - o(1)$ . Furthermore, we have  $\ln(s_0/s_{\min}) = \ln(1/(4\tilde{p}^3 n^2 \ln^3 n)) = \ln(1/(\tilde{p}^3 n^2)) - O(\ln \ln n)$ , which is  $(1 - o(1)) \ln(1/(\tilde{p}^3 n^2))$  by our assumption on  $\tilde{p}$ . If we can prove that  $1/\delta = (1 - o(1))(1 - p)^{-n}(1/p)$ , the proof is complete.

To bound  $\delta$ , we use Lemma 6.7. Note that  $i \leq s_{\max}$  holds in our simplified process. Using the lemma and recalling that  $1/\tilde{p} \leq 1/p$ , we get

$$\begin{aligned} \frac{\mathbb{E}(X^{(t)} - X^{(t+1)} \mid X^{(t)} = i)}{i} &\leq p \left(1 - p + \frac{s_{\max} p^2}{1 - p}\right)^{n - s_{\max}} \\ &\leq p \left(1 - p + \frac{1}{n \ln n}\right)^{n - s_{\max}} \\ &\leq p \left((1 - p) \left(1 + \frac{2}{n \ln n}\right)\right)^{n - s_{\max}} \\ &= (1 + o(1))p(1 - p)^n, \end{aligned}$$

where we have used  $p \leq 1/2$  and  $(1 + 2/(n \ln n))^n = 1 + o(1)$  and

$$(1 - p)^{-s_{\max}} = (1 - p)^{-1/(2\tilde{p}^2 n \ln n)} = 1 + o(1).$$

Hence,  $1/\delta \geq (1 - o(1))(1/p)(1 - p)^{-n}$  as suggested, which completes the proof.  $\square$

Finally, we remark that the expected optimization time of the (1+1) EA with  $p = 1/n$  on ONEMAX is known to be  $en \ln n - \Theta(n)$  [4]. Hence, in conjunction with Theorems 5.1 and 6.2, we obtain for  $p = 1/n$  that the expected optimization time of the (1+1) EA varies by at most an additive term  $\Theta(n)$  within the class of linear functions.

## 7. Conclusions

We have presented new bounds on the expected optimization time of the (1+1) EA on the class of linear functions. These bounds are now tight up to lower-order terms, which applies to any mutation probability  $p = O((\ln n)/n)$ . This means that  $1/n$  is the optimal mutation probability on any linear function. We have for the first time studied the case  $p = \omega(1/n)$  and proved a transition from polynomial to exponential running time in the region  $\Theta((\ln n)/n)$ . The lower bounds show that ONEMAX is the easiest linear function for all  $p \leq 1/2$ , and they apply not only to the (1+1) EA but also to the large class of mutation-based EAs. This means that the (1+1) EA is an optimal mutation-based algorithm on linear functions. The upper bounds hold with high probability. As proof techniques, we have employed multiplicative drift in conjunction with adaptive potential functions. In the future, we hope to see these techniques applied to the analysis of other randomized search heuristics.

We finish with an open problem. Even though our proofs of upper bounds would simplify for the function BINVAL, this function is often considered as a worst case. Is it true that the runtime of the (1+1) EA on BINVAL is stochastically largest within the class of linear functions, thereby complementing the result that the runtime on ONEMAX is stochastically smallest?

## Acknowledgements

The author thanks Benjamin Doerr, Timo Kötzing, Per Kristian Lehre and Carola Winzen for constructive discussions on the subject. Moreover, he thanks Daniel Johannsen for proofreading a draft of Sections 4 and 5 and pointing out a simplification of the proof of Theorem 4.1. Finally, he thanks Dirk Sudholt, who suggested studying the class of mutation-based EAs.

## References

- [1] Auger, A. and Doerr, B. (2011) *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, World Scientific.
- [2] Bäck, T. (1993) Optimal mutation rates in genetic search. In *Proc. International Conference on Genetic Algorithms, ICGA '93*, Morgan Kaufmann, pp. 2–8.
- [3] Doerr, B., Fouz, M., and Witt, C. (2010). Quasirandom evolutionary algorithms. In *Proc. Genetic and Evolutionary Computation Conference, GECCO '10*, ACM Press, pp. 1457–1464.
- [4] Doerr, B., Fouz, M., and Witt, C. (2011). Sharp bounds by probability-generating functions and variable drift. In *Proc. Genetic and Evolutionary Computation Conference, GECCO '11*, ACM Press, pp. 2083–2090.

- [5] Doerr, B. and Goldberg, L. A. (2010). Adaptive drift analysis. In *Proc. Parallel Problem Solving from Nature XI, PPSN '10*, Vol. 6238 of *Lecture Notes in Computer Science*, Springer, pp. 32–41.
- [6] Doerr, B. and Goldberg, L. A. (2012). Adaptive drift analysis. *Algorithmica*, in press. Available online: <http://www.springerlink.com/content/v0n75x65k016r477>
- [7] Doerr, B., Johannsen, D., and Winzen, C. (2010). Drift analysis and linear functions revisited. In *Proc. Congress on Evolutionary Computation, CEC '10*, IEEE Press, pp. 1–8.
- [8] Doerr, B., Johannsen, D., and Winzen, C. (2010). Multiplicative drift analysis. In *Proc. Genetic and Evolutionary Computation Conference, GECCO '10*, ACM Press, pp. 1449–1456.
- [9] Droste, S., Jansen, T., and Wegener, I. (2000). A natural and simple function which is hard for all evolutionary algorithms. In *Proc. 26th Annual Conference of the IEEE Industrial Electronics Society, IECON '00*, pp. 2704–2709.
- [10] Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoret. Comput. Sci.* **276** 51–81.
- [11] Hajek, B. (1982). Hitting-time and occupation-time bounds implied by drift analysis with applications. *Adv. Appl. Probab.* **13** 502–525.
- [12] He, J. and Yao, X. (2001). Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence* **127** 57–85.
- [13] He, J. and Yao, X. (2004). A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing* **3** 21–35.
- [14] Jägersküpper, J. (2007). Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoret. Comput. Sci.* **379** 329–347.
- [15] Jägersküpper, J. (2008). A blend of Markov-chain and drift analysis. In *Proc. Parallel Problem Solving from Nature X, PPSN '08*, Vol. 5199 of *Lecture Notes in Computer Science*, Springer, pp. 41–51.
- [16] Jägersküpper, J. (2011). Combining Markov-chain analysis and drift analysis: The (1+1) evolutionary algorithm on linear functions reloaded. *Algorithmica* **59** 409–424.
- [17] Lehre, P. K. and Witt, C. (2012). Black-box search by unbiased variation. *Algorithmica* **64** 623–642.
- [18] Marshall, A. W., Olkin, I. and Arnold, B. (2011). *Inequalities: Theory of Majorization and its Applications*, second edition, Springer.
- [19] Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms*, Cambridge University Press.
- [20] Mühlenbein, H. (1992). How genetic algorithms really work I: Mutation and hillclimbing. In *Proc. Parallel Problem Solving from Nature I, PPSN '92*, Elsevier, pp. 15–26.
- [21] Neumann, F. and Witt, C. (2010). *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*, Natural Computing Series, Springer.
- [22] Sudholt, D. (2010). General lower bounds for the running time of evolutionary algorithms. In *Proc. Parallel Problem Solving from Nature XI, PPSN '10*, Vol. 6238 of *Lecture Notes in Computer Science*, Springer, pp. 124–133.
- [23] Sudholt, D. (2012). A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, to appear. Preprint: <http://dx.doi.org/10.1109/TEVC.2012.2202241>
- [24] Wegener, I. (2001). Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In *Evolutionary Optimization* (R. Sarker, M. Mohammadian and X. Yao, eds), Kluwer.
- [25] Wegener, I. and Witt, C. (2005). On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. *J. Discrete Algorithms* **3** 61–78.



- [26] Wegener, I. and Witt, C. (2005). On the optimization of monotone polynomials by simple randomized search heuristics. *Combin. Probab. Comput.* **14** 225–247.
- [27] Witt, C. (2012). Optimizing linear functions with randomized search heuristics: The robustness of mutation. In *Proc. Symposium on Theoretical Aspects of Computer Science, STACS '12*, Leibniz International Proceedings in Informatics, pp. 420–431. Open access: <http://drops.dagstuhl.de/opus/volltexte/2012/3392/>.