Design and Analysis of an Asymmetric Mutation Operator

Thomas Jansen and Dirk Sudholt

No. CI-234/07 (revised version of CI-195/05)

Technical Report ISSN 1433-3325 November 2007

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the Technical University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

Design and Analysis of an Asymmetric Mutation Operator[*]

Thomas Jansen and Dirk Sudholt

FB Informatik, Univ. Dortmund, 44221 Dortmund, Germany
{`Thomas.Jansen, Dirk.Sudholt`}`@udo.edu`

Revised November 2007

Abstract

Evolutionary algorithms are general randomized search heuristics and typically perform an unbiased random search that is guided only by the fitness of the search points encountered. However, in practical applications there is often problem-specific knowledge that suggests some additional bias. The use of appropriately biased variation operators may speed-up the search considerably. Problems defined over bit strings of finite length often have the property that good solutions have only very few 1-bits or very few 0-bits. A specific mutation operator tailored towards such situations is studied under different perspectives and in a rigorous way discussing its assets and drawbacks. This is done by considering illustrative example functions as well as function classes. The main focus is on theoretical run time analysis yielding asymptotic results. These findings are accompanied by the results of empirical investigations that deliver additional insights.

1 Introduction

General randomized search heuristics are often applied in the context of optimization when there is not enough knowledge, time, or expertise to design problem-specific algorithms. One popular example belonging to this class of algorithms are evolutionary algorithms. When analyzing such algorithms, one typically assumes that nothing is known about the objective function at hand and that function evaluations are the only way to gather knowledge about it. This optimization scenario is called black-box optimization [7] and it leads to the well-known no free lunch theorem (NFL) when taken to its extreme: when there is no structural knowledge at all, then all algorithms have

[*]This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

equal performance [11]. In practical applications, such a scenario is hardly ever realistic since there is almost always some knowledge about typical solutions. It is well known that incorporating problem-specific knowledge can be crucial for the success and the efficiency of evolutionary algorithms (see [3, 18, 4] for some recent studies).

Here, we consider one specific mutation operator for binary strings that is plausible when good solutions contain either very few bits with value 0 or very few bits with value 1. Many real-world problems share this specific property. One example is the problem of computing a minimum spanning tree [2]. A bit string $x \in \{0,1\}^n$ represents an edge set where each bit corresponds to exactly one edge of the graph and the selected edges correspond to bits with value 1. Most graphs with m nodes contain $\Theta(m^2)$ edges whereas trees contain only $m-1$ edges. So in this case all feasible solutions contain only a very small number of bits with value 1.

The most common mutation operator for bit strings of length n flips each bit independently with probability $1/n$. But for the minimum spanning tree problem, this standard mutation operator is not well suited. After some time, one typically has some feasible solutions and waits for different feasible solutions to be found via mutations. In case we have got a bit string representing a spanning tree and we wait for another spanning tree to be generated by mutation, the operator tends to create offspring with more than $m-1$ edges. Biasing the search towards strings with few 1-bits leads to a significant speed-up for this problem. Motivated by this finding, Neumann and Wegener [16] suggested the asymmetric mutation operator analyzed here. This mutation operator tends, on average, to preserve the number of 1-bits.

In the search space $\{0,1\}^n$ one can think of all points with exactly i 1-bits as forming the ith level. Clearly, for $i = O(1)$ and $i = n - O(1)$ the levels contain only a polynomial number of points whereas the levels with $i \approx n/2$ are exponentially large. Imagine a random walk on $\{0,1\}^n$ induced by repeated standard bit mutations only. Standard bit mutations flip each bit independently with some fixed mutation probability, typically $1/n$ for bit strings of length n. Thus, standard bit mutations tend to sample the search space uniformly. This implies that the random walk induced by repeated standard bit mutations spends most of the time on levels with $i \approx n/2$. When reaching a search point x with either very few or lots of 1-bits, there is a strong tendency to return to levels $i \approx n/2$ since these levels have a much larger size.

The asymmetric mutation operator considered here is likely to preserve the current level, on average. However, considering a random walk induced by repeated asymmetric mutations, variance lets the random walk change the current level. Since there is no tendency to the medium levels, the random walk is more likely to reach levels with very few or lots of 1-bits than the random walk based on standard bit mutations.

In an optimization process, if the fitness values encountered guide the search towards areas of the search space where the number of 1-bits is either small or large, this mutation operator is more efficient in generating other such search points at random. However, the mutation operator is not custom-built with one specific application in mind. It is still a quite general mutation operator that we consider to be a natural choice when it is known that good solutions to the optimization problem at hand have either very few or lots of 1-bits. It has to be noted, though, that it is not an unbiased operator as defined by Droste and Wiesmann [8] (assuming Hamming distance to be a natural metric in $\{0,1\}^n$).

This paper is not about one specific mutation operator for a specific kind of problem and the demonstration of its usefulness. Our aim is to present a broad and informative analysis of this mutation operator. We consider its performance on illustrative example functions and on interesting classes of functions. All example functions considered here have been introduced elsewhere and for completely different reasons. Thus, they are not designed with this mutation operator in mind. With this approach we are able to demonstrate the assets and drawbacks of this specific mutation operator in a clear and intuitive, yet rigorous way.

Our analysis has a strong emphasis on theoretical results leading to proven theorems. However, we aim at presenting an interesting and readable report on the investigation of a specific mutation operator. Therefore, we make the reader familiar with the central ideas for proving the presented results and refer the interested reader to the appendix for formal proofs.

Following the usual approach in the analysis of (randomized) algorithms, the difficulty of the analysis is reduced by not taking care of the algorithms in all tiny details but concentrating on the major effects. This leads to asymptotic results that are expressed using the usual notions for (expected) run times. We accompany these theoretical findings with the results of empirical investigations. Clearly, such empirical data can neither prove nor disprove the theoretical findings: proving general statements by empirical data only is not possible for logical reasons. While in general theorems can be falsified by a single counter-example, no empirical data can represent a counter-example for purely asymptotical results. Nevertheless, the presentation of such empirical data has several advantages. It gives a clearer impression of the actual run times for realistic values of n. It demonstrates the influence of the lower order terms that are hidden in the asymptotic notation. And it gives insights to effects that are difficult to deal with in a theoretical analysis. We believe that the combination of theoretical results with empirical findings delivers a more complete picture[1]. While presenting a concrete analysis for one concrete mutation operator we hope that

[1]A preliminary version with parts of the theoretical results without the empirical data was presented at a conference [13].

this analysis can serve as an example of how a thorough analysis of new operators and variants of evolutionary algorithms can be presented.

In the following section, we define the mutation operator, the evolutionary algorithm we consider, and our analytical framework. In Section 3, we concentrate on the performance on simply structured example functions and demonstrate that the operator shows increased efficiency as expected. This helps to build a more concrete intuition of the properties of the asymmetric mutation operator. In a more general context, we explain in Section 4 that the performance on a broad and interesting class of functions does not differ from that of an unbiased mutation operator. Section 5 presents an example where the bias introduced by the mutation operator has an immense negative impact. Finally, we conclude in Section 6 with some remarks about possible future research. As already remarked above, the appendix contains the formal proofs for the results in the different sections.

2 Definitions

In order to concentrate on the effects of the mutation operator we consider an evolutionary algorithm that is as simple as possible. This leads us to the well-known (1+1) EA, a kind of stochastic hill-climber.

Algorithm 1 ((1+1) EA)**.**

```
1. Initialization
   Choose $x \in \{0,1\}^n$ uniformly at random.
2. Mutation
   $y := mutate(x)$.
3. Selection
   If $f(y) \geq f(x)$, $x := y$.
4. Stopping Criterion
   If the stopping criterion is not met, continue at line 2.
```

Most often the (1+1) EA is applied using standard bit mutations. We give a formal definition of this mutation operator.

Mutation Operator 1 (Standard Bit Mutations)**.** *Independently for each bit in $x \in \{0,1\}^n$, flip the bit with probability $1/n$.*

The asymmetric mutation operator that we consider aims at leaving the number of bits with value 1 unchanged. This can be achieved by letting the probability to mutate a bit depend on its value. In order to give a formal definition, we introduce the following notations. For a bit string $x = x_1 x_2 \cdots x_n$ we denote the number of bits with value 1 in x by $|x|_1$, i. e.,

$|x|_1 = \sum_{i=1}^{n} x_i$. Analogously, $|x|_0$ denotes the number of bits with value 0, i. e., $|x|_0 = n - |x|_1$.

Mutation Operator 2 (Asymmetric Bit Mutations)**.** *Independently for each bit in $x \in \{0,1\}^n$, flip the bit with probability $1/(2\,|x|_1)$ if it has value 1 and with probability $1/(2\,|x|_0)$ otherwise.*

In the following, we refer to the (1+1) EA with asymmetric bit mutations as the asymmetric (1+1) EA. We use $1/(2|x|_i)$ as mutation probability instead of $1/|x|_i$ to prevent the mutation operator from becoming deterministic in the special case of exactly one bit with value 0 or 1. In this deterministic case the property that any $y \in \{0,1\}^n$ can be reached by any $x \in \{0,1\}^n$ in one mutation is not preserved. The value 2 is a straightforward choice since for any x with $0 < |x|_1 < n$ the expected number of flipping bits is 1. This coincides with the expected number of flipping bits for standard bit mutations.

One may think that using a different initialization with asymmetric mutations makes more sense. Using the asymmetric mutation operator is motivated by the wish to bias the search towards regions of the search space with either lots of or few 0-bits. Thus, it seems reasonable to choose the initial population in this region. While such considerations are sensible in practical applications they are rather disadvantageous, here. Since we aim at a fair and meaningful comparison of two different mutation operators, we rather keep any other aspect of the evolutionary algorithm serving as a framework unchanged.

Theoretical results concerned with the performance of evolutionary algorithms as optimizers often concentrate on the expected optimization time, i. e., the expected run time until some optimal point in the search space is found. As usual we consider the number of function evaluations to be an accurate measure for the actual run time.

The results on the expected run time are asymptotic ones. They use the well-known notation for the order of growth (see for example [2]). For the sake of completeness, we give a definition.

Definition 1. *Let $f, g\colon \mathbb{N}_0 \to \mathbb{R}$ be two functions. We say f grows at most as fast as g and write $f = O(g)$ iff there exist $n_0 \in \mathbb{N}$ and $c \in \mathbb{R}^+$ such that for all $n \geq n_0$ we have $f(n) \leq c \cdot g(n)$. We say f grows at least as fast as g and write $f = \Omega(g)$ iff $g = O(f)$. We say f and g have the same order of growth and write $f = \Theta(g)$ iff $f = O(G)$ and $f = \Omega(g)$. We say f grows faster then g and write $f = \omega(g)$ iff $\lim\limits_{n\to\infty} g(n)/f(n) = 0$. We say f grows slower than g and write $f = o(g)$ iff $g = \omega(f)$.*

For many objective functions, mutations of single bits turn out to be important leading from x to a so-called Hamming neighbor y. For $x, y \in$

$\{0,1\}^n$ the Hamming distance $\mathrm{H}(x,y)$ equals the number of bits where x and y differ, i. e., $\mathrm{H}(x,y) = \sum_{i=1}^{n} (x_i \oplus y_i)$ where $x_i \oplus y_i$ denotes the exclusive or of x_i and y_i. Using standard bit mutations, steps to Hamming neighbors have probability $\Theta(1/n)$. It is important to note that asymmetric bit mutations do not decrease the probability of such mutations significantly. They may, however, increase the probability for such steps significantly.

Lemma 1. *Let $x, y \in \{0,1\}^n$ with $H(x,y) = 1$ be given. The probability to mutate x into y in one asymmetric mutation is bounded below by $1/(8\,|x|_i)$ if a bit with value i needs to flip.*

Proof. Assume that one 0-bit in x needs to flip; the other case is symmetric. For $x \neq 0^n$, the probability to flip exactly this bit equals

$$\frac{1}{2\,|x|_0}\left(1-\frac{1}{2\,|x|_0}\right)^{|x|_0-1}\left(1-\frac{1}{2\,|x|_1}\right)^{|x|_1} \geq \frac{1}{8\,|x|_0}$$

since $(1-1/(2k))^k \geq 1/2$ for $k \in \mathbb{N}$. For $x = 0^n$ we obtain $1/(4\,|x|_0) > 1/(8\,|x|_0)$ as lower bound in the same way. □

Like for standard bit mutations, the probability to flip k bits simultaneously in one mutation decreases drastically with k. We show exemplarily that the probability of flipping $\Omega(n^\varepsilon)$ bits at once is exponentially small. The proof is a simple application of Chernoff bounds, it can be found in the appendix.

Lemma 2. *Let $\varepsilon > 0$. The probability that the asymmetric mutation operator flips $\Omega(n^\varepsilon)$ bits in one step is bounded above by $2^{-\Omega(n^\varepsilon \log n)}$.*

When using standard bit mutations, the probability to mutate some $x \in \{0,1\}^n$ into some $y \in \{0,1\}^n$ is determined only by the number of bits with different values in x and y, i. e., their Hamming distance $\mathrm{H}(x,y)$. Therefore, one easy way of generalizing results on specific example functions to results on function classes is to group functions that are essentially equal but differ only in their "coding." We adopt the definition and notation of [5] where the generalization of objective functions is considered in the context of black-box complexity.

Definition 2. *For $f\colon \{0,1\}^n \to \mathbb{R}$ and $a \in \{0,1\}^n$ we define $f_a\colon \{0,1\}^n \to \mathbb{R}$ by $f_a(x) := f(x \oplus a)$ for all $x \in \{0,1\}^n$ where $x \oplus a$ denotes the bit-wise exclusive or of x and a.*

Since the (1+1) EA with standard bit mutations is insensitive to the number of 1-bits in the current bit string and since it treats 1-bits and 0-bits symmetrically, it exhibits the same behavior on f as on f_a for any a. So, the class of functions f_a is a straightforward generalization of f. When we use asymmetric bit mutations instead, this is not necessarily the case. Transforming x to $x \oplus a$ does in general change the number of 1-bits and therefore alters the mutation probabilities. This is one way to describe how the asymmetric mutation operator biases the search. We will consider this in greater detail in the following section. Note, however, that the (1+1) EA with asymmetric bit mutations behaves the same on f_a and $f_{\overline{a}}$ where $\overline{a}$ denotes the bit-wise complement of a. This is due to the symmetrical roles of 0 and 1 as bit values if one replaces all 0s by 1s and vice versa. Therefore, it suffices to consider functions f_a with $|a|_1 \leq n/2$, only. We adopt the widely used notation b^i for the i concatenations of the letter b. Thus, the all-one bit string of length n can be written as 1^n.

We start our analysis with some very general bounds on the expected optimization time of the asymmetric (1+1) EA that only rely on basic properties of the asymmetric mutation operator. The same considerations have also been made for the (1+1) EA with standard bit mutations, so we have a direct comparison of both mutation operators. For the (1+1) EA with standard bit mutations it is known that the expected optimization time is bounded from above by n^n for all functions [6] and bounded from below by $\Omega(n \log n)$ if the global optimum is unique [12].

Theorem 1. *The expected optimization time of the asymmetric (1+1) EA on any function $f\colon \{0,1\}^n \to \mathbb{R}$ is at most $(2n)^n$. If f has a unique global optimum, the expected optimization time is bounded below by $n/2$.*

Proof. As long as the current population x is not a global optimum, the probability to hit an optimum x^* by a direct jump is bounded below by $(1/(2n))^n$. Hence the expected time to hit a global optimum is at most $(2n)^n$.

For the lower bound we observe that the expected number of flipping bits when mutating x equals 1 for $x \notin \{0^n, 1^n\}$ and 1/2 for $x \in \{0^n, 1^n\}$. This implies that the expected progress towards the optimum in one generation is at most 1. As the initial population has expected Hamming distance $n/2$ to the unique global optimum, a drift analysis (similar to the technique presented by He and Yao [10]) shows that the expected optimization time is at least $n/2$. □

We only remark that a lower bound $\Omega(n)$ can also be shown for functions with multiple global optima as long as the number of global optima is $2^{o(n)}$.

Comparing standard bit mutations to asymmetric mutations, Theorem 1 shows no clear advantage for any operator. We also see that the resulting bounds are weak due to the weak assumptions made on the fitness function. This motivates the investigation of concrete functions for which much stronger results can be obtained.

There is a number of well-known example functions that we want to consider in the following sections. The function OneMax(x) simply counts the number of 1-bits in x. $\textsc{Needle}(x)$ takes value 1 if $x = 1^n$ and 0 otherwise. Since the function does not give any hints to find the optimum, it's like looking for a needle in a haystack. The function $\mathrm{LO}(x)$ counts the number of leading ones in x. On $\textsc{Ridge}(x)$, the function value increases on a ridge of search points 1^i0^{n-i} into the direction of the global optimum and all other search points give hints to reach the start of the ridge. $\textsc{Plateau}(x)$ is very similar to $\textsc{Ridge}$. The only difference is that all search points on the ridge except the global optimum 1^n form a plateau of search points with equal fitness.

We give precise formal definitions.

Definition 3.

$$\begin{aligned}
\textsc{OneMax}(x) &:= |x|_1\\
\textsc{Needle}(x) &:= \prod_{i=1}^{n} x_i\\
\mathrm{LO}(x) &:= \sum_{i=1}^{n}\prod_{j=1}^{i} x_j\\
\textsc{Ridge}(x) &:= \begin{cases} n - |x|_1 & \text{if } x \notin \{1^i0^{n-i} \mid 0 \le i \le n\}\\ n + i & \text{if } x \in \{1^i0^{n-i} \mid 0 \le i \le n\}\end{cases}\\
\textsc{Plateau}(x) &:= \begin{cases} n - |x|_1 & \text{if } x \notin \{1^i0^{n-i} \mid 0 \le i \le n\}\\ n + 1 & \text{if } x \in \{1^i0^{n-i} \mid 0 \le i < n\}\\ n+2 & \text{if } x = 1^n\end{cases}
\end{aligned}$$

A point $x \in \{0,1\}^n$ is a local optimum of a function $f\colon \{0,1\}^n \to \mathbb{R}$ if $f(x) \ge f(y)$ holds for all Hamming neighbors y of x (i. e. $H(x,y) = 1$). A function $f\colon \{0,1\}^n \to \mathbb{R}$ is unimodal iff it has exactly one local optimum.

We now cite results on the expected optimization time of the (1+1) EA with standard bit mutations for the example functions from Definition 3. These results are used as a bottom-line for the comparison when we use asymmetric bit mutations instead.

Theorem 2. *Let $E(T_f)$ denote the expected optimization time of the (1+1) EA with standard bit mutations on the function $f\colon \{0,1\}^n \to \mathbb{R}$.*

$E(T_{\textsc{OneMax}}) = \Theta(n \log n)$ *[6]*

$E(T_{\textsc{Needle}}) = \Theta(2^n)$ *[9]*

$E(T_{\mathrm{LO}}) = \Theta(n^2)$ *[6]*

$E(T_f) = O(nd)$ *for unimodal functions f with d different function values [6]*

$E(T_{\textsc{Ridge}}) = \Theta(n^2)$ *[14]*

$E(T_{\textsc{Plateau}}) = O(n^3)$ *[14]*

It is important to remember that $\mathrm{E}(T_f) = \mathrm{E}(T_{f_a})$ holds for the (1+1) EA with standard bit mutations for any a. We will see that this is different for the (1+1) EA with asymmetric bit mutations and that the performance gap on f and f_a can be exponentially large.

3 Assets of the Asymmetric Mutation Operator

The asymmetric bit mutation operator preserves, on average, the number of 1-bits in the parent. This makes this mutation operator very different from standard bit mutations if the number of 1-bits is either very small or very large. Thus, we expect to obtain best results when good search points have this property and when good search points lead the algorithm to the global optimum. The well-known fitness function OneMax has all these properties. It is therefore not surprising that the asymmetric bit mutation operator can lead to a considerable speed-up.

Theorem 3. *The expected optimization time of the asymmetric (1+1) EA on* OneMax *is* $\Theta(n)$.

Proof. As long as the current search point x differs from the global optimum, there are $|x|_0$ Hamming neighbors with a larger fitness value. Due to Lemma 1, the probability of increasing the fitness value is at least $|x|_0 \cdot 1/(8|x|_0) = 1/8$ and the expected time to increase the fitness is at most 8. Since the fitness value has to be increased at most n times, $8n$ is an upper bound on the expected optimization time. The lower bound follows from Theorem 1. □

Asymmetric mutations outperform the standard mutation operator by a factor of the order of $\log n$ here. However, this relies heavily on the fact that the unique global optimum is the all-one bit string. Clearly, the objective function OneMax can be described as minimizing the Hamming distance to the unique global optimum. This is equivalent to maximizing the Hamming distance to the bit-wise complement of the unique global optimum. We can preserve this property but move the global optimum x^* somewhere else by defining the fitness as $n - \mathrm{H}(x, x^*)$. This leads us exactly to OneMax_a with $a = \overline{x^*}$. One may fear that the advantage of asymmetric bit mutations for OneMax might be counterbalanced by a disadvantage when the global optimum is far away from 1^n. However, this is not the case if one considers asymptotic expected optimization times.

Theorem 4. *For any a the expected optimization time of the asymmetric (1+1) EA on* OneMax_a *is* $\Theta(n \log(\min\{|a|_0, |a|_1\} + 2))$.

A complete proof can be found in the appendix. It is important to observe that the Hamming distance to the unique global optimum cannot increase during the run. This is due to the elitist selection employed and the direct correspondence between function value and Hamming distance. Using this observation, the proof of the upper bound is a straightforward estimation of the expected waiting times for events increasing the current fitness. A lower bound $\Omega(n)$ follows from Theorem 1, proving the claimed lower bound in case $|a|_0 = O(1)$ or $|a|_1 = O(1)$. If the unique global optimum has larger Hamming distance to 0^n and 1^n, we can use a different way of reasoning. We observe that, typically, a linear number of bits needs to flip in order to reach this optimum. It is not difficult to prove a constant lower bound on the probability that there is at least one of these bits that never flips in $O(n \ln |a|_1)$ generations. This implies the desired lower bound on the expected optimization time.

We see that, asymptotically, there is no disadvantage for the (1+1) EA with asymmetric bit mutations in comparison with standard bit mutations on OneMax_a. We complement these asymptotical bounds by the results of some experiments. All reported results are averages of 100 independent runs. For OneMax_a, we choose a with $|a|_1 = c{\cdot}n$ and $c \in \{0, .05, .1, .15, .2, \dots, .95, 1\}$. We choose n, the length of the bit strings, from $\{100, 200, 300, \dots, 1900, 2000\}$. The average optimization times can be seen in Fig. 1. Note that these empirical findings have illustrative purposes. Therefore, we refrain from a statistical analysis that does not yield additional insights.

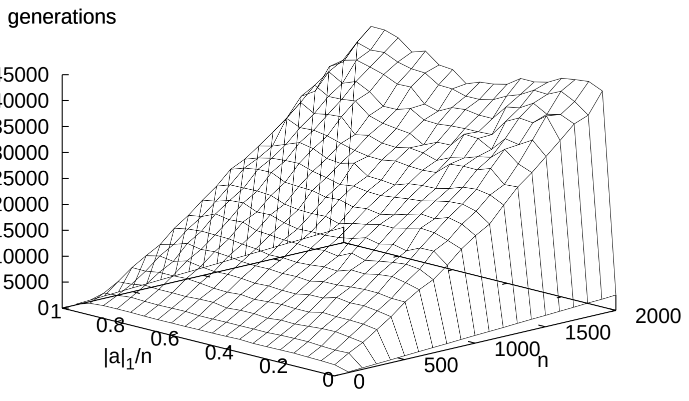

Figure 1: Average run times on OneMax_a for $n \in \{100, 200, \dots, 2000\}$ and $|a|_1 \in \{0, .05n, .1n, \dots, .95n, n\}$.

Two aspects of the empirical data displayed in Fig. 1 deserve some explanation. First, we observe that the average run times are maximal in some distance to the extreme values 0 and n and decrease towards $|a|_1 = n/2$. This effect is not contained in our bound. Note, that the observed differences are small enough not to be visible in the asymptotic notation $\Theta(\cdot)$. Observing concrete runs for $|a|_1 = c{\cdot}n$ for small values of c, one observes that the asymmetric mutation operator has the tendency to lead the algorithm too close to 1^n making final steps back towards the unique global optimum necessary. This causes an increased optimization time.

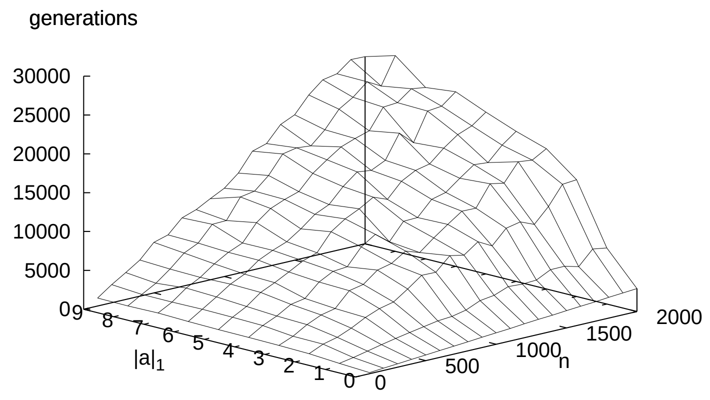

Figure 2: Average run times on OneMax_a for $n \in \{100, 200, \ldots, 2000\}$ and $|a|_1 \in \{0, 1, 2, \ldots, 9\}$.

The second aspect that catches our attention is the steep increase of the average optimization time at the boundaries. It seems to increase from a value very small for $|a|_1 \in \{0, n\}$ to something clearly larger almost immediately. This seems to contradict our theoretical bound $\Theta(n \log(|a|_1 + 2))$. Remembering that $c = .05$ implies $|a|_1 = n/20 = \Theta(n)$ we see that this is not really the case. More values for a close to 0 and n are helpful to see this more clearly. We present average run times for $|a|_1 \in \{0, 1, 2, \ldots, 9\}$ in Fig. 2. There we see that the run time increases with $|a|_1$ quite smoothly as our bound predicts. We learn that the investigation of observed run times alone without a theoretical analysis may be quite misleading.

The reader might conclude from our findings that the search is not too clearly biased by asymmetrical bit mutations. However, for OneMax_a, the function values point into the direction of the global optimum so clearly that the relatively small bias introduced by the asymmetric mutations is not important when compared to the clear bias introduced by selection. Thus, optimization is quite efficient regardless of the location of the unique global optimum.

In the following, we show that there is a clear bias due to asymmetric bit mutations that can have a great impact on the performance of the asymmetric (1+1) EA. We consider the asymmetric (1+1) EA on a flat fitness function: we consider Needle. Since all non-optimal search points are equally fit, we exclude the effects of selection on the optimization process and, as long as the needle is not found, the search process equals the random walk induced be repeated asymmetrical bit mutations. So, by considering the function Needle with the needle in 1^n, we can learn more about the bias induced by asymmetric bit mutations.

As long as the needle has not been found, the asymmetric (1+1) EA is symmetric w. r. t. bit positions. Hence, we can describe the asymmetric (1+1) EA as a discrete Markov chain where the transition probabilities depend on the number of 1-bits in the current bit string, only. The main observation is that, as long as no search point in $\{0^n, 1^n\}$ is reached, this process is a martingale (see [1]), i. e., the expected change in the number of 1-bits over time is 0: let $x \in \{0,1\}^n$ with $0 < |x|_1 < n$ and x' be a random variable describing the result of an asymmetric bit mutation of x. Then

$$\mathrm{E}\left(|x'|_1 \mid x\right) = |x|_1 \cdot \left(1 - \frac{1}{2\,|x|_1}\right) + |x|_0 \cdot \frac{1}{2\,|x|_0} = |x|_1 .$$

We therefore observe a Markov chain that is similar to a fair random walk on $\{0, 1, \ldots, n\}$ with the main difference that arbitrary step sizes may occur. For a fair random walk $\{X_t\}_{t\geq 0} = X_0, X_1, \ldots$ with step size fixed to 1 it is well known that the expected time to reach either 0 or n is bounded by $X_0(n-X_0)$. We are able to generalize this result to a large class of stochastic processes. Since this may be of independent interest, we state the lemma here and refer to the appendix for a proof.

Lemma 3. *Consider a stochastic process $\{X_t\}_{t\geq 0}$ on $\{0, \ldots, n\}$. Define $T := \inf\{t \mid X_t \in \{0, n\}\}$. If the process $\{X_t\}_{t\geq 0}$ is a martingale (i. e. $E(X_{t+1} \mid X_0, \ldots, X_t) = X_t$) and $0 < X_t < n$ implies $X_{t+1} \neq X_t$ for all $0 \leq t < T$, then $E(T) \leq X_0(n - X_0)$.*

Theorem 5. *For any constant $k \in \mathbb{N}_0$ and all $a \in \{0,1\}^n$ with either at most k 0-bits or at most k 1-bits, the expected optimization time of the asymmetric (1+1) EA on Needle$_a$ is bounded above by $O(n^2 + n^{k+1})$.*

The full proof can be found in the appendix. As long as the needle has not been found, the asymmetric (1+1) EA is symmetric w. r. t. bit positions. This motivates to consider only steps where the number of 1-bits in the current search point changes. The process induced by these steps satisfies the conditions of Lemma 3 implying that either state 0 or state n is reached in expected time $O(n^2)$. Moreover, the expected time to travel between states 0 and n can be bounded by $O(n^2)$, too, which proves the claim for $k = 0$.

For $k > 0$ we consider the case where the needle has $n - k$ 1-bits, the other case is symmetric. We estimate the expected time until the needle is hit by a direct jump from 1^n. The probability for such a jump is $\Omega(n^{-k})$ for constant k and after an unsuccessful attempt the expected time to return to 1^n is $O(n)$. Together, the bound $O(n^2 + n^{k+1})$ follows.

In order to get a closer picture of the actual performance, we consider the results of 100 independent runs for $k = 0$ and $n \in \{100, 200, \dots, 2000\}$. We report the number of runs where the unique global optimum was found within cn^2 steps for $c \in \{1, 2, \dots, 6\}$ in Table 1.

n	$1n^2$	$2n^2$	$3n^2$	$4n^2$	$5n^2$	$6n^2$
100	72	90	95	100	100	100
200	73	95	100	100	100	100
300	75	90	97	99	100	100
400	78	97	99	99	100	100
500	74	92	96	99	99	100
600	73	92	96	99	100	100
700	74	93	97	100	100	100
800	74	96	100	100	100	100
900	73	96	99	100	100	100
1000	71	94	98	100	100	100
1100	71	90	95	97	100	100
1200	80	94	96	98	99	100
1300	67	90	97	99	100	100
1400	79	93	98	99	100	100
1500	70	91	99	100	100	100
1600	79	91	97	99	100	100
1700	71	89	99	99	100	100
1800	83	90	95	98	99	100
1900	75	94	97	98	99	100
2000	76	92	100	100	100	100

Table 1: Number of runs where the needle was found within the given time bound.

Let $N := \{\text{NEEDLE}_a \mid a \in \{0,1\}^n\}$ be the class of needle-functions with the global optimum at some point $\overline{a}$ in the search space. It is known from results on the black-box complexity of function classes [5] that any search heuristic needs at least $2^{n-1} + 1/2$ function evaluations on N on average. Thus, while the asymmetric (1+1) EA performs very well on NEEDLE_a with a close to 0^n or 1^n, it performs poorly on other functions NEEDLE_a with a far from 0^n and 1^n. This is another hint that the search process of the asymmetrical (1+1) EA is clearly biased.

Note, that the class $N = \{\text{NEEDLE}_a \mid a \in \{0,1\}^n\}$ is closed under permutations of the search space. Thus, the same conclusion seems to be implied by the NFL: averaged over all such functions all algorithms make an equal number of different function evaluations [11]. However, this result has only limited relevance with respect to the expected optimization time since it does not take into account re-sampling of points in the search space.

4 Analysis for Unimodal Functions

The results from Section 3 proved the asymmetric mutation operator to be advantageous for objective functions where good bit strings have either many or few 1-bits. Clearly, OneMax and Needle are both artificial examples that do not have much in common with problems encountered in practical applications. In order to gain a broader perspective, results on more general function classes are needed. Here, we compare the performance of the asymmetric (1+1) EA with the (1+1) EA with standard bit mutations on a whole class of interesting and important functions, namely on unimodal functions. It is interesting to note that the class of unimodal functions is closed under the transformation of objective functions considered here. I. e., for any $a \in \{0,1\}^n$, f_a is unimodal iff f is. Thus, again we can move the unique global optimum anywhere in the search space.

An important property of unimodal functions is that they can be optimized via mutations of single bits, i. e., hill-climbers are guaranteed to be successful. Starting with an arbitrary search point, there is a path of Hamming neighbors to the unique global optimum with strictly increasing fitness. Note, however, that paths to the unique global optimum may be exponentially long, making such functions difficult to optimize. In fact, it is known that any search heuristic needs in the worst case an exponential number of function evaluations to optimize a unimodal function [7].

Using Lemma 1, it is easy to obtain a general upper bound on the expected optimization time for unimodal functions with d different function values. The upper bound as well as its short proof are not different from the corresponding result for standard bit mutations.

Theorem 6. *Let $f\colon \{0,1\}^n \to \mathbb{R}$ be a unimodal function with d different function values. The expected optimization time of the asymmetric (1+1) EA on f is bounded above by $O(n \cdot d)$.*

Proof. We know from Lemma 1 that the probability to increase the function value of the current population is bounded below by $1/(8n)$. This yields $8n$ as upper bound on the expected time to increase the fitness. Clearly, at most $d-1$ fitness increases are sufficient to reach the global optimum. $\square$

We see that asymmetric bit mutations deliver the same upper bound on an important class of functions as standard bit mutations. Of course, in both cases, the upper bound is not necessarily tight. However, it is known to be tight for standard bit mutations for some functions. One example of such a function is LO, an example function where the (1+1) EA with standard bit mutations has expected optimization time $\Theta(n^2)$. Moreover, deviations from the expected value by some constant factors are extremely unlikely. The use of asymmetric bit mutations, however, leads to a considerable speed-up.

Theorem 7. *The expected optimization time of the asymmetric (1+1) EA on* LO *is bounded by* $O(n^{3/2})$.

A proof of the theorem is given in the appendix. The idea is to partition a run into two phases with the first phase comprising the beginning of the run as long as the function value is bounded above by $n^{1/2}$ and the second phase comprising the rest of the run. Since in the first phase the fitness value needs to be increased at most $n^{1/2}$ times and always single bit mutations are sufficient to do so, clearly, its expected length is bounded by $O(n^{3/2})$. For the second phase, we observe that all bits following the first bit that differs from the optimum have not yet had an impact on the fitness and thus are subject to a random process. Since we have $n^{1/2}$ leading ones that cannot be flipped in an accepted step, the number of 0-bits in the current bit string constitutes a supermartingale with an expected decrease of $\Omega(n^{-1/2})$. A drift analysis shows that the expected time until the number of 0-bits reaches 0 is $O(n^{3/2})$.

As OneMax, the function LO has the property that the unique global optimum is the all-one bit string 1^n. Obviously, this fosters the finding of the global optimum using asymmetric bit mutations. Therefore, it makes sense to investigate the expected optimization time on LO_a. We would like to see whether there are some $a \in \{0,1\}^n$ such that the expected optimization time of the (1+1) EA using asymmetric mutations is $\omega(n^{3/2})$.

We consider a where the optimum has linear Hamming distance to both 0^n and 1^n. We will see that then the asymmetric (1+1) EA's tendency towards some kind of bits is more hindering than helpful.

Theorem 8. *Given some constant* $0 < c < 1$*, let* $a \in \{0,1\}^n$ *be chosen uniformly at random among all search points with* cn *1-bits. The expected (w. r. t. the random bits of* a *and the algorithm's decisions) optimization time of the asymmetric (1+1) EA on* $\mathrm{LO}_a(x)$ *is* $\Theta(n^2)$.

Again, the full proof is placed in the appendix and we only present the main ideas, here. Since the upper bound follows from Theorem 6, only the lower bound needs to be proved. We consider the first point of time where the first $n/2$ bits of x match the global optimum $\overline{a}$. Then typically (w. r. t. the choice of a) the current population subsequently has a linear number of 1-bits and 0-bits on these positions and the expected time to increase the LO_a-value by flipping the leftmost bit differing from the optimum $\overline{a}$ is $\Omega(n)$.

If $a = 0^n$ then the drift arguments from the proof of Theorem 7 reveal that the bias by asymmetric mutations can increase the number of 1-bits and hence the LO_a considerably in one step. However, positions i with

$a_i \neq a_{i+1}$ can slow down this process: as long as the LO_a-value of the current population x is smaller than $i-1$ the asymmetric (1+1) EA is invariant to the bit positions of a_i and a_{i+1}, hence $\mathrm{Prob}\,(x_i = 1 \wedge x_{i+1} = 0) = \mathrm{Prob}\,(x_i = 0 \wedge x_{i+1} = 1)$. The probability that $x_i x_{i+1} = \overline{a_i}\,\overline{a_{i+1}}$ is thus bounded by 1/2. Due to the choice of a there are typically $\Omega(n)$ such constellations, hence $\Omega(n)$ increases of the LO_a-value are needed in expectancy to find the optimum.

Here, we consider experiments for LO_a for $n \in \{40, 80, \ldots, 600\}$ and different bit strings a. We fix $|a|_1 \in \{0, .05n, .1n, \ldots, .95n, n\}$ and choose one such a uniformly at random for each value of $|a|_1$. In Fig. 3 we observe a clear increase of the average optimization time as $|a|_1$ moves towards $n/2$ as could be expected.

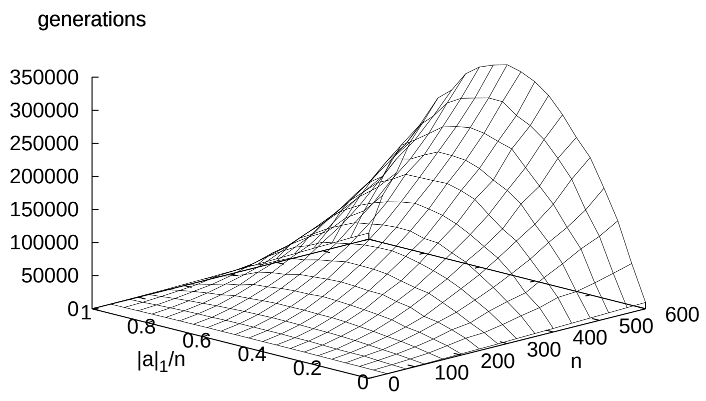

Figure 3: Average run times on LO_a for $n \in \{40, 80, \ldots, 600\}$ and $|a|_1 \in \{0, .05n, .1n, \ldots, .95n, n\}$.

When looking for a known example function where the general upper bound for unimodal functions is tight, without applying any transformation, one may think of RIDGE [17]. Whereas the unique global optimum is 1^n (like for ONEMAX and LO), the algorithm cannot benefit from additional 1-bits that happen to be present in the trailing bits. The definition of RIDGE forces these bits to be 0 on the ridge. This makes it seem unlikely that the (1+1) EA with asymmetric bit mutations can outperform its counterpart with standard bit mutations on RIDGE. We prove this intuition to be correct.

Theorem 9. *The expected optimization time of the asymmetric (1+1) EA on* RIDGE *is* $\Theta(n^2)$. *The same holds for* RIDGE_a *and any* $a \in \{0,1\}^n$.

Proof. The upper bound follows from Theorem 6. With probability $1 - 2^{-\Omega(n)}$ the initial search point has Hamming distance at least $n/3$ from the unique global optimum. Offspring closer to the optimum with a fitness value smaller than $n+1$ are rejected. Thus, with probability $1-2^{-\Omega(n)}$ the first accepted search point x^* where $f(x^*) \geq n+1$ has Hamming distance $\Omega(n)$ to the unique global optimum.

Let $S = (s_0, \ldots, s_{n-1})$ be the sequence of Hamming neighbors such that $f(s_i) = n+i$ for all $0 \leq i \leq n-1$. Then for every a there is a coherent subsequence $S' = (s'_1, \ldots, s'_m)$ of S of length $m = \Omega(n)$ such that $f(s'_1) \geq f(x^*)$ and both $|s'_i|_1 = \Omega(n)$ and $|s'_i|_0 = \Omega(n)$ hold. Due to the definition of $\text{R}\textsc{idge}_a$, this subsequence has to be traversed in order to optimize $\text{R}\textsc{idge}_a$. The expected decrease in Hamming distance to the global optimum on this subsequence in one mutation is $O(1/n)$. Hence applying drift arguments yields the lower bound $\Omega(n^2)$ on the expected optimization time. $\square$

Considering the results of experiments for RIDGE designed in the same way as the experiments for LO we see that our asymptotical bounds are confirmed (see Fig. 4). There is hardly any difference for different choices of a.

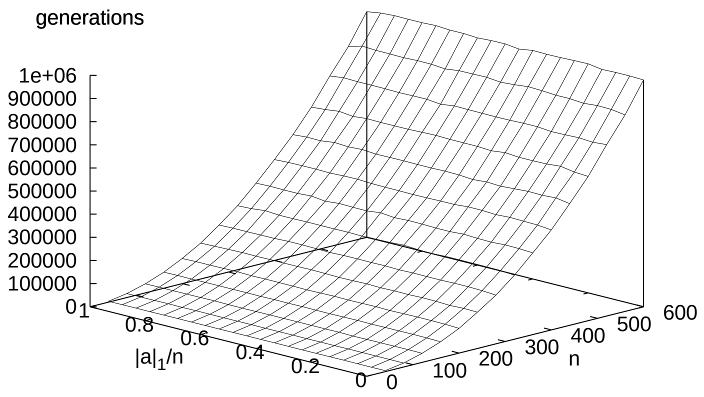

Figure 4: Average run times on $\text{R}\textsc{idge}_a$ for $n \in \{40, 80, \ldots, 600\}$ and $|a|_1 \in \{0, .05n, .1n, \ldots, .95n, n\}$.

The performance of the (1+1) EA with standard bit mutations and asymmetric bit mutations are asymptotically equal on $\text{R}\textsc{idge}_a$. Even the proofs of the bounds are very similar [14]. So far, we have seen only advantages for the asymmetric mutations and many similarities to standard bit mutations. In the following section, we consider an example where the asymmetric mutation operator leads to an extreme decline in performance.

5 Drawbacks of the Asymmetric Mutation Operator

The function PLATEAU is very similar to RIDGE. The function values differ only for n out of 2^n points in the search space. These n points are the most important ones, though. For RIDGE, the increase in function values of these ridge points leads into the direction of the global optimum. For PLATEAU, the function values are constant and the evolutionary algorithm has to perform a kind of blind random walk on this plateau. It is known that standard bit mutations complete this random walk successfully on average in $O(n^3)$ steps. Asymmetric bit mutations fail to be efficient in any sense, here.

Theorem 10. *The probability that the asymmetric (1+1) EA optimizes* PLATEAU *within $2^{o(n^{1/6})}$ steps is bounded above by $2^{-\Omega(n^{1/6})}$.*

Our discussion of the performance of asymmetric (1+1) EA showed a clear bias towards 0^n and 1^n. We have seen already there that this bias is stronger towards the nearer of the two extremes in the search space. On PLATEAU, the first point found on the plateau is likely to be close to 0^n, whereas the unique global optimum is 1^n. The bias induced by asymmetric bit mutations makes it hard to perform a random walk towards 1^n. This intuitive reasoning is made rigorous in a proof that can be found in the appendix.

Note, however, that this immense drawback is due to the special definition of PLATEAU. In particular, we can transform the landscape in a way that does not influence the (1+1) EA with standard bit mutations at all but is important for the asymmetric (1+1) EA. This leads to a function where we can prove upper bounds on the expected optimization time of the two algorithms of equal order.

Theorem 11. *For even n we define $a_{01} := 010101\cdots01 \in \{0,1\}^n$. The expected optimization time of the asymmetric (1+1) EA on* PLATEAU$_{a_{01}}$ *is $O(n^3)$.*

Proof. It follows from the result on ONEMAX$_a$ (Theorem 4) that some point on the plateau will be found on average within the first $O(n \log n)$ steps. Then, the plateau cannot be left again. For each $i \in \{0, \ldots, n-1\}$ and some search point x on the plateau we show the following claim. If it is possible to create search points x^{+i}, x^{-i} on the plateau out of x such that the Hamming distance to the unique global optimum is increased or decreased by i, resp., then x^{+i} and x^{-i} are reached with equal probability. This is due to the choice of a_{01} since all plateau points of PLATEAU$_{a_{01}}$ have either $n/2$ or $(n/2)+1$ 1-bits and points with $n/2$ and $(n/2)+1$ 1-bits are alternating on the plateau. Therefore, $\left|x^{+i}\right|_1 = \left|x^{-i}\right|_1$ holds and both the number of flipping 1-bits and the number of flipping 0-bits is the same for x^{+i} and x^{-i}. Furthermore, the choice of a_{01} implies that $|x|_1 = n/2 + O(1)$ for all x on the plateau yielding a probability of $\Theta(1/n)$ to flip any bit in x.

By these arguments, an upper bound $O(n^3)$ on the expected optimization time can be shown analogously to the results in Jansen and Wegener [14]. $\square$

Again, we add to the findings of our theoretical analysis by considering empirical data. We consider experiments for PLATEAU$_{a_{01}}$ measuring the average run time of the asymmetric (1+1) EA over 100 runs for each $n \in \{10, 20, \ldots, 200\}$. The results are shown in Figure 5. A regression analysis with functions $c \cdot n^3$ yielded a good fit for $c = 1.33951$, thus suggesting that the upper bound $O(n^3)$ is tight.

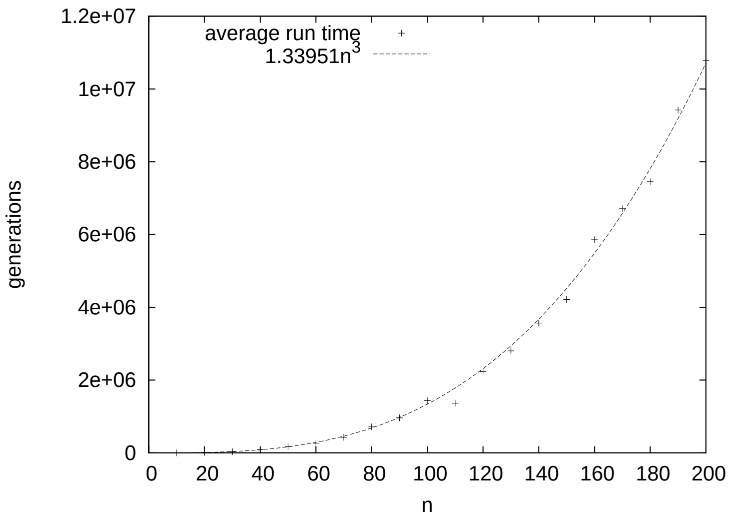

Figure 5: Average run times on PLATEAU$_{a_{01}}$ for $n \in \{10, 20, \ldots, 200\}$ and the fitted function $1.33951n^3$.

Next, we consider experiments for the asymmetric (1+1) EA on the function PLATEAU. Since our theoretical result predicts overly large run times, we measure the number of runs where the optimum is found within $16n^3$ steps. The factor 16 is chosen for the following reasons according to the empirical results on PLATEAU$_{a_{01}}$. First, this factor is more than ten times larger than the factor $c = 1.33951$ of the fitted function for PLATEAU$_{a_{01}}$. Moreover, all runs on PLATEAU$_{a_{01}}$ found the global optimum within $16n^3$ steps.

The results on PLATEAU are shown in Table 2. The search space dimension $n = 10$ is too small for the bias of asymmetric mutations to have a significant impact on the plateau. However, for larger n the asymmetric (1+1) EA clearly fails on PLATEAU.

n	successful runs	n	successful runs
10	60	110	0
20	0	120	0
30	0	130	0
40	0	140	0
50	0	150	0
60	0	160	0
70	0	170	0
80	0	180	0
90	0	190	0
100	0	200	0

Table 2: Number of runs where the optimum of $\text{PLATEAU}_{a_{01}}$ is found within $16n^3$ steps.

6 Conclusions and Future Work

We presented a mutation operator for bit strings that flips bits with a probability that depends on the number of 1-bits. The operator is designed in a way that on average the number of 1-bits is not changed. This helps to bias the search towards areas of the search space with bit strings containing either very few 0-bits or very few 1-bits. Such a mutation operator is motivated by applications where good solutions are known or at least thought of having this property.

We presented a rigorous and detailed analysis of this mutation operator by comparing it with standard bit mutations flipping each bit independently with probability $1/n$. For OneMax, a speed-up of order $\log n$ is proved. For Needle, there is even an exponential advantage for the asymmetric mutation operator.

For the class of unimodal functions we proved the same general upper bound as known for standard bit mutations. For LO, a speed-up of order $n^{1/2}$ is proved in comparison to standard bit mutations. However, both mutation operators lead to run times of equal order on a simple transformation of LO and on a class of ridge functions. These results show that the general upper bound for unimodal functions can be tight and that both algorithms can have similar performance on broad classes of functions.

Contrarily, we demonstrated a clear weakness of the asymmetric bit mutations on a function where an unbiased random walk on a plateau is needed in order to be successful. We showed that there is an exponential gap between the performance of this asymmetric mutation operator and standard bit mutations. However, a simple transformation of the landscape lets both mutation operators lead to polynomial expected optimization times for this objective function.

All theoretical analyses come along with additional experimental results. The empirical data reveals details of the performance that are invisible in the asymptotic notation of the theorems. It is possible to obtain theorems at a greater level of detail that may cover some parts of the observations. The decision where the effort of such additional analytical work is worthwhile is not easy to make.

While motivated by practical applications, our presentation is purely theoretical. In fact, Neumann and Wegener [16] introduced asymmetric bit mutations motivated by the minimum spanning tree problem. Note, however, that this is a classical text book problem where evolutionary algorithms are not used in practice. It would be interesting to compare the performance of the asymmetric mutation operator in real applications where one suspects that good solutions have only a few bits with value 0 or 1.

References

[1] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, 1998.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2002.

[3] B. Doerr, N. Hebbinghaus, and F. Neumann. Speeding up evolutionary algorithms through restricted mutation operators. In *Parallel Problem Solving from Nature (PPSN IX)*, pages 978–987, 2006. LNCS 4193.

[4] B. Doerr and D. Johannsen. Adjacency list matchings – an ideal genotype for cycle covers. In *Genetic and Evolutionary Computation Conference (GECCO 2007)*, pages 1203–1210. ACM, New York, 2007.

[5] S. Droste, T. Jansen, K. Tinnefeld, and I. Wegener. A new framework for the valuation of algorithms for black-box optimization. In *Foundations of Genetic Algorithms 7 (FOGA)*, pages 253–270, San Francisco, CA, 2003. Morgan Kaufmann.

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

[7] S. Droste, T. Jansen, and I. Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 39(4):525–544, 2006.

[8] S. Droste and D. Wiesmann. On the design of problem-specific evolutionary algorithms. In *Advances in Evolutionary Computing: Theory and Applications*, pages 153–173. Springer-Verlag New York, Inc., 2003.

[9] J. Garnier, L. Kallel, and M. Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203, 1999.

[10] J. He and X. Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(1):21–35, 2004.

[11] C. Igel and M. Toussaint. A no-free-lunch theorem for non-uniform distributions of target functions. *Journal of Mathematical Modelling and Algorithms*, 3(4):313–322, 2004.

[12] T. Jansen, K. A. De Jong, and I. Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13:413–440, 2005.

[13] T. Jansen and D. Sudholt. Design and analysis of an asymmetric mutation operator. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 497–504. IEEE Press, 2005.

[14] T. Jansen and I. Wegener. Evolutionary algorithms — how to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Transactions on Evolutionary Computation*, 5:589–599, 2001.

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

[16] F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. In *Genetic and Evolutionary Computation Conference (GECCO 2004)*, pages 713–724. Springer, 2004. LNCS 3102.

[17] R. J. Quick, V. J. Rayward-Smith, and G. D. Smith. Fitness distance correlation and ridge functions. In *Parallel Problem Solving from Nature (PPSN V)*, pages 77–86. Springer, 1998.

[18] G. R. Raidl, G. Koller, and B. A. Julstrom. Biased mutation operators for subgraph-selection problems. *IEEE Transactions on Evolutionary Computation*, 10(2):145–156, 2006.

[19] C. Scheideler. *Probabilistic Methods for Coordination Problems.* HNI-Verlagsschriftenreihe 78, University of Paderborn, 2000. Habilitation Thesis, available at `http://www14.in.tum.de/personen/scheideler/index.html.en`.

Appendix

This appendix contains the complete formal proofs of the results presented in this paper. To increase readability, we repeat the results here using the same numbering as in the paper.

Lemma 2. *Let $\varepsilon > 0$. The probability that the asymmetric mutation operator flips $\Omega(n^\varepsilon)$ bits in one step is bounded above by $2^{-\Omega(n^\varepsilon \log n)}$.*

Proof. Let x be the current search point and let Z_x be the random variable describing the number of bits flipping in a mutation of x. The expected number of flipping bits equals $\mathrm{E}(Z_x) = 1$ if $x \notin \{0^n, 1^n\}$ and $\mathrm{E}(Z_x) = 1/2$ if $x \in \{0^n, 1^n\}$, so $1/2 \le \mathrm{E}(Z_x) \le 1$. Let c be the constant hidden in the big-Omega notation of $\Omega(n^\varepsilon)$. By Chernoff bounds,

$$\begin{aligned}\mathrm{Prob}\left(Z_x > cn^\varepsilon\right) &< \left(\frac{e^{cn^\varepsilon - 1}}{(cn^\varepsilon)^{cn^\varepsilon}}\right)^{\mathrm{E}(Z)} \\ &\le \frac{e^{cn^\varepsilon/2}}{e^{1/2}} \cdot (cn)^{-\varepsilon/2 \cdot cn^\varepsilon} = 2^{-\Omega(n^\varepsilon \log n)}.\end{aligned}$$

□

Theorem 4. *For any a the expected optimization time of the asymmetric (1+1) EA on* $\textsc{OneMax}_a$ *is $\Theta(n \log(\min\{|a|_0, |a|_1\} + 2))$.*

Proof. W. l. o. g., $|a|_1 \le n/2$. The unique global optimum of $\textsc{OneMax}_a$ is $\overline{a}$. We begin with a proof of the upper bound and partition a run into two phases: the first phase starts with the beginning of the run and ends when we have a search point with at most $2|\overline{a}|_0$ 0-bits for the first time. The second phase starts after the first phase and ends when the global optimum is found.

Let x be some current search point in the first phase, then there are at least $|x|_0 - |\overline{a}|_0$ positions i where $x_i = 0$ and $\overline{a}_i = 1$. Clearly, a 1-bit-mutation flipping such a bit increases the fitness. During the first phase $|x|_0 > 2|\overline{a}|_0$ holds. Thus, $|x|_0 - |\overline{a}|_0 > |x|_0/2$ and by Lemma 1 the probability of a 1-bit-mutation flipping one of the considered bits is at least $|x|_0/2 \cdot 1/(8|x|_0) = 1/16$. Therefore, the expected length of phase 1 is $O(n)$.

At the beginning of the second phase, $|x|_0 \le 2|\overline{a}|_0$ holds. Moreover, we have $\mathrm{H}(x, \overline{a}) \le 3|\overline{a}|_0$ since $\mathrm{H}(x, 1^n) = |x|_0 \le 2|\overline{a}|_0$ and $\mathrm{H}(1^n, \overline{a}) = |\overline{a}|_0$. This implies $\textsc{OneMax}_a(x) \ge n - 3|\overline{a}|_0$ for the rest of phase 2. Thus, the fitness has yet to be increased at most $3|\overline{a}|_0 = 3|a|_1$ times. There are $n - \textsc{OneMax}_a(x)$ Hamming neighbors with function value larger than x. By Lemma 1, the probability to reach a specific Hamming neighbor by a direct mutation is bounded below by $1/(8n)$. Thus, the expected length of the second phase is bounded above by

$$\sum_{i=1}^{3|a|_1} \frac{8n}{i} = 8n \sum_{i=1}^{3|a|_1} \frac{1}{i} = O(n \log(|a|_1 + 2)).$$

For the lower bound we distinguish three cases with respect to $|a|_1$. In case $|a|_1 \leq 1$ the lower bound $\Omega(n)$ follows from Theorem 1. Consider the case $2 \leq |a|_1 \leq n/4$. Since the mutation operator is symmetric with respect to bit positions, we can assume $a = 1^{|a|_1} 0^{|a|_0}$ without loss of generality. With probability at least 1/2, the initial population contains at least $|a|_1/2$ 1-bits among the first $|a|_1$ positions. Analogously, it contains at least $|a|_0/2$ 1-bits among the last $|a|_0$ positions with probability at least 1/2. Consider the case where both events occur which happens with probability at least 1/4. Since the Hamming distance to the global optimum $\overline{a} = 0^{|a|_1} 1^{|a|_0}$ cannot increase, the number of 1-bits is bounded below by $|a|_0/2 - |a|_1 \geq n/8$ during the run. Thus, the probability that a 1-bit is flipped is bounded above by $4/n$. We have at least $|a|_1/2$ 1-bits that all need to flip at least once. The probability that not all of these bits flip within $((n/4)-1)\ln|a|_1$ mutations is bounded below by

$$\begin{aligned} & 1 - \left(1 - \left(1 - \frac{4}{n}\right)^{((n/4)-1)\ln|a|_1}\right)^{|a|_1/2} \\ \geq\ & 1 - \left(1 - e^{-\ln|a|_1}\right)^{|a|_1/2} \geq 1 - e^{-1/2}. \end{aligned}$$

Thus, the expected optimization time is bounded below by

$$\frac{1}{4} \cdot \left(1 - e^{-1/2}\right) \cdot \left(\frac{n}{4} - 1\right) \ln|a|_1 = \Omega(n \log(|a|_1 + 2))$$

in this case.

Finally, we consider the case $n/4 < |a|_1 \leq n/2$. Chernoff bounds [15] yield that with probability $1 - 2^{-\Omega(n)}$ the initial Hamming distance to the unique global optimum is bounded below by $n/8$. Moreover, it is easy to see (similar to Lemma 2) that with probability $1 - 2^{-\Omega(n)}$ we only have mutations with $o(n)$ bits flipping simultaneously within the first $O(n \log n)$ generations. Thus, we may consider the situation at the end when the Hamming distance to the optimum is decreased to $n/8 - o(n)$. As the Hamming distance cannot increase during a run, the number of 1-bits in the current population is always bounded below by $n/8$ and bounded above by $7n/8$. This implies that for each bit (regardless of its value) the probability to flip it is bounded above by $4/n$. Now we are in a situation very similar to the second case. Repeating the line of thought from there completes the proof. □

Lemma 3. *Consider a stochastic process* $\{X_t\}_{t\geq 0}$ *on* $\{0,\dots,n\}$. *Define* $T := \inf\{t \mid X_t \in \{0,n\}\}$. *If the process* $\{X_t\}_{t\geq 0}$ *is a martingale (i. e.* $E(X_{t+1} \mid X_0,\dots,X_t) = X_t$*) and* $0 < X_t < n$ *implies* $X_{t+1} \neq X_t$ *for all* $0 \leq t < T$, *then* $E(T) \leq X_0(n - X_0)$.

Proof. Let $\mathrm{E}_t(Z)$ abbreviate $\mathrm{E}(Z \mid X_0,\dots,X_t)$. We define $\{Y_t\}_{t\geq 0}$ by $Y_t := (X_t)^2 - \sum_{k=0}^{t-1} \mathrm{E}_k\left((X_{k+1}-X_k)^2\right)$ and consider

$$\mathrm{E}_t(Y_{t+1}) = \mathrm{E}_t\left((X_{t+1})^2\right) - \sum_{k=0}^{t} \mathrm{E}_t\left(\mathrm{E}_k\left((X_{k+1}-X_k)^2\right)\right).$$

Regarding the $\sum$-term, the summand for $k = t$ equals

$$\mathrm{E}_t\left(\mathrm{E}_k\left((X_{k+1}-X_k)^2\right)\right) = \mathrm{E}_k\left((X_{k+1}-X_k)^2\right)$$

and for $k < t$ we have

$$\mathrm{E}_t\left(\mathrm{E}_k\left((X_{k+1}-X_k)^2\right)\right) = \mathrm{E}_k\left((X_{k+1}-X_k)^2\right)$$

since the right-hand side is $X_0,\dots,X_t$-measurable. Secondly, by the formula $\mathrm{E}\left(Z^2\right) = (\mathrm{E}(Z))^2 + \mathrm{E}\left((Z-\mathrm{E}(Z))^2\right)$ we have

$$\begin{aligned}\mathrm{E}_t\left((X_{t+1})^2\right) &= (\mathrm{E}_t(X_{t+1}))^2 + \mathrm{E}_t\left((X_{t+1}-\mathrm{E}_t(X_{t+1}))^2\right)\\ &= (X_t)^2 + \mathrm{E}_t\left((X_{t+1}-X_t)^2\right).\end{aligned}$$

Together,

$$\begin{aligned}\mathrm{E}\left(Y_{t+1} \mid X_0^t\right) &= (X_t)^2 + \mathrm{E}_t\left((X_{t+1}-X_t)^2\right)\\ &\quad - \sum_{k=0}^{t} \mathrm{E}_k\left((X_{k+1}-X_k)^2\right)\\ &= (X_t)^2 - \sum_{k=0}^{t-1} \mathrm{E}_k\left((X_{k+1}-X_k)^2\right)\\ &= Y_t.\end{aligned}$$

Thus, $\{Y_t\}_{t\geq 0}$ is a martingale with respect to $\{X_t\}_{t\geq 0}$.

In every state $0 < X_t < n$ there is a positive probability to get closer to the closest state from $\{0,n\}$, hence $T < \infty$ follows. Since for $t \leq T$

$$|Y_t| = \left| X_t^2 - \sum_{k=0}^{t-1} \mathrm{E}_k\left((X_{k+1}-X_k)^2\right) \right| < 4n^2 + Tn^2 < \infty$$

holds, we can apply the optional stopping theorem [1]. This yields $\mathrm{E}(Y_T) = \mathrm{E}(Y_0) = (X_0)^2$ on the one hand and, along with $\mathrm{E}_k\left((X_{k+1}-X_k)^2\right) \geq 1$,

$$\begin{aligned}\mathrm{E}(Y_T) &= \mathrm{E}\left(X_T^2\right) - \mathrm{E}\left(\sum_{k=0}^{T-1} \mathrm{E}_k\left((X_{k+1}-X_k)^2\right)\right)\\ &\leq \mathrm{E}\left(X_T^2\right) - \mathrm{E}(T)\end{aligned}$$

on the other hand, which implies $\mathrm{E}(T) \leq \mathrm{E}\left(X_T^2\right) - (X_0)^2 = \mathrm{Prob}(X_T = n) \cdot n^2 - (X_0)^2$. Applying the optional stopping theorem again w. r. t. $\{X_t\}_{t\geq 0}$ yields $\mathrm{E}(X_T) = X_0$. Along with $\mathrm{E}(X_T) = \mathrm{Prob}(X_T = n) \cdot n$ we obtain $\mathrm{Prob}(X_T = n) = X_0/n$ and $\mathrm{E}(T) \leq X_0(n - X_0)$ follows. □

Theorem 5. *For any constant $k \in \mathbb{N}_0$ and all $a \in \{0,1\}^n$ with either at most k 0-bits or at most k 1-bits, the expected optimization time of the asymmetric (1+1) EA on* $\textsc{Needle}_a$ *is bounded above by $O(n^2 + n^{k+1})$.*

Proof. W. l. o. g. we assume that the needle has $n-k$ 1-bits. Call a step of the asymmetric (1+1) EA *essential* if the number of ones in the current search point changes. Since the asymmetric (1+1) EA is symmetric w. r. t. the position of ones in the current search point, it is sufficient to consider the random process of essential steps, only. Consider one step of the asymmetric (1+1) EA and let $p_{i,j}$ denote the probability that some y with $|y|_1 = j$ is created out of some x with $|x|_1 = i$. We have $p_{i,i} \leq 3/4$ since $p_{i,i} = (1-1/(2n))^n \leq e^{-1/2}$ if $i \in \{0,n\}$ and $p_{i,i} \leq 1 - (p_{i,i-1} + p_{i,i+1}) \leq 3/4$ by Lemma 1 otherwise. It follows that the probability of an essential step is at least 1/4 and the expected total number of steps is at most by a factor of 4 larger than the expected number of essential steps.

The process of essential steps fulfills the conditions of Lemma 3, thus starting with an initial search point with i ones the expected number of essential steps until some x^* with $|x^*|_1 \in \{0,n\}$ is reached is $i(n-i) \leq n^2/4$. With probability 1/2 we have $|x^*|_1 = n$ and are done. Otherwise, let $T_{0\to n}$ be the random time for the asymmetric (1+1) EA to reach 1^n starting from 0^n. Let $i \geq 1$ be the number of ones of the search point reached from 0^n in the first essential step. We know from Lemma 3 that the expected number of essential steps to next reach some x^{**} with $|x^{**}|_1 \in \{0,n\}$ is bounded by $i(n-i)$. Moreover, from the proof of Lemma 3 we know that $\mathrm{Prob}(x^{**} = n) = i/n$, hence we return to 0^n with probability $1 - i/n$. This implies the following recursion.

$$\mathrm{E}(T_{0\to n}) \leq 1 + \sum_{i=1}^{n} p_{0,i}\left(i(n-i) + \left(1 - \frac{i}{n}\right) \cdot \mathrm{E}(T_{0\to n})\right)$$

By rearranging we obtain

$$\begin{aligned}
\mathrm{E}\left(T_{0\to n}\right) &\le \frac{1+\sum_{i=1}^{n} p_{0,i}\cdot i(n-i)}{\sum_{i=1}^{n} p_{0,i}\cdot \frac{i}{n}}\\
&\le n+\frac{\sum_{i=1}^{n} p_{0,i}\cdot in}{\sum_{i=1}^{n} p_{0,i}\cdot \frac{i}{n}} = n+n^2.
\end{aligned}$$

This implies that for $k=0$ the expected number of essential steps is at most $n^2/4+1/2\cdot \mathrm{E}\left(T_{0\to n}\right)\le 3n^2/4+n/2$ and, taking into account non-essential steps, the bound $3n^2+2n$ follows for $k=0$.

For $k>0$ the time until the needle is found is clearly bounded by the expected time to reach the needle by a direct jump from 1^n. The probability for such a jump is $(1/(2n))^k\cdot(1-1/(2n))^{n-k}\ge e^{-1/2}\cdot(2n)^{-k}$, hence the expected number of trials is $O((2n)^k)$. The expected return time from 1^n to 1^n in terms of essential steps is bounded by

$$\begin{aligned}
&1+\sum_{i=0}^{n} p_{n,i}\left(i(n-i)+\left(1-\frac{i}{n}\right)\cdot \mathrm{E}\left(T_{0\to n}\right)\right)\\
&\le 1+\sum_{i=0}^{n-1} p_{n,i}\left(i(n-i)+(n-i)(n+1)\right)\\
&\le 1+2n\cdot\sum_{i=0}^{n-1} p_{n,i}(n-i).
\end{aligned}$$

The $\sum$-term describes the expected number of flipping bits when mutating 1^n which can be more easily computed as $n\cdot 1/(2n)=1/2$. This results in the bound $1+n$. As the expected number of steps between two trials is $O(n)$, we obtain the bound $O(n^2+n\cdot(2n)^k)=O(n^2+n^{k+1})$ for constant k. □

Theorem 7. *The expected optimization time of the asymmetric (1+1) EA on* LO *is bounded by $O(n^{3/2})$.*

Proof. We partition a run into two phases: the first phase starts with the beginning of the run and ends when some search point x^* with $\mathrm{LO}(x^*)\ge n^{1/2}$ is reached for the first time. The second phase starts after the first phase and ends when the global optimum is found.

Due to Lemma 1, the expected length of phase 1 is bounded by $O(n^{3/2})$ since there always are Hamming neighbors with larger fitness and the fitness has to be increased at most $n^{1/2}$ times to reach the end of phase 1.

For the investigation of phase 2, we apply drift analysis arguments presented by He and Yao [10] and choose $|x|_0$ as a distance function to the optimum.

Let x be the current population and x' be the population of the next generation. Then $x' = x$ or x' is an accepted mutant of x. We obtain

$$\begin{aligned} \mathrm{E}\left(\left|x'\right|_0\right) &= \mathrm{E}\left(\left|x'\right|_0 \mid x' \text{ is accepted}\right) \cdot \mathrm{Prob}\left(x' \text{ is accepted}\right) \\ &\quad + |x|_0 \cdot (1 - \mathrm{Prob}\left(x' \text{ is accepted}\right)). \end{aligned}$$

Since the first $\mathrm{LO}(x)$ 1-bits cannot flip to 0 in an accepted step,

$$\begin{aligned} &\mathrm{E}\left(\left|x'\right|_0 \mid x' \text{ is accepted}\right) \\ &= |x|_0 \cdot \left(1 - \frac{1}{2\,|x|_0}\right) + (|x|_1 - \mathrm{LO}(x)) \cdot \frac{1}{2\,|x|_1} \\ &= |x|_0 - \frac{\mathrm{LO}(x)}{2\,|x|_1}. \end{aligned}$$

A necessary and sufficient condition for the acceptance of x' is that the first $\mathrm{LO}(x)$ 1-bits do not flip. Thus we have

$$p := \mathrm{Prob}\left(x' \text{ is accepted}\right) = \left(1 - \frac{1}{2\,|x|_1}\right)^{\mathrm{LO}(x)}$$

which is bounded below by $1/2$ since $\mathrm{LO}(x) \leq |x|_1$. Together,

$$\begin{aligned} \mathrm{E}\left(\left|x'\right|_0\right) &= \left(|x|_0 - \frac{\mathrm{LO}(x)}{2\,|x|_1}\right) \cdot p + |x|_0 \cdot (1-p) \\ &= |x|_0 - p \cdot \frac{\mathrm{LO}(x)}{2\,|x|_1} \\ &\leq |x|_0 - \frac{\mathrm{LO}(x)}{4\,|x|_1}. \end{aligned}$$

Hence,

$$\mathrm{E}\left(|x|_0 - \left|x'\right|_0\right) \geq \frac{\mathrm{LO}(x)}{4\,|x|_1} \geq \frac{n^{-1/2}}{4}$$

and using drift arguments by He and Yao [10], we obtain an upper bound $n/(n^{-1/2}/4) = 4n^{3/2}$ on the expected time to complete phase 2. □

Theorem 8. *Given some constant* $0 < c < 1$*, let* $a \in \{0,1\}^n$ *be chosen uniformly at random among all search points with* cn *1-bits. The expected (w. r. t. the random bits of* a *and the algorithm's decisions) optimization time of the asymmetric (1+1) EA on* $\mathrm{LO}_a(x)$ *is* $\Theta(n^2)$.

Proof. The upper bound follows from Theorem 6.

W. l. o. g., n is even and $c \leq 1/2$. Let $k = cn$. Consider a pair of neighbored bits in a, say (a_i, a_{i+1}), such that $a_i \neq a_{i+1}$. Such a constellation is called a transition. Let T be the number of transitions in the whole bit string and T_ℓ, T_r be the number of transitions within the bits $a_1, \dots, a_{n/2}$

and $a_{n/2+1}, \ldots, a_n$, resp. We first estimate $\mathrm{E}(T)$. We have $\binom{n}{k}$ possibilities to choose a. The event $a_i \neq a_{i+1}$ for some $1 \leq i \leq n-1$ occurs if exactly one of these variables is 1 and the remaining $k-1$ 1-bits are distributed among the remaining $n-2$ bits. Hence

$$\mathrm{Prob}\left(a_i \neq a_{i+1}\right) = 2\binom{n-2}{k-1}\binom{n}{k}^{-1} = \frac{2k(n-k)}{n(n-1)}.$$

By the linearity of expectation, along with $k \leq n/2$, we have

$$\mathrm{E}(T) = (n-1) \cdot \frac{2k(n-k)}{n(n-1)} = \frac{2k(n-k)}{n} \geq k.$$

Due to symmetry T_ℓ and T_r are due to the same probability distribution and since $(a_{n/2}, a_{n/2+1})$ is excluded we have $\mathrm{E}(T_\ell) = \mathrm{E}(T_r) \geq \mathrm{E}(T)/2 - 1 \geq k/2 - 1$.

We claim that T_ℓ and T_r are strongly concentrated. Observing

$$|\mathrm{E}(T_\ell \mid a_1, \ldots, a_i) - \mathrm{E}(T_\ell \mid a_1, \ldots, a_{i-1})| \leq 1$$

we can apply the method of bounded martingale differences (see, e. g., [19]) yielding

$$\mathrm{Prob}\left(T_\ell \leq \frac{k}{2} - \frac{k}{4}\right) \leq e^{-\left(\frac{k}{4}-1\right)^2/n} = e^{-\Omega(n)}.$$

Hence, with overwhelming probability $T_\ell \geq k/4$ and $T_r \geq k/4$.

Assuming $T_\ell \geq k/4$ implies that both the number of 1-bits and the number of 0-bits among the bits $a_1, \ldots, a_{n/2}$ is bounded below by $k/8$ as every bit contributes to at most 2 transitions. Recalling that $\overline{a}$ is the global optimum, a direct consequence is that whensoever a search point x with $\mathrm{LO}_a(x) \geq n/2$ is mutated the mutation probability for a specific bit is at most $4/k$.

Let x^* be the first search point reached during the optimization process where $\mathrm{LO}_a(x^*) \geq n/2$. We bound the expected optimization time by the expected time to find the optimum starting with x^*. Let

$$d(x) := |\{i \mid \mathrm{LO}_a(x) + 1 < i < n \wedge a_i \neq a_{i+1}\}|$$

be the number of transitions to the right of the leftmost bit differing in x and $\overline{a}$. We now apply a drift analysis in order to estimate the expected time until the current population's d-value has decreased to 0, which is necessary to find the optimum.

Let x be the current population and x' be the population in the next generation. A necessary condition for $d(x') < d(x)$ is that the leftmost differing bit flips which has probability at most $4/k$. Moreover, the positions of the bits to the right have not yet had any influence on the fitness up to now, hence for any j with $\mathrm{LO}_a(x) + 1 < j < n$

$$\mathrm{Prob}\left(x_j = 1 \wedge x_{j+1} = 0\right) = \mathrm{Prob}\left(x_j = 0 \wedge x_{j+1} = 1\right) \leq \frac{1}{2}.$$

We observe that in case the LO_a-value increases the two bits of the following transition both match the optimum $\overline{a}$ with probability at most 1/2. Transitions may overlap, however two matching bits can decrease the d-value by at most 2. In case both these bits match we consider the first transition in $x_{j+2}, \ldots, x_n$ and repeat the argumentation with independent events. If the considered bits do not both match, the d-value decreases by at most 2. We arrive at

$$\mathrm{E}\left(d(x) - d(x')\right) \leq \frac{4}{k} \cdot \left(2\sum_{i=0}^{\infty} 2^{-i}\right) = \frac{16}{k}.$$

By the same arguments $\mathrm{E}\left(d(x^*)\right) \geq T_r - 4 \geq k/4 - 4$ follows and drift analysis yields the bound

$$\frac{\mathrm{E}\left(d(x^*)\right)}{16/k} \geq \left(\frac{k}{4} - 4\right) \cdot \frac{k}{16} = \frac{c^2 n^2}{64} - O(n).$$

Note that this bound also holds when multiplying with the probability $1 - e^{-\Omega(n)}$ for $T_\ell, T_r \geq k/4$ and we have proved the theorem. □

Theorem 10. *The probability that the asymmetric (1+1) EA optimizes* Plateau *within $2^{o(n^{1/6})}$ steps is bounded above by $2^{-\Omega(n^{1/6})}$.*

Proof. By Lemma 2, the probability to flip at least $n^{1/4}$ bits in one mutation is bounded above by $2^{-\Omega(n^{1/4}\log n)}$. Thus, the probability that such a mutation occurs within $2^{o(n^{1/6})}$ generations is bounded above by $2^{-\Omega(n^{1/4}\log n)}$.

With probability $1 - 2^{-\Omega(n^{2/3}\log n)}$, the first population on the plateau contains at most $3n^{2/3}$ 1-bits. We call a step *relevant* if the offspring y replaces its parent x, $y \neq x$, and x is a search point on the plateau. This implies that in a relevant step, we have a movement on the plateau. Secondly, we consider phases of $t = n^{5/12}$ relevant steps. Let $S := \{1^i 0^{n-i} \mid 2n^{2/3} \leq i \leq 3n^{2/3}\}$. The first phase starts when a point in S is reached for the first time. The i-th phase, $i \geq 2$, starts after the $(i-1)$-th phase has ended and when a point in S is reached for the next time. We ignore non-relevant steps and steps between two phases in our bound as they can only increase the optimization time. We remark that during a phase, if no more than $n^{1/4}$ bits flip in one mutation, only search points in $\{1^i 0^{n-i} \mid n^{2/3} \leq i \leq 4n^{2/3}\}$ are traversed.

Let E^+ denote the event that one relevant step increases the number of 1-bits in the population. Analogously, let E^- denote the event that one relevant step decreases the number of 1-bits in the population. As long as

$|x|_1 = \Theta(n^{2/3})$ holds, we can find an upper bound on $\text{Prob}\,(E^+)$ as follows. On the plateau, for each value of i there is at most one search point with Hamming distance i and a larger number of 1-bits. This yields

$$\text{Prob}\left(E^+\right) \leq \sum_{i=1}^{|x|_0} \left(\frac{1}{2\,|x|_0}\right)^i < \sum_{i=1}^{\infty} n^{-i} = O(1/n).$$

Clearly, the probability for a direct mutation to a Hamming neighbor on the plateau is a lower bound on $\text{Prob}\,(E^-)$. By the proof of Lemma 1, we have $\text{Prob}\,(E^-) \geq 1/(8\,|x|_1) = \Omega(n^{-2/3})$.

Thus, the conditional probability of $|y|_1 > |x|_1$ given a relevant step creating y from x is

$$\text{Prob}\left(E^+ \mid E^+ \cup E^-\right) = \frac{O(1/n)}{\Omega(n^{-2/3})} \leq cn^{-1/3} =: q$$

for some constant $c > 0$.

Let x_{start} be the first search point and x_{end} be the last search point within the current phase. In steps towards the optimum, the maximal step size is $n^{1/4}$ and in all other relevant steps, we move away from the optimum by at least 1. If the algorithm makes $r < n^{1/6}/2$ steps towards the optimum, this implies

$$|x_{\text{end}}|_1 \leq |x_{\text{start}}|_1 + r \cdot n^{1/4} - (t - r) \leq |x_{\text{start}}|_1 .$$

Thus, a necessary condition for $|x_{\text{start}}|_1 > |x_{\text{end}}|_1$ is to have at least $m := n^{1/6}/2$ steps towards the global optimum. W. l. o. g. $m \in \mathbb{N}$, then we have

$$\begin{aligned}
&\text{Prob}\,(|x_{\text{end}}|_1 > |x_{\text{start}}|_1) \\
&\leq \binom{t}{m} \cdot q^m \leq \left(\frac{t}{m} \cdot q\right)^m \\
&= \left(2c \cdot n^{-1/12}\right)^{n^{1/6}/2} = 2^{-\Omega(n^{1/6})}.
\end{aligned}$$

Since $|x_{\text{start}}|_1 > |x_{\text{end}}|_1$ is a necessary event for finding the optimum, we have proved the theorem. □