UNIVERSITY OF DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods

Design and Analysis of an Asymmetric Mutation Operator

Thomas Jansen and Dirk Sudholt

No. CI-195/05

Technical Report ISSN 1433-3325 May 2005

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/XI
44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, “Computational Intelligence,” at the 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

Abstract

Evolutionary algorithms as general randomized search heuristics typically perform a random search that is biased only by the fitness of the search points encountered. In practical applications the use of biased variation operators suggested by problem-specific knowledge 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 one-bits or very few zero-bits. Here, one specific mutation operator that is tailored towards such situations is defined and analyzed. The assets and drawbacks of this mutation operator are discussed. This is done by presenting analytical results on illustrative example functions as well as on function classes.

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

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

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 [3] 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 equal performance [6]. 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.

Here, we consider one specific mutation operator that is plausible when it is known that good solutions typically contain either very few bits with value zero or almost only bits with this value. Many real-world problems share this specific property. One example is the problem of computing a minimal spanning tree [9]. 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 edges selected correspond to bits with value 1. Most graphs with m nodes contain $\Theta(m^2)$ edges whereas trees contain only $m-1$ edges. The most common mutation operator for bit strings of length n flips each bit independently with probability $1/n$. In case we have got a bit string representing one such tree and we wait for another tree to be generated by mutation, this operator is quite slow since it tends to create offspring with a larger number of ones. Biasing the search towards strings with few one-bits may lead to a significant speed-up.

We introduce a mutation operator that, on average, preserves the number of one-bits. Considering the search space $\{0,1\}^n$ one can think of all points with exactly i one-bits as forming the i-th 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. Standard bit mutations tend to sample the search space uniformly. This implies that the random walk 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 one-bits, there is a strong tendency to return to levels $i \approx n/2$ since these levels have got a much larger size. The mutation operator defined here tends, on average, not to change a level. This implies that the random walk induced by repeating these asymmetric bit mutations spends more time on levels with very few or lots of one-bits since the above-mentioned tendency to levels with medium numbers of one-bits is missing.

At first glance, this behavior seems contradictory. We have designed an operator where, on average, the current level of a random walk is preserved. But since there is no tendency to the medium levels, the random walk based on the asymmetric bit mutations is more likely to reach levels with very few or lots of one-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 one-bits is either small or large, this mutation operator is more efficient in generating other such search points at random. The mutation operator is, however, 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 one-bits. It has to be noted, though, that it is not an unbiased operator as defined by Droste and Wiesmann [4] (assuming Hamming distance to be a natural metric in $\{0,1\}^n$). This paper, however, is not about the design of a specific mutation operator for a specific kind of problem and the demonstration of its usefulness. Our aim is to present a broad and informative theoretical 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 prove the assets and drawbacks of this specific mutation operator in a clear and intuitive, yet rigorous way. In addition to this concrete analysis this is meant to be 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 analyze the performance on simply structured example functions and prove that the operator shows increased efficiency as expected. In a more general context, we prove 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.

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., $|x|_1 = \sum\limits_{i=1}^{n} x_i$. 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 $1/|x|_i$ since this avoids that the mutation operator becomes deterministic for 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 but other constants $c > 1$ may be used instead.

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. Sometimes, the expected optimization time is biased by rare events that yield overly large run times. Thus, we accept the success probability after t steps, i. e., the probability to find a global optimum within the first t function evaluations, as additional robust measure of the efficiency of algorithms. Wegener [10] links these two measures by presenting a general restart scheme with the following property. A randomized search heuristic with a success probability that is bounded below by a term converging to 0 polynomially fast within a polynomial number of steps is turned into a randomized search heuristic with expected polynomial optimization time.

When using standard bit mutations, the probability for mutating 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 $\mathrm{H}(x,y) = \sum_{i=1}^{n}(x_i \oplus y_i)$ where $x_i \oplus y_i$ denotes the exclusive or of x_i and y_i. 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 [1] where the generalization of objective functions is considered in the context of black-box complexity.

Definition 1. *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 ones in the current bit string and since it treats one-bits and zero-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 one-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 zeros by ones 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.

There is a number of well-known example functions that we want to consider in the following sections. We give precise formal definitions here and cite results on the expected optimization time of the (1+1) EA with standard bit mutations. These results are used as a bottom-line for the comparison when we use the asymmetric bit mutations instead.

Definition 2. *We define the following functions on* $\{0,1\}^n$*.*

$$\begin{aligned}
\textsc{OneMax}(x) &:= |x|_1 \\
\textsc{Needle}(x) &:= \prod_{i=1}^{n} x_i \\
\textsc{Ridge}(x) &:= \begin{cases} n - |x|_1 & \textit{if } x \notin \{1^i 0^{n-i} \mid 0 \leq i \leq n\} \\ n + i & \textit{if } x \in \{1^i 0^{n-i} \mid 0 \leq i \leq n\} \end{cases} \\
\textsc{Plateau}(x) &:= \begin{cases} n - |x|_1 & \textit{if } x \notin \{1^i 0^{n-i} \mid 0 \leq i \leq n\} \\ n + 1 & \textit{if } x \in \{1^i 0^{n-i} \mid 0 \leq i < n\} \\ n + 2 & \textit{if } x = 1^n \end{cases}
\end{aligned}$$

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

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

$E(T_{\textsc{OneMax}}) = \Theta(n \log n)$ *[2]*

$E(T_{\textsc{Needle}}) = \Theta(2^n)$ *[5]*

$E(T_f) = O(n \cdot d)$ *for unimodal functions f with d different function values [2]*

$E(T_{\textsc{Ridge}}) = \Theta(n^2)$ *[7]*

$E(T_{\textsc{Plateau}}) = O(n^3)$ *[7]*

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 zero-bits in the parent. This makes this mutation operator very different from standard bit mutations if the number of zero-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 2. *The expected optimization time of the (1+1) EA with asymmetric bit mutations on* OneMax *is bounded above by* $5.1n$.

Proof. As long as the current search point x is different from the unique global optimum, its fitness can be increased by a mutation of at least one zero-bit and no one-bit. The probability for such a mutation is given by

$$p(x) := \left(1 - \left(1 - \frac{1}{2\,|x|_0}\right)^{|x|_0}\right) \cdot \left(1 - \frac{1}{2\,|x|_1}\right)^{|x|_1}.$$

Using the inequalities

$$\frac{1}{2} \leq \left(1 - \frac{1}{2k}\right)^{k} \leq e^{-1/2}$$

for all $k \in \mathbb{N}$, we obtain $p(x) \geq (1 - e^{-1/2}) \cdot (1/2)$. Since for each fitness value $i \in \{0, 1, \ldots, n-1\}$ the fitness needs to be increased from i to some larger value at most once,

$$\sum_{i=0}^{n-1} \frac{2}{1 - e^{-1/2}} < 5.1n$$

is an upper bound on the expected optimization time. □

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. 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 $\textsc{OneMax}_a$ with $a = \overline{x^*}$. One may fear that the advantage of asymmetric bit mutations for OneMax is counterbalanced by a disadvantage when the global optimum is far away from 1^n. This, however, is not the case if one considers asymptotic expected optimization times.

Theorem 3. *Let $a \in \{0,1\}^n$ with $|a|_1 \leq n/2$. The expected optimization time of the (1+1) EA with asymmetric bit mutations on $\textsc{OneMax}_a$ is $\Theta(n \log(2 + |a|_1))$.*

Proof. The unique global optimum of $\textsc{OneMax}_a$ is $\overline{a}$. Let T denote the optimization time of the (1+1) EA with asymmetric bit mutations on $\textsc{OneMax}_a$. We begin with a proof of an upper bound on $\mathrm{E}(T)$.

We 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 least $|\overline{a}|_1 - |a|_1 = n - 2\,|a|_1$ one-bits for the first time. The second phase starts after the first phase and ends when the global optimum is found. Let T_1 and T_2 denote the lengths of the two phases. Thus, $T = T_1 + T_2$ holds.

We call a position i where $x_i = 0$ and $\overline{a}_i = 1$ an improving position for x. Clearly, if x is the current population, a mutation that flips only x_i increases the fitness. During the first phase $|x|_1 < |\overline{a}|_1 - |a|_1$ holds. Thus, there are at least $|a|_1$ improving positions for x. Analogously to the proof of Theorem 2, we obtain a lower bound $\Omega(1)$ on the probability to increase the fitness. Therefore, $\mathrm{E}\left(T_1\right) = O(n)$ follows.

The special case $|a|_1 = 0$ is dealt with in Theorem 2. Thus, we can assume $|a|_1 > 0$ here. In the second phase, $|x|_1 \geq n-2\,|a|_1$ holds. We have $\mathrm{H}\left(x, \overline{a}\right) \leq 3\,|a|_1$ since we have $\mathrm{H}\left(x, 1^n\right) = |x|_0 \leq 2\,|a|_1$ and $\mathrm{H}\left(1^n, \overline{a}\right) = |a|_1$. This implies $\textsc{OneMax}_a(x) \geq n - 3\,|a|_1$. There are $n - \textsc{OneMax}_a(x)$ Hamming neighbors with function value larger than x. It is not difficult to see that the probability to reach a specific Hamming neighbor by a direct mutation is bounded below by $1/(8n)$ (see Lemma 1 in the following section for a proof). Thus, the expected length of the second phase is bounded above by

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

For the lower bound we distinguish four cases with respect to $|a|_1$. Since the mutation operator is symmetric with respect to bit positions, we can assume that $a = 1^{|a|_1}0^{|a|_0}$ holds without loss of generality. It is important to remember that the Hamming distance to the unique global optimum cannot increase during a run. Chernoff bounds [8] yield that for any constant ε with $0 < \varepsilon < 1/2$ with probability $1 - 2^{-\Omega(n)}$ the initial Hamming distance to the unique global optimum is bounded below by $((1/2) - \varepsilon)n$ and bounded above by $((1/2) + \varepsilon)n$.

We begin with the special case $|a|_1 = 1$. With probability $1 - 2^{-\Omega(n)}$ the initial population has Hamming distance at most $(3/5)n - 1$ from the unique global optimum. Then, the probability to flip a zero-bit to one is bounded above by $1/(2 \cdot (3/5)n) = (5/6)n$. With probability $1/2$, the initial value of the left-most bit is 0. Remember that we have $a = 10^{n-1}$. Thus, the expected optimization time is bounded below by $(6/5)n$ in this case. This yields $\Omega(n)$ as lower bound.

For the special case $|a|_1 = 0$, we need to be more precise. Let $p_{i,j}$ (with $i < j$) denote the probability that a mutation of x with $|x|_1 = i$ leads to x' with $|x'|_1 = j$. Since at least $j - i$ zero-bits need to mutate, we have

$$p_{i,j} \leq \binom{|x|_0}{j-i} \left(\frac{1}{2\,|x|_0}\right)^{j-i} \leq \left(\frac{1}{2}\right)^{j-i}$$

as upper bound on $p_{i,j}$. With probability $1 - 2^{-\Omega(n)}$ the initial population has Hamming distance at least $(2/5)n$ and at most $(3/5)n$ from the unique global optimum. Let D denote the decrease in Hamming distance from the global optimum in one mutation. We can bound $\mathrm{E}\,(D)$ from above and have

$$\mathrm{E}\,(D) = \sum_{d=1}^{|x|_0} p_{|x|_1,|x|_1+d} \cdot d < \sum_{d=1}^{\infty} 2^{-d} \cdot d = 2$$

independent of the current population x. Note, that this independence is true for our bound, not for the actual expectation. Due to this homogeneity we have that the expected decrease in the Hamming distance in t generations is bounded above by $2t$. By Markov's inequality, the probability to have a decrease in Hamming distance by at least $4t$ in t generations is bounded above by $1/2$. Thus, with probability at least $(1/2)\cdot(1-2^{-\Omega(n)})$ the optimum is not reached within $n/10$ generations. This yields

$$\left(1 - 2^{-\Omega(n)}\right) \cdot \frac{1}{2} \cdot \frac{n}{10} = \Omega(n)$$

as lower bound on the expected optimization time.

Now we consider the case $1 < |a|_1 \leq n/4$. With probability at least $1/2$, the initial population contains at least $|a|_1/2$ one-bits in the first $|a|_1$ positions. Analogously, it contains at least $|a|_0/2$ one-bits in the last $|a|_0$ positions with probability at least $1/2$. In the following, we consider the case where both events occur which happens with probability at least $1/4$. Since the Hamming distance to the global optimum $0^{|a|_1}1^{|a|_0}$ can only decrease, the number of one-bits is bounded below by $|a|_0/2 - |a|_1 \geq n/8$ during the run. Thus, the probability that a one-bit is flipped is bounded above by $4/n$.

We have at least $|a|_1/2$ one-bits that all need to flip at least once. The probability that this does not happen 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

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

in this case.

Finally, we consider the case $n/4<|a|_1\leq n/2$. It is easy to see 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 $n/8-o(n)$. Then, the number of ones 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 third case. Repeating the line of thought from there completes the proof. □

We see that, asymptotically, there is no disadvantage for the (1+1) EA with asymmetrical bit mutations in comparison with standard bit mutations on $\textsc{OneMax}_a$. The reader might conclude from this result that the search is not clearly biased by asymmetrical bit mutations. However, for $\textsc{OneMax}_a$, 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.

In the following, we show that there is a clear bias due to asymmetric bit mutations which 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 have got the same fitness value, 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.

It will turn out to be important to consider the probabilities to increase and decrease the number of zero-bits depending on its current value. We compare standard bit mutations and asymmetric bit mutations with respect to this property. Figure 1 shows clear differences. More important than the different size of the probabilities (indicated by the different scaling) are the different shapes of the curves. This leads to a performance on NEEDLE that is surprising.

Theorem 4. *For any constant $k \in \mathbb{N}_0$ and all $a \in \{0,1\}^n$ with either at most k zero-bits or at most k one-bits, the success probability of the asymmetric (1+1) EA on NEEDLE_a after $O(n^2)$ steps is bounded below by $\Omega\left(n^{-k}\right)$. Making appropriate use of restarts, the expected optimization time is bounded above by $O(n^{k+2})$.*

Proof. Since the proof is somewhat involved and contains some tedious technical details, we concentrate on the main proof ideas here and refer to technical lemmas in the appendix for the details. First, we prove that the expected number of steps the asymmetric (1+1) EA needs to reach some $x \in \{0^n, 1^n\}$ for the first time is bounded above by $O(n^2)$. The probability that some string with either at most k zero-bits or at most k one-bits is an intermediate population in these $O(n^2)$ steps is bounded below $\Omega(1)$. Proving that each of these strings is the actual intermediate population with equal probability completes the proof for the success probability.

Let Z_t denote the random number of zero-bits in the current population of the asymmetric (1+1) EA in the t-th generation. We consider the random changes of Z_t during one run of the asymmetric (1+1) EA. Therefore, we are interested in the probability to increase and decrease Z_t. Let B_z^+ denote the event that the number of zero-bits is increased from z by at least 1 in one mutation. Analogously, let B_z^- denote the event that the number of zero-bits is decreased from z by at least 1 in one mutation. Due to symmetry reasons, $\text{Prob}\left(B^+_{\lfloor n/2\rfloor - i}\right) = \text{Prob}\left(B^-_{\lceil n/2\rceil + i}\right)$ holds for all values of i. This implies $\text{Prob}\left(B^+_{\lfloor n/2\rfloor}\right) = \text{Prob}\left(B^-_{\lceil n/2\rceil}\right)$ as a special case. We know from Lemma 3 (see appendix) that $\text{Prob}\left(B_z^+\right) \geq \text{Prob}\left(B_z^-\right)$ holds for $z \geq n/2$.

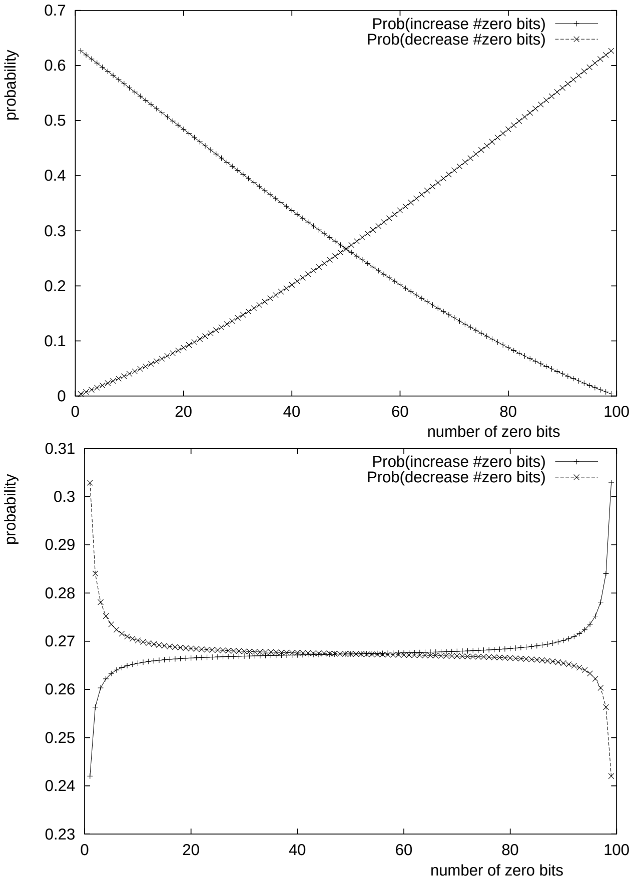

Figure 1: Probability to increase and decrease the number of zero-bits for standard bit mutations (top) and asymmetric bit mutations (bottom) with $n = 100$.

Clearly, $\mathrm{Prob}\left(B_z^+\right) + \mathrm{Prob}\left(B_z^-\right) = \Omega(1)$ holds. Thus, we may restrict ourselves to the consideration of generations where the value of Z_t changes. This does not change the order of magnitude of the number of generations needed for the minimum value of t with $Z_t \in \{0, n\}$. Let T denote this value of t. Let ${B'}_z^+$ and ${B'}_z^-$ denote the conditional probabilities to increase resp. decrease the number of zero-bits given that this number changes. It is well known that having $\mathrm{Prob}\left({B'}_z^+\right) = 1/2$ implies $\mathrm{E}\left(T\right) = O(n^2)$. We have $\mathrm{Prob}\left({B'}_z^+\right) \geq 1/2$ for $z \geq n/2$ and $\mathrm{Prob}\left({B'}_z^-\right) \geq 1/2$ for $z \leq n/2$. This tendency to increase the majority value of bits can only decrease $\mathrm{E}\left(T\right)$. This completes the proof for $k = 0$.

For $k > 0$, we concentrate on the case of a target point $\overline{a}$ with exactly k one-bits. The other case is symmetric. There are exactly $\binom{n}{k}$ bit strings with exactly k one-bits. For $k = O(1)$, $\binom{n}{k} = \Theta(n^k)$ holds. Consider the case when a string with k ones becomes the current population of the asymmetric (1+1) EA. For symmetry reasons, all strings with k ones have equal probability. The probability to change the number of one-bits in one generation by exactly one is bounded below by $\Omega(1)$. In addition, the probability to mutate i bits decreases exponentially with i. This implies that $\overline{a}$ becomes current population with probability $\Omega(1/n^k)$ within the first $O(n^2)$ generations.

For the statement on the expected optimization time, we stop a run after $O(n^2)$ generations and restart it. On average, $O(n^k)$ restarts are sufficient. □

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 [1] 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 [6]. 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 that meet the assumption that good bit strings have either many or few zero-bits. 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 function is closed under the transformation of objective functions considered here. I. e., for any $a \in \{0,1\}^n$, f_a is unimodal if and only if f is. Thus, the unique global optimum may be 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. Therefore, we are interested in the probability to reach a specific Hamming neighbor.

Lemma 1. *Let $x, x' \in \{0,1\}^n$ with $H(x,x') = 1$ be given. The probability to mutate x into x' in one asymmetric mutation is bounded below by $1/(8n)$.*

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

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

For $x = 0^n$ we obtain $1/(4n) > 1/(8n)$ as lower bound in the same way. □

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 [3].

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 proof are not different from the corresponding result for standard bit mutations.

Theorem 5. *Let $f\colon \{0,1\}^n \to \mathbb{R}$ be a unimodal function with d different function values. The expected optimization time of the (1+1) EA on f with asymmetric bit mutations 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 by at least 1 is bounded below by $1/(8n)$.

This yields $8n$ as upper bound for one mutation increasing the function value. Clearly, at most $d-1$ such mutations are sufficient to reach the global optimum. □

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. We consider RIDGE as one example and analyze the performance of the (1+1) EA with asymmetric bit mutations on RIDGE.

Theorem 6. *The expected optimization time of the (1+1) EA with asymmetric bit mutations 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 5. 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 point x^* with $f(x^*) \geq n+1$ that becomes current search point 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 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 RIDGE_a, this subsequence has to be traversed in order to optimize RIDGE_a. The expected decrease in Hamming distance to the global optimum on this subsequence in one mutation is $O(1/n)$. Using the same arguments as in the proof of Theorem 3 we obtain $\Omega(n^2)$ as lower bound on the expected optimization time. □

The performance of the (1+1) EA with standard bit mutations and asymmetric bit mutations are asymptotically equal on RIDGE_a. Even the proofs of the bounds are very similar [7]. 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 this ridge points into the direction of the global optimum. For PLATEAU, the function values are constant and the evolutionary algorithms 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 7. *The probability that the (1+1) EA with asymmetric bit mutations optimizes* PLATEAU *within* $2^{O(n^{1/4})}$ *steps is bounded above by* $2^{-\Omega(n^{1/4})}$.

Proof. It is easy to see that the probability to flip more than $n^{1/4}$ bits in one mutation is bounded above by $2^{-\Omega(n^{1/4}\log n)}$. Thus, the probability that such a mutation occurs within $2^{O(n^{1/4})}$ generations is bounded above by $2^{-\Omega(n^{1/4}\log n)}$.

With probability $1-2^{-\Omega(n)}$, the first population on the plateau contains at most $2n^{2/3}$ one-bits. We ignore steps on the plateau where the number of one-bits is smaller than $n^{2/3}$. This can only decrease the optimization time. We consider a phase of length $t = n^{5/4}$. Let E^+ denote the event that one mutation increases the number of one-bits in the population. Let L^+ denote the sum of the step lengths of all these steps in the phase. Analogously, let E^- denote the event that one mutation decreases the number of one-bits in the population. Let L^- denote the sum of the step lengths of all these steps in the phase. We make a crude worst case assumption: each step leading towards the global optimum has the maximal step length $n^{1/4}$ whereas each step leading away from the global optimum has the minimal step length 1. Obviously, this can only decrease the optimization time. Our assumption yields

$$\mathrm{E}\left(L^+\right) \leq \mathrm{Prob}\left(E^+\right) \cdot t \cdot n^{1/4} = \mathrm{Prob}\left(E^+\right) \cdot n^{3/2}$$

and

$$\mathrm{E}\left(L^-\right) \geq \mathrm{Prob}\left(E^-\right) \cdot t = \mathrm{Prob}\left(E^+\right) \cdot n^{5/4}.$$

As long as $|x|_1 = \Theta(n^{2/3})$ holds, we can find a upper bound on $\text{Prob}\left(E^+\right)$ as follows. On the plateau, there is for each value of i at most one search point with Hamming distance i and a larger number of one-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 $\text{Prob}\left(E^-\right)$. By the proof of Lemma 1, we have $\text{Prob}\left(E^-\right) \geq 1/(8\,|x|_1) = \Omega(n^{-2/3})$. Taking the condition into account that at most $n^{1/4}$ bits flip simultaneously does not change the order of growth of these bounds. Let L'^+ and L'^- denote values corresponding to L^+ and L^- under the condition that at most $n^{1/4}$ bits flip simultaneously. Then $\text{E}\left(L'^+\right) = O(n^{1/2})$ and $\text{E}\left(L'^-\right) = \Omega(n^{7/12})$. Applications of Chernoff bounds yield that with probability $1-2^{-\Omega(n^{1/2})}$, the random values of L'^+ and L'^- are within the same order of growth as their expectations. Thus, with probability $1 - 2^{-\Omega(n^{1/4}\log n)}$, the Hamming distance to the optimum is not decreased at the end of this phase. This completes the proof. □

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

Proof. It follows from the result on OneMax_a (Theorem 3) that some point on the plateau will on average be found 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 the following holds. 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 $|x^{+i}|_1 = |x^{-i}|_1$. Furthermore, the choice of a_{01} implies that $|x|_1 = n/2 - O(1)$ 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 [7]. □

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 one-bits. The operator is designed in a way that on average the number of one-bits is not changed. This helps to bias the search towards areas of the search space with bit strings containing either very few zero-bits or very few one-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 mutation 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. Furthermore, we demonstrated exemplarily for one unimodal function that this upper bound can be tight.

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 and 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.

We believe that our analysis draws a clear picture of the advantages and disadvantages that come with the asymmetric mutation operator. However, many questions remain open. Clearly, the result on Needle is weak. While discussing that there is some bias towards the “end” of the search space we have not been able to give good bounds on the size of this bias. More precise results would be useful to extend our result on success probabilities to a result on the expected optimization time without restarts. Furthermore, there are other interesting example functions where a careful analysis of the effects of asymmetric mutations may be worthwhile.

While motivated by practical applications, our presentation is purely theoretical. It would be interesting to compare the performance of the asymmetric mutation operator in applications where one suspects that good solutions have only a few bits with value zero or one.

References

[1] Stefan Droste, Thomas Jansen, Karsten Tinnefeld, and Ingo 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.

[2] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[3] Stefan Droste, Thomas Jansen, and Ingo Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 2005. To appear. http://www.springerlink.com/index/10.1007/s00224-004-1177-z.

[4] Stefan Droste and Dirk Wiesmann. Metric based evolutionary algorithms. In *Proceedings of the Third European Conference on Genetic Programming (EuroGP 2000)*, pages 29–43, 2000.

[5] Josselin Garnier, Leila Kallel, and Marc Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203, 1999.

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

[7] Thomas Jansen and Ingo 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.

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

[9] Frank Neumann and Ingo 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.

[10] Ingo Wegener. Simulated annealing beats metropolis in combinatorial optimization. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005)*, 2005. To appear. Technical report: http://eccc.uni-trier.de/eccc-reports/2004/TR04-089.

A Technical Lemmas

Definition 3. *Let $p_z(i)$ denote the probability to mutate exactly i out of z zero-bits. Let $q_z(i)$ denote the probability to mutate exactly i out of $n - z$ one-bits. Let B_z^+ denote the event that the number of zero-bits is increased from z by at least 1 in one mutation. Analogously, let B_z^- denote the event that the number of zero-bits is decreased from z by at least 1 in one mutation.*

Clearly,

$$p_z(i) = q_{n-z}(i) = \binom{z}{i} \left(\frac{1}{2z}\right)^i \left(1 - \frac{1}{2z}\right)^{z-i}$$

holds. Due to our definitions of $p_z(i)$ and $q_z(i)$, we have the following probabilities:

$$\begin{aligned} \text{Prob}\left(B_z^+\right) &= \sum_{i=1}^{n-z} \sum_{j=0}^{\min\{i-1,z\}} p_z(j) q_z(i) \\ \text{Prob}\left(B_z^-\right) &= \sum_{i=1}^{z} \sum_{j=0}^{\min\{i-1,n-z\}} p_z(i) q_z(j) \end{aligned}$$

Lemma 2. *For any $z \geq \lfloor n/2 \rfloor$ and $i \geq 2$*

$$\begin{aligned} p_z(1) &\leq p_{z+1}(1) \\ p_z(i) &\geq p_{z+1}(i) \\ q_z(1) &\geq q_{z+1}(1) \\ q_z(i) &\leq q_{z+1}(i) \end{aligned}$$

hold.

Proof. Since we have $p_z(i) = q_{n-z}(i)$ for all z and i, it suffices to prove the statements on $p_z(i)$. We consider $p_{z+1}(i)/p_z(i)$ for some $z > \lfloor n/2 \rfloor$ and some $i < z - 1$.

$$\begin{aligned} \frac{p_{z+1}(i)}{p_z(i)} &= \frac{\binom{z+1}{i} \left(\frac{1}{2(z+1)}\right)^i \left(1 - \frac{1}{2(z+1)}\right)^{z+1-i}}{\binom{z}{i} \left(\frac{1}{2z}\right)^i \left(1 - \frac{1}{2z}\right)^{z-i}} \\ &= \frac{z+1}{z-i+1} \cdot \left(\frac{z}{z+1}\right)^i \cdot \left(\frac{z+1/2}{z+1} \cdot \frac{z}{z-1/2}\right)^{z-i} \cdot \frac{z+1/2}{z+1} \\ &= \frac{z+1}{z-i+1} \cdot \left(\frac{z(z+1/2)}{(z+1)(z-1/2)}\right)^z \cdot \left(\frac{z-1/2}{z+1/2}\right)^i \cdot \frac{z+1/2}{z+1} \\ &= \frac{z-1/2}{z-i+1} \cdot \left(\frac{z(z+1/2)}{(z+1)(z-1/2)}\right)^z \cdot \left(\frac{z-1/2}{z+1/2}\right)^{i-1} \end{aligned}$$

Note that $(z(z+1/2)/((z+1)(z-1/2)))^z$ converges to 1 from above. Due to our restriction to cases with $z > \lfloor n/2 \rfloor$ we have z large and may ignore this term. Since i is bounded above by z, $((z-1/2)/(z+1/2))^{i-1}$ is bounded below by $1/e$. Clearly, $(z-1/2)/(z-i+1)$ is increasing with i and becomes larger than e for $i \geq (1-1/e)z + 1 + 1/(2e)$. Thus, for such values of i the proof is complete. Using $i < (1-1/e)z + 1 + 1/(2e)$ yields a better lower bound on $((z-1/2)/(z+1/2))^{i-1}$ and we can iterate our argument. We see that we are looking for a smallest value of i such that $p_{z+1}(i)/p_z(i) \geq 1$ holds. For $i = 1$ we get

$$\frac{p_{z+1}(1)}{p_z(1)} = \frac{z-1/2}{z} \cdot \left(\frac{z(z+1/2)}{(z+1)(z-1/2)}\right)^z < 1$$

For $i = 2$ we have

$$\frac{p_{z+1}(2)}{p_z(2)} = \frac{z-1/2}{z-1} \cdot \left(\frac{z(z+1/2)}{(z+1)(z-1/2)}\right)^z \cdot \frac{z-1/2}{z+1/2} > 1$$

completing the proof. □

Lemma 3. *For $z > \lfloor n/2 \rfloor$, Prob$(B_z^+) \geq$ Prob(B_z^-) holds.*

Proof. Since we are looking for an asymptotic result that is valid for sufficiently large n, we may assume that z is sufficiently large. This allows us to use the limits of terms for $n \to \infty$ as approximation.

The number of flipping zero-bits is binomially distributed with parameters z and $1/(2z)$. Since z is sufficiently large, this distribution can be approximated by the Poisson distribution with parameter $\lambda = 1/2$. This yields $p_z(i) \approx 1/(2^i \cdot e^{1/2} \cdot i!)$ and we see that $p_z(i)$ decreases with i exponentially fast. The number of flipping one-bits is binomially distributed with parameters $n-z$ and $1/(2(n-z))$. The approximation of this distribution by the Poisson distribution is only valid if z is not too large. However, clearly $q_z(i)$ decreases with i exponentially fast, too. Thus, whether Prob(B_z^+) is larger or smaller than Prob(B_z^-) is determined by the first few terms of the sum.

We have

$$p_z(i) \approx \frac{1}{2^i \cdot e^{1/2} \cdot i!}$$

and compare this with

$$q_z(i) = \binom{n-z}{i} \left(\frac{1}{2(n-z)}\right)^i \left(1 - \frac{1}{2(n-z)}\right)^{n-z-i}$$

for small values of i. Clearly,

$$q_z(i) > \frac{1}{2^i \cdot i!} \cdot \left(\frac{n-z-i}{n-z}\right)^{i-1} \left(1 - \frac{1}{2(n-z)}\right)^{n-z-i}$$

holds. We know that $(1 - 1/(2(n-z)))^{n-z-i} > e^{-1/2}$ holds. For $i = 1$, we have $q_z(1) > p_z(1)$ as an immediate consequence. We know from Lemma 2 that with increasing values of z this tendency increases. This implies that $\text{Prob}\,(B_z^+)$ grows with z for $z \geq n/2$. □