Runtime Analysis of the (μ+1) EA on Simple Pseudo-Boolean Functions

Carsten Witt*
FB Informatik, LS 2, Universität Dortmund,
44221 Dortmund, Germany
carsten.witt@cs.uni-dortmund.de

November 14, 2005

Abstract

Although Evolutionary Algorithms (EAs) have been successfully applied to optimization in discrete search spaces, theoretical developments remain weak, in particular for population-based EAs. This paper presents a first rigorous analysis of the (μ+1) EA on pseudo-Boolean functions. Using three well-known example functions from the analysis of the (1+1) EA, we derive bounds on the expected runtime and success probability. For two of these functions, upper and lower bounds on the expected runtime are tight, and on all three functions, the (μ+1) EA is never more efficient than the (1+1) EA. Moreover, all lower bounds grow with μ. On a more complicated function, however, a small increase of μ provably decreases the expected runtime drastically.

This paper develops a new proof technique that bounds the runtime of the (μ+1) EA. It investigates the stochastic process for creating family trees of individuals; the depth of these trees is bounded. Thereby, the progress of the population towards the optimum is captured. This new technique is general enough to be applied to other population-based EAs.

1 Introduction

Evolutionary Algorithms (EAs) are successfully applied to optimization tasks, but theoretical knowledge is still far behind practical experience. In recent years, advances have been made in the theoretical analysis of the computational time complexity of EAs, in particular for the optimization in discrete search spaces induced by pseudo-Boolean functions $f\colon \{0,1\}^n \to \mathbb{R}$. However, this kind of theory often concentrates on simple single-individual EAs such as the (1+1) EA (e. g. Garnier et al., 1999; Droste et al., 2002; Wegener, 2002) and its coevolutionary variants (Jansen and Wiegand, 2004). Therefore, it does not explain the utility of populations employed in many real-world EAs. Theoretical analyses of the impact of crossover operators in population-based EAs (e. g. Jansen and Wegener, 2001c) do not necessarily explain why a large population might be beneficial (Storch and Wegener, 2003). Hence, in this paper, we consider instead population-based EAs where mutation is the only search operator. Such EAs have been studied by Jansen and Wegener (2001b) and Witt (2003), who prove that a population can be beneficial with fitness-proportional selection schemes. Recently, Storch (2004) analyzed steady-state EAs using uniform selection and diversity-maintaining operators. Moreover, He and Yao

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

(2002) studied some variants of (μ+μ) EAs. However, results on the time complexity of standard (μ+λ) EAs with uniform selection are not available for $\mu > 1$, i. e., non-trivial sizes of the parent population. Up to now, only analyses of (1+λ) EAs (Jansen et al., 2005) are known.

The aim of this paper is to contribute to a theory of standard (μ+λ) EAs with $\mu > 1$ in discrete search spaces. Here, we start with the simple case $\lambda = 1$ and consider a (μ+1) EA that is a generalization of the (1+1) EA for the search space $\{0,1\}^n$. In particular, we study the behavior of the (μ+1) EA on example functions and compare the obtained results with those for the (1+1) EA. To this end, a new and general proof technique for bounding the expected runtime of the (μ+1) EA is developed. An advantage of the new technique is that it has not been designed for a special mutation operator. In particular, we are able to analyze the (μ+1) EA with a global search operator that may flip many bits. In other contexts, analysis of EAs is much more difficult with a global than with a local search operator (for such examples see, e. g., Wegener and Witt, 2005).

The paper is structured as follows. In Section 2, we define the (μ+1) EA and the three example functions considered. Furthermore, we introduce the notion of family trees which will be used later in the new proof techniques. In Section 3, upper bounds on the expected runtime of the (μ+1) EA on the functions are presented. In Section 4, we describe how family trees can be applied to derive a general lower bound on the expected runtime and on the success probability of the (μ+1) EA. This tool is used again in Section 5 to prove more specific lower bounds. These bounds are tight for two examples and show for all three examples that the (μ+1) EA is asymptotically no more efficient than the (1+1) EA. However, it is a common belief that a population helps to better explore the search space, and one should find a situation where the (μ+1) EA with $\mu > 1$ outperforms the (1+1) EA. Indeed, a function where a small increase of μ decreases the expected runtime drastically is identified in Section 6. We then finish with some conclusions.

2 Definitions

We obtain the (μ+1) EA for the maximization of functions $f\colon \{0,1\}^n \to \mathbb{R}$ as a generalization of the well-known (1+1) EA (see Droste et al., 2002). As in continuous search spaces, a pure (μ+1) evolution strategy should do without recombination and should employ a uniform selection for reproduction. As usual, a truncation selection is applied for replacement. The mutation operator should be able to search globally, i. e., to flip many bits in a step. Therefore, a standard mutation flipping each bit with probability $1/n$ seems sensible. These arguments lead to the following definition.

Definition 1 ((μ+1) EA)

1. *Choose μ individuals $x^{(i)} \in \{0,1\}^n$, $1 \le i \le \mu$, uniformly at random. Let the multiset $X^{(0)} = \{x^{(1)}, \dots, x^{(\mu)}\}$ be the population at time 0. Let $t := 0$.*
2. *Repeat*
   - (a) *Choose some x from the population $X^{(t)}$ at time t uniformly at random.*
   - (b) *Create x' by flipping each bit of x independently with probability $1/n$. Let X' be the population obtained by adding x' to $X^{(t)}$.*
   - (c) *Create the multiset $X^{(t+1)}$, the population at time $t+1$, by deleting an individual with lowest f-value from X' uniformly at random.*
   - (d) *Set $t := t + 1$.*

We have kept the definition of the (μ+1) EA as simple as possible and refrain from employing diversity-maintaining mechanisms. Therefore, the (μ+1) EA with $\mu = 1$ is very similar to the (1+1) EA, yet differs in one respect. If an individual created by mutation has the same f-value as its parent, either one is retained with equal probability.

As usual in theoretical investigations, we leave the stopping criterion of the (μ+1) EA unspecified and analyze the number of iterations (also called *steps*) of the infinite loop until for the first time the current population contains an optimal individual, i. e., one that maximizes f. We then say that the (μ+1) EA *has reached the optimum.* The sum of this number of steps and the population size μ is denoted as the *runtime* of the (μ+1) EA and corresponds to the number of function evaluations executed so far (a common approach in black-box optimization, cf. Droste et al., 2006). Throughout the paper, we consider only $\mu = \mathit{poly}(n)$, i. e., values of μ bounded by a polynomial of n. For super-polynomial values, even initialization alone would produce a super-polynomial runtime.

As mentioned in the introduction, we study the (μ+1) EA on well-known functions already considered with the (1+1) EA. These functions are well structured and serve as a starting point for the analysis w. r. t. to more complicated problems. Since the following three functions are meant to exhibit typical behavior of EAs, they are often called example functions. The first one is the famous function

$$\textsc{OneMax}(x) \;=\; x_1 + \cdots + x_n,$$

which counts the number of ones of a search point $x \in \{0,1\}^n$. The second,

$$\textsc{LeadingOnes}(x) \;=\; \sum_{i=1}^{n}\prod_{j=1}^{i} x_j,$$

counts the number of leading ones. Finally, we investigate the function

$$\mathrm{SPC}(x) \;=\; \begin{cases} n+1 & \text{if } x = 1^i0^{n-i} \text{ for } i < n,\\ 2n & \text{if } x = 1^n,\\ n - \textsc{OneMax}(x) & \text{otherwise} \end{cases}$$

(SPC stands for *short path with constant fitness)*, which was introduced by Jansen and Wegener (2001a). The search points that can be written as 1^i0^{n-i}, $i < n$, form a so-called plateau, i. e., a region of the search space that is connected via 1-bit flips and has the same fitness, where the unique optimum is connected to one end of the plateau. This function is of particular interest since the (μ+1) EA will have to search within the plateau of constant fitness without guidance by fitness values. An interesting question is whether or not this exploration of the plateau benefits from the population.

To elucidate the utility of the (μ+1) EA's population, throughout the paper, we compare the (μ+1) EA with μ parallel runs of the (1+1) EA. The total cost (neglecting initialization cost) of t steps of the (μ+1) EA corresponds to the cost of performing μ parallel runs of the (1+1) EA until time t/μ. Thus, if we consider the (μ+1) EA at time t, we denote μ parallel runs of the (1+1) EA considered at time t/μ as the *corresponding parallel run.* In particular, we are interested in examples where the (μ+1) EA is significantly more efficient than its corresponding parallel run.

In order to derive runtime bounds for the (μ+1) EA, it is helpful to consider the so-called *family trees* of the individuals from the initial population. This concept has been introduced in a different context by Rabani et al. (1998) and has already been studied for different population-based EAs in discrete and continuous search spaces by Witt (2003) and Jägersküpper and Witt (2005), respectively. Let x_0 be an arbitrary individual from the initial population. We visualize the descendants of x_0 created by time $t \geq 0$ in the family tree $T_t(x_0)$ whose nodes denote time steps and are labeled with individuals generated at these steps; here it comes in handy that at each time step, exactly one individual is created. The edges of the tree correspond to mutation steps and model direct parent-child relations between the individuals labeling the nodes. In many of our considerations, the time steps where individuals are created do not matter, and for convenience, we will identify nodes of family trees with the individuals labeling these nodes. For instance, we will sometimes say that nodes of family trees *are* the individuals that label them. In this notion, $T_t(x_0)$ *contains* all descendants of x_0 produced by direct mutation and indirect mutation of x_0 until time t. Although formally, the tree is undirected, the points of time where individuals are created lead to an ordering of the corresponding nodes. Hence, descendants of x_0 are called *successors* of the root.

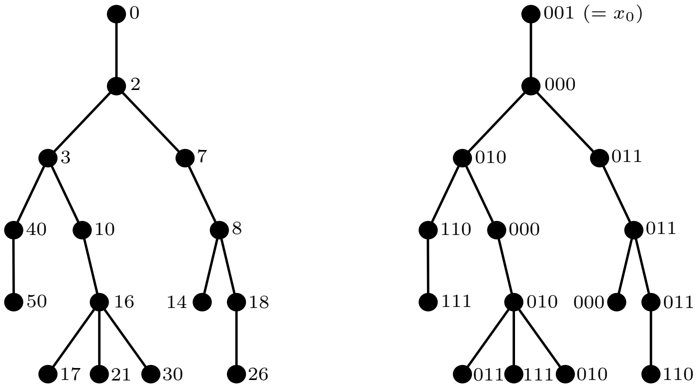

Figure 1: In a family tree, nodes denote the points of time where individuals appear (left picture) and are labeled with the search point of the individuals (right picture).

Formally, each $T_t(x_0)$ is an undirected graph (V_t, E_t) with $V_t \subseteq \mathbb{N}_0$ and a mapping $c_t\colon V_t \to \{0,1\}^n$. These components are obtained inductively. For $T_0(x_0)$, it holds that $V_0 = \{0\}$, $E_0 = \emptyset$ and $c_0(0) = x_0$. If at time $t \geq 0$, the $(\mu{+}1)$ EA chooses the individual x_0 for mutation and creates y, $V_{t+1} := V_t \cup \{t+1\}$, c_{t+1} is obtained from c_t by extending it via $c_{t+1}(t{+}1) := y$, and $E_{t+1} := E_t \cup \{\{0, t{+}1\}\}$. If at time t, the individual created in the mutation step at time $t' < t$ is chosen and $t' + 1 \in V_t$ holds, V_{t+1} and c_{t+1} are defined as before and $E_{t+1} := E_t \cup \{\{t'+1, t+1\}\}$. (Note that due to technical reasons, the individual created at time t labels node $t + 1$. Hence, the condition $t' + 1 \in V_t$ means that the individual created at time t' is contained in the tree at time t.) Otherwise $V_{t+1} := V_t$, $E_{t+1} := E_t$ and $c_{t+1} := c_t$; hence, in this case, $T_{t+1}(x_0) = T_t(x_0)$ because at time t, the $(\mu{+}1)$ EA chooses an individual contained in a different family tree. At any time t, there are μ different family trees $T_t(x_0)$, namely one tree for each x_0 from the initial population. Family trees whose roots are individuals not belonging to the initial population can be identified with subtrees of trees $T_t(x_0)$ for some individual x_0 from the initial population.

Since the $(\mu{+}1)$ EA describes an infinite stochastic process, the process growing the family trees $T_t(x_0)$ for $t \geq 0$ is infinite as well. Note, however, that $T_t(x_0)$ can contain individuals that have already been deleted from the population at time t. It can even happen that a given tree only contains deleted individuals. On the other hand, since at least one x_0 from the initial population always has alive descendants, the tree $T_t(x_0)$ of this x_0 contains arbitrarily many nodes for growing t. Finally, when studying family trees (and subtrees thereof), we omit the time index t when interested in the infinite stochastic process rather than a specific point of time.

Figure 1 shows one exemplary family tree with its labeling. The left-hand picture displays the node numbers while the right-hand picture contains the labels from $\{0,1\}^3$ belonging to the nodes. While the node numbers are monotonically increasing on paths, this need not be the case for the value of the goal function. If the goal function for the picture is, e. g., OneMax, the individuals labeling nodes 2 and 10 are deleted first. In the example, there are several points of time where the (μ+1) EA chooses from the family tree of another individual from the initial population. For example, this holds for $t = 0$ and $t = 3$ since nodes 1 and 4 are missing.

In this paper, family trees are used to capture the progress of the population towards the optimum of the goal function. Suppose that at some large time t, the depth of any family tree (w. r. t. to its root) is still very small. This depth bounds the length of every path from the root to a leaf in the tree, and, as mentioned above, such a path corresponds to a sequence of mutations that finally creates the individual at the leaf, starting from the individual at the root. Hence, if the depth is small, this means that all leaves are individuals that are with a high probability similar to the root. This makes the optimization of even simple functions very unlikely, which may lead to lower bounds on the runtime. On the other hand, lower bounds on the depth of family trees guarantee some progress and may imply upper bounds on the runtime. We will lower bound the depth of family trees in the following section in order to derive one specific upper bound on the runtime. In contrast, we will upper bound the depth of the trees in Section 4 to derive a very general lower bound on the runtime. More specific upper bounds on the depth of family trees and lower bounds on the runtime are shown in subsequent sections. Hence, while the family trees are useful in both respects, applications for lower bounding the runtimes of population-based EAs seem to be more general and common.

3 Upper Bounds

In this section, we will derive upper bounds on the expected runtime of the (μ+1) EA for the example functions. One may conjecture that these bounds are larger than the corresponding bounds for the (1+1) EA by at most a factor μ. Although this turns out to be true for the three examples, this need not always be the case as will be shown in Section 6.

The first two of the following bounds are not too difficult to obtain. The proofs use potential functions, a straightforward generalization of the proof technique of artificial fitness layers (cf. Wegener, 2002). We start with the problem for which we can present the easiest of the three proofs.

Theorem 1 *Let $\mu = poly(n)$. Then the expected runtime of the (μ+1) EA on* LeadingOnes *is bounded above by $\mu + 3en \cdot \max\{\mu \ln(en), n\} = O(\mu n \log n + n^2)$.*

Proof: Note that the term μ reflects the cost of initialization. We measure the progress to the optimum by the potential L, defined as the maximum LeadingOnes value of the current population's individuals. To increase L, it is sufficient to select an individual with value L (hereinafter called a *best individual*) and to flip only the leftmost zero. The probability of the latter equals $(1/n)(1-1/n)^{n-1}$. Hence, if there are i best individuals, the probability of the considered event is at least

$$\frac{i}{\mu} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{i}{e\mu n},$$

and the waiting time is at most $e\mu n/i$. The potential has to increase at most n times. Estimating i by 1 would lead to an upper bound of $\mu + e\mu n^2$ on the expected runtime.

However, the (μ+1) EA can produce replicas of individuals. If the number of best individuals is i, the probability of creating a replica of a best individual is bounded below by

$$\frac{i}{\mu} \cdot \left(1 - \frac{1}{n}\right)^n \ \geq\ \frac{i}{2e\mu}.$$

Furthermore, if $i < \mu$, this replica replaces a worse individual and increases the number of best ones. Assume pessimistically that L does not increase until we have at least $\min\{n/\ln(en), \mu\}$ best individuals. The expected time for this is at most

$$\sum_{i=1}^{\lceil n/\ln(en)\rceil - 1} \frac{2e\mu}{i} \ \leq\ 2e\mu\bigl(\ln(n/\ln(en)) + 1\bigr) \ \leq\ 2e\mu\ln(en)$$

since we can bound the k-th Harmonic number according to $\sum_{i=1}^{k} 1/i \leq (\ln k) + 1$. Now the expected time to increase L is at most $e\mu n/(\min\{n/\ln(en), \mu\})$. Altogether, the expected runtime is at most

$$\mu + n \cdot \left(2e\mu\ln(en) + \frac{e\mu n}{\min\{n/\ln(en), \mu\}}\right) \ \leq\ \mu + 3en \cdot \max\{\mu\ln(en), n\}$$

as suggested. □

By the preceding proof, we have also shown the following corollary, which will turn out to be useful in Section 6.

Corollary 1 *Let $\mu = poly(n)$. Then the expected time until the (μ+1) EA on* LeadingOnes *creates an individual with k leading ones is bounded above by $\mu + 3ek \cdot \max\{\mu\ln(en), n\}$.*

Now we study the well-known OneMax problem. Interestingly, we require somewhat more complicated arguments than before to show upper bounds on the expected runtime. This was also the case when Jansen et al. (2005) showed corresponding upper bounds for the (1+λ) EA.

Theorem 2 *Let $\mu = poly(n)$. Then the expected runtime of the (μ+1) EA on* OneMax *is bounded above by $\mu + 5e\mu n + en\ln(en) = O(\mu n + n\log n)$.*

Proof: The idea of the proof is similar as in Theorem 1. Let L be the maximum OneMax value of the current population. In contrast to LeadingOnes, the probability of increasing L depends on L itself. Since each individual has at least $n - L$ zeros, the considered probability is bounded below by

$$\frac{i}{\mu} \cdot \frac{n-L}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \ \geq\ \frac{i(n-L)}{e\mu n}$$

if the population contains at least i individuals with value L.

We are interested in the expected time until the population contains at least $\min\{n/(n-L), \mu\}$ individuals with value L. Using the same elementary calculations as in the proof of Theorem 1, this time is bounded above by

$$\sum_{i=1}^{\lceil n/(n-L)\rceil - 1} \frac{2e\mu}{i} \ \leq\ 2e\mu\ln(en/(n-L))$$

if L does not increase before. If we sum up these expected waiting times for all values of L, we obtain (using Stirling's formula to estimate $n! \geq (n/e)^n$) a total expected waiting time of at most

$$2e\mu \sum_{L=0}^{n-1} \ln\left(\frac{en}{n-L}\right) \;=\; 2e\mu \ln\left(\frac{e^n n^n}{n!}\right) \;\leq\; 2e\mu \ln(e^{2n}) \;=\; 4e\mu n.$$

After the desired number of individuals with value L has been obtained, the expected time for increasing L is at most

$$\frac{e\mu n}{\min\{\mu, n/(n-L)\} \cdot (n-L)} \;=\; \frac{e\mu n}{\min\{\mu(n-L), n\}}.$$

Hence, the expected waiting time for all L-increases is at most

$$\sum_{L=0}^{n-1} \left(\frac{e\mu n}{\mu(n-L)} + \frac{e\mu n}{n}\right) \;\leq\; en\ln(en) + e\mu n,$$

and the total expected runtime, therefore, at most $\mu + en\ln(en) + 5e\mu n$. □

For SPC, we can only prove a (seemingly) trivial upper bound. Surprisingly, it will turn out later that this bound is at least almost tight. For the proof, the family trees come into play for the first time.

Theorem 3 *Let $\mu = poly(n)$. Then the expected runtime of the (μ+1) EA on* SPC *is bounded by $O(\mu n^3)$.*

Before the proof, we will first introduce some notions. We call the members of the initial population *initial individuals*. In the run of the (μ+1) EA on SPC, there is a first point of time where all individuals of the current population are so-called plateau points, i. e., individuals of shape $1^i 0^{n-i}$, $0 \leq i \leq n$. (Note that we allow $i = n$ whereas the point 1^n is not from the plateau of constant fitness. One could call these points "path points" since $1^i 0^{n-i}$ forms a connected path; however, we stick to the notion of plateau points to avoid confusions with paths within family trees.) Let t_{pla} denote this point of time. The random variable t_{pla} has finite expectation and can take even extreme values such as 0, i. e., all initial individuals are plateau points. With high probability, however, t_{pla} occurs later, and the optimum is not reached until (and including) time t_{pla}.

Steps occurring after time t_{pla} and producing individuals outside the plateau, i. e., not of shape $1^i 0^{n-i}$, immediately delete the created individual since its fitness must be worse than that of a plateau point. Let x_0 be an arbitrary individual from a population at some time $s \geq t_{\text{pla}}$. We now study the sequence of family trees $T_t(x_0)$, $t \geq s$, whose common root is labeled with x_0. If $s > 0$, x_0 can have predecessors in the family tree of an initial individual, i. e., $T_t(x_0)$ is a subtree of its tree. Let y be an individual in $T_t(x_0)$. If y has been deleted by time t, we call the node labeled with y and the path from the root to y dead, and alive otherwise. There is always at least one alive path in some family tree. Now consider an alive path at some time. The event that the path is currently alive imposes conditions on the mutations creating the nodes on the path. After time t_{pla}, e. g., a path cannot be alive if it ends with an individual outside the plateau. Our goal is to show that on alive paths in family trees, we can rediscover a run of the (1+1) EA on SPC. This is formalized by the following lemma.

Lemma 1 *Consider the (μ+1) EA on* SPC*. Let $t_{\text{pla}} \leq s \leq t$, let x_0 be an arbitrary individual from the population at time s, and at time t, let p be an arbitrary path* *starting at the root of $T_t(x_0)$. Let $x_0, \ldots, x_\ell$ denote the individuals along p. If p is alive and the optimum has not been reached yet, the sequence $x_0, \ldots, x_\ell$ has the same distribution as the sequence of search points of the (1+1) EA on* SPC *in the time steps $0, \ldots, \ell$ provided that the (1+1) EA starts with x_0 and that it until (and including) time ℓ only creates search points of shape $1^j 0^{n-j}$ where $0 \leq j < n$.*

Proof: For notational convenience, we without loss of generality assume $s = 0$. Let $0, t_1, \ldots, t_\ell$ denote the nodes along p. We prove the claim by induction on $i \in \{0, \ldots, \ell\}$, where the base case $i = 0$ is trivial since the (1+1) EA is assumed to start with x_0. For the induction step, let $0 \leq i < \ell$ and assume that the node t_i, labeled with x_i, has been created and is alive. The latter is necessary for p to be alive at time t. Now the lemma imposes exactly the following set of conditions on the events creating the alive path $0, t_1, \ldots, t_{i+1}$ considered in the induction step.

1. A step appends to t_i some t' labeled with $x' = 1^k 0^{n-k}$ for $k < n$.
2. If $i + 1 < \ell$, t' stays alive at least until a step chooses t' and produces t_{i+2}; otherwise, t' stays alive at least until time t.
3. $t' = t_{i+1}$.

The last two conditions follow immediately from the event considered. The first condition holds since the optimum has not yet been reached and since we consider a point of time after t_{pla}. If x' was not a plateau point, x' would be deleted immediately, contradicting the second condition. Let A be the intersection of the three conditions. If A occurs, $x' = x_{i+1}$.

Let $y_0, \ldots, y_\ell$ be the random current search points of the (1+1) EA in the steps to time ℓ; by assumption, $y_0 = x_0$. Analogously to the first condition, the lemma requires all mutations of the (1+1) EA until time ℓ to create plateau points different from 1^n. By the definition of the (1+1) EA, all these mutants are accepted. Hence, both the (μ+1) EA's mutation generating x_{i+1} and the mutation of the (1+1) EA generating y_{i+1} are conditioned on the event of creating search points of shape $1^j 0^{n-j}$ for $0 \leq j < n$.

We investigate the distribution of x_{i+1} more carefully. Here it is crucial that the (μ+1) EA chooses the individual to be deleted uniformly from the set of worst individuals. Since all individuals of shape $1^j 0^{n-j}$, $j \neq n$, have the same SPC-value, the event that such an individual is deleted is independent of the value of j. Hence, the second of the above conditions is independent of k, the number of ones in x'. It follows that the distribution of k is the same under only the first condition as under Condition A. The (1+1) EA uses the same mutation operator as the (μ+1) EA and, according to the last paragraph, it creates y_{i+1} under that condition holding for x'. If x_i and y_i take the same value $1^j 0^{n-j}$, their successors have the same distribution. Since, by induction hypothesis, x_i is identically distributed as y_i, the induction step follows. □

Having established the correspondence to the (1+1) EA on SPC, we are ready to prove Theorem 3. In Section 5.2, Lemma 1 will be applied again to derive lower bounds on the runtime.

Proof of Theorem 3: Let x_0 be an arbitrary individual from the population at time t_{pla} and let $T(x_0)$ be its family tree. We want to show that the following property (P) holds: the expected time until at least one path in $T(x_0)$ reaches length k or until all paths in $T(x_0)$ are dead is bounded above by $4e\mu k$ for all $k \geq 1$, or the optimum is reached before. This property will entail the theorem for the following reasons. We will show that $\text{E}(t_{\text{pla}}) = O(\mu n \log n)$. Moreover, we will apply Lemma 1 in the following way. We consider the event that in $T(x_0)$, a path is created where at least one node is labeled with 1^n. It follows that the probability of obtaining 1^n on a path of length $\ell + 1$ within $T(x_0)$ is bounded below by the probability that the (1+1) EA with initial search point x_0 creates 1^n within $\ell + 1$ steps. (Here, even unsuccessful steps of the (1+1) EA do not count.) By the results of Jansen and Wegener (2001a), this implies that the optimum is obtained after an expected path length of $O(n^3)$ in a family tree. By Markov's inequality, a path length of $O(n^3)$ is sufficient to reach the optimum with probability at least 1/2. Since at least one path remains alive forever, this implies according to (P) that with probability at least 1/2, the optimum is reached after $O(\mu n^3)$ expected steps. In the case of a failure, we can repeat the above argumentation with an individual from some time $t > t_{\text{pla}}$ instead of x_0. The expected number of repetitions is bounded above by 2 such that the optimum is reached after an expected number of $O(\mu n^3)$ steps.

To prove (P), we only have to consider the case that there is always at least one alive path in $T(x_0)$ until time $t_{\text{pla}} + 4e\mu k$. (Otherwise, there is nothing to show.) For any point of time $t \geq t_{\text{pla}}$, we define the following potential L_t. Let S_t be the set of those currently alive successors of x_0 that will always have an alive descendant until time $t_{\text{pla}} + 4e\mu k$. Then L_t is defined to denote the maximum depth of S_t-individuals in $T(x_0)$, i. e., the length of a longest path leading from the root to an S_t-individual. A special property of the potential L_t is that it depends on the future, but is a function mapping each current point of time to a value. Moreover, by definition, L_t cannot shrink in the run of the (μ+1) EA, and there is the following sufficient condition for increasing the L-value. An individual x' defining the current L-value is chosen for mutation, a child being a plateau point is produced, and x' is deleted from the population before its child is deleted. The probability is $1/\mu$ for the first event, $(1 - 1/n)^n \geq 1/(2e)$ for the second event since it is sufficient to produce a replica, and 1/2 for the third one since the considered individuals have the same SPC-value. Hence, the expected time to increase the L-value is bounded above by $4e\mu$, implying that an alive path reaches length k an expected number of at most $4e\mu k$ steps after time t_{pla}.

We still have to show that $\mathrm{E}(t_{\text{pla}}) = O(\mu n \log n)$. If the population contains at least one plateau point, there is the following sufficient condition for increasing the number of plateau points: the (μ+1) EA chooses a plateau point and produces a replica of it. Hence, by similar arguments as in the proof of Theorem 1, the population is filled up by plateau points after an expected number of $O(\mu \log \mu) = O(\mu \log n)$ steps. Otherwise, the SPC-value of any individual x of the population is given by $n - |\textsc{OneMax}(x)|$. Then the expected time until creating a plateau point is bounded above by the expected time until the number of ones has been minimized for at least one individual, i. e., by $O(\mu n \log n)$. □

4 A General Lower Bound Technique

For lower bounds on the runtime of (μ+1) EA, we consider the growth of the family tree $T(x_0)$ for an arbitrary initial individual x_0. As mentioned in Section 2, upper bounds on the depth of family trees can imply lower bounds on the runtime. Such upper bounds follow from the selection mechanism of the (μ+1) EA, which always selects the individual to be mutated uniformly from the current population. Recall that a path in a family tree has been defined as a monotonically increasing sequence $0, t_1, \ldots, t_\ell$ in which t_i denotes the point of time where the i-th node is present for the first time. The probability that the path $0, t_1, \ldots, t_\ell$ is created and labeled with the individuals $x_0, x_1, \ldots, x_\ell$ is upper bounded by

$$\prod_{i=0}^{\ell-1} \frac{\mathrm{mut}(x_i, x_{i+1})}{\mu} \;=\; \left(\frac{1}{\mu}\right)^{\ell} \cdot \prod_{i=0}^{\ell-1} \mathrm{mut}(x_i, x_{i+1}),$$

where $\mathrm{mut}(x, y)$ denotes the probability of the mutation operator creating $y \in \{0,1\}^n$ from $x \in \{0,1\}^n$. This holds since both the selection and the mutation operator in a step of the (μ+1) EA work independently of previous steps and since the probability of choosing node t_i at time $t_{i+1} - 1$ is either 0 (if the individual created at time $t_i - 1$ has been deleted in the meantime) or $1/\mu$. Regardless of the labeling, this implies an upper bound on the depth of family trees, which holds with overwhelming probability.

Lemma 2 *Let $D(t)$ denote the depth of a family tree of the (μ+1) EA at time t. For all $t \geq 0$, $\mathrm{Prob}(D(t) \geq 3t/\mu) = 2^{-\Omega(t/\mu)}$.*

Proof: Let $\ell := \lceil 3t/\mu \rceil$. We show the claim by considering all possible paths of length exactly ℓ. The event that at least one such path emerges equals the event $D(t) \geq 3t/\mu$.

We already know that the probability of a fixed path $0, t_1, \ldots, t_\ell$ is upper bounded by $(1/\mu)^\ell$. The number of different paths of length ℓ possible until time t is bounded above by $\binom{t}{\ell}$ since this equals the number of choices such that $1 \leq t_1 < \cdots < t_\ell \leq t$. Hence, the considered probability is bounded above by

$$\binom{t}{\ell} \left(\frac{1}{\mu}\right)^{\ell} \;\leq\; \binom{\ell\mu/3}{\ell} \left(\frac{1}{\mu}\right)^{\ell}.$$

Stirling's formula implies $\binom{n}{k} \leq (\frac{ne}{k})^k$ for $0 \leq k \leq n$. Hence, we can finally upper bound the probability by

$$\left(\frac{\ell\mu e/3}{\ell}\right)^{\ell} \left(\frac{1}{\mu}\right)^{\ell} \;=\; \left(\frac{e}{3}\right)^{\ell} \;=\; 2^{-\Omega(\ell)} \;=\; 2^{-\Omega(t/\mu)}$$

since $e/3 < 1$. □

Lemma 2 states that with overwhelming probability, a family tree of the (μ+1) EA becomes asymptotically no deeper than the total number of mutations performed in a single run of the corresponding parallel run (cf. Section 2). The tree can become wide, but a flat tree means that on any path from the root, few mutations occur. As mentioned above, intuitively, this makes the optimization of even simple functions very unlikely. The following theorem makes this precise by even covering some functions whose set of global optima has exponential size. Essentially, we assume the set of global optima to have size $2^{o(n)}$ or that all global optima have a linear Hamming distance to search points with $n/2$ ones. The runtime bounds derived are tight for some simple functions such as OneMax (if μ is not too small). To formulate the theorem, we introduce the abbreviation $|x| = x_1 + \cdots + x_n$ for $x \in \{0,1\}^n$.

Theorem 4 *Let $\mu = poly(n)$, $f\colon \{0,1\}^n \to \mathbb{R}$, $S_{\mathrm{opt}} := \mathrm{argmax}\{f(x) \mid x \in \{0,1\}^n\}$, $o_{\min} := \min\{|x| \mid x \in S_{\mathrm{opt}}\}$ and $o_{\max} := \max\{|x| \mid x \in S_{\mathrm{opt}}\}$. If at least one of the three conditions*

$$|S_{\mathrm{opt}}| = 2^{o(n)}, \quad o_{\min} = n/2 + \Omega(n), \quad o_{\max} = n/2 - \Omega(n)$$

holds, the expected runtime of the (μ+1) EA on f is at least $\Omega(\mu n)$, and the success probability in $c\mu n$ steps is $2^{-\Omega(n)}$ if the constant $c > 0$ is chosen appropriately small. If $|S_{\mathrm{opt}}| = 1$, the expected runtime on f is even $\Omega(\mu n + n \log n)$.

Proof: The lower bound $\Omega(n \log n)$ in the case $|S_{\text{opt}}| = 1$ needs only be shown for $\mu \le c' \log n$ where $c' > 0$ is an arbitrary constant. Now let $|S_{\text{opt}}| = 1$ and $\mu \le \log n/2$. We argue according to a generalization of the Coupon Collector's Theorem that has been described by Droste et al. (2002) for linear functions and the (1+1) EA. To generalize their argumentation, we estimate the probability that bit $i \in \{1, \dots, n\}$ is in all initial individuals different from the unique optimal assignment. Since $\mu \le \log n/2$, this probability is at least $n^{-1/2}$. By Chernoff bounds (see Motwani and Raghavan, 1995), at least $\sqrt{n}/2$ bits are wrong in all initial individuals with probability $1 - 2^{-\Omega(\sqrt{n})}$. Therefore, assuming that there are $\sqrt{n}/2$ such bits, the probability that at least one of these bits is never flipped within $t := \lfloor (n-1)(\ln n)/2 \rfloor$ steps is bounded below by

$$1 - \left(1 - \left(1 - \frac{1}{n}\right)^t\right)^{\sqrt{n}/2} \;\ge\; 1 - \left(1 - \frac{1}{\sqrt{n}}\right)^{\sqrt{n}/2} \;\ge\; 1 - e^{-1/2},$$

which implies that t steps are required with probability at least $1 - e^{-1/2} - 2^{-\Omega(\sqrt{n})} = \Omega(1)$. Hence, the lower bound $\Omega(n \log n)$ on the expected runtime follows.

For the lower runtime bound $\Omega(\mu n)$ under one of the three conditions, we set up a phase of length $s := \lfloor c\mu n \rfloor$ for some constant $c > 0$ and show that the (μ+1) EA requires at least s steps with probability $1 - 2^{-\Omega(n)}$ if c is small enough. The idea for the proof is as follows. Within s steps, each family tree created by the (μ+1) EA (with high probability) does not contain nodes labeled with optimal individuals. To prove this, we study the maximal Hamming distance between the individuals at the nodes in the tree and the individual at the root. Afterwards, we show that with high probability, the Hamming distance of the root individual to all optimal individuals is greater.

Let x_0 be an arbitrary initial individual and let $T_t(x_0)$ denote its random family tree at time t. According to Lemma 2, the probability that $T_s(x_0)$ reaches depth greater than $\lfloor 3cn \rfloor$ is $2^{-\Omega(n)}$. Now the aim is to prove that also with probability $2^{-\Omega(n)}$, there is a node in $T_s(x_0)$ at depth at most $3cn$ and labeled with an individual whose Hamming distance to x_0 is bounded below by $8cn$. Let us consider a sequence of $\lfloor 3cn \rfloor$ points where each point is the result of a mutation of its predecessor by means of the (μ+1) EA's mutation operator. Each bit in each point in this sequence is flipped independently with probability $1/n$. Hence, the expected Hamming distance of any two points in this sequence is at most $3cn$, and, by applying Chernoff bounds w. r. t. to the upper bound $4cn$, it is at least $8cn$ with probability bounded above by $e^{-4cn/3}$. We call a path bad if it starts at the root of $T_s(x_0)$, has length $\ell \le 3cn$ and contains a label with Hamming distance at least $8cn$ w. r. t. x_0. Together with our considerations from the beginning of this section, we obtain that the probability of creating a specific bad path $0, t_1, \dots, t_\ell$ is bounded above by $(1/\mu)^\ell \cdot e^{-4cn/3}$.

We now are able to estimate the probability of $T_s(x_0)$ containing a bad path. Since the number of paths of length at most ℓ is bounded above by $\binom{s}{\ell}$, the probability is at most

$$\begin{aligned}\sum_{\ell=1}^{3cn} \binom{s}{\ell} \left(\frac{1}{\mu}\right)^\ell \cdot e^{-4cn/3} &\le 3cn \max_{\ell=1}^{3cn} \left\{ \left(\frac{ce\mu n}{\ell}\right)^\ell \left(\frac{1}{\mu}\right)^\ell e^{-4cn/3} \right\} \\ &= 3cne^{-4cn/3} \max_{\ell=1}^{3cn} \left(\frac{cen}{\ell}\right)^\ell .\end{aligned}$$

Since the expression $((cen)/\ell)^\ell$ is maximized for $\ell = cn$, we obtain the upper bound

$$3cne^{cn}e^{-4cn/3} = 3cne^{-cn/3} = 2^{-\Omega(n)}.$$

The last estimation holds since c is assumed to be a positive constant.

We finally show that the Hamming distance of x_0 to all optimal individuals with high probability is at least $8cn$ if c is small enough. First we consider the case $S_{\text{opt}} = 2^{o(n)}$. For any $y \in S_{\text{opt}}$, it holds that x_0 has an expected Hamming distance $n/2$ to y and by Chernoff bounds, the Hamming distance is at least $n/3$ with probability $1 - 2^{-\Omega(n)}$. Since $S_{\text{opt}} = 2^{o(n)}$, the Hamming distance from x_0 to all $y \in S_{\text{opt}}$ is also at least $n/3$ with probability $1 - 2^{-\Omega(n)}$. In the case $o_{\min} = n/2 + \Omega(n)$, we apply Chernoff bounds w. r. t. to the expectation $\mathrm{E}(|x_0|) = n/2$. Hence, the Hamming distance from x_0 to any optimal individual is bounded below by $\Omega(n)$ with probability $1 - 2^{-\Omega(n)}$. The case $o_{\max} = n/2 - \Omega(n)$ is symmetrical. Therefore, in all three cases, choosing c small enough results in $8cn$ being smaller than the lower bound $\Omega(n)$. All in all, for a fixed initial individual x_0, $T_s(x_0)$ with probability $1 - 2^{-\Omega(n)}$ does not contain nodes labeled with optimal individuals. Since $\mu = \mathit{poly}(n)$, this also holds for all initial individuals together. □

Theorem 4 covers the wide range of unimodal functions. For certain subclasses of the class of unimodal functions (including linear functions), the (1+1) EA's expected runtime is at most $O(n \log n)$. With respect to this class, Theorem 4 states that the (μ+1) EA is at most by a factor of $O(\log n)$ more efficient than the corresponding parallel run.

For more difficult functions (which means that the (1+1) EA has a larger expected optimization time than $O(n \log n)$), the proof concept of Theorem 4 can be carried over to show larger lower bounds also for the (μ+1) EA. However, we have to derive better lower bounds on the depth that family trees need for optimization. Thus, more structure of the function f and of the encountered search points comes into play.

5 Special Lower Bounds

In this section, we show specialized lower bounds on the runtime of the (μ+1) EA. As in the proof of Theorem 4, the key idea here is to bound the depth of family trees.

5.1 Lower Bound for LeadingOnes

Theorem 5 *Let $\mu = poly(n)$. Then the expected runtime of the (μ+1) EA on* LeadingOnes *is $\Omega(\mu n \log n + n^2)$. Moreover, the success probability within some $c\mu n \log n$ steps, $c > 0$, is $2^{-\Omega(n)}$.*

Proof: The bound $\Omega(n^2)$ follows by means of the analysis of the (1+1) EA on LeadingOnes described by Droste et al. (2002, Theorem 17). This analysis can directly be applied to the potential L from the proof of Theorem 1, i. e., the maximum LeadingOnes value of the (μ+1) EA's current population. The probability of increasing L is at most $1/n$ and at least the rightmost $n - L - 1$ bits are uniformly distributed in each individual of the population. This is again a consequence of the (μ+1) EA's deletion operator. Therefore, the estimations for the number of so-called free-riders carry over to the potential L.

The basic idea for the bound $\Omega(\mu n \log n)$ is the same as in Theorem 4. We will show that for some small enough constant $c > 0$, the (μ+1) EA requires at least $s := \lfloor c\mu n \log n \rfloor$ steps with probability $1 - 2^{-\Omega(n)}$. To this end, we consider the family tree $T_s(x_0)$ obtained after s steps for an arbitrary initial individual x_0. Using Lemma 2 and $\mu = \mathit{poly}(n)$, it only remains to show that with probability $1 - 2^{-\Omega(n)}$, no node at depth at most $3cn \log n$ in $T_s(x_0)$ is optimal.

For notational convenience, let $f :=$ LeadingOnes. Since all initial individuals are uniform over $\{0,1\}^n$, $f(x_0) \leq n/2$ with probability $1 - 2^{-\Omega(n)}$. As mentioned above (shown by Droste et al., 2002), it holds that in each individual of any population, the bits after the leftmost zero of the individual are uniformly distributed. This implies that with probability $1 - 2^{-\Omega(n)}$, along any fixed path in $T_s(x_0)$, it is not enough to increase the f-value at most $n/6$ times to obtain a total increase in f-value of at least $n/2$. We call a path bad if the f-value increases at least $n/6$ times on the path. Hence, we have to estimate the probability that a specific bad path $0, t_1, \dots, t_\ell$ is created. The probability of increasing the f-value is bounded above by $1/n$ since the leftmost zero has to flip. Hence, using the estimation from the beginning of Section 4, the probability of creating such a path is bounded above by

$$\left(\frac{1}{\mu}\right)^{\ell} \left(\frac{1}{n}\right)^{n/6}.$$

Finally, we estimate the probability of $T_s(x_0)$ containing a bad path by the same method as in the proof of Theorem 4. Since the number of paths of length at most ℓ is bounded above by $\binom{s}{\ell}$, the probability is at most

$$\sum_{\ell=1}^{3cn\log n} \binom{s}{\ell} \left(\frac{1}{\mu}\right)^{\ell} \left(\frac{1}{n}\right)^{n/6} \leq 3cn(\log n) \left(\frac{1}{n}\right)^{n/6} \max_{\ell=1}^{3cn\log n} \left(\frac{cen\log n}{\ell}\right)^{\ell}.$$

Since the expression $((cen\log n)/\ell)^\ell$ is maximized for $\ell = cn\log n$, we obtain the upper bound

$$3cn(\log n) \cdot e^{cn\log n} \cdot e^{((\ln 2)/6)n\log n} = 3cn(\log n) \cdot e^{(c-(\ln 2)/6)n\log n}.$$

We choose c small enough such that the last expression is $2^{-\Omega(n\log n)}$.

Altogether, $T_s(x_0)$ does not contain an optimal node with probability $1-2^{-\Omega(n)}$. Finally, since $\mu = \mathit{poly}(n)$, this also holds for all initial individuals together. □

We have seen that the waiting time $\Theta(n)$ required by the (1+1) EA for an increase of the LeadingOnes-value translates into a factor $\Theta(\mu\log n)$ within the runtime of the (μ+1) EA provided that $\mu = \Omega(n/\log n)$. A similar correspondence seems to hold for a generalization of LeadingOnes called LOB_b (*leading ones blocks*) studied by Jansen and Wiegand (2004). For constant $b \in \mathbb{N}$, let $\text{LOB}_b(x) := n\sum_{i=1}^{n/b}\sum_{j=1}^{bi} x_j - \text{OneMax}(x)$. The OneMax part of $\text{LOB}_b(x)$ implies that after a short time and with high probability, each current individual of the (1+1) EA and the (μ+1) EA is of shape $1^{bi}0^{n-bi}$ for some i. To create a search point with higher LOB_b-value, a mutation of probability $\Theta(n^{-b})$ is necessary and sufficient. It seems that the arguments from the proofs of Theorem 1 and Theorem 5 can be generalized in a straightforward manner. We conjecture that the expected runtime of the (μ+1) EA on LOB_b equals $\Theta(n^b + \mu n\log n)$.

5.2 Lower Bound for SPC

It is interesting to study in how far the exploration of the plateau of constant fitness posed by the function SPC benefits from the population of the (μ+1) EA. Therefore, in this section, we will derive a lower bound on the expected runtime of the (μ+1) EA on SPC. In fact, as in the upper bound of Theorem 3, the main idea will be to analyze situations where the whole population consists of individuals from the plateau of constant fitness. Unfortunately, the following lower runtime bound does not match the upper bound from the above-mentioned theorem. Note, however, that the lower bound is $\Omega(n^3)$, implying that the (μ+1) EA is asymptotically never more efficient than the (1+1) EA on SPC.

Theorem 6 *Let $\mu = poly(n)$. Then the expected runtime of the (μ+1) EA on* SPC *is lower bounded by $\Omega(\mu n^3/\log\mu)$. Moreover, the success probability within some $c\mu n^3/\log\mu$ steps, $c>0$, is $2^{-\Omega(\log^2\mu)}$.*

The main proof idea is the same as in the proof of Theorem 5. By Lemma 2, the probability of a family tree's depth reaching at least $3cn^3/\log\mu$ within a phase of $\lfloor c\mu n^3/\log\mu\rfloor$ steps is $2^{-\Omega(n^3/\log\mu)}$. However, we will prove that a depth of at least $3cn^3/\log\mu$ is necessary for optimization with probability $1-2^{-\Omega(\log^2\mu)}$ in every family tree.

To this end, we fix an arbitrary initial individual x_0 and consider the following event. In the family tree $T(x_0)$, starting from the root, a path $p=(0,t_1,\ldots,t_\ell)$ is created that is alive at time t_ℓ. Some nodes of p might be labeled with the optimal search point 1^n; however, we will show that this is unlikely for well-chosen values $\ell=\Theta(n^3/\log\mu)$. Note that in contrast to the proofs of Theorem 4 and Theorem 5, we restrict our attention to alive paths in family trees. There are two reasons for this. First, we will be able to reapply Lemma 1. Second, the number of alive paths in a family tree always is at most μ, i. e., is bounded by a polynomial rather than exponential value. For technical reasons, we are interested in nodes of p that are far away from both 1^n and 0^n and study only a subpath $\tilde{p}$ of p with the following property: all search points on $\tilde{p}$ are of shape 1^i0^{n-i} with $n/4\le i\le 3n/4$. Of course, there is a positive probability that no such subpath $\tilde{p}$ exists. However, our goal is to show that with high probability, such a $\tilde{p}$ exists and has length at least $\Omega(n^3/\log\mu)$. The latter is proven in two steps. First, we show that with high probability, the labels of the nodes change $\Omega(n^2/\log\mu)$ times along $\tilde{p}$. Second, the distance of two nodes with different labels is $\Omega(n)$ with probability $\Omega(1)$.

To be able to investigate $\tilde{p}$, we concentrate on points of time where the population contains plateau points, i. e., individuals of shape 1^i0^{n-i}, $0\le i\le n$. Let t_{pl1} denote the first point of time such that at least one plateau point exists. Moreover, let t_{pla} still denote the first point of time such that μ plateau points are in the population. Recall that the optimum may have been reached before. We prove that this is very unlikely.

Lemma 3 *With probability $1-2^{-\Omega(\sqrt{n})}$, all populations until (and including) time t_{pla} contain only individuals with at most $3n/5$ ones.*

Proof: First of all, we prove that with probability $1-2^{-\Omega(n)}$, all individuals until time t_{pl1} have at most $4n/7$ ones. To show this, the following arguments are used. Since the initial individuals are drawn uniformly at random and $\mu=poly(n)$, the probability that there exists an initial individual with more than $5n/9$ ones is bounded above by $2^{-\Omega(n)}$ according to Chernoff bounds. Before time t_{pl1}, the goal function SPC(x) equals $n-\sum_{i=1}^{n}x_i$, also called ZeroMax. Hence, the selection mechanism of the (μ+1) EA implies that the minimum number of zeros in the population cannot decrease until time $t_{\mathrm{pl1}}-1$. By the same arguments as in the proof of Theorem 3, we have $\mathrm{E}(t_{\mathrm{pl1}})=O(\mu n\log n)$ and, by Markov's inequality and the fact that independent phases may be repeated, $t_{\mathrm{pl1}}=O(\mu n^2\log n)$ with probability $1-2^{-\Omega(n)}$. Since flipping a linear number of bits in at least one out of at most $O(\mu n^2\log n)$ steps of the (μ+1) EA has probability $2^{-\Omega(n\log n)}$ due to $\mu=poly(n)$, the plateau point created at time t_{pl1} has at most $4n/7$ ones, and the other individuals have at most $5n/9$ ones with probability $1-2^{-\Omega(n)}$. In the following, we assume this event to have happened.

By similar arguments as in the proof of Theorem 1, the expected number of steps from time t_{pl1} until time t_{pla} is $O(\mu\log\mu)$. Moreover, this time is bounded by $O(\mu\sqrt{n}\log\mu)$ with probability $1-2^{-\Omega(\sqrt{n})}$ since we can apply Markov's inequality and repeat independent phases. According to Lemma 2, no family tree becomes deeper than $O(\sqrt{n}\log\mu)$ in this number of steps with probability $1-2^{-\Omega(\sqrt{n})}$, therefore we can estimate the number of ones in the individuals (which are plateau points) in the population at time t_{pla} using the arguments from the proof of Theorem 4. Replacing the value $\lfloor c\mu n\rfloor$ used there for s with some value $O(\mu\sqrt{n}\log\mu)$ and considering the family trees for the population at time t_{pl1}, we obtain that no individual at time t_{pla} has more than $3n/5$ ones with probability $1-2^{-\Omega(\sqrt{n})}$. $\square$

Now we can study $\tilde{p}$, a well-structured subpath of the above-mentioned path p. In the following lemma, we may assume p to be as long as needed for the events considered there. Only if p is long enough, the events can occur. To cope with the case that μ does not grow with n, we have to be careful when applying O-notation with respect to μ.

Lemma 4 *Consider an arbitrary alive path $p=(0,t_1,\ldots,t_\ell)$ in $T(x_0)$ and suppose that the optimum is not reached by time t_ℓ. With probability at least $\max\{1-2^{-\Omega(\log^2\mu)},\Omega(1)\}$, there is a subpath $\tilde{p}$ of p such that all nodes of $\tilde{p}$ are created at time t_{pla} or later, $\tilde{p}$ contains only labels 1^i0^{n-i} for $n/4\le i\le 3n/4$, and the labels along $\tilde{p}$ change $\Omega(n^2/\log\mu)$ times.*

Proof: Let v be the label of the last node of p that is created until (and including) time t_{pla}. According to Lemma 3, it contains at most $3n/5$ ones with probability $1-2^{-\Omega(\sqrt{n})}$. From time t_{pla} on, an individual that is not a plateau point is deleted immediately after its creation. We identify nodes in family trees with their labels. Hence, v and all its successors on p are plateau points since p could not be alive at time t_ℓ otherwise. Since flipping a linear number of bits in one step out of polynomially many steps has probability $2^{-\Omega(n\log n)}$, it follows that with probability $1-2^{-\Omega(n)}$, there is a successor v^* of v on p that has at least $n/3$ and still at most $2n/3$ ones. Hence, with probability $1-2^{-\Omega(\sqrt{n})}$, we have identified a start point v^* for $\tilde{p}$ that is created at time t_{pla} or later. The path $\tilde{p}$ ends at the first node that is not of shape 1^i0^{n-i} with $n/4\le i\le 3n/4$. Now we have to show that with high probability, at least $\Omega(n^2/\log\mu)$ nodes on $\tilde{p}$ are different plateau points than their parents.

The start node v^* is a search point 1^i0^{n-i}, $n/3\le i\le 2n/3$, with Hamming distance $\Omega(n)$ to the optimal search point 1^n. We want to show that the random walk describing the different plateau points on $\tilde{p}$ is similar to a fair random walk on the line of length $\Theta(n)$, i. e., where the probability of moving to the left on the line equals the probability of moving to the right (Feller, 1968). This is also a key idea in the proof of the upper bound for the (1+1) EA on SPC (Jansen and Wegener, 2001a). For the described random walk, it is well known that distance $\Omega(n)$ to its start point is not overcome within $\Theta(n^2/\log\mu)$ steps with probability $1-2^{-\Omega(\log^2\mu)}$.

Since we study paths that are alive at at least one point of time and the optimum is assumed to be reached after time t_ℓ, we can apply Lemma 1 on the sequence of family trees rooted at v^* and argue about v^*'s successors. We consider the first at most $s:=\lfloor n^2/(50(\log\mu+1))\rfloor$ nodes on p that are different search points than their parents. Each considered search point is the result of a mutation of some 1^i0^{n-i} into some 1^j0^{n-j}, $j\neq i$. We also consider these at most s mutations. In contrast to the above-mentioned random walk, it may be the case that $|j-i|>1$, i. e., steps of size larger than 1 occur. Therefore, we investigate the event that the (μ+1) EA flips the i-th bit of a search point 1^i0^{n-i}. Then it is most likely to obtain $1^{i-1}0^{n-i+1}$ by the mutation. To create a plateau point $1^{i-1-k}0^{n-i+1+k}$ for some $k\ge 1$, the bits $i-k,\ldots,i-1$, so-called additionally flipping bits, have to be flipped simultaneously. The corresponding probability is bounded above by $1/n^k$ for the (μ+1) EA's mutation operator. An analogous statement holds for the case that the $(i+1)$-st bit is flipped. The expected number of additionally flipping bits in s steps is, therefore, bounded above by $\sum_{i=1}^{n} s/n^i \leq 2n/(50(\log\mu+1))$, and the probability of $n/24$ additionally flipping bits is $2^{-\Omega(n)}$ according to Chernoff bounds. This will be used in the next paragraph.

In the following, we assume no mutation step within s steps to flip at least $n/4$ bits, which holds with probability $1-2^{-\Omega(n\log n)}$. Now, given the event that some 1^i0^{n-i}, where $n/4 \leq i \leq 3n/4$, is mutated to a different plateau point, the probability of flipping bit i and bit $i+1$ is the same. Hence, the probability of obtaining a point with more than i ones is the same as the probability of obtaining less ones, and we have established the correspondence to the fair random walk. In s steps, we expect $s/2$ steps increasing and $s/2$ steps decreasing the number of ones provided no search point with less than $n/4$ or more than $3n/4$ ones is encountered in these steps. We call the latter event a failure. According to Chernoff bounds, the probability that there is a surplus of $n/25$ increasing or decreasing steps is bounded above both by $2^{-\Omega(\log^2\mu)}$ and by $1-\Omega(1)$ if no failure occurs. Recall that v^* has at least $n/3$ and at most $2n/3$ ones. Using the above analysis of additionally flipping bits, we obtain that the probability of creating 1^i0^{n-i}, where $i < n/4$ or $i > 3n/4$, within the s mutations is bounded above by $\max\{2^{-\Omega(\log^2\mu)}, 1-\Omega(1)\}$, which justifies the assumption that no failure occurs. □

Now we only need to estimate the distance of two nodes on $\tilde{p}$ that are different points from their parents. We call such nodes *action nodes.* To bound the distance of two action nodes, we again make use of Lemma 1.

Lemma 5 *With probability* $\Omega(1)$*, the distance of any two consecutive action nodes on* $\tilde{p}$ *is* $\Omega(n)$*. This holds independently of previous action nodes on* $\tilde{p}$*.*

Proof: First of all, we consider the superpath p, which, by our assumptions, is alive before reaching the optimum. Let v be an arbitrary node on p created at time t_{pla} or later and let v' be its successor on the considered part of p. The path from v to v' is alive at least one point of time such we can apply Lemma 1 with $x_0 := w$. Again, we identify nodes with their labels. Hence, we know that v' is the result of a mutation under the condition that its predecessor, a plateau point, is changed again into a plateau point different from 1^n.

The unconditional probability of changing a plateau point 1^i0^{n-i} into a different plateau point is bounded above by $2/n$ since the i-th or $(i+1)$-st bit has to flip. The probability of changing a plateau point again into a plateau point is bounded below by $(1-1/n)^n \geq 1/(2e)$ since a replica is sufficient. Hence, the probability that v' differs from v is bounded above by $4e/n$. Consequently, the first $n/(8e)$ successors of v on p are the same plateau point as v with probability at least $1/2$.

Actually, rather than p, we consider nodes on the subpath $\tilde{p}$. By definition, these are created at time t_{pla} or later as desired. However, also by definition, only nodes of shape 1^i0^{n-i}, where $n/4 \leq i \leq 3n/4$, are created on $\tilde{p}$, and the search points along $\tilde{p}$ change $\Omega(n^2/\log\mu)$ times. The first condition only increases the probability of creating a replica, and the second one does not influence the distance of action nodes on $\tilde{p}$. Finally, the above statement on the successors of v holds also if v is a action node since the definition of an action node only implies a condition on its predecessor. Therefore, the distance of an action node and the next action node on $\tilde{p}$ is at least $n/(8e)$ with probability at least $1/2$. This estimation holds independently of previous action nodes since the mutation operator of the (μ+1) EA works independently of earlier steps and creating a replica does not violate the properties of $\tilde{p}$. □

Finally, we can put the preceding three lemmas together.

Proof of Theorem 6: Assuming that the event of Lemma 3 holds, we only introduce an error term of $2^{-\Omega(\sqrt{n})}$. Assuming that the optimum has not yet been reached, we may apply the Lemmas 4 and 5. We start with the trivial case $\mu = O(1)$. Then by Lemma 4, there exists a constant $c > 0$ such that with probability $\Omega(1)$, at least cn^2 mutations changing plateau points into different plateau points are necessary to reach the optimum. Each such mutation has probability $O(1/n)$, and the theorem follows by Chernoff bounds.

Now let $\mu = \omega(1)$. Then we combine Lemma 4 and Lemma 5. Let p still be an arbitrary alive path in a fixed family tree $T(x_0)$. Assume that the labels along an appropriate subpath $\tilde{p}$ according to Lemma 4 change $\Omega(n^2/\log\mu)$ times. Then the expected number of consecutive action nodes at distance $\Omega(n)$ is $\Omega(n^2/\log\mu)$. Since Lemma 5 holds independently of previous action nodes, we can apply Chernoff bounds. We obtain that with probability $1 - 2^{-\Omega(n^2/\log\mu)}$, the number of consecutive action nodes at distance $\Omega(n)$ is $\Omega(n^2/\log\mu)$, and therefore, $\tilde{p}$'s length is at least $\Omega(n^3/\log\mu)$. By the prerequisites of Lemma 4, we obtain that p's length is $\Omega(n^3/\log\mu)$ with probability $1 - 2^{-\Omega(\log^2\mu)}$. The order of magnitude of the failure probability is governed by the one from Lemma 4.

By Lemma 2, the probability of a family tree's depth reaching at least $u := 3cn^3/\log\mu$ within a phase of $\lfloor c\mu n^3/\log\mu \rfloor$ steps, where $c > 0$ is some constant, is $2^{-\Omega(n^3/\log\mu)} = 2^{-\Omega(\log^2\mu)}$ since $\mu = \mathit{poly}(n)$. Trivially, at each point of time, the number of alive paths in all family trees equals μ. We apply our lower bound on the length of alive paths needed to reach the optimum. Hence, the probability that at least one path of length less than $\ell := c'n^3/\log\mu$, $c' > 0$ some constant, contains an optimal node is still at most $\mu 2^{-\Omega(\log^2\mu)} = 2^{-\Omega(\log^2\mu)}$. If we choose c small enough and n large enough, u is smaller than ℓ. □

The lower and upper bounds for the (μ+1) EA on SPC derived so far are still apart by a factor of $\Theta(1/\log\mu)$. We should discuss what an asymptotically tight bound could look like. To this end, it is helpful to study μ parallel runs of the (1+1) EA on SPC. It follows by a modification of the Theorem of DeMoivre/Laplace (Raab and Steger, 1998) that the probability of at least $n/2+\alpha(n)\sqrt{n}$ successes in n independent Bernoulli trials with success probability 1/2 each is still bounded below by $1 - 2^{-\Omega(\alpha(n)^2\log\alpha(n))}$ if $\alpha(n) = \omega(1)$ and $\alpha(n) = o(n^{1/3})$. This means that the success probability of a single run of the (1+1) EA within some $cn^3\sqrt{\log\log\mu/\log\mu}$ steps, where $c > 0$ a constant, is still $\Omega(1/\mu)$, and the expected runtime of the parallel runs $O(\mu n^3\sqrt{\log\log\mu/\log\mu})$. This might suggest that an asymptotically tight bound for the (μ+1) EA is also of order $\Theta(\mu n^3/\mathit{polylog}(\mu))$, supporting the bound of Theorem 6. To show this, however, one would have to prove that in the family trees of the (μ+1) EA, there are $\Omega(\mu)$ almost independent paths of length $\Omega(n^3/\mathit{polylog}(\mu))$. This seems to require an analysis of the lifetime of paths in family trees. We leave the open problem of analyzing the runtime of the (μ+1) EA on SPC even more exactly. In any case, the (μ+1) EA is, up to logarithmic factors w. r. t. μ, as inefficient as the corresponding parallel run here.

6 An Example where $\mu > 1$ is Essential

In the previous sections, we have shown, for some example functions, that the (μ+1) EA can only be slightly more efficient than its corresponding parallel run. Moreover, it is never more efficient than a single run of the (1+1) EA on all of these functions, and it becomes less and less efficient for increasing values of μ.

However, it is common belief that the population of the (μ+1) EA helps to better explore the search space. We can make this precise in some respect for an example function which is similar to that considered by Witt (2003) for an elitist steady-state GA with fitness-proportional selection. Assume that in a subspace $\{0,1\}^\ell$ of the search space, an optimal setting for LeadingOnes is sought, while in the subspace $\{0,1\}^{n-\ell}$, an optimal setting for OneMax is sought. If ℓ is not too small, the (1+1) EA normally finds the optimal setting for OneMax faster than for LeadingOnes. On the other hand, by the results from the Sections 3–5, the expected runtime of the (μ+1) EA is $O(\mu\ell\log n+\ell n)$ for the LeadingOnes part and $\Omega(\mu(n-\ell))$ for the OneMax part. For $\ell=\sqrt{n}$ and $\mu=\Omega(n)$, e. g., this means that the (μ+1) EA is faster on the LeadingOnes part. This can be explained since now the subspace of the OneMax part is better explored but less exploited than the other subspace. If the function leads to an isolated local optimum when the OneMax part is optimized first, the (1+1) EA is expected to behave inefficiently. Moreover, if a global optimum is reached when the LeadingOnes part is optimized first, we expect the (μ+1) EA to be efficient.

The following function has been defined with this idea in mind. Let search points $x\in\{0,1\}^n$ be divided into a prefix $(x_1,\dots,x_m)$ of length m and a suffix $(x_{m+1},\dots,x_n)$ of length ℓ. Let $\ell:=\lceil n^{1/2}/45\rceil$, i. e., $m=n-o(n)$. Without loss of generality, m is divisible by 3. For any search point $x\in\{0,1\}^n$, we define $\mathrm{PO}(x):=x_1+\dots+x_m$; PO stands for *prefix ones.* Let $\mathrm{LSO}(x):=\sum_{i=0}^{\ell-1}\prod_{j=0}^{i}x_{m+1+j}$; LSO stands for *leading suffix ones.* Finally, let $b:=2m/3+\lceil n^{1/2}/(55\log^2 n)\rceil$. Then let

$$f(x)\ :=\ \begin{cases}\mathrm{PO}(x)+n^2\cdot\mathrm{LSO}(x) & \text{if } \mathrm{PO}(x)\le 2m/3,\\ n^2\ell-n\cdot|\mathrm{PO}(x)-b|+\mathrm{LSO}(x)-\ell-1 & \text{otherwise.}\end{cases}$$

The structure of f needs to be explained. The first case occurs if x has few POs. Then the f-value is strongly influenced by the number of LSOs. The optimum f-value of $n^2\ell+2m/3$ is achieved if $\mathrm{LSO}(x)=\ell$ and $\mathrm{PO}(x)=2m/3$. However, if $\mathrm{PO}(x)\le 2m/3$ and $\mathrm{LSO}(x)<\ell$, the f-value is at most $n^2(\ell-1)+2m/3$, which is less than $n^2\ell-nb-\ell-1$, a lower bound on the value in the second case, i. e., if $\mathrm{PO}(x)>2m/3$. If $\mathrm{PO}(x)=b$ and $\mathrm{LSO}(x)=\ell$, we have a locally optimal search point with f-value $n^2\ell-1$. The set of better search points consists of all x' where $\mathrm{LSO}(x')=\ell$ and $\mathrm{PO}(x')\le 2m/3$. Hence, the Hamming distance from x to a better point is $b-2m/3=\Omega(n^{1/2}/\log^2 n)$, which suggests that escaping from a local optimum is hard. In fact, the (1+1) EA is likely to get stuck here, and even multistarts do not help. For the following theorem, recall that there is a slight difference between the (1+1) EA and the (μ+1) EA with $\mu=1$.

Theorem 7 *With probability at least $1-2^{-\Omega(n^{1/2}/\log n)}$, on the function f, both the runtime of the (1+1) EA and of the (μ+1) EA with $\mu=1$ is $2^{\Omega(n^{1/2}/\log n)}$.*

Proof: The proof idea is not too complicated. We show that the evolutionary algorithm is likely to create a search point with b POs at some time. Afterwards, it has to overcome a Hamming distance of at least $b-2m/3$ in one step in order to create an optimal search point. This takes a long time with high probability. In the following, we conduct the proof only for the (1+1) EA. One can easily verify that all considerations remain valid also for the (μ+1) EA with $\mu=1$ since it does not matter whether equal-valued search points are rejected or not.

The initial search point of the (1+1) EA has at most $\ell/2$ LSOs with probability $1-2^{-\ell/2}=1-2^{-\Omega(n^{1/2}/\log n)}$. We assume we start with such a point. In the following, we divide the run into consecutive phases of length $\lceil 4e(n-1)\rceil$ and consider the event of the (1+1) EA increasing the number of POs to b in one of the phases. We have to be careful since the number of POs of the current search point of the (1+1) EA can decrease if a so-called bad step that increases the number of LSOs occurs. Assume for the moment that a so-called good phase of length $\lceil 4e(n-1)\rceil$, i. e., without bad steps, occurs and that the LSO-value is still less than ℓ. Then at most b decreases of the distance $|\mathrm{PO}(x)-b|$ are sufficient to create a search point with b POs. The probability of a decrease of this absolute value is bounded below by $((m-b)/n)(1-1/n)^{n-1} \geq 1/(4e)$ (for n large enough) since there are always at least $m-b \geq n/4$ bits available whose flipping decreases $|\mathrm{PO}(x)-b|$ and increases the f-value. Hence, according to Chernoff bounds, the phase contains at least b decreases with probability at least $1-2^{-\Omega(n)}$.

We still have to analyze the effects of bad steps. The probability of a step being bad is bounded by $1/n$ since it is necessary that the leftmost suffix zero flips. Hence, the probability of a good phase is bounded below by $(1-1/n)^{\lceil 4e(n-1)\rceil} \geq e^{-4e-1} = \Omega(1)$. This implies that the good phase is preceded by at most $\ell/(32e)$ bad phases with probability at most $(1-e^{-4e-1})^{\ell/(32e)} = 2^{-\Omega(\ell)}$. The total number of steps in this number of phases is bounded above by $\ell n/8$. Since $\ell n/8$ mutations flip a total number of at most $\ell/6$ bits with probability $1-2^{-\Omega(\ell)}$, we can work under the assumption of observing at most $\ell/(32e)$ bad phases and a total number of at most $\ell/6$ bad steps before the good phase. This assumption does not increase the probability of a bad phase.

It can still happen that $\ell/6$ bad steps increase the number of LSOs to ℓ since an LSO-increasing step can contain many free-riders, i. e., 1-bits after the leftmost suffix zero. However, as already mentioned in the above analyses w. r. t. LeadingOnes, Droste et al. (2002) have shown that the entries right of the leftmost suffix zero always form a uniformly distributed random search point. This implies that the probability of at least $\ell/3$ free-riders is $2^{-\Omega(\ell)}$. Together with the assumption of having at most $\ell/2$ LSOs in the beginning, we have proved that the number of LSOs does not reach $\ell/2+\ell/3+\ell/6=\ell$ until the good phase. Up to here, we have proved that with probability $1-2^{-\Omega(\ell)}$, a search point with b POs is created before the optimum is reached.

The definition of f implies that no search point with more than $2m/3$ (except b) POs is accepted when a point with b POs has been reached. Hence, it is necessary that at least $2m/3-b \geq n^{1/2}/(55\log^2 n)$ bits flip in one step of the (1+1) EA. The probability of this event is bounded above by

$$\binom{n}{2m/3-b} \cdot \left(\frac{1}{n}\right)^{2m/3-b} \leq \frac{1}{(n^{1/2}/(55\log^2 n))!} = 2^{-\Omega(n^{1/2}/\log n)}$$

using Stirling's formula. Therefore, the probability of observing this event within some $2^{\varepsilon n^{1/2}/\log n}$ steps is $2^{-\Omega(n^{1/2}/\log n)}$ if $\varepsilon > 0$ is chosen small enough. Altogether, we have proved that the runtime is at least $2^{\Omega(n^{1/2}/\log n)}$ with probability $1-2^{-\Omega(n^{1/2}/\log n)}$. □

On the other hand, if the population size is a small polynomial, f is optimized in expected polynomial time. This expected time is $O(n^2/\log n)$ if μ is chosen appropriately.

Theorem 8 *Let $\mu \geq n/\ln(en)$ and $\mu = poly(n)$. With a probability of at least $1-2^{-\Omega(n^{1/2}/\log n)}$, the ($\mu$+1) EA optimizes f within $O(\mu n)$ steps. Its expected runtime is bounded by $O(\mu n)$.*

Proof: For the first claim, we use the same proof idea as for Theorem 1. Assume for the moment that all individuals have at most $2m/3$ POs at any time, which means that the first case of the definition of f always holds. Then we can use the potential L, defined as the maximum number of LSOs in the current population. Due to the properties of f in its first case, the potential cannot decrease, and no individual with L LSOs can be deleted if the current population still contains individuals with less LSOs. Hence, we are in the same situation as if optimizing LeadingOnes such that the expected time to creating an individual with ℓ LSOs (hereinafter called a success) is bounded by $\mu+3e\mu\ell\ln(en)$ according to Corollary 1. (Here, $\mu \geq n/\ln(en)$ is essential.) By Markov's inequality, the time is bounded by $t^* := \lfloor 7e\mu\ell\ln(en)\rfloor$ with probability at least 1/2. As the analysis from the proof of Theorem 1 holds for arbitrary initial populations, we can repeat independent phases of length t^*. This means for any constant $c > 0$ that the probability of no success within a number of $\lfloor cn^{1/2}/\ln(en)\rfloor$ phases of length t^*, i. e., within at most $7ce\mu\ell n^{1/2} = O(\mu n)$ steps, is at most $2^{-cn^{1/2}/\log n}$ (if n is large enough).

After the first success, at least one individual with ℓ LSOs always remains in the current population. It is sufficient to reach the optimum by increasing the number of POs of such an individual to $2m/3$. By the standard arguments used throughout the paper, we obtain that the expected value of the time for this to happen is bounded by $O(\mu n)$. Moreover, since the probability of increasing the number of POs is always a constant in the events considered, we can apply Chernoff bounds. Hence, the optimization time is bounded above by $O(\mu n)$ with a probability of $1 - 2^{-\Omega(n^{1/2}/\log n)}$.

We estimate the probability of the event that no individual ever has more than $2m/3$ POs within $s := \lfloor 7ce\mu\ell n^{1/2}\rfloor$ steps, i. e., until the time where a success occurs with high probability. We reuse the argumentation described in the proof of Theorem 4. Replacing the value $c\mu n$ used there for s with our actual s, we obtain that within s steps and for any fixed family tree, no individual in the tree has a Hamming distance of at least $56ce\ell n^{1/2}$ to the initial individual at the root. With probability $1 - 2^{-\Omega(n)}$, no initial individual has at least $7m/12$ POs. If we choose, e. g., $c \leq 1/51$ and n large enough, we have

$$\lfloor 56ce\ell n^{1/2}\rfloor \;\leq\; 153c\left(\frac{n^{1/2}}{45}+1\right)n^{1/2} \;\leq\; \left(\frac{1}{15}+o(1)\right)n \;<\; \frac{m}{12},$$

i. e., the probability of a Hamming distance of at least $m/12$ to the initial individual has been bounded by $2^{-\Omega(n)}$. Altogether, since $\mu = \mathit{poly}(n)$, this bounds the probability of having some individual with more than $2m/3$ POs within s steps by $2^{-\Omega(n)}$. The probability of obtaining more than $2m/3$ POs before ℓ LSOs has altogether been bounded by $2^{-n^{1/2}(c-o(1))/\log n}$. Since the sum of all considered failure probabilities is $2^{-\Omega(n^{1/2}/\log n)}$, the first statement of the theorem has been proven.

For the statement on the expected runtime, we have to consider the case that an individual has more than $2m/3$ POs at some time. By similar arguments as in the proof of Theorem 7, it is easy to see that such a situation is reached after an expected number of $O(\mu n)$ steps if the optimum is not reached before. Similarly as in the proof of Theorem 7, we can show that an individual with b POs is created after another $O(\mu n)$ expected steps. It follows by standard arguments that afterwards, $O(\mu\ell)$ expected steps suffice to create a locally optimal individual, and another $O(\mu\log\mu) = O(\mu\log n)$ expected steps are enough to create a population full of locally optimal individuals. Here, we pessimistically assume the optimum is not reached before.

The Hamming distance of a locally and a globally optimal individual is $h := b - 2m/3 \leq n^{1/2}/(55\log^2 n)+1$. Hence, it is sufficient to flip at most $n^{1/2}/(55\log^2 n)+1$ selected bits in order to create a globally optimal search point from a locally optimal one. The corresponding probability is bounded below by

$$\left(\frac{1}{n}\right)^h\left(1-\frac{1}{n}\right)^{n-h} \;=\; 2^{(-\log n)(n^{1/2})/(55\log^2 n)-O(\log n)} \;=\; 2^{-n^{1/2}/(55\log n)-O(\log n)}.$$

The expected waiting time for such a mutation, therefore, is upper bounded by $2^{n^{1/2}/(55\log n)+O(\log n)}$. We have calculated above that the probability that the runtime is not $O(\mu n)$ is bounded above by $2^{-n^{1/2}(1-o(1))/(51\log n)}$ for our choice of c. The product of this upper bound on the failure probability and the upper bound on the expected time in the failure case is $o(1)$. The total expected runtime of the (μ+1) EA on f is, therefore, at most $O(\mu n) + o(1) = O(\mu n)$. □

7 Conclusions

We have presented a first analysis of the (μ+1) EA for pseudo-Boolean functions by studying the expected runtime on three well-known example functions. For two of these, we have derived asymptotically tight bounds, and in all three cases, $\mu = 1$ leads asymptotically to the lowest runtime. However, we have also identified a function where the (μ+1) EA outperforms the (1+1) EA and its multistart variants drastically provided that $\mu \geq n/\ln(en)$.

To bound the runtime of the population-based (μ+1) EA, we have developed the new technique of analyzing randomized family trees. This technique is not limited to this EA. For the case of fitness-proportional selection, similar family trees were investigated before. Since the upper bounds on the depth of the (μ+1) EA's family trees are independent of the mutation operator and even of the search space, the new technique has already been applied to continuous search spaces by Jägersküpper and Witt (2005). Nevertheless, the most interesting direction seems to be an extension to (μ+λ) strategies by a combination with the existing theory on the (1+λ) EA.

References

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

Droste, S., Jansen, T., and Wegener, I. (2006). Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*. To appear, preliminary version in Proceedings of FOGA'02.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications*. Wiley.

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

He, J. and Yao, X. (2002). From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5):495–511.

Jägersküpper, J. and Witt, C. (2005). Rigorous runtime analysis of a (μ+1) ES for the Sphere function. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 849–856. ACM Press.

Jansen, T., De Jong, K., and Wegener, I. (2005). On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440.

Jansen, T. and Wegener, I. (2001a). 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.

Jansen, T. and Wegener, I. (2001b). On the utility of populations. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'01)*, pages 1034–1041. Morgan Kaufmann.

Jansen, T. and Wegener, I. (2001c). Real royal road functions – where crossover provably is essential. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'01)*, pages 375–382. Morgan Kaufmann.

Jansen, T. and Wiegand, R. P. (2004). The cooperative coevolutionary (1+1) EA. *Evolutionary Computation*, 12(4):405–434.

Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms*. Cambridge University Press.

Raab, M. and Steger, A. (1998). Balls into bins – a simple and tight analysis. In *Proceedings of the 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'98)*, volume 1518 of *Lecture Notes in Computer Science*, pages 159–170. Springer.

Rabani, Y., Rabinovich, Y., and Sinclair, A. (1998). A computational view of population genetics. *Random Structures and Algorithms*, 12(4):313–334.

Storch, T. (2004). On the choice of the population size. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'04)*, volume 3102 of *Lecture Notes in Computer Science*, pages 748–760. Springer.

Storch, T. and Wegener, I. (2003). Real royal road functions for constant population size. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'04)*, volume 2724 of *Lecture Notes in Computer Science*, pages 1406–1417.

Wegener, I. (2002). Methods for the analysis of evolutionary algorithms on pseudo-boolean functions. In Sarker, R., Yao, X., and Mohammadian, M., editors, *Evolutionary Optimization*, pages 349–369. Kluwer.

Wegener, I. and Witt, C. (2005). On the optimization of monotone polynomials by simple randomized search heuristics. *Combinatorics, Probability and Computing*, 14:225–247.

Witt, C. (2003). Population size vs. runtime of a simple EA. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)*, volume 3, pages 1996–2003. IEEE Press.