

**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

---

An Analysis of the  $(\mu+1)$  EA on Pseudo-Boolean  
Functions

Carsten Witt

No. CI-170/04

Technical Report

ISSN 1433-3325

May 2004

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.

# An Analysis of the $(\mu+1)$ EA on Pseudo-Boolean Functions

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

May 18, 2004

## Abstract

Evolutionary Algorithms (EAs) are successfully applied for optimization in discrete search spaces, but theory is still weak in particular for population-based EAs. Here, a first rigorous analysis of the  $(\mu+1)$  EA on pseudo-Boolean functions is presented. For three example functions well-known from the analysis of the  $(1+1)$  EA, bounds on the expected runtime and success probability are derived. For two of these functions, upper and lower bounds on the expected runtime are tight, and on all three functions, the  $(\mu+1)$  EA is never more efficient than the  $(1+1)$  EA. Moreover, all lower bounds grow with  $\mu$ . On a more complicated function, however, a small increase of  $\mu$  provably decreases the expected runtime drastically.

For the lower bounds, a novel proof technique is developed. The stochastic process creating family trees of individuals is investigated and relationships with well-known models of random trees, e. g., uniform random recursive trees, are established. Thereby, a known theory on random trees is transferred to the analysis of EAs. Moreover, generalizations of the technique are applicable to more complex 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 in the theoretical analysis of the computational time complexity of EAs have been made. However, this kind of theory often focuses on simple single-individual EAs such as the  $(1+1)$  EA (e. g., Garnier, Kallel, and Schoenauer

---

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

(1999), Droste, Jansen, and Wegener (2002), Wegener (2002)). Therefore, it does not explain the utility of populations employed in many real-world EAs. In the context of multi-objective optimization, populations are used to maintain a set of solutions close to Pareto-optimal solutions or even a subset of a Pareto set. Indeed, for simple multi-objective evolutionary algorithms, first runtime analyses have been presented recently (e.g., Laumanns, Thiele, Zitzler, Welzl, and Deb (2002), Giel (2003)). These analyses, however, do not explain the utility of populations satisfactorily if we consider the scenario of single-objective optimization. There, a population is meant to maintain implicit parallelism and to explore different regions of the search space.

We focus on single-objective optimization in discrete search spaces, in particular on the maximization of pseudo-Boolean functions  $f: \{0, 1\}^n \rightarrow \mathbb{R}$ . Here, some analyses of the impact of crossover operators on the runtime are known (e.g., Storch and Wegener (2003), Jansen and Wegener (2001c)). Compared with this, results on population-based EAs where mutation is the only search operator are somewhat weak. Runtime analyses for steady-state EAs using fitness-proportional selection and a standard mutation as single search operator have been performed by Jansen and Wegener (2001b) and Witt (2003). Recently, Storch (2004) presented runtime analyses for steady-state EAs using uniform selection and diversity-maintaining operators. Moreover, He and Yao (2002) studied some variants of  $(\mu+\mu)$  EAs. However, results on the time complexity of standard  $(\mu+\lambda)$  EAs using uniform selection for reproduction hardly exist for  $\mu > 1$ , i.e., non-trivial sizes of the parent population. Up to now, only results on corresponding  $(1+\lambda)$  EAs (Jansen and De Jong (2002)) are known.

The aim of this paper is to contribute to a theory of standard  $(\mu+\lambda)$  EAs with  $\mu > 1$  in discrete search spaces. Here, we start with the simple case  $\lambda = 1$  and consider a  $(\mu+1)$  EA that is a generalization of the  $(1+1)$  EA for the search space  $\{0, 1\}^n$ , and follow the research line started for this  $(1+1)$  EA. Therefore, we study the behavior of the  $(\mu+1)$  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  $(\mu+1)$  EA from below 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  $(\mu+1)$  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 (2003), He and Yao (2003)).

The paper is structured as follows. In Section 2, we define the  $(\mu+1)$  EA and the considered example functions. Moreover, we introduce the tool of family trees that is essential throughout the paper. In Section 3, simple upper bounds on the expected runtime of the  $(\mu+1)$  EA on the example functions are presented. In Section 4, we describe the new lower bound technique. We apply the technique in Section 5 to prove lower bounds on the expected runtime and bounds on the success probability. These bounds are tight for two of the examples. Moreover, they show that on all three examples, the  $(\mu+1)$  EA is asymptotically never more efficient than the  $(1+1)$  EA. However, it is a com-

mon belief that a population helps to better explore the search space, and it is important to find an example where the  $(\mu+1)$  EA with  $\mu > 1$  outperforms the  $(1+1)$  EA. Therefore, a function where an increase of  $\mu$  by a sublinear factor decreases the expected runtime drastically, namely from exponential to polynomial, is identified in Section 6. We finish with some conclusions.

## 2 Definitions

We obtain the  $(\mu+1)$  EA for the maximization of functions  $f: \{0,1\}^n \rightarrow \mathbb{R}$  as a generalization of the well-known  $(1+1)$  EA (see Droste, Jansen, and Wegener (2002)). As for continuous search spaces, a pure  $(\mu+1)$  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 the most sensible. These arguments lead to the following definition of the  $(\mu+1)$  EA.

### Definition 1 ( $(\mu+1)$ EA)

1. Choose  $\mu$  individuals  $x^{(i)} \in \{0,1\}^n$ , where  $i \in \{1, \dots, \mu\}$ , uniformly at random. Let the multiset  $X^{(0)} = \{x^{(1)}, \dots, x^{(\mu)}\}$  be the so-called initial population at time 0. Let  $t := 0$ .
2. Repeat infinitely
  - (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  $X^{(t+1)}$ , 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  $(\mu+1)$  EA as simple as possible and refrain from employing diversity-maintaining mechanisms. The  $(\mu+1)$  EA with  $\mu = 1$  is very similar to the  $(1+1)$  EA but differs in one respect. If an individual created by mutation has the same  $f$ -value as its father, either of both is retained with equal probability.

As usual in theoretical investigations, we leave the stopping criterion of the  $(\mu+1)$  EA unspecified and analyze the number of iterations (also called *steps*) of the infinite loop until the current population for the first time contains an optimal individual, i. e., one that maximizes  $f$ . The sum of this number and the population size  $\mu$  is denoted as the *runtime* of the  $(\mu+1)$  EA and corresponds to the number of function evaluations executed so far (a common approach in black-box optimization, cf. Droste, Jansen, Tinnefeld, and Wegener (2003)).

Throughout the paper, we consider only  $\mu = \text{poly}(n)$ , i. e., values of  $\mu$  bounded by a polynomial of  $n$ .

We study the  $(\mu+1)$  EA on the following example functions. The well-known function  $\text{ONEMAX}(x) = x_1 + \dots + x_n$  counts the number of ones of a string  $x \in \{0, 1\}^n$ , and  $\text{LEADINGONES}(x) = \sum_{i=1}^n \prod_{j=1}^i x_j$  counts the number of leading ones. The function  $\text{SPC}(x)$  (*short path with constant fitness*) introduced by Jansen and Wegener (2001a) equals  $n - \text{ONEMAX}(x)$  if  $x$  cannot be written as  $1^i 0^{n-i}$  for any  $i$ . It equals  $2n$  if  $x = 1^n$  and  $n+1$  otherwise.  $\text{SPC}$  is of particular interest since it is necessary to search within a plateau of constant fitness to find an optimal string. Here, it should be studied whether the exploration of the plateau benefits from the population.

To elucidate the utility of the  $(\mu+1)$  EA's population, throughout the paper, we compare the  $(\mu+1)$  EA with  $\mu$  parallel runs of the  $(1+1)$  EA. The total cost (neglecting initialization cost) of  $t$  steps of the  $(\mu+1)$  EA corresponds to the cost raised by  $\mu$  parallel runs of the  $(1+1)$  EA up to time  $t/\mu$ . Thus, if we consider a  $(\mu+1)$  EA at time  $t$ , we denote  $\mu$  parallel runs of the  $(1+1)$  EA considered at time  $t/\mu$  as the *corresponding parallel run*.

In order to derive runtime bounds for the  $(\mu+1)$  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, Rabinovich, and Sinclair (1998) and already been analyzed for a different population-based EA by Witt (2003)). Fix an arbitrary such individual  $x$ . If  $x$  is mutated, a descendant of  $x$  is produced. More generally, we can visualize the descendants of  $x$  and their descendants by the family tree  $T_t(x)$  at time  $t$  as follows.  $T_0(x)$  contains only the root  $x$ .  $T_t(x)$  contains  $T_{t-1}(x)$  and the additional edge  $\{v, w\}$  if  $w$  is the result of a mutation of the individual  $v$  at time  $t-1$  and  $v$  is contained in  $T_{t-1}(x)$ . The tree  $T_t(x)$  can contain individuals that have already been deleted from the population at time  $t$ . When studying family trees (and subgraphs of them), we often omit the time index  $t$ . Note that the  $(\mu+1)$  EA describes an infinite stochastic process, implying that the process growing the family trees  $T_t(x)$  is infinite as well for at least one  $x$  from the initial population.

Finally, all nodes in a family tree are different individuals, but many individuals in a family tree may represent the same search point  $s \in \{0, 1\}^n$ . For an individual  $x$  (in a family tree), we call the search point  $s \in \{0, 1\}^n$  that is associated with  $x$  *the string of  $x$*  or say that  $x$  *is/carries the string  $s$* .

### 3 Upper Bounds

The first two of following upper bounds on the expected runtime of the  $(\mu+1)$  EA 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)).

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

**Proof:** Note that the term  $\mu$  reflects the cost of initialization. We measure the progress to the optimum by the potential  $L$ , defined as the maximum LEADING-ONES 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 the leftmost zero. The selection and the mutation operator of the  $(\mu+1)$  EA are independent. 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  $(\mu+1)$  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

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

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.  $\square$

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

**Corollary 1** *Let  $\mu = \text{poly}(n)$ . Then the expected time until the  $(\mu+1)$  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 to show upper bounds on the expected runtime. This was also the case when Jansen and De Jong (2002) showed corresponding upper bounds for the  $(1+\lambda)$  EA.

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

**Proof:** The proof idea 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  $L$ .

By the same elementary calculations as in the proof of Theorem 1, the time until the population contains at least  $\min\{n/(n-L), \mu\}$  individuals with value  $L$  is bounded above by  $2e\mu \ln(en/(n-L))$  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) 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  $\mu + en \ln(en) + 5e\mu n$ .  $\square$

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 tool of family trees comes into play.

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

Before preceding to the proof, we introduce some notions. In the run of the  $(\mu+1)$  EA on SPC, there is a first point of time where all individuals of the current population are so-called plateau strings, i. e., individuals of shape  $1^i 0^{n-i}$ ,  $i \in \{0, \dots, n\}$ . (Note that we allow  $i = n$  whereas the string  $1^n$  is not from the plateau of constant fitness. Maybe it would be more precise to call these points path points since  $1^i 0^{n-i}$  is a connected path. We stick to the notion of plateau strings to avoid confusions with paths within family trees studied later.) Let  $t_{\text{pla}}$  denote this point of time. The random variable  $t_{\text{pla}} = 0$  has finite expectation and may take even extreme values such as 0, i. e., the initial population consists

only of plateau strings. With high probability, however,  $t_{\text{pla}}$  occurs later, and the population at time  $t_{\text{pla}}$  does not yet contain an optimal string.

Steps that occur after time  $t_{\text{pla}}$  and produce individuals outside the plateau, i. e., not of shape  $1^i 0^{n-i}$ , immediately delete the created individual. We now study the family trees of the population at some time  $t \geq t_{\text{pla}}$ . Let  $x$  be such an individual and let  $y$  be a node in its family tree. If the individual corresponding to  $y$  has been deleted by time  $t$ , we call the path from  $x$  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  $t_{\text{pla}}$ , e. g., a path cannot be alive if it contains individuals outside the plateau. Moreover, the following Lemma 1 is crucial. To avoid any misunderstandings in its formulation, we emphasize here that we consider a random object without any further conditions if we speak of “any object” in the context of a set of random objects. Hence, e. g., “any path” in Lemma 1 will mean that we consider a random path that can be thought of as being chosen uniformly from the corresponding family tree.

**Lemma 1** *Consider the  $(\mu+1)$  EA on SPC. Let  $x^{(0)}$  be any individual from the population at time  $t \geq t_{\text{pla}}$ , and let  $p^*$  be any path in the family tree of  $x^{(0)}$  at time  $t' \geq t$ . Let  $s^{(0)}, \dots, s^{(\ell)}$  denote the strings along  $p^*$ . If  $p^*$  is alive and  $s^{(i)} \neq 1^n$  for  $0 \leq i \leq \ell$ , the sequence  $s^{(0)}, \dots, s^{(\ell)}$  has the same distribution as the sequence of strings of the  $(1+1)$  EA on SPC in the time steps  $0, \dots, \ell$  if it starts with  $s^{(0)}$  and does not create  $1^n$  by time  $\ell$ .*

**Proof:** Let  $x^{(0)}, \dots, x^{(\ell)}$  be the nodes of the path  $p^*$ . We prove the claim by induction on  $i \in \{0, \dots, \ell\}$ , where the base case  $i = 0$  is trivial. For the induction step, let  $0 \leq i < \ell$  and assume that the node  $x^{(i)}$ , carrying  $s^{(i)}$ , 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 for the induction step.

- A step occurs that chooses  $x^{(i)}$  for mutation, creates a candidate  $s' \neq 1^n$  for  $s^{(i+1)}$  by the mutation and does not delete the node  $x'$  carrying  $s'$ .
- If  $i + 1 < \ell$ ,  $x'$  is not deleted until a step occurs that chooses  $x'$  and produces  $x^{(i+2)}$ ; if  $i + 1 = \ell$ ,  $x'$  is not deleted by time  $t'$ .
- The node  $x'$  equals the node  $x^{(i+1)}$  on the path  $p^*$ .

Let  $A$  denote this set of conditions. If  $A$  holds,  $s'$  is called  $s^{(i+1)}$ . W.l.o.g., we assume that the first node  $x'$  fulfilling the first two properties also satisfies the third property since we do not make any assumptions on the time steps between  $t$  and  $t'$  where  $p^*$ 's nodes are created and since the mutations in different steps of the  $(\mu+1)$  EA are independent of each other. Since we study time  $t' \geq t_{\text{pla}}$ , Condition  $A$  forces  $s'$  to be a plateau string different from  $1^n$ . This also holds for the current strings of the  $(1+1)$  EA up to time  $\ell$ . Hence, in the mutation



producing  $s^{(i+1)}$ , both the mutation operator of the  $(\mu+1)$  EA and the  $(1+1)$  EA are conditioned on the event that some  $1^k 0^{n-k}$ ,  $k \neq n$ , is produced.

To study the distribution of  $s^{(i+1)}$  more carefully, recall that the  $(\mu+1)$  EA chooses for deletion uniformly from the set of worst individuals. Now, since all individuals of shape  $1^j 0^{n-j}$ ,  $j \neq n$ , have the same SPC value, the event that a path with such an individual at its end dies is independent of the value of  $j$ . Hence, Condition  $A$  does not depend on  $j$ . Let  $p(j, k)$  be the probability of the  $(\mu+1)$  EA's mutation operator changing  $s^{(i)}$  of shape  $1^j 0^{n-j}$  into some  $s'$  of shape  $1^k 0^{n-k}$ , where  $j, k \neq n$ . By the definition of independence, we obtain  $p(j, k)|A = p(j, k)$ . This means that  $k$  follows the same distribution both with and without Condition  $A$ . The  $(1+1)$  EA has the same mutation operator as the  $(\mu+1)$  EA, and a mutation changing  $1^j 0^{n-j}$  into  $1^k 0^{n-k}$ , where  $j, k \neq n$ , is always accepted. Since  $s^{(i)}$  has the same distribution as the corresponding string of the  $(1+1)$  EA, the induction step follows.  $\square$

**Proof of Theorem 3:** Let  $x$  be an arbitrary individual from the population at time  $t_{\text{pla}}$  and let  $T(x)$  be its family tree. We want to show that the following property  $P$  holds. The expected time until at least one path in  $T(x)$  reaches length  $k$  or until all paths in  $T(x)$  are dead is bounded above by  $4e\mu k$  for all  $k$ , or the optimum is reached before. This will imply the theorem for the following reasons. We will show that  $E(t_{\text{pla}}) = O(\mu n \log n)$ . Moreover, we will apply Lemma 1 in the following way. We consider a random path within  $T(x)$  up to the first occurrence of  $1^n$ . It follows that the probability of obtaining  $1^n$  on a path of length  $\ell + 1$  within  $T(x)$  equals the probability that the  $(1+1)$  EA with starting point  $x$  creates  $1^n$  within  $\ell + 1$  steps. 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  $t \geq t_{\text{pla}}$ . 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 for  $x$  within  $4e\mu k$  steps. For a current point of time, we define the following potential  $L$ . Let  $S$  be the set of those currently alive successors of  $x$  that will always have an alive descendant within  $4e\mu k$  steps. Then  $L$  is defined to denote the length of a longest path leading from  $x$  to an individual from  $S$ . A special property of the potential  $L$  is that it depends on the future, but is a function mapping each current point of time to a value. Moreover, by definition,  $L$  cannot shrink in the run of the  $(\mu+1)$  EA, and there is the following sufficient condition for increasing  $L$ . An individual  $x'$  defining the current  $L$ -value is chosen for mutation, a child from the plateau 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  $L$  is bounded above by  $4e\mu$ , implying that an alive path reaches length  $k$  after an expected number of at most  $4e\mu k$  steps.

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

## 4 A General Lower Bound Technique

For lower bounds on the runtime, we consider the growth of the family tree for any initial individual of the  $(\mu+1)$  EA. Upper bounds on the depth of family trees always follow from the selection mechanism of the  $(\mu+1)$  EA, which selects the individual to be mutated uniformly from the current population. Therefore, it is possible to model the stochastic process growing a family tree as follows.

**Definition 2 ( $1/\mu$ -tree)** *Let  $p := p_{t,u}$ ,  $t, u \geq 0$ , be a sequence of probability distributions such that the support of  $p_{t,u}$  is a subset of  $\{0, 1, \dots, u\}$ . A  $p$ -tree at time 0 consists only of the root. A  $p$ -tree  $T_t$  at time  $t \geq 1$  is obtained from a  $p$ -tree  $T_{t-1}$  as follows. Let  $u$  be the number of nodes of  $T_{t-1}$ . Sample  $v$  by  $p_{t-1,u}$ . If  $v > 0$ , append a new leaf to the  $v$ -th inserted node of  $T_{t-1}$ ; otherwise, let  $T_t := T_{t-1}$ .*

*A  $p$ -tree is called a  $1/\mu$ -tree if  $p_{t,u}(v) \leq 1/\mu$  for all  $v > 0$ .*

A  $p$ -tree at time  $t$  can have less than  $t + 1$  nodes since  $p$  can put some probability on 0. In principle, the growth of a family tree of the  $(\mu+1)$  EA on a given function can be modeled by the stochastic process called  $p$ -tree. However, we cannot specify the distributions  $p_{t,u}$  exactly since it is too difficult to predict whether and, if so, which individuals corresponding to nodes from a tree are deleted. Recall that nodes corresponding to deleted individuals are nevertheless kept in the family tree forever but cannot produce any descendants. Hence, in the  $(\mu+1)$  EA, deleted nodes have probability 0 of being chosen and alive nodes have probability  $1/\mu$ . This is modeled by the class of processes that we call  $1/\mu$ -tree.

The following lemma contains an interesting result for the depth of  $1/\mu$ -trees. Its technical proof is included in Appendix A.2.

**Lemma 2** *Let  $D(t)$  denote the depth of a  $1/\mu$ -tree at time  $t$ . For all  $t \geq 0$  and  $d \geq 0$ ,  $\text{Prob}(D(t) \geq d) \leq (t/\mu)^d/d!$ . Moreover,  $\text{Prob}(D(t) \geq 3t/\mu) = 2^{-\Omega(t/\mu)}$ .*

Lemma 2 states that with overwhelming probability, a family tree of the  $(\mu+1)$  EA becomes asymptotically no deeper than the total number of mutations performed in a single run of the corresponding parallel run. The tree can become wide, but a flat tree means that few mutations lie on any path from the root to a node in the tree. Hence, if the depth is small, this means that a leaf is an individual that is with a high probability similar to the root. This makes the optimization of even simple functions very unlikely if the tree is not deep enough. The following result is tight for some simple functions such as ONEMAX (if  $\mu$  is not too small).

**Theorem 4** *Let  $\mu = \text{poly}(n)$  and let  $f$  be a function with a unique global optimum. Then the expected runtime of the  $(\mu+1)$  EA on  $f$  is  $\Omega(\mu n + n \log n)$ . Moreover, the success probability within some  $c\mu n$  steps,  $c > 0$ , is  $2^{-\Omega(n)}$ .*

**Proof:** The lower bound  $\Omega(n \log n)$  follows for  $\mu \leq \log n/2$  by a generalization of the Coupon Collector's Theorem that has been described by Droste, Jansen, and Wegener (2002, Lemma 10) for linear functions and the  $(1+1)$  EA. To adapt the corresponding argumentation, we have to estimate the probability that some bit  $i \in \{1, \dots, n\}$  has the wrong value in all initial individuals. Since  $\mu \leq \log n/2$ , 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 positions, the probability that no mutation flips a bit from at least one of these positions within  $t := (n-1)(\ln n)/2$  steps is bounded below by

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

which implies that  $t$  steps are required with probability at least  $1 - e^{-1/2} - 2^{-\Omega(\sqrt{n})}$ .

For the lower runtime bound  $\Omega(\mu n)$ , we set up a phase of length  $s := \lfloor c\mu n \rfloor$  for some constant  $c > 0$  and show that the  $(\mu+1)$  EA requires at least  $s$  steps with probability  $1 - 2^{-\Omega(n)}$  if  $c$  is small enough. The proof idea is as follows. In  $s$  steps, a family tree created by the  $(\mu+1)$  EA has to reach (with high probability) a certain depth to optimize  $f$ . However, the probability of reaching this depth is very small.

Let  $x$  be an arbitrary initial individual. We consider the infinite random process of building its family tree. Let  $T_t(x)$  denote the tree at time  $t$ . According to Lemma 2, the probability that  $T_s(x)$  reaches depth at least  $3cn$  is  $2^{-\Omega(n)}$ . Now the aim is to prove that with probability  $1 - 2^{-\Omega(n)}$ , a depth of at least  $3cn$  is necessary for optimization (if  $c$  is small enough).

In the infinite process of building the tree  $T(x)$ , we consider the event that a node  $v$  with optimal  $f$ -value is inserted. The aim is to show that each such  $v$  is at a large depth with overwhelming probability. Each initial individual is created uniformly at random. By Chernoff bounds, the root  $x$  has Hamming distance at least  $n/3$  to the unique optimal string (represented by  $v$ ) with probability

$1 - 2^{-\Omega(n)}$ . Consider the path  $p_v$  from  $x$  to  $v$ . We claim that with probability  $1 - 2^{-\Omega(n)}$ , its length is at least  $n/4$ . This follows by considering a sequence of  $n/4$  strings where each string is the result of a mutation of its predecessor by means of the  $(\mu+1)$  EA's mutation operator. Each bit in each string in this sequence is flipped independently with probability  $1/n$ . Hence, the expected Hamming distance of any two strings in this sequence is at most  $n/4$ , and, by Chernoff bounds, it is less than  $n/3$  with probability  $1 - 2^{-\Omega(n)}$ . Since the nodes on each path in the trees  $T_t(x)$  form such a random sequence of strings, the claim follows.

The  $(\mu+1)$  EA does not stop when the optimum is reached. Therefore, in  $T_s(x)$ , there may be several paths leading from  $x$  to an optimal node. Since we study lower bounds on their length, we only have to consider paths whose end point is the first optimal node on the path. Let  $p_v$  be such a path with end point  $v$ . We obtain that the optimal node  $v$  cannot be successor or predecessor of the optimal node  $v'$  on another path  $p_{v'}$  of this kind. Hence, at most  $\mu$  different paths of this kind can be created even if we investigate the set of all family trees  $T_s(x)$  for each individual  $x$  of the initial population. Since  $\mu = \text{poly}(n)$ , the probability that there is a node with optimal  $f$ -value at depth less than  $n/4$  in a family tree is still  $2^{-\Omega(n)}$ . If  $c$  is small enough,  $n/4$  is at least  $3cn$ . Since the sum of all failure probabilities is  $2^{-\Omega(n)}$ , the proof of the theorem is complete.  $\square$

Theorem 4 covers the wide range of unimodal functions. For a larger subclass of the class 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  $(\mu+1)$  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  $(\mu+1)$  EA. However, we have to derive better lower bounds on the depth of a family tree such that more structure of the function  $f$  and of the encountered search points comes into play. In various circumstances, we have already exploited that many individuals in a family tree can be the same string  $s \in \{0, 1\}^n$ . This leads to the definition of so-called monochromatic subtrees. Note that our notion of subtrees refers to connected subgraphs of a tree, which do not need to contain any leaves of the original tree.

**Definition 3 (Monochromatic Subtree (MST))** *A subtree of a family tree is called a monochromatic subtree if all of its nodes are the same string.*

Obviously, all nodes in a monochromatic subtree have the same  $f$ -value. It is interesting that the stochastic process creating an MST sometimes equals the process for a so-called (uniform) random recursive tree, a model of random trees well known from the literature (e. g., Pittel (1994); Smythe and Mahmoud (1995)). This relationship will allow us to apply the known theory on RRTs. We obtain an RRT of any desired size by means of the following stochastic process.

**Definition 4 (Random Recursive Tree (RRT))** *An RRT at time 0 consists only of the root. An RRT  $T_t$  at time  $t \geq 1$  is obtained from an RRT  $T_{t-1}$  by uniformly at random choosing one of its nodes and appending a new leaf to it.*

Note that the RRT at time  $t \geq 0$  consists of exactly  $t + 1$  nodes. The processes generating MSTs and RRTs coincide only if the  $(\mu+1)$  EA is able to choose uniformly from the set of nodes of the MST. Since deleted individuals are nevertheless kept in the family tree, this property can only be guaranteed if the individuals of the considered MST are still present in the population.

**Lemma 3** *Let  $T^*$  be a monochromatic subtree of a family tree and let  $V$  be the set of nodes of  $T^*$ . If the  $(\mu+1)$  EA does not delete any individual from  $V$  until the creation of the last node of  $T^*$  then  $T^*$  is an RRT.*

**Proof:** We consider the process of building  $T^*$ , studying only the steps  $t$  of the  $(\mu+1)$  EA choosing nodes from the subset  $N_t \subseteq V$  of already created nodes. Let  $r \in \{0, 1\}^n$  be the string of the first node that is inserted into  $T^*$ , i. e.,  $r$  is the color of the MST  $T^*$ . Since all nodes, i. e., individuals, in  $V$  remain present in the population by the end of the procedure, the selection mechanism of the  $(\mu+1)$  EA implies that each of these steps chooses the father of some new node uniformly at random from the set  $N_t$ .

However, the result of such a step can be a node that is a different string than  $r$ . We call this event the creation of an external node. External nodes are not attributed to  $T^*$ . Since all nodes in  $T^*$  are the same string  $r$ , the event of creating an external node is independent of the choice of the father. Therefore, we can ignore the steps creating external nodes. The next step complies again with the process of creating an RRT. By induction, the monochromatic subtree  $T^*$  is created according to the rules for an RRT.  $\square$

If the  $(\mu+1)$  EA deletes individuals of an MST from the population, it chooses these, by the definition of an MST and the  $(\mu+1)$  EA, uniformly from the alive nodes of the MST. Hence, the earliest inserted nodes have the highest chances of having been deleted by any fixed time  $t$ . Early inserted nodes are close to the root. This implies that an MST that is affected by deletion steps is typically deeper than an RRT of the same size. We can make this precise by considering generalized RRTs, namely so-called  $p$ -marked random trees ( $p$ -marked RTs).

**Definition 5 ( $p$ -marked RT)** *Let  $p_{t,u}$ ,  $t, u \geq 0$ , be a sequence of probability distributions such that the support of  $p_{t,u}$  is a subset of  $\{0, \dots, u\}$ . A  $p$ -marked RT at time 0 consists only of the unmarked root. A  $p$ -marked RT  $T_t$  at time  $t \geq 1$  is obtained from a  $p$ -marked RT  $T_{t-1}$  in two steps. First, an unmarked node is chosen uniformly at random and a new, unmarked leaf is appended. Let  $U$  denote the set of unmarked nodes after this step. Then  $u^*$  is sampled according to  $p_{t-1, |U|-1}$ , a subset  $S^* \subseteq U$  of size  $u^*$  is chosen uniformly at random, and all nodes in  $S^*$  are marked.*

Again, a tree at time  $t$  has exactly  $t + 1$  nodes, only the unmarked ones of which can become fathers of new nodes. The concept of marked nodes has been introduced since it seems to be the easiest way of modeling those nodes of family trees that correspond to deleted individuals. It is crucial that for all  $p_{t,u}$ , the set of newly marked nodes is, by definition, uniform over the yet unmarked ones. We postulate that always at least one node remains unmarked since the tree could not grow any more otherwise.

**Lemma 4** *A monochromatic subtree of a family tree is a  $p$ -marked RT.*

**Proof:** Like in the proof of Lemma 3, we concentrate on the steps choosing nodes from the subset of already created nodes of  $T^*$ . Likewise, we can ignore steps creating external nodes.

Between the steps adjoining nodes to the already created part of  $T^*$ , it is possible for the  $(\mu+1)$  EA to eliminate one or more individuals, i. e., nodes from the subtree. Since all nodes of  $T^*$  are the same string, the set of deleted nodes is chosen uniformly. Finally, the next step chooses uniformly from the set of undeleted nodes. Hence, the whole process of growing  $T^*$  is modeled by Definition 5.  $\square$

We will show that the probability of a  $p$ -marked RT with  $t$  nodes reaching depth  $d$  is, for any  $p$ , at least as large as the respective probability of an RRT. Let for a  $p$ -marked RT and an RRT at time  $t$  the measures  $D^*(t, i)$  resp.  $D(t, i)$  denote the depth of the node that was inserted at time  $i$ . Then the following technical lemma holds. Its proof is contained in Appendix A.1.

**Lemma 5** *For all  $t, i, d \geq 0$  and  $i \leq t$ ,  $\text{Prob}(D^*(t, i) \geq d) \geq \text{Prob}(D(t, i) \geq d)$ .*

Since lower bounds on the depth of ordinary RRTs are well known (see, e. g., Lemma 9 in Appendix A.1), we have developed new tools for lower bounding the depth of MSTs and, therefore, of family trees. Upper bounds are still provided by Lemma 2.

## 5 More Special Lower Bounds

Combining the tools developed so far yields a new proof method for the analysis of the  $(\mu+1)$  EA. In Section 5.1, we exemplify the new proof method for the well-studied LEADINGONES function. In Section 5.2, we apply the technique again to prove lower bounds for the function SPC.

### 5.1 Lower Bound for LeadingOnes

**Theorem 5** *Let  $\mu = \text{poly}(n)$ . Then the expected runtime of the  $(\mu+1)$  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, Jansen, and Wegener (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  $(\mu+1)$  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  $(\mu+1)$  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 show that for some small enough constant  $c > 0$ , the  $(\mu+1)$  EA requires at least  $s := \lfloor c\mu n \log n \rfloor$  steps with probability  $1 - 2^{-\Omega(n)}$ . Now we consider the family tree  $T_s(x)$  obtained after  $s$  steps for an arbitrary initial individual  $x$ . By Lemma 2, it suffices to show that a depth of at least  $3cn \log n$  is necessary for optimization with probability  $1 - 2^{-\Omega(n)}$ .

For notational convenience, let  $f := \text{LEADINGONES}$ . During the process of building the tree  $T(x)$ , we consider the event that a node  $v$  with optimal  $f$ -value  $n$  is inserted. Since all initial individuals are uniform over  $\{0, 1\}^n$ , the root  $x$  has an  $f$ -value of at most  $n/2$  with probability  $1 - 2^{-\Omega(n)}$ . Consider the path  $p_v$  from  $x$  to  $v$ . W.l.o.g., the  $f$ -value is non-decreasing along  $p_v$ . As mentioned above (shown by Droste, Jansen, and Wegener (2002)), the bits after the leftmost zero are, in each string on  $p_v$ , uniformly distributed. Moreover, the  $f$ -value has to increase by at least  $n/2$  along  $p_v$  with probability  $1 - 2^{-\Omega(n)}$ . Together with the above-mentioned analysis of free-riders, it follows that at least  $n/6$  different strings lie on  $p_v$  with probability  $1 - 2^{-\Omega(n)}$ . We call the nodes that are different strings than their fathers *subtree roots*. For a subtree root  $r$ , we denote by  $T^*(r)$  the maximal MST rooted at  $r$ . Observe that all nodes of  $T^*(r)$  are below  $r$  in the tree  $T(x)$ . Now we work under the assumption that  $p_v$  contains at least  $n/6$  subtree roots.

Consider the event that a subtree root  $r \neq v$  is created, and let  $r'$  denote the random next subtree root on  $p_v$ . By Lemma 4, the MST  $T^*(r)$  is a  $p$ -marked RT, and  $r'$  is some node inserted into (but not not attributed to) a  $p$ -marked RT. Considering the construction of  $T^*(r)$ , we prove that  $r'$  is likely to be created late during this process. The probability of mutating a string with value  $f(r)$  to a better string is bounded above by  $1/n$ . Hence, with probability at least  $1/2$ , the first  $n/2$  steps that choose a father in the already existing MST create nodes with at most the same value as the root. Since producing a replica of a string has probability  $(1 - 1/n)^n \geq 1/(2e)$ , the expected number of replicas within  $n/2$  steps is at least  $n/(4e)$ . By Chernoff bounds, with probability at least  $1/2 - 2^{-\Omega(n)}$ ,  $T^*(r)$  receives at least  $n/(8e)$  nodes before an individual with larger value than  $f(r)$  is appended. Hence, with probability at least  $1/2 - 2^{-\Omega(n)}$ , the node  $r'$  has a distance to  $r$  that is bounded below by the depth of the at least  $n/(8e)$ -th node of a  $p$ -marked RT.

How deep is the at least  $k$ -th node such that  $k \geq n/(8e)$  within a  $p$ -marked RT? We know it if the tree is an ordinary RRT. Then, according to Lemma 9 and Chebyshev's inequality, the depth is at least  $(\log n)/2$  with probability

at least  $1/2$  (for  $n$  large enough). By Lemma 5, the same statement holds also for a  $p$ -marked RT. Altogether, the distance of  $r$  and  $r'$  on  $p_v$  is at least  $(\log n)/2$  with probability at least  $1/4 - o(1)$ . Since the process creating  $T^*(r')$  is independent of the process creating  $T^*(r)$ , we can apply Chernoff bounds. Since at least  $n/6$  choices for  $r$  are available on  $p_v$ , at least  $n/25$  subtree roots have their successive subtree roots at distance at least  $(\log n)/2$  with probability  $1 - 2^{-\Omega(n)}$ . Altogether, the length of  $p_v$  is at least  $n(\log n)/50$  with probability  $1 - 2^{-\Omega(n)}$ .

By the same arguments as in the proof of Theorem 4, we have to study at most  $\mu$  paths leading to an optimal node. Hence, since  $\mu = \text{poly}(n)$ , the probability that there is a node with  $f$ -value  $n$  at depth less than  $n(\log n)/50$  in a family tree is still  $2^{-\Omega(n)}$ . If  $c$  is small enough, the bound  $n(\log n)/50$  is at least  $3cn \log n$ . Finally, the sum of all failure probabilities is  $2^{-\Omega(n)}$ .  $\square$

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  $(\mu+1)$  EA provided that  $\mu = \Omega(n/\log n)$ . A similar correspondence seems to hold for a generalization of LEADINGONES called  $\text{LOB}_b$  (*leading ones blocks*) studied by Jansen and Wiegand (2003). For constant  $b \in \mathbb{N}$ , let  $\text{LOB}_b(x) := \sum_{i=1}^{n/b} \sum_{j=1}^{bi} x_j - \text{ONEMAX}(x)$ . The ONEMAX part of  $\text{LOB}_b(x)$  implies that after a short time and with high probability, each current string of the (1+1) EA and the  $(\mu+1)$  EA is of shape  $1^{bi}0^{n-bi}$  for some  $i$ . To create a string with higher  $\text{LOB}_b$ -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  $(\mu+1)$  EA on  $\text{LOB}_b$  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  $(\mu+1)$  EA. Therefore, in this section, we will derive a lower bound on the expected runtime of the  $(\mu+1)$  EA on SPC. In fact, 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 Theorem 3. Note, however, that the lower bound is  $\Omega(n^3)$ , implying that the  $(\mu+1)$  EA is asymptotically never more efficient than the (1+1) EA on SPC.

**Theorem 6** *Let  $\mu = \text{poly}(n)$ . Then the expected runtime of the  $(\mu+1)$  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  $x$ . We consider the event that while creating its family tree  $T(x)$ , a path  $p^*$  emerges from the root to a node that is the optimal string  $1^n$ . For technical reasons, we are interested in nodes that are still far away from  $1^n$  and  $0^n$  and study only a subpath  $p^{**}$  of  $p^*$  with the following properties: All strings on  $p^{**}$  are of shape  $1^i 0^{n-i}$  with  $n/4 \leq i \leq 3n/4$ . Of course, there is a non-zero probability that no such path  $p^{**}$  exists. However, our goal is to show that with high probability, such a  $p^{**}$  exists and has length at least  $3cn^3/\log \mu$  if  $c$  is small enough. The latter is proven in two steps. First, we show that with high probability, the strings of the nodes change  $\Omega(n^2/\log \mu)$  times along  $p^{**}$ . Second, the distance of two nodes with different strings is  $\Omega(n)$  with probability  $\Omega(1)$ .

To be able to investigate  $p^{**}$ , we concentrate on points of time where the population contains plateau strings, i. e., individuals of shape  $1^i 0^{n-i}$ ,  $i \in \{0, \dots, n\}$ . Let  $t_{\text{p11}}$  denote the first point of time such that at least one plateau string exists. Moreover, let  $t_{\text{pla}}$  still denote the first point of time such that  $\mu$  plateau strings are in the population. Recall that the optimum may have been reached before. We prove that this is very unlikely.

**Lemma 6** *With probability  $1 - 2^{-\Omega(\sqrt{n})}$ , all populations up to (and including) time  $t_{\text{pla}}$  contain only individuals with at most  $3n/5$  ones.*

**Proof:** First of all, we prove that all individuals up to time  $t_{\text{p11}}$  have at most  $4n/7$  ones with probability  $1 - 2^{-\Omega(n)}$ . To show this, the following arguments are needed. Since the initial individuals are drawn uniformly at random and  $\mu = \text{poly}(n)$ , 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  $t_{\text{p11}}$ , the fitness function  $\text{SPC}(x)$  equals  $n - \sum_{i=1}^n x_i$ , also called ZERO-MAX. Hence, the selection mechanism of the  $(\mu+1)$  EA implies that the minimum number of zeros in the population cannot decrease up to time  $t_{\text{p11}} - 1$ . By the same arguments as in the proof of Theorem 3, we have  $E(t_{\text{p11}}) = O(\mu n \log n)$  and, by Markov's inequality and the fact that independent phases may be repeated,  $t_{\text{p11}} = O(\mu n^2 \log n)$  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  $(\mu+1)$  EA has probability  $2^{-\Omega(n \log n)}$  due to  $\mu = \text{poly}(n)$ , the plateau string created at time  $t_{\text{p11}}$  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 from time  $t_{\text{p11}}$  until time  $t_{\text{pla}}$  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. Since no family tree becomes deeper than  $O(\sqrt{n} \log \mu)$  in this number of steps with probability  $1 - 2^{-\Omega(\sqrt{n})}$  according to Lemma 2, we can estimate the number of ones in the individuals (which are plateau strings) in the population at time  $t_{\text{pla}}$ . Since

$\mu = \text{poly}(n)$  holds and  $O(\sqrt{n} \log n)$  steps flip only a total number of  $O(\sqrt{n} \log n)$  bits with probability  $1 - 2^{-\Omega(\sqrt{n})}$  according to Chernoff bounds, no individual at time  $t_{\text{pla}}$  has more than  $3n/5$  ones with probability  $1 - 2^{-\Omega(\sqrt{n})}$ .  $\square$

Now we can study  $p^{**}$ , a subpath of the path  $p^*$  with some nice structure. To cope with the case that  $\mu$  does not grow with  $n$ , we always have to be careful when applying  $O$ -notation with respect to  $\mu$ .

**Lemma 7** *With a probability of at least  $\max\{1 - 2^{-\Omega(\log^2 \mu)}, \Omega(1)\}$ , there is a subpath  $p^{**}$  of  $p^*$  such that all nodes of  $p^{**}$  are created at time  $t_{\text{pla}}$  or later,  $p^{**}$  contains only strings  $1^i 0^{n-i}$  where  $n/4 \leq i \leq 3n/4$  and the strings along  $p^{**}$  change  $\Omega(n^2/\log \mu)$  times.*

**Proof:** Let  $v$  be the last string on  $p^*$  that is created up to (and including) time  $t_{\text{pla}}$ . According to Lemma 6, it contains at most  $3n/5$  ones with probability  $1 - 2^{-\Omega(n^{1/2})}$ . From time  $t_{\text{pla}}$  on, an individual that is outside the plateau is deleted immediately after its creation. Hence,  $v$  and all its successors on  $p^*$  are plateau strings since  $p^*$  could not lead to an optimal string otherwise. Since flipping a linear number of bits in one step 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  $1 - 2^{-\Omega(n^{1/2})}$ , we have identified a starting point  $v^*$  for  $p^{**}$  that is created at time  $t_{\text{pla}}$  or later. The path  $p^{**}$  ends at the first string that is not of shape  $1^i 0^{n-i}$  with  $n/4 \leq i \leq 3n/4$ . Now we have to show that with high probability, at least  $\Omega(n^2/\log \mu)$  nodes on  $p^{**}$  are different plateau strings than their fathers.

The string  $1^i 0^{n-i}$ ,  $n/3 \leq i \leq 2n/3$ , of the starting node  $v^*$  has Hamming distance  $\Omega(n)$  to the optimal string  $1^n$ . We want to show that the random walk describing the different strings of nodes on  $p^{**}$  is similar to a fair random walk on the line of length  $\Theta(n)$ . For the latter random walk, it is well known that a distance  $\Omega(n)$  to its starting point is not overcome within  $\Theta(n^2/\log \mu)$  steps with probability  $1 - 2^{-\Omega(\log^2 \mu)}$ . We consider the first at most  $s := \lfloor n^2/(50(\log \mu + 1)) \rfloor$  nodes on  $p^{**}$  that are different strings than their fathers. Each considered string is the result of a mutation of some  $1^i 0^{n-i}$  into some  $1^j 0^{n-j}$ ,  $j \neq i$ . In contrast to the mentioned random walk, it may be the case that  $|j - i| > 1$ .

We consider the event that the  $i$ -th bit of a string  $1^i 0^{n-i}$  is flipped. Then it is most likely to obtain  $1^{i-1} 0^{n-i+1}$  by the mutation. To create a plateau string  $1^{i-1-k} 0^{n-i+1+k}$  for some  $k \geq 1$ , the bits  $i-k, \dots, i-1$ , so-called additionally flipping bits, have to be flipped simultaneously. The corresponding probability is bounded above by  $1/n^k$  for the  $(\mu+1)$  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 later on.

Next, we assume no mutation to flip at least  $n/4$  bits at least once within  $s$  steps, which holds with probability  $1 - 2^{-\Omega(n \log n)}$ . Now, given the event that some  $1^i 0^{n-i}$ , where  $n/4 \leq i \leq 3n/4$ , is mutated to a different plateau string,

the probability of flipping bit  $i$  and bit  $i + 1$  is the same. Hence, the probability of obtaining a string with more than  $i$  ones is the same as the probability of obtaining less ones, and we have established a correspondence to a fair random walk. In  $s$  steps, we expect  $s/2$  steps increasing and  $s/2$  steps decreasing the number of ones provided no string with less than  $n/4$  or more than  $3n/4$  ones is encountered in these steps. We call the latter event a failure. The probability that there is a surplus of  $n/24$  increasing or decreasing steps is bounded both by  $2^{-\Omega(\log^2 \mu)}$  and by  $1 - \Omega(1)$  according to Chernoff bounds 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  $1^i 0^{n-i}$ , where  $i < n/4$  or  $i > 3n/4$ , within the  $s$  mutations is  $\max\{2^{-\Omega(\log^2 \mu)}, 1 - \Omega(1)\}$ , which justifies the assumption that no failure occurs.  $\square$

Now it remains to estimate the distance of two nodes on  $p^{**}$  that are different strings than their fathers. Like in the proof of Theorem 5, we call such nodes *subtree roots*. Let  $r$  be a subtree root other than the last one on  $p^{**}$ . Like in the previous section, we can again investigate the maximal monochromatic subtree  $T^*(r)$  rooted at  $r$ . To lower bound its depth, the probability of mutating a plateau string  $1^i 0^{n-i}$  into a different plateau string is essential. It is bounded above by  $2/n$  since bit  $i$  or bit  $i + 1$  has to flip. Asymptotically, this equals the upper bound on the probability of creating a new subtree root in the proof of Theorem 5. Hence, by the arguments there, we could immediately derive the lower bound  $\Omega(\mu n^2)$  on the expected runtime of the  $(\mu+1)$  EA on SPC. However, the distance of two subtree roots can be bounded below by more direct arguments.

**Lemma 8** *With probability  $\Omega(1)$ , the distance of any two consecutive subtree roots on  $p^{**}$  is  $\Omega(n)$ . This holds independently of previous subtree roots on  $p^{**}$ .*

**Proof:** First of all, we consider the superpath  $p^*$ , excluding the last node (which is  $1^n$ ). Let  $r$  be a random node on  $p^*$  created at time  $t_{\text{pla}}$  or later. Now, we can apply Lemma 1 with  $x^{(0)} := r$ . Hence, we know that the successor of  $r$  is the result of a mutation under the condition that its predecessor, a plateau string, is changed again into a plateau string different from  $1^n$ .

We already know that the unconditional probability of changing a plateau string into a different plateau string is bounded above by  $2/n$ . The probability of changing a plateau string into a plateau string is bounded below by  $(1 - 1/n)^n \geq 1/(2e)$  since a replica is sufficient. Hence, the probability that the successor of  $r$  on  $p^*$  differs from its father is bounded above by  $4e/n$ . Consequently, the first  $n/(8e)$  successors of  $r$  on  $p^*$  are the same string as  $r$  with probability at least  $1/2$ .

Actually, rather than  $p^*$ , we consider the subpath  $p^{**}$ . By definition, this is created at time  $t_{\text{pla}}$  or later as desired. However, also by definition, only strings of shape  $1^i 0^{n-i}$ , where  $n/4 \leq i \leq 3n/4$ , are created on  $p^{**}$ , and the strings along  $p^{**}$  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 subtree roots on  $p^{**}$ . Finally, the above statement on the successors

of  $r$  holds also if  $r$  is a subtree root since the definition of a subtree root only implies a condition on its predecessor. Therefore, the distance of a subtree root and the next subtree root on  $p^{**}$  is at least  $n/(8e)$  with probability at least  $1/2$ . This holds independently of previous subtree roots due to the independence of the steps of the  $(\mu+1)$  EA.  $\square$

**Proof of Theorem 6:** Assuming that the event of Lemma 6 holds, we only introduce an error term of  $2^{-\Omega(\sqrt{n})}$ . We start with the trivial case  $\mu = O(1)$ . Then by Lemma 7, there exists a constant  $c > 0$  such that with probability  $\Omega(1)$ , at least  $cn^2$  mutations changing plateau strings into different plateau strings are necessary. Each such mutation has probability  $O(1/n)$ . The theorem follows by Chernoff bounds.

Now let  $\mu = \omega(1)$ . Then we combine Lemma 7 and Lemma 8. Let  $p^*$  still be a path that leads to an optimal node in some family tree  $T(x)$ . Assume that the strings along an appropriate subpath  $p^{**}$  according to Lemma 7 change  $\Omega(n^2/\log \mu)$  times. Then the expected number of consecutive subtree roots at distance  $\Omega(n)$  is  $\Omega(n^2/\log \mu)$ . Since Lemma 8 holds independently of previous subtree roots, we can apply Chernoff bounds. We obtain that with probability  $1 - 2^{-\Omega(n^2/\log \mu)}$ , the number of consecutive subtree roots at distance  $\Omega(n)$  is  $\Omega(n^2/\log \mu)$ , and therefore,  $p^*$ 's length is at least  $\Omega(n^3/\log \mu)$ . By the prerequisites of Lemma 7, we obtain that  $p^*$ 's length is  $\Omega(n^3/\log \mu)$  with probability  $1 - 2^{-\Omega(\log^2 \mu)}$ .

By the same arguments as in the proof of Theorem 4, we have to study at most  $\mu$  paths leading to an optimal node. Hence, the probability of there being at least one such path of length less than  $\ell := c'n^3/\log \mu$ ,  $c' > 0$  some constant, is still  $2^{-\Omega(\log^2 \mu)}$ . 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  $\mu = \text{poly}(n)$ . If we choose  $c$  small enough and  $n$  large enough,  $u$  is smaller than  $\ell$ .  $\square$

The lower and upper bounds for the  $(\mu+1)$  EA on SPC derived so far are still by a factor of  $\Theta(1/\log \mu)$  away. We should discuss what an asymptotically tight bound could look like. First of all, it is helpful to study  $\mu$  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  $1 - 2^{-\Omega(\alpha(n)^2)}$  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  $cn^3/\sqrt{\log \mu}$  steps,  $c > 0$  a constant, is still  $\Omega(1/\mu)$ , and the expected runtime of the parallel runs  $O(\mu n^3/\sqrt{\log \mu})$ . This might suggest that an asymptotically tight bound for the  $(\mu+1)$  EA is also of order  $\Theta(\mu n^3/\text{polylog}(\mu))$ , supporting the bound of Theorem 6. In any case, the  $(\mu+1)$  EA is, up to logarithmic factors w. r. t.  $\mu$ , as inefficient as the corresponding parallel run.

On the other hand, one can gain insight on the truth by looking for alternative proofs of a lower bound on the expected runtime of the  $(\mu+1)$  EA on SPC. Now we want to identify subtrees of a family tree that die early and hence do

not contain predecessors of an optimal node. Consider a population  $P_t$  at some time  $t \geq t_{\text{pla}}$  and before reaching the optimum, assume that all individuals in  $P_t$  have still at most  $3n/4$  ones and let  $x \in P_t$  be arbitrary. We study the creation of the family tree  $T(x)$ , ignoring nodes that are outside the plateau. Note that  $T(x)$  is not assumed to be monochromatic. We model the random walk that describes the number of alive nodes within  $T(x)$  by the Markov chain  $A$  on the state space  $\{0, \dots, \mu\}$  with starting state 1. In this random walk, we only count steps changing the number of alive nodes within  $T(x)$ . Suppose that the number of alive nodes in  $T(x)$  equals  $i > 1$ . Then both the probability of increasing the number of its alive nodes and the probability of decreasing it equal  $(i/\mu)((\mu - i)/(\mu + 1))$ , i. e., the random walk is symmetric at this state. However, state 1 is special. The definition of the  $(\mu+1)$  EA implies that there is always at least one subtree that is not completely dead. In the Markov chain  $A$ , we ignore this. Under this simplifying assumption, we obtain two absorbing states for the random walk.

We are interested in the time until absorption. This is the fair Gambler's Ruin problem (Feller (1971)). The expected time until absorption equals  $\mu$  in this scenario, and, by Markov's inequality, the time is at most  $\mu^2$  with probability  $1 - 1/\mu$ . If state  $\mu$  is reached, this means that the family tree rooted at any other  $x' \in P_t$  is completely dead. Hence, under the simplifying assumption, each family tree  $T(x)$ , where  $x \in P_t$ , is dead after  $O(\mu^2)$  steps with probability at least  $1 - 1/\mu$ . Indeed at least one of these subtrees must remain alive forever. For any  $x \in P_t$ , this means that its family tree is still alive after  $\mu^2$  steps with probability at most  $2/\mu$ .

We consider two cases w. r. t.  $\mu$ . Let  $\mu = O(n^{3/2-\varepsilon})$  for some constant  $\varepsilon > 0$ . Then  $\mu^2$  is asymptotically smaller than the minimum depth of the subtree  $T(x)$  that is required for optimization with probability  $1 - o(1/\text{poly}(n))$  according to Lemma 7 and Lemma 8. This means that with probability  $1 - O(1/\mu)$ ,  $T(x)$  dies completely before  $1^n$  can be created in the tree. This would imply that as long as the current population is still far away from optimality, an expected fraction of  $\Omega((\mu - 1)/\mu)$  of all steps is wasted. Hence, a lower bound  $\Omega(\mu n^3)$  on the expected runtime of the  $(\mu+1)$  EA on SPC seems to hold in this case.

Unfortunately, this proof idea breaks down if  $\mu = \omega(n^{3/2}/\log \mu)$ . However, for large values of  $\mu$ , e. g.,  $\mu = \Omega(n^{3+\varepsilon})$  for some constant  $\varepsilon > 0$ , one can observe the following. By the Gambler's Ruin model, it holds that with probability  $1/\mu^{3+\varepsilon/2}$ ,  $T(x)$  contains  $\mu^{3+\varepsilon/2}$  alive nodes at least once before dying completely. To return from state  $\mu^{3+\varepsilon/2}$  to state 0, it takes  $\Omega(\mu^{6+\varepsilon})$  steps with constant probability, which is larger than  $O(\mu n^3)$ , the upper bound on the expected runtime. If we consider all individuals  $x \in P_t$ , we obtain that an expected number  $\Omega(\mu^{\varepsilon/2})$  of its individuals always have an alive descendant within  $O(\mu n^3)$  steps. Of course, the random walks describing the number of alive nodes within different subtrees  $T(x)$  are not independent since a large number of alive nodes in one tree implies a small number in all other trees. But if  $\mu$  is large enough, it seems to be the case that with high probability, a polynomial number of  $P_t$ 's individuals always has an alive descendant within  $O(\mu n^3)$  steps. This is a situation similar to the above-mentioned parallel run. Additionally using the

arguments concerning property  $P$  from the proof of Theorem 3, one would arrive at an upper bound of  $O(\mu n^3 / \text{polylog}(\mu))$  on the expected runtime.

We conjecture that an asymptotically tight bound on the expected runtime of the  $(\mu+1)$  EA on SPC is of order  $\Theta(\mu n^3 / \max\{\log(\mu/f(n)), 1\})$  for some function  $f(n)$  similar to  $n^3$ .

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

In the previous sections, we have shown for example functions that the  $(\mu+1)$  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  $\mu$ .

However, it is common belief that the population of the  $(\mu+1)$  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  $\ell$  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  $(\mu+1)$  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  $(\mu+1)$  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 if the ONEMAX part is optimized first, the  $(1+1)$  EA is expected to behave inefficiently. Moreover, if a global optimum is reached if the LEADINGONES part is optimized first, we expect the  $(\mu+1)$  EA to be efficient.

The following function has been defined with this idea in mind. Let strings  $x \in \{0, 1\}^n$  be divided into a prefix  $(x_1, \dots, x_m)$  of length  $m$  and a suffix  $(x_{m+1}, \dots, x_n)$  of length  $\ell$ . Let  $\ell := \lceil n^{1/2} \rceil$ , i. e.,  $m = n - o(n)$ . W. l. o. g.,  $m$  is divisible by 3. For a string  $x \in \{0, 1\}^n$ , we define  $\text{PO}(x) := x_1 + \dots + x_m$  as the number of **prefix ones**. Let  $\text{LSO}(x) := \sum_{i=0}^{\ell-1} \prod_{j=0}^i x_{m+1+j}$  be the number of **leading suffix ones**. Finally, let  $b := 2m/3 + \lceil n^{1/2} / (800 \log^2 n) \rceil$ . Then let

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

We have to discuss the structure of  $f$ . 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  $\text{LSO}(x) = \ell$  and  $\text{PO}(x) = 2m/3$ . However, if  $\text{PO}(x) \leq 2m/3$  and  $\text{LSO}(x) < \ell$ , the  $f$ -value is at most  $n^2(\ell - 1) + 2m/3$ , which is less than  $n^2 \ell - nb$ , a lower bound on the value in the second case, i. e., if  $\text{PO}(x) > 2m/3$ . If  $\text{PO}(x) = b$  and  $\text{LSO}(x) = \ell$ , we have a locally optimal

string with  $f$ -value  $n^2\ell + \ell$ . The Hamming distance to any better string 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  $(\mu+1)$  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  $(\mu+1)$  EA with  $\mu = 1$  are  $2^{\Omega(n^{1/2}/\log n)}$ .*

**Proof:** The proof idea is very simple. We show that the evolutionary algorithm is likely to create a string 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 string. 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  $(\mu+1)$  EA with  $\mu = 1$  since it does not matter whether equal-valued strings are rejected or not.

The initial string 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 to start with such a string. In the following, we divide the run into consecutive phases of length  $4e(n-1)$  and expect the (1+1) EA to increase the number of POs to  $b$  in one of the phases. We have to be careful since the number of POs of the current string 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  $4e(n-1)$ , i. e., without bad steps, occurs and that the LSO-value is still less than  $\ell$ . Then at most  $b$  decreases of the distance  $|\text{PO}(x) - b|$  are sufficient to create a string 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  $|\text{PO}(x) - b|$  and increases the  $f$ -value. Hence, the phase contains at least  $b$  decreases with probability at least  $1 - 2^{-\Omega(n)}$  according to Chernoff bounds.

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  $(1-1/n)^{4e(n-1)} \geq e^{-4e} = \Omega(1)$ . This implies that the good phase is preceded by at most  $\ell/8$  bad phases with probability at most  $(1 - e^{-4e})^{\ell/8} = 2^{-\Omega(\ell)}$ . Since  $\ell n/8$  mutations flip at most  $\ell/6$  bits with probability  $1 - 2^{-\Omega(\ell)}$ , we can work under the assumption of observing at most  $\ell/8$  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  $\ell$  since an LSO-increasing step can contain many free-riders, i. e., 1-bits after the leftmost suffix zero. However, as a consequence of the analysis of LEADINGONES by Droste, Jansen, and Wegener (2002), the entries right of the leftmost suffix zero always form a uniformly distributed random string. 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 string with  $b$  POs is created before the optimum is reached.

The definition of  $f$  implies that no string with more than  $2m/3$  (except  $b$ ) POs is accepted when a string with  $b$  POs has been reached. Hence, it is necessary that at least  $2m/3 - b \geq n^{1/2}/(800 \log^2 n)$  bits flip in one step of the (1+1) EA. The probability of this event is bounded above by

$$\binom{n}{2m/3 - b} \cdot \left(\frac{1}{n}\right)^{2m/3 - b} \leq \frac{1}{(n^{1/2}/(800 \log^2 n))!} = 2^{-\Omega(n^{1/2}/\log n)}$$

according to 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  $1 - 2^{-\Omega(n^{1/2}/\log n)}$ .  $\square$

On the other hand, a polynomial-size population allows an efficient optimization.

**Theorem 8** *Let  $\mu \geq n/\ln(en)$  and  $\mu = \text{poly}(n)$ . With probability at least  $1 - 2^{-\Omega(n^{1/2}/\log n)}$ , the  $(\mu+1)$  EA optimizes  $f$  within  $O(\mu n^{3/2}/\log n)$  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 always the first case of the definition of  $f$  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 until creating an individual with  $\ell$  LSOs (hereinafter called a success) is bounded by  $\mu + 3e\mu n^{1/2} \ln(en)$  according to Corollary 1. (Here,  $\mu \geq n/\ln(en)$  is essential.) By Markov's inequality, the time is bounded by  $t^* := \lceil 7e\mu n^{1/2} \ln(en) \rceil$  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  $7ce\mu n$  steps, is at most  $2^{-cn^{1/2}/\log n}$  (if  $n$  is large enough).

After the first success, at least one individual with  $\ell$  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  $m$ . 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, by the same arguments as in the preceding paragraph, this time is bounded above by  $O(\mu n^{3/2}/\log n)$  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  $s := \lceil 7ce\mu n \rceil$  steps, i. e., until the time where a success occurs with high probability. We do this by a similar approach as in the proof of Theorem 4. According to Lemma 2, no family tree reaches a depth of at least  $21cen$  with probability  $1 - 2^{-\Omega(n)}$ . No initial individual has at least  $7m/12$  POs with probability  $1 - 2^{-\Omega(n)}$ . If we choose, e. g.,  $c \leq 1/768$ , we have  $\lfloor 21cen \rfloor \leq 63n/768 < 64n/768 = n/12$ , i. e., the probability that at most  $\lfloor 21cen \rfloor$  mutations flip a total number of at least  $m/12$  bits is bounded by  $2^{-\Omega(n)}$  according to Chernoff bounds. Altogether, this bounds the probability of more than  $2m/3$  POs within  $s$  steps by  $2^{-\Omega(n)}$ . We remark here that the probability of obtaining more than  $2m/3$  POs before  $\ell$  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 not to be reached before.

The Hamming distance of a locally optimal individual and a globally optimal one is  $h := b - 2m/3 \leq n^{1/2}/(800 \log^2 n) + 1$ . Hence, it is sufficient to flip at most  $n^{1/2}/(800 \log^2 n) + 1$  selected bits in order to create a globally optimal string from a locally optimal one. The corresponding probability is bounded below by

$$\left(\frac{1}{n}\right)^h \left(1 - \frac{1}{n}\right)^{n-h} = 2^{(-\log n)(n^{1/2})/(800 \log^2 n) - O(1)} = 2^{-n^{1/2}/(800 \log n) - O(1)}.$$

The expected waiting time for such a mutation, therefore, is bounded above  $2^{n^{1/2}/(800 \log n) + O(1)}$ . We have calculated above that the probability that an individual with more than  $2m/3$  POs exists before  $\ell$  LSOs are obtained is at most  $2^{-n^{1/2}(1-o(1))/(768 \log n)}$  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  $(\mu+1)$  EA on  $f$  is, therefore, at most  $O(\mu n) + o(1) = O(\mu n)$ .  $\square$

## Conclusions

We have presented a first analysis of the  $(\mu+1)$  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. In contrast to this, we have identified a function where the  $(\mu+1)$  EA outperforms the  $(1+1)$  EA and its multistart variants drastically provided that  $\mu \geq n/\ln(en)$ .

To prove lower bounds, we have developed a new technique. This technique is not only limited to the  $(\mu+1)$  EA. The upper bounds on the depth of family trees are independent of the mutation operator and even of the search space, and the lower bounds derived in the proofs of Theorem 4 and Theorem 5 hold for every selection operator choosing uniformly from individuals of the same fitness. For different selection-for-reproduction mechanisms, the concept of  $1/\mu$ -trees can be adapted. Nevertheless, the most interesting direction seems to be an extension to  $(\mu+\lambda)$  strategies by a combination with the existing theory on the  $(1+\lambda)$  EA.

## Acknowledgements

The author thanks Thomas Jansen and Ingo Wegener for fruitful discussions on the proof techniques.

## A Analyses of Special Random Trees

### A.1 Depth Estimations for Generalized RRTs

Definition 5 contains a model of a stochastic process generating random trees, so-called  $p$ -marked RTs. If  $p_{t,u} = 0$  for all  $t$  and  $u$ , a  $p$ -marked RT is an ordinary RRT as stated in Definition 4. In fact, no marking process  $p$  can prevent the trees from getting at least as deep as ordinary RRTs. To show this, we consider for a  $p$ -marked RT at time  $t$  the measure  $D^*(t, i)$ , namely the random depth of the node that was inserted at time  $i$ ,  $i \in \{0, \dots, t\}$ . Let  $D(t, i)$  denote the corresponding measure for an ordinary RRT. For the sake of completeness, we state results on its expectation and variance. For a proof, see Smythe and Mahmoud (1995).

**Lemma 9** *Let  $H_i$  denote the  $i$ -th Harmonic number. Then  $E(D(t, i)) = H_i$ . Moreover,  $\text{Var}(D(t, i)) \leq H_i$ .*

Lemma 5 relates tail bounds for  $D^*(t, i)$  and  $D(t, i)$ .

**Proof of Lemma 5:** The proof relies on properties of the selection mechanism growing the  $p$ -marked RT. For a  $p$ -marked RT at time  $t$ , let  $C_t$  be the random variable that equals  $j$  if the next inserted node (which is not present until the tree at time  $t + 1$ ) gets the node that was inserted at time  $j$  as its father. For ordinary RRTs, the distribution of the corresponding random variables is uniform over  $\{0, \dots, t\}$ . However, the model of a  $p$ -marked RT implies that  $\text{Prob}(C_t = j) \geq \text{Prob}(C_t = j - 1)$  for all  $j \in \{1, \dots, t\}$ . This follows from the distributions  $p_{t,u}$ , which select the  $u^*$  nodes to be marked uniformly from the set of unmarked nodes. Therefore, the node inserted at time  $i - 1$  has at least the same probability of being marked by time  $t$  as the node inserted at time  $i$ .

The node inserted at time  $i$  has depth at least  $d$  iff its father has depth at least  $d - 1$ . By the law of total probability, we obtain a recurrence,

$$\text{Prob}(D^*(t, i) \geq d) = \sum_{j=0}^{i-1} \text{Prob}(C_{i-1} = j) \cdot \text{Prob}(D^*(t, j) \geq d - 1),$$

and prove the claim of the lemma by induction on  $i$  for arbitrary  $t$  and  $d$ . The base cases  $i = 0$  and  $i = 1$  are trivial. For  $i \geq 2$ , the induction hypothesis allows us to write

$$\text{Prob}(D^*(t, i) \geq d) \geq \sum_{j=0}^{i-1} \text{Prob}(C_{i-1} = j) \cdot \text{Prob}(D(t, j) \geq d - 1).$$

Below, we will prove that the  $\text{Prob}(D(t, j) \geq d - 1)$  are non-decreasing w. r. t.  $j$ . Since the same holds for the  $\text{Prob}(C_{i-1} = j)$  and since the  $\text{Prob}(C_{i-1} = j)$  sum up to 1, the above sum is bounded below by

$$\frac{1}{i} \sum_{j=0}^{i-1} \text{Prob}(D(t, j) \geq d - 1) = \text{Prob}(D(t, i) \geq d),$$

which proves the induction step.

We still have to prove that  $\text{Prob}(D(t, j) \geq d - 1) \geq \text{Prob}(D(t, j - 1) \geq d - 1)$ . The claim follows by a generalization of the following argument. Given an urn with blue and red balls, the probability of choosing a red ball cannot decrease if, after drawing a ball, we add another ball of the color of the drawn ball to the urn. Formally, the right-hand side of the inequality equals the probability  $p_r$  of the event  $E_r$  of choosing a node at depth at least  $d - 2$  from the tree present at time  $j - 2$ , and the left-hand side equals the probability  $p_l$  of the corresponding event  $E_l$  at time  $j - 1$ . According to the rules for ordinary RRTs,  $p_r$  can be written as  $p_r = k/(j - 1)$  for some  $k \in \{0, \dots, j - 1\}$ . If  $E_r$  occurs, the probability of  $E_l$  in the next step is increased, namely to  $(k + 1)/j$ . Otherwise, the probability is at least  $k/j$ . Since  $E_r$  occurs with probability  $p_r$ , the law of total probability yields

$$\begin{aligned} p_l &\geq p_r \cdot \frac{k + 1}{j} + (1 - p_r) \cdot \frac{k}{j} = \frac{k(k + 1)}{j(j - 1)} + \frac{k}{j} - \frac{k^2}{j(j - 1)} \\ &= \frac{k^2 + k + k(j - 1) - k^2}{j(j - 1)} = \frac{k(1 + j - 1)}{j(j - 1)} = \frac{k}{j - 1} = p_r. \end{aligned}$$

This proves the claim and, therefore, the lemma.  $\square$

## A.2 Depth Estimations for $1/\mu$ -trees

We use a combinatorial approach to estimate the depth of  $1/\mu$ -trees, introduced in Definition 2, and study the following measures.

**Definition 6** Let  $L(t, d)$  denote the number of nodes at depth  $d$  in a  $1/\mu$ -tree at time  $t$ . Let  $E(t, d) := \mathbb{E}(L(t, d))$  denote its expectation.

The following basic relations hold.

**Lemma 10** Consider a  $1/\mu$ -tree. Then

$$\begin{aligned} E(t, 0) &= 1 && \text{for } t \geq 0, \\ E(0, d) &= 0 && \text{for } d \geq 1, \\ E(t, d) &\leq \sum_{i=0}^{t-1} \frac{E(i, d-1)}{\mu} && \text{for } t \geq 1, d \geq 1. \end{aligned}$$

**Proof:** The first two equalities follow by the definition of  $p$ -trees. For the third relation, observe that  $L(i, d)$  cannot decrease for increasing values of  $i$ . Moreover,  $L(i+1, d)$  differs from  $L(i, d)$  if and only if a node from depth  $d-1$  is chosen at time  $i$  to be the father of the newly inserted node. If  $L(i, d-1)$  equals  $j$ , this happens with probability at most  $j/\mu$  according to the definition of  $1/\mu$ -trees. Hence, by the law of total probability,

$$E(t, d) \leq \sum_{i=0}^{t-1} \frac{1}{\mu} \sum_{j=1}^{\infty} \text{Prob}(L(i, d-1) = j) \cdot j.$$

The inner sum represents the definition of  $E(i, d-1)$ , proving the inequality.  $\square$

Using these relations, we are able to derive tail bounds on  $D(t)$ , denoting the depth of a  $1/\mu$ -tree at time  $t$ . These bounds are stated in Lemma 2.

**Proof of Lemma 2:** For  $D(t) \geq d$  to occur,  $L(t, d) \geq 1$  is necessary and sufficient. Since, by Markov's inequality,  $\text{Prob}(L(t, d) \geq 1) \leq E(t, d)$ , the lemma follows if we establish the inequality

$$E(t, d) \leq \frac{1}{d!} \left( \frac{t}{\mu} \right)^d \tag{1}$$

for any  $t \geq 0$  and  $d \geq 0$ .

For  $d = 0$ , the statement holds according to Lemma 10. For the induction step from  $d$  to  $d+1$ , we apply the inequality from Lemma 10 and the induction hypothesis to obtain

$$E(t, d+1) \leq \frac{1}{\mu} \sum_{i=0}^{t-1} \frac{1}{d!} \left( \frac{i}{\mu} \right)^d \leq \frac{1}{d! \mu^{d+1}} \int_0^t i^d di = \frac{1}{(d+1)!} \left( \frac{t}{\mu} \right)^{d+1}.$$

The estimation by the integral holds since we can identify its lower sum  $\sum_{i=0}^{t-1} i^d$ .

The second claim of the lemma is an easy consequence of the first one by means of Stirling's formula. Setting  $d := 3t/\mu$ , we obtain

$$\text{Prob}(D(t) \geq d) \leq \frac{1}{d!} \left( \frac{d}{3} \right)^d \leq \frac{1}{\sqrt{2\pi}} \cdot e^d \cdot d^{-d-1/2} \cdot \left( \frac{d}{3} \right)^d = 2^{-\Omega(d)}.$$

$\square$

As a simple corollary of Lemma 2, we obtain that the expected value of  $D(t)$  is bounded by  $O(t/\mu)$ .

## References

- Droste, S., Jansen, T., Tinnefeld, K., and Wegener, I. (2003). A new framework for the valuation of algorithms for black-box optimization. In *Proc. of Foundations of Genetic Algorithms 7 (FOGA 2002)*, 253–270. Morgan Kaufmann.
- Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, **276**, 51–81.
- Feller, W. (1971). *An Introduction to Probability Theory and its Applications*. Wiley, New York.
- Garnier, J., Kallel, L., and Schoenauer, M. (1999). Rigorous hitting times for binary mutations. *Evolutionary Computation*, **7**(2), 173–203.
- Giel, O. (2003). Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proc. of the 2003 Congress on Evolutionary Computation*, 1918–1925. IEEE Press.
- 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.
- He, J. and Yao, X. (2003). Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, **145**(1–2), 59–97.
- Jansen, T. and De Jong, K. (2002). An analysis of the role of offspring population size in EAs. In *Proc. of GECCO 2002, Genetic and Evolutionary Computation Conference*, 238–246.
- 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 *Proc. of GECCO 2001, Genetic and Evolutionary Computation Conference*, 1034–1041.
- Jansen, T. and Wegener, I. (2001c). Real royal road functions – where crossover provably is essential. In *Proc. of GECCO 2001, Genetic and Evolutionary Computation Conference*, 375–382.
- Jansen, T. and Wiegand, R. P. (2003). Exploring the explorative advantage of the cooperative coevolutionary (1+1) EA. In *Proc. of GECCO 2003, Genetic and Evolutionary Computation Conference*, no. 2723 in LNCS, 310–321. Springer.

- Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., and Deb, K. (2002). Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Proc. of PPSN VII*, no. 2439 in LNCS, 44–53. Springer.
- Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms*. Cambridge University Press.
- Pittel, B. (1994). Note on the heights of random recursive trees and random m-ary search trees. *Random Structures and Algorithms*, **5**(2), 337–348.
- Raab, M. and Steger, A. (1998). Balls into bins – a simple and tight analysis. In *Proc. of RANDOM '98*, no. 1518 in LNCS, 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.
- Smythe, R. T. and Mahmoud, H. M. (1995). A survey of recursive trees. *Theory of Probability and Mathematical Statistics*, **51**, 1–27.
- Storch, T. (2004). On the choice of the population size. In *Proc. of GECCO 2004, Genetic and Evolutionary Computation Conference*. To appear.
- Storch, T. and Wegener, I. (2003). Real royal road functions for constant population size. In *Proc. of GECCO 2003, Genetic and Evolutionary Computation Conference*, no. 2724 in LNCS, 1406–1417.
- Wegener, I. (2002). Methods for the analysis of evolutionary algorithms on pseudo-boolean functions. In Sarker, R., Yao, X., and Mohammadian, M. (eds.), *Evolutionary Optimization*, 349–369. Kluwer.
- Wegener, I. and Witt, C. (2003). On the optimization of monotone polynomials by the (1+1) EA and randomized local search. In *Proc. of GECCO 2003, Genetic and Evolutionary Computation Conference*, no. 2723 in LNCS, 622–633.
- Witt, C. (2003). Population size vs. runtime of a simple EA. In *Proc. of the Congress on Evolutionary Computation 2003*, vol. 3, 1996–2003. IEEE Press.