

Theoretical Analysis of Fitness-Proportional Selection: Landscapes and Efficiency

Frank Neumann
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany

Pietro S. Oliveto
University of Birmingham
Birmingham, UK

Carsten Witt
Technical University of Denmark
Kgs. Lyngby, Denmark

ABSTRACT

We investigate theoretically how the fitness landscape influences the optimization process of population-based evolutionary algorithms using fitness-proportional selection. Considering the function ONEMAX, we show that it cannot be optimized in polynomial time with high probability regardless of the population size. This is proved by a generalization of drift analysis. For populations of at most logarithmic size, the negative result transfers to any function with unique optimum. Based on these insights, we investigate the effect of scaling the objective function in combination with a population that is not too small and show that then such algorithms compute optimal solutions for a wide range of problems in expected polynomial time. Finally, relationships with $(1+\lambda)$ EAs and $(1,\lambda)$ EAs are described.

Categories and Subject Descriptors: F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity

General Terms: Theory, Algorithms, Performance

Keywords: Genetic algorithms, Running time analysis, Selection, Theory

1. INTRODUCTION

Evolutionary computation methods have been shown to be very successful for a wide range of optimization problems. Especially, if there are no good problem-specific algorithms available for a new problem at hand, they often produce good results without much development effort. To apply an evolutionary algorithm to a new problem it is just necessary to think about a suitable representation for the solution space, some variation operators, and a selection method which determines the individuals that produce offspring and/or constitute the population of the next generation.

The aim of this paper is to contribute to the theoretical understanding of selection methods within an evolutionary

algorithm. We focus on fitness-proportional selection which has originally been proposed for the use in genetic algorithms [4]. Recently, it has been shown that fitness-proportional selection does not work on linear functions with non-zero weights when the population size is 1 [3]. The reason for this is that the search process is not guided towards optimal solutions. Since the mechanism was proposed to work with population based algorithms, a natural question that arises when considering this work is to ask whether larger populations can help to make such an approach efficient. We consider a simple population- and mutation-based genetic algorithm and examine whether this algorithm achieves an optimal solution efficiently for ONEMAX and a general class of functions if the population is at least polynomially bounded. Our results show that larger populations still cannot help to solve even the simple ONEMAX problem efficiently. For the first time, widely recognized problematic issues of the fitness-proportional selection mechanism [2] are rigorously proved. In fact it is generally accepted that as the process evolves, the selection pressure becomes lower and lower until random search occurs.

To achieve this result we make use of several probabilistic tools. One way of analyzing evolutionary algorithms with respect to their runtime behavior is by carrying out a drift analysis [7]. In the past, such analyses mainly focused on one single individual in the population. In the context of populations, the analysis of probabilistic family trees is known [10], however, this technique is mainly suited for uniform selection and does not deliver super-polynomial lower bounds. In this paper, we extend the method of drift analysis to a whole population by measuring the quality of a population using a potential function. It turns out that a trivial potential function that looks at only the best individual from a population is not sufficient for a general lower bound. Our potential is defined more carefully in order to capture the status of the whole population in a single random variable. Therefore, the techniques developed in this paper may be of independent interest for analyzing population-based evolutionary algorithms on more complicated problems in the future.

After having presented these negative results, we consider how to make evolutionary algorithms based on fitness-proportional selection successful. In many experimental studies it is reported that fitness-proportional selection is very sensitive with respect to the differences of the different function values that a function may attain. A popular method to compensate the weakness of fitness-proportional selection is to use fitness-scaling methods which increase the proba-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'09, July 8–12, 2009, Montréal, Québec, Canada.
Copyright 2009 ACM 978-1-60558-325-9/09/07 ...\$5.00.

bility of having good search points in the population. See [2] for a review of popular scaling functions and methods.

We examine how the fitness landscape changes by using a simple scaling mechanism based on the original function value and the population size. We rigorously prove that such an algorithm using fitness-proportional selection obtains an optimal solution for the ONEMAX function quickly. Afterwards, we generalize these ideas and show that the algorithm is also efficient on each function where a direct application of the method of fitness-based partitions leads to a polynomial upper bound for the well-known (1+1) EA. Finally, we look more deeply into the selection pressure. For a high selection pressure, we point out situations where simple evolutionary algorithms using fitness-proportional selection considerably outperform the elitist version of the algorithm. This is based on relationships with $(1+\lambda)$ EAs and $(1,\lambda)$ EAs.

The rest of the paper is organized as follows. In Section 2, we introduce the algorithm that is subject to our investigations. Section 3 introduces the drift analysis for populations and proves the lower bound on ONEMAX for populations of almost arbitrary size as well as for small populations in the case of functions with unique optimum. In Section 4, we show how to change the landscape by a simple scaling mechanism such that the algorithm becomes successful on a wide range of problems, and in Section 5 we discuss the impact of selection pressure for the optimization process. We finish with concluding remarks.

2. SIMPLE GENETIC ALGORITHM

Our goal is to examine the use of fitness-proportional selection within a simple population based algorithm. We consider a simple genetic algorithm based on this selection method. The algorithm does not use recombination operators because we want to focus on the impact of fitness-proportional selection on populations. Hence we keep the algorithm as simple as possible.

As a convention, all populations are assumed as multisets, i. e., may contain duplicates of search points.

ALGORITHM 1 (SIMPLE GENETIC ALGORITHM (SGA)).

1. Create a parent population P consisting of μ randomly chosen individuals.
2. $C := \emptyset$.
3. While $|C| < \mu$ do
 - Select one individual x of P according to fitness-proportional selection.
 - Create an offspring x' of x by flipping each bit of x independently with probability $1/n$.
 - $C := C \cup \{x'\}$.
4. Set $P := C$ and go to 2.

The algorithm starts with a parent population P consisting of μ individuals that are chosen uniformly at random for the considered search space $\{0,1\}^n$. In each iteration the new parent population is created by choosing μ times one individual x of $P = \{x_1, \dots, x_\mu\}$ according to fitness-proportional selection. Using fitness proportional selection, each individual x is selected for mutation with probability

$f(x)/\sum_{i=1}^{\mu} f(x_i)$ in a single selection step. Afterwards, the selected individual is mutated by flipping each of its bits independently with probability $1/n$. The μ offspring created in this way constitute the population of the next generation.

To measure the efficiency of our algorithm, we consider the number of fitness evaluations until our algorithm has sampled an optimal search point for the first time. In order to show that the algorithm cannot solve a specific problem, we show that the runtime is exponential with probability exponentially close to one. To this end, in Section 3 we show that the number of generations is exponential with overwhelming probability. It is straightforward that the number of fitness evaluations will be even higher with the same probability. Showing that the algorithm is efficient we prove that, in expectation, a polynomial number of fitness evaluations are sufficient to obtain an optimal search point. The expected number of fitness evaluations to reach an optimal search point will be also called the “expected optimization time” when proving polynomial upper bounds of the SGA.

3. LOWER BOUNDS

We first consider the SGA defined as Algorithm 1 on the test problem $\text{ONEMAX}(x) := |x| = x_1 + \dots + x_n$. The aim is to demonstrate that the population drifts away from the optimum of the function. To this end, a potential function capturing the status of the whole population is defined. This is achieved by exponentially increasing the significance of the levels of the ONEMAX function, which results in a potential function of exponential range. The simplified drift theorem, which was recently published by Oliveto and Witt [8], will finally be applied to the logarithm of the potential. However, we need the following slight modification with a parameter $r(N)$ that is allowed to grow with the state space and a weaker second condition where $\text{Prob}(\Delta_t(i) \leq -j)$ is used instead of $\text{Prob}(\Delta_t(i) = -j)$. The theorem is proven along the lines of the proof by Oliveto and Witt [8].

THEOREM 1 (SIMPLIFIED DRIFT THEOREM). *Let X_t , $t \geq 0$, be the random variables describing a Markov process over a finite state space $S \subseteq [0, N]$ and denote $\Delta_t(i) := (X_{t+1} - X_t \mid X_t = i)$ for $i \in S$ and $t \geq 0$. Suppose there exist an interval $[a, b]$ in the state space, two constants $\delta, \varepsilon > 0$ and, possibly depending on $\ell := b - a$, a function $r(\ell)$ satisfying $1 \leq r(\ell) = o(\ell/\log(\ell))$ such that for all $t \geq 0$ the following two conditions hold:*

1. $E(\Delta_t(i)) \geq \varepsilon$ for $a < i < b$,
2. $\text{Prob}(\Delta_t(i) \leq -j) \leq \frac{r(\ell)}{(1+\delta)^j}$ for $i > a$ and $j \in \mathbb{N}_0$.

Then there is a constant $c^ > 0$ such that for $T^* := \min\{t \geq 0: X_t \leq a \mid X_0 \geq b\}$ it holds $\text{Prob}(T^* \leq 2^{c^* \ell / r(\ell)}) = 2^{-\Omega(\ell / r(\ell))}$.*

For the definition of our potential function, the following notion is handy.

DEFINITION 2. *Given a population P , we define its spectrum $(k_0, \dots, k_n) \in \{0, \dots, \mu\}^n$ as the vector consisting of the frequencies of the $n + 1$ possible ONEMAX values in the population.*

Obviously, any spectrum (k_0, \dots, k_n) satisfies $k_0 + \dots + k_n = \mu$. Since we analyze ONEMAX, only the spectra of populations will be relevant for the optimization process.

DEFINITION 3. The potential $\phi_c(P)$ of a population is defined by $\sum_{i=0}^n k_i \cdot c^i$, where (k_0, \dots, k_n) is the spectrum of P and $c > 1$ is a constant.

The idea behind the exponential scaling is to bias the potential towards good individuals and to make sure that the potential reflects the “top” of the population sufficiently.

In the following, we will drop the index c for the sake of readability. The following lemma shows that the potential captures the “important” individuals of the population. Its straightforward proof is omitted from this submission.

LEMMA 4. If $\phi(P) \leq c^r$ the maximum fitness of individuals in P is bounded from above by r . If $\phi(P) \geq c^r$ the maximum fitness is at least $r - \log_c \mu$.

Our aim is to show: if $\phi(P) > c^{\alpha n}$ for a sufficiently large constant $\alpha < 1$, e. g., $\alpha = 0.995$ then the ϕ -value drifts away from the optimum. We first concentrate on single individuals. Since the drift theorem will be applied to $\log(\phi)$ in the end, we consider the relative drift of the population, i. e., the expected factor by which the potential is decreased. Let x be an individual with i one-bits and let x' be the result of a mutation applied to x . The expected change in potential w. r. t. x is given by $E(R_i) := E(c^{|x'|})/c^i = E(c^{|x'| - i})$ as the potential of x equals c^i . A drift away from the optimum is equivalent to $E(R_i) < 1$. In order to compensate the bias introduced by selection, we show that even $E(R_i) < 1/2$ is satisfied for appropriate choices of c and α . Note that the following proof follows a similar structure as was necessary for bounding the moment-generating function of the absolute drift, see, e. g., Giel and Wegener [1].

LEMMA 5. Choosing $c = 8$ as the base for $\phi_c(\cdot)$ and $\alpha = 0.995$, it holds for all $n \geq 5$ and $i \geq \alpha n$ that $E(R_i) \leq 0.48$.

PROOF. We prepare ourselves by bounding the relevant transition probabilities. Let $p(i, j)$ be the probability of creating a string with j ones from a string with i ones. We start with the case $j > i$. Since it is necessary to flip at least $j - i$ zero-bits for an increase of size $j - i$, we obtain

$$j \geq i \Rightarrow p(i, j) \leq \binom{n-i}{n}^{j-i} \leq (1-\alpha)^{j-i},$$

where the last inequality follows from $i \geq \alpha n$.

For the non-increasing steps, i. e., $j \leq i$, it is necessary to flip either exactly $i - j$ out of i one-bits and no other bit or at least one zero-bit. Hence, $j \leq i$ implies

$$\begin{aligned} p(i, j) &\leq \binom{i}{i-j} \left(\frac{1}{n}\right)^{i-j} \left(1 - \frac{1}{n}\right)^{n-(i-j)} + (1-\alpha) \\ &\leq \frac{1}{(i-j)!} \left(1 - \frac{1}{n}\right)^{n-(i-j)} + (1-\alpha), \end{aligned}$$

where the last inequality follows from $i \leq n$. Moreover, since $n \geq 5$ is assumed,

$$\begin{aligned} p(i, j) &\leq \frac{1}{(i-j)!} \left(1 - \frac{1}{n}\right)^n \left(\frac{n}{n-1}\right)^{i-j} + (1-\alpha) \\ &\leq \frac{(5/4)^{i-j}}{(i-j)!} \cdot e^{-1} + (1-\alpha) \end{aligned}$$

for $j \leq i$. Note that the estimation holds also for $i = j$, where it breaks down to $p(i, i) \leq e^{-1} + (1-\alpha)$.

Our aim is to prove

$$E(R_i) = \frac{\sum_{j=0}^n p(i, j) \cdot c^j}{c^i} = \sum_{j=0}^n p(i, j) \cdot c^{j-i} < 0.48.$$

We concentrate on the last sum and treat its terms for $j > i$ and $j \leq i$ differently according to the above estimations. In the first case, we have

$$\begin{aligned} \sum_{j=i+1}^n p(i, j) \cdot c^{j-i} &\leq \sum_{j=i+1}^n ((1-\alpha) \cdot c)^{j-i} \\ &\leq \sum_{k=1}^{\infty} ((1-\alpha) \cdot c)^k = \frac{(1-\alpha) \cdot c}{1 - (1-\alpha) \cdot c}. \end{aligned}$$

For the second case, we obtain

$$\begin{aligned} \sum_{j=0}^i p(i, j) \cdot c^{j-i} &\leq \sum_{j=0}^i \left(\left(\frac{5}{4c}\right)^{i-j} \cdot \frac{e^{-1}}{(i-j)!} + \frac{1-\alpha}{c^{i-j}} \right) \\ &\leq \sum_{k=0}^{\infty} \left(\left(\frac{5}{4c}\right)^k \cdot \frac{e^{-1}}{k!} + \frac{1-\alpha}{c^k} \right) = e^{5/(4c)-1} + \frac{1-\alpha}{1-1/c}. \end{aligned}$$

Altogether, the sum is at most

$$\frac{(1-\alpha) \cdot c}{1 - (1-\alpha) \cdot c} + e^{5/(4c)-1} + \frac{1-\alpha}{1-1/c},$$

which becomes less than 0.48 by, e. g., choosing $\alpha = 0.995$ and $c = 8$. \square

In the following, the choice $c = 8$ is fixed.

Now for the selection. We fix a population P with spectrum (k_0, \dots, k_n) and denote by P' its offspring population. Let the random variables S_0, \dots, S_n – called *selection frequencies* – denote how many individuals are chosen for mutation from the respective levels of the ONEMAX function. Clearly, $k_i = 0$ implies $S_i = 0$. We will prove that the best and worst fitnesses from a population stay in a ratio of at most 2 for an exponential number of steps. The reason is that the maximum fitness is n and the minimum fitness after initialization is close to $n/2$. Under this assumption, the expected S_i -values are not too different from the frequencies in the parent population.

LEMMA 6. With probability $1 - 2^{-n^{1-o(1)}}$, best and worst fitness for all populations up to time $t^* = 2^{n^{1-\gamma(n)}}$, $\gamma(n)$ an appropriate function satisfying $\gamma(n) = o(1)$, are in ratio at most 2. Under this assumption, $E(S_i) \leq 2k_i$ for $0 \leq i \leq n$.

PROOF. Let $\ell = 0.499n$ and $u = 0.997n$. We will show that all fitness values (i. e., ONEMAX values) observed up to time $t^* = 2^{n^{1-\gamma(n)}}$ are in the interval $[\ell, u]$ with probability $1 - 2^{-n^{1-o(1)}}$ if $\gamma(n)$ goes to zero slowly enough. For fitness-proportional selection, this immediately implies $E(S_i) \leq 2k_i$ for $0 \leq i \leq n$. The rest of the proof is divided into the obvious parts regarding the lower and the upper bound.

For the lower bound, we first note that due to Chernoff bounds and the assumption $\mu = \text{poly}(n)$, all individuals from the initial population have fitness at least ℓ with probability $1 - 2^{-\Omega(n)}$. Observe that fitness-proportional selection prefers individuals with larger number of ones. Let $p(i, j)$ denote the probability of creating an individual with $j < i$ ones

from one with i ones. It is well known that this probability is monotone decreasing w. r. t. i (Wegener and Witt [9]), i. e., $p(i+1, j) < p(i, j)$ for all $i \geq j$. Hence, if we replace fitness-proportional selection by uniform selection, the probability of creating an individual with less than ℓ ones in a generation only increases. So the problem breaks down to showing that Algorithm 1 with uniform selection does not produce individuals with less than ℓ ones within t^* steps.

The latter statement follows by fixing, in the modified algorithm, an arbitrary individual x from the random population at time t^* . The individual has a lineage of length t^* which, as the fitness function is not used, corresponds to a purely random walk on the search space $\{0, 1\}^n$. Oliveto and Witt [8] consider this scenario implicitly in their application of the new drift theorem w. r. t. the needle-in-a-haystack function. They show that a random walk of length even 2^{c^*n} , c^* sufficiently small, with probability $1 - 2^{-\Omega(n)}$ does not create individuals with more than $n - \ell$ ones or, symmetrically, more than $n - \ell$ zeroes. The last perspective bounds the probability of less than ℓ ones for all individuals in a single lineage of length t^* . Since there are at most $\mu = \text{poly}(n)$ individuals, the probability of once creating an individual with less than ℓ ones is bounded by $2^{-\Omega(n)}$ also for all individuals. This completes the consideration of the lower bounds.

The upper bound that is needed here is actually the result that we prove in the upcoming Theorem 8: The maximum number of ones in all individuals up to time t^* is bounded from above by $u = 0.997n$ with probability $1 - 2^{-n^{1-o(1)}}$. We can use this result already at this place without creating a loop in the final proof. For each step, the probability that the upper bound u or the lower bound ℓ on the fitness does not hold is $2^{-n^{1-o(1)}}$. By inductively summing up failure probabilities, the probability that a bound does not hold after t steps is at most $t2^{-n^{1-o(1)}}$. This is still $2^{-n^{1-o(1)}}$ for all $t \leq t^* = 2^{n^{1-\gamma(n)}}$ if $\gamma(n)$ goes to zero slowly enough. \square

Of course, the S_i are not independent because always $S_0 + \dots + S_n = \mu$. However, this will not be a problem since we can apply the linearity of expectation in the following. We show that within the interval $[8^{0.996n}, \mu 8^n]$, the potential really drifts towards smaller values. Again the relative change is estimated since the drift theorem will finally be applied to the logarithm of ϕ .

LEMMA 7 (MAIN LEMMA).

1. If $\phi(P) \geq 8^{0.996n}$ then $E(\phi(P')/\phi(P)) = 1 - \Omega(1)$.
2. $\text{Prob}(\phi(P')/\phi(P) \geq 8^k) \leq n^{O(1/\log \log n)} \cdot 2^{-k}$ for arbitrary P and all $k \geq 0$.

PROOF. To prove the first statement of the lemma, we only consider the individuals with at least $\alpha n = 0.995n$ one-bits. Since $\phi(P) \geq 8^{0.996n}$ is assumed and ϕ is exponential in the number of one-bits, the relative contribution of the remaining individuals to the potential is $2^{-\Omega(n)}$. This holds also for the next generation population P' since flipping $(0.996n - 0.995n)/2 = n/2000$ bits in a step has probability $2^{-\Omega(n)}$. So, pessimistically assuming $\phi(P')/\phi(P) = 1 - \Omega(1)$, the relative contribution of the individuals with at most $0.9955n$ one-bits to $\phi(P')$ is still $2^{-\Omega(n)}$.

We fix an arbitrary outcome of the selection frequencies and obtain for the expected potential of P' that

$$E(\phi(P') \mid S_0, \dots, S_n) = \sum_{i=0}^n S_i \cdot (8^i \cdot E(R_i)),$$

where $8^i \cdot E(R_i)$ equals the expected potential of an offspring with parent from level i , i. e., with i one-bits. By the law of total expectation and the linearity of expectation,

$$E(\phi(P')) = E(E(\phi(P') \mid S_0, \dots, S_n)) = \sum_{i=0}^n E(S_i) \cdot 8^i \cdot E(R_i),$$

which is at most

$$\sum_{i=0}^n 2k_i \cdot 8^i \cdot E(R_i)$$

due to Lemma 6.

Taking into account that only individuals with at least αn one-bits matter, we have

$$E(\phi(P')) \leq \sum_{i=\alpha n}^n 2k_i \cdot 8^i \cdot E(R_i) + 2^{-\Omega(n)} \cdot \phi(P).$$

Finally applying Lemma 5, we obtain

$$\begin{aligned} E(\phi(P')) &\leq \sum_{i=\alpha n}^n 2k_i \cdot 8^i \cdot 0.48 + 2^{-\Omega(n)} \cdot \phi(P) \\ &\leq \sum_{i=\alpha n}^n 0.96 \cdot k_i \cdot 8^i + 2^{-\Omega(n)} = (1 - \Omega(1)) \cdot \phi(P), \end{aligned}$$

which proves the first part of the lemma.

To prove the second part, we assume arbitrary $\phi(P)$ and estimate $\phi(P')/\phi(P)$ from above using the following pessimistic assumptions: If an individual is chosen for mutation, each flipped bit leads to an increase of the individual's potential. Then the relative increase of the individual's potential by the j -th mutation, $1 \leq j \leq \mu$, of a generation is bounded from above by a factor 8^{X_j} , where X_j denotes the random number of flipping bits in the j -th mutation. We bound this factor for every mutation from above by 8^{X^*} , where $X^* := \max\{X_1, \dots, X_\mu\}$ is the μ -th order statistic, i. e., the maximum of these random variables. Since $\text{Prob}(X_i \geq k) \leq \binom{n}{k} (1/n)^k \leq (n^k/k!) \cdot (1/n)^k$, an application of the union bound yields $\text{Prob}(X^* \geq k) \leq \mu/k!$.

To bound the increase of the whole potential, we have to take into account that individuals can be chosen more than once. As before, let (k_0, \dots, k_n) denote the spectrum of population P and let S_i , $0 \leq i \leq n$, denote the random number of individuals chosen from level i . Under our pessimistic assumptions, the potential of the offspring population is at most

$$\begin{aligned} \phi(P') &\leq \sum_{i=0}^n S_i \cdot 8^{i+X^*} = \sum_{i=0}^n k_i \cdot \frac{S_i}{k_i} \cdot 8^{i+X^*} \\ &\leq 8^{X^* + \max_i \{S_i/k_i\}} \cdot \phi(P). \end{aligned}$$

Hence, it remains to bound $S^* := \max\{S_i/k_i\}$. We already know that S_i is dominated by a binomial distribution $B(\mu, 2k_i/\mu)$ with parameters μ and $2k_i/\mu$, implying that S_i is dominated by $B(\mu, 2/\mu)$. Therefore, $\text{Prob}(S_i/k_i \geq k) \leq \binom{\mu}{k} (2/\mu)^k \leq 2^k/k!$ for any i similarly as above. Finally, using the union bound, we have $\text{Prob}(S^* \geq k) \leq n2^k/k!$.

Combining the previous insights, we consider the random variable $8^{X^*+S^*}$, which is an upper bound on the relative increase of the potential. For $8^{X^*+S^*} \geq 8^k$, or, equivalently, $X^*+S^* \geq k$ to happen, $X^* \geq k/2$ or $S^* \geq k/2$ is necessary. Hence,

$$\begin{aligned} \text{Prob}(X^* + S^* \geq k) &\leq 2 \cdot \left(\frac{\mu}{(k/2)!} + \frac{n2^{k/2}}{(k/2)!} \right) \\ &\leq \frac{e^{k/2} \cdot 2^{k/2+1} \cdot (\mu + n)}{(k/2)^{k/2}} \leq \frac{2^{2k+3}(\mu + n)}{k^{k/2}} =: g(k), \end{aligned}$$

which in turn will be bounded from above by

$$\min \left\{ 1, \left(\frac{1}{2} \right)^{k-8 \log(\mu+n)/\log \log(\mu+n)} \right\}$$

if $\beta := \mu + n$ is chosen large enough. The last estimation holds since $k \geq 8 \log \beta / \log \log \beta$ implies

$$\begin{aligned} k^{k/2} &= k^{k/4} \cdot k^{k/4} \geq \left(\frac{\log \beta}{\log \log \beta} \right)^{\frac{2 \log \beta}{\log \log \beta}} \cdot k^{k/4} \\ &\geq 2^{(\log \log \beta - \log \log \log \beta) \cdot \frac{2 \log \beta}{\log \log \beta} + (\log k) \cdot \frac{k}{4}} \\ &\geq 2^{(3/2) \log \beta + 3k + 4} \geq \frac{2^{2k+4}(\mu + n)}{2^{-k}}, \end{aligned}$$

(note that all inequalities assume growing β), which means $g(k) = 2^{2k+3}(\mu + n)/k^{k/2} \leq 1/2$ for all $k \geq 8 \log \beta / \log \log \beta$ and β large enough. Moreover, obviously $g(k+1)/g(k) \leq 1/2$ for $k \geq 64$, implying $g(k) \leq (1/2)^{k-8 \log \beta / \log \log \beta}$ for appropriate k and β . For smaller values of k , we use the trivial bound 1 on probabilities. Now,

$$\begin{aligned} \left(\frac{1}{2} \right)^{k-8 \log \beta / \log \log \beta} &\leq \left(\frac{1}{2} \right)^k \cdot (\mu + n)^{8/\log \log \beta} \\ &= \left(\frac{1}{2} \right)^k n^{O(1/\log \log n)} \end{aligned}$$

since $\mu = n^{O(1)}$. This completes the proof of the second part. \square

Now the following lower bound is merely an application of the drift theorem w. r. t. the main lemma.

THEOREM 8. *Let $\mu = \text{poly}(n)$. Then the SGA needs on ONEMAX with probability $1 - 2^{-n^{1-o(1)}}$ at least $2^{n^{1-o(1)}}$ generations to create a search point with more than 99.7% one-bits.*

PROOF. As motivated above, we capture the potential of a population by $\psi(P) := \log_8(\mu 8^n) - \log_8(\phi(P))$ and identify the random ψ -value at time t with the random variables X_t from Theorem 1. We have $0 \leq \psi(P) \leq n + \log_8(\mu) = n + o(n)$ for all populations P and ψ takes only values from a finite set. Moreover, the following two conditions hold:

1. $E(\psi(P') - \psi(P)) = E(-\log_8(\phi(P')/\phi(P)))$ is at least $-\log_8(E(\phi(P')/\phi(P)))$ due to Jensen's inequality. According to Lemma 7, this is $\Omega(1)$ for all P such that $\psi(P) \leq 0.004n + \log_8(\mu)$.
2. Also according to Lemma 7, $\text{Prob}(\psi(P') \leq \psi(P) - k) = \text{Prob}(\log_8(\phi(P')/\phi(P)) \geq k) \leq n^{O(1/\log \log n)} \cdot 2^{-k}$.

Choosing $N := n + \log_8(\mu)$, $r(n)$ as any function of order $n^{O(1/\log \log n)}$, $a := 0.003n$ and $b := 0.004n$, Theorem 1 is in force. We obtain

$$\begin{aligned} \text{Prob}(T^* \leq 2^{e^*(b-a)/r(N)}) \\ &= \text{Prob}(T^* \leq 2^{e^* \cdot (n/1000) \cdot n^{-O(1/\log \log n)}}) \\ &= \text{Prob}(T^* \leq 2^{n^{1-o(1)}}) = 2^{-n^{1-o(1)}}. \end{aligned}$$

Additionally using Chernoff bounds to bound $\psi(P) \geq n/3$ for the initial population, we obtain a total failure probability of $2^{-n^{1-o(1)}}$. \square

REMARK 9. *Theorem 8 holds for any function where maximum and minimum fitness value are in ratio at most 2.*

It does not seem obvious to generalize the previous result to the case of arbitrary linear functions, i. e., functions of the kind $f(x) = w_0 + w_1x_1 + \dots + w_nx_n$, $w_i \in \mathbb{N}^+$ and $0 \leq i \leq n$. However, a lower bound can be shown for populations of size $\mu \leq (\log n)/4$ even for functions with a unique optimum, which, w. l. o. g., is assumed to be the all-ones string. The idea is that even a $(1, \mu)$ EA, which is a degenerate case of the SGA for extreme functions (see Sections 4 and 5 for details), would not be able to optimize such functions efficiently. We consider a much simpler potential function now, namely $\phi(P) := \max\{|x_i| \mid i = 1, \dots, \mu\}$ for a population $P = \{x_1, \dots, x_\mu\}$. In other words, the potential measures the maximum number of one-bits (i. e., correctly set bits) for the individuals in the population. Moreover, in the analysis we pessimistically assume that only individuals with $\phi(P)$ one-bits are chosen for mutation; for this assumption the above-mentioned monotonicity of the mutation operator is used. Although our new, simple and straightforward choice for $\phi(P)$ does not seem to work for the proof of Theorem 8, it is sufficient now due to the restricted population size and a strong drift away from the optimum, i. e., the all-ones string. The following proof is a significant simplification of a corresponding statement for the $(1, \mu)$ EA (see [5]).

Our aim is to bound the drift of the potential, i. e., $\Delta(P) := \phi(P') - \phi(P)$, where P' is the offspring population of population P . Since the potential only depends on the number of ones, it suffices to study $\Delta_i := \Delta(\phi(P') - \phi(P^*))$, where P^* is a population consisting of individuals with i ones only. We show a negative drift for Δ_i given that i is large enough.

LEMMA 10. *Given $\mu \leq (\log n)/4$ and $i \geq n - \sqrt{n}$, it holds $E(\Delta_i \mid \Delta_i \neq 0) = -1 + o(1)$ and $\text{Prob}(\Delta_i \geq k \mid \Delta_i \neq 0) = O((n^{1/4} \log n)/k!)$.*

PROOF. In a single mutation, the probability of not decreasing the number of one-bits is at most $(1 - \frac{1}{n})^{n-\sqrt{n}} + \frac{\sqrt{n}}{n} \leq \frac{2}{5}$ for large enough n since it is necessary to either maintain all one-bits or to flip at least one zero-bit. Hence, with probability at least

$$\left(1 - \frac{2}{5}\right)^\mu \geq \left(\frac{1}{2}\right)^{(\log n)/4} = \frac{1}{n^{1/4}}$$

the offspring population contains only individuals with less than i ones, which means $\text{Prob}(\Delta_i < 0) \geq n^{-1/4}$. On the other hand, the probability that a zero-bit flips at all within $\mu = O(\log n)$ trials, a necessary condition for $\Delta_i > 0$, is at most $\mu/\sqrt{n} = o(n^{-1/4})$. This means

1. $\text{Prob}(\Delta_i > 0) = o(\text{Prob}(\Delta_i < 0))$
2. $\text{Prob}(\Delta_i \neq 0) = \text{Prob}(\Delta_i < 0) + \text{Prob}(\Delta_i > 0) = (1 - o(1)) \cdot (\text{Prob}(\Delta_i < 0))$.

The unconditional drift is bounded from above according to

$$E(\Delta_i) \leq -\text{Prob}(\Delta_i < 0) + \text{Prob}(\Delta_i > 0) \cdot \left(1 + \frac{\sqrt{n}}{n}\right),$$

where we pessimistically assume a decrease of only -1 in the case $\Delta_i < 0$ and, in the other case, allow for an additional number of \sqrt{n} zero-bits to be flipped independently of the zero-bits that flip necessarily for $\Delta_i > 0$ to hold. Hence, $E(\Delta_i) = -(1 - o(1)) \cdot \text{Prob}(\Delta_i < 0)$. For the drift under the condition $\Delta_i \neq 0$ we obtain

$$\begin{aligned} E(\Delta_i \mid \Delta_i \neq 0) &= \frac{E(\Delta_i)}{\text{Prob}(\Delta_i \neq 0)} \\ &= \frac{-(1 - o(1)) \cdot \text{Prob}(\Delta_i < 0)}{(1 + o(1)) \cdot \text{Prob}(\Delta_i < 0)} = -1 + o(1), \end{aligned}$$

which proves the first statement of the lemma. The second statement follows similarly as in Lemma 7 by considering the maximum number of flipping bits within $\mu = O(\log n)$ trials along with $\text{Prob}(\Delta_i \neq 0) \geq n^{-1/4}$. \square

An application of Theorem 1 yields:

THEOREM 11. *If $\mu \leq (\log n)/4$, the SGA needs with probability $1 - 2^{-\Omega(n^{3/4}/\log n)}$ at least $2^{\Omega(n^{3/4}/\log n)}$ generations to optimize a function with unique optimum.*

PROOF. Counting only the steps where $\Delta_i \neq 0$, we establish the conditions of Theorem 1 by choosing $r(N)$ appropriately s. t. $r(N) = O(n^{1/4} \log n)$, $\delta := 1$, $\varepsilon := 1/2$, $a := \sqrt{n}/2$, $b := \sqrt{n}$. Furthermore, given population P_t at time t , we set $X_t := n - \phi(P_t)$. Since the initial number of ones is less than $n - \sqrt{n}$ with probability $1 - 2^{-\Omega(n)}$, the total failure probability is $2^{-\Omega(n^{3/4}/\log n)}$. \square

4. SCALING

The first result from the previous section applies to ONEMAX, which is a function with a small polynomial range, namely $\{0, \dots, n\}$. We now consider whether exponential scaling helps to make the algorithm successful. Scaling the objective function and turning it into a fitness function that can distinguish between search points in a better way can only help if the population is large enough. In the extreme case $\mu = 1$ the algorithm performs only random search.

To obtain an efficient evolutionary algorithm using fitness-proportional selection, we require two properties that have to be fulfilled. On the one hand, the population size has to be large enough such that the algorithm does not perform random search. On the other hand, the scaling function should ensure that at least some fraction of the individuals with the highest fitness value are chosen for mutation.

Let g be the original fitness function. Then, for each input x , the scaled function f is

$$f(x) = (\mu - 1)^{g(x)}.$$

Let x and y be two individuals with different fitness values such that $g(x) > g(y)$. Then, the following property of the scaled function holds:

$$f(x) \geq (\mu - 1) \cdot f(y).$$

Using this property, we can lower bound the probability of choosing one individual with highest fitness value in a selection step.

LEMMA 12. *Let f be a scaled fitness function of g such that $f(x) = (\mu - 1)^{g(x)}$ for each input x . The probability of the SGA choosing an individual with the highest fitness value for mutation in a single selection step when optimizing f is at least $1/2$.*

PROOF. Let $f(x^*)$ be the largest fitness value currently present in the population P , $x^* \in P$ an individual attaining this value and r be the number of individuals in P having fitness $f(x^*)$. Furthermore, let $f(y)$ be the highest possible fitness value such that $f(x^*) > f(y)$ and consider the individuals x_i with $1 \leq i \leq \mu$ sorted increasingly according to fitness. We calculate the probability of choosing an individual in the population of fitness $f(x^*)$. We get

$$\begin{aligned} &\frac{r \cdot f(x^*)}{r \cdot f(x^*) + \sum_{i=1}^{\mu-r} f(x_i)} \\ &\geq \frac{f(x^*)}{f(x^*) + \sum_{i=1}^{\mu-1} f(x_i)} \geq \frac{f(x^*)}{f(x^*) + (\mu - 1)f(y)}. \end{aligned}$$

Using $f(x^*) \geq (\mu - 1) \cdot f(y)$, the last expression is bounded from below by $\frac{(\mu-1) \cdot f(x^*)}{2(\mu-1)f(y)} = 1/2$. \square

Let ONEMAX_s be the function that is obtained by scaling the fitness values of ONEMAX in the described way. Hence

$$\text{ONEMAX}_s(x) = (\mu - 1)^{\text{ONEMAX}(x)}.$$

Note that the value of a search point with respect to ONEMAX_s depends on the value of ONEMAX and the population size used in the algorithm. Using the scaled function ONEMAX_s instead of ONEMAX in the evolutionary algorithm, we show that the SGA turns into an efficient algorithm for the problem if the population size is at least logarithmic in the problem size. Note that Theorem 11 yields an exponential lower bound for functions with unique optimum if the population size is at most logarithmic. In general, logarithmic population size is necessary to make a $(1, \mu)$ EA simulate a $(1 + \mu)$ EA (see [5]), which provides for the elitism that the proof techniques from this section are based upon.

THEOREM 13. *Choosing $\mu \geq c \log n$, c an appropriate constant, the expected time until the algorithm has produced an optimal search point for the function ONEMAX_s is $O(\mu n)$.*

PROOF. From Lemma 12 it follows that an individual with the highest fitness value in the population is chosen with a probability of at least $1/2$. The probability of producing a solution with at least the same fitness from a solution x is at least $(1 - 1/n)^n \geq 1/4$ for $n \geq 2$.

Therefore, the probability of not having produced an individual of fitness at least $f(x^*)$ in an iteration of the algorithm is upper bounded by

$$\left(1 - \frac{1}{8}\right)^\mu = \alpha^\mu,$$

where $\alpha = 7/8$ is a constant. Hence, with probability at least $1 - \alpha^\mu$ the fitness of the best individual does not decrease in a single iteration.

Let $\text{ONEMAX}(x^*) = n - k$, i.e., k bits can be flipped to achieve an improvement. Let E_1 be the event that the current best individual is selected for mutation and an improvement towards the optimum is achieved. The probability of E_1 is estimated in a similar way as in the analysis of the $(1+\lambda)$ EA presented by Jansen, De Jong and Wegener [6]. Since the best individual is selected for mutation with probability at least $1/2$, the probability that event E_1 happens is at least $k/(2en)$ for each selected individual. Since μ individuals are selected for mutation in each generation, the probability that event E_1 does not happen for any of them is less than $(1 - k/(2en))^\mu$. Hence at each step the probability an improvement occurs is bounded from below by

$$\begin{aligned} 1 - \left(1 - \frac{k}{2en}\right)^\mu &\geq 1 - e^{-k\mu/(2en)} \\ &\geq 1 - \frac{1}{1 + k\mu/(2en)} = \frac{k\mu}{2en + k\mu}. \end{aligned}$$

This implies that the expected number of iterations under the condition that the fitness of the best individual in the population does not decrease is upper bounded by

$$\sum_{k=1}^n \frac{2en + k\mu}{k\mu} = \sum_{k=1}^n \left(1 + \frac{2en}{k\mu}\right) \leq n + \frac{c'n \log n}{\mu}$$

with c' an appropriate constant. Since there are μ function evaluations in each iteration and μ is at least $c \log n$ we get an upper bound of $c'\mu n$ if no failure has happened before.

Consider a typical run of $2c'\mu n$ steps. Assuming that the value of the best individual in the population has not decreased, the optimal solution is produced with probability $p_1 \geq 1/2$ using Markov's inequality. With probability at least $p_2 = (1 - \alpha^\mu)^{2c'\mu n}$ the fitness of the best individual in the population has not decreased during the considered phase. Since $\mu \geq c \cdot \log n$ and c is an appropriate constant, we get $p_2 = \Omega(1)$. Hence, with probability $p_1 \cdot p_2 = \Omega(1)$ the optimal solution has been obtained during that phase. We have not made any assumption on the population at the beginning of the considered phase. This implies that the expected optimization time is upper bounded by $O((c'\mu n)/(p_1 \cdot p_2)) = O(\mu n)$. \square

It is worth to point out that the result presented in Theorem 13 shows that once the objective function is scaled, the SGA has the same asymptotic runtime for ONEMAX as that of the $(\mu+1)$ EA [10] and of the $(1+\mu)$ EA [6] as long as the population size of the algorithms is at least logarithmic in the problem size. For smaller population sizes the two latter algorithms have the same performance as that of the $(1+1)$ EA for ONEMAX.

The ideas for proving that scaling helps to make the algorithm effective for the ONEMAX-problem use the following property. With a good probability, the algorithm makes during a phase of fixed length always progress towards an optimal solution once the function is scaled. Considering problems where the fitness values can be grouped into classes such that the probability of producing search points of a higher fitness class with a good probability leads to the method of fitness-based partitions which is very popular

for analyzing simple evolutionary algorithms such as the $(1+1)$ EA [7] and the $(1+\mu)$ EA [6]. In the following, we want to relate runtime bounds achievable by the method of fitness-based partitions and the $(1+1)$ EA to the runtime of the simple genetic algorithm using the scaling mechanism introduced in this section. Again we will need populations of at least logarithmic size since this allows the algorithm to create duplicates of the best individual with probability $\Omega(1)$.

We consider a problem where the function values can be partitioned into different classes A_1, \dots, A_m such that for two search point x and y where $g(x) \in A_i$ and $g(y) \in A_j$, $i < j$, $g(x) < g(y)$ holds. Additionally, we assume that A_m is not empty and contains only the optimal function value. Using the method of fitness-based partitions for evolutionary algorithms using elitism, a lower bound p_i has to be found on the probability of producing from any solution x with $g(x) \in A_i$ a solution x' with $g(x') \in A_j$, where $j > i$. Having computed these probabilities the expected runtime of the whole algorithm to achieve an optimal search point can then be bounded by $T^* = \sum_{i=1}^{m-1} p_i^{-1}$.

THEOREM 14. *Let g be a function that can be partitioned into m fitness levels A_i , $1 \leq i \leq m$. Let f be the scaled function obtained from g such that $f(x) = (\mu - 1)^{g(x)}$. Furthermore, let p_i be a lower bound on the probability of producing from a solution $x \in A_i$ a solution $x' \in A_j$ where $j > i$ and set $T^* := \sum_{i=1}^{m-1} p_i^{-1} = \text{poly}(n)$. Choosing $\mu \geq c \log n$, c an appropriate constant, the expected optimization time of the SGA on f is $O(m \cdot \mu + T^*)$.*

PROOF. Let $x^* \in P$ be a solution with the highest fitness value in the population. If $f(x^*) \in A_i$, the probability of producing from x^* a solution x' with $f(x') \in A_j$, where $j > i$, is at least p_i . We already know that the fitness of the best individual in the population does not decrease with probability $1 - \alpha^\mu$ (see proof of Theorem 13). We consider the probability that one of the best individuals is chosen for mutation and that an improvement is achieved. This is done similarly as in the proof of Theorem 13 and in [6]: In each selection step a best individual is selected for mutation with probability at least $1/2$ (Lemma 12). Hence, the probability that this happens in one specific iteration is at least

$$\begin{aligned} 1 - (1 - p_i/2)^\mu &\geq 1 - e^{-p_i \cdot \mu/2} \\ &\geq 1 - \frac{1}{1 + p_i \cdot \mu/2} = \frac{p_i \mu}{2 + p_i \mu}. \end{aligned}$$

Considering the population size μ and summing up over the different values of i , we get

$$\mu \cdot \sum_{i=1}^{m-1} \frac{2 + p_i \mu}{p_i \mu} = \mu \cdot \sum_{i=1}^{m-1} \left(1 + \frac{2}{p_i \mu}\right) = \mu \cdot m + 2 \cdot T^*.$$

In a phase of $2(\mu m + 2 \cdot T^*)$ steps, an optimal solution is produced with probability $p_1 \geq 1/2$. With probability $p_2 = (1 - \alpha^\mu)^{2(\mu m + 2 \cdot T^*)}$ the fitness of the best individual has not decreased during that phase. We assume $T^* = \text{poly}(n)$. Hence, choosing $\mu \geq c \cdot \log n$, c an appropriate constant, we get $p_2 = \Omega(1)$. Therefore, the expected optimization time is upper bounded by $O((m\mu + 2 \cdot T^*)/(p_1 \cdot p_2)) = O(m\mu + T^*)$. \square

5. SELECTION PRESSURE

In the previous section, we have shown that scaling helps to make the algorithm efficient. The results presented in Theorems 13 and 14 are based on increasing the selection pressure by using the scaled function instead of the original one. We have shown that the selection pressure may be increased by changing the fitness landscape such that the algorithm can “hill-climb” up ONEMAX or improve from one fitness-based partition to another.

In this section, we want to discuss scaling mechanisms and their impact on the selection pressure a bit further. As the selection pressure is increased the SGA resembles very closely a $(1, \mu)$ EA (we still use μ instead of λ for the offspring population size). The $(1, \mu)$ EA has a parent population of one single individual. In each generation it creates μ new individuals by mutating the parent and then selects the best one for the next generation. Hence, the behaviour of the SGA will approach that of the $(1, \mu)$ EA as the probability of selecting the best individual gets closer to 1 (i. e., the selection pressure is increased). Consider the following scaling function with higher selection pressure than that used in the previous section: $f(x) = (\mu - 1)^{2g(x)}$.

Then for any two individuals x and y such that $g(x) > g(y)$ it holds that $f(x) \geq (\mu - 1)^2 \cdot f(y)$. By using similar arguments to those of Lemma 12 we get that the probability of selecting the best current individual x^* is at least

$$\begin{aligned} \frac{f(x^*)}{f(x^*) + (\mu - 1)f(y)} &\geq \frac{(\mu - 1)^2 \cdot f(y)}{(\mu - 1)^2 f(y) + (\mu - 1)f(y)} \\ &= \frac{\mu - 1}{\mu - 1 + 1} = 1 - \frac{1}{\mu} \end{aligned}$$

If μ is not too small this gives a probability of $1 - o(1)$ of selecting the best individual for mutation and there is a constant probability that all the individuals for the next generation are obtained by mutating the current best.

Once the selection pressure is high enough the SGA will work exactly as the $(1, \mu)$ EA. A runtime analysis comparison between the $(1, \mu)$ EA and the $(1 + \mu)$ EA has already been performed [5]. In the paper a function called CLIFF is presented for which the $(1, \mu)$ EA is efficient for logarithmic values of μ while the $(1 + \mu)$ EA is inefficient for any μ value.

The CLIFF function is the following:

$$\text{CLIFF}(x) = \begin{cases} \text{ONEMAX}(x) - \lfloor n/3 \rfloor & \text{if } |x| \geq n - \lfloor n/3 \rfloor \\ \text{ONEMAX}(x) & \text{otherwise.} \end{cases}$$

As proved in [5] for the $(1 + \mu)$ EA, an elitist selection algorithm attempting to optimize this function will get trapped on top of the cliff located at $\lfloor 2/3n \rfloor$ one-bits with exponentially high probability. Then the expected optimization time is exponential because $n/3$ bits need to be flipped for the optimum to be found. It is worth to point out here that the described proof idea also holds for a $(\mu + \lambda)$ EA with parent and offspring populations up to polynomial size.

On the other hand, if μ is of logarithmic size, then the $(1, \mu)$ EA has a good enough probability of quickly falling down the cliff and then climbing up to the optimum efficiently without hardly ever going back to the cliff. As discussed previously if the selection pressure of the SGA is high enough the results obtained in [5] for the $(1, \mu)$ EA can be extended to the SGA considered in this paper. Hence, a family of functions has been presented where a simple GA

using fitness-proportional selection outperforms elitist selection algorithms.

6. CONCLUSIONS

Selection methods play a crucial role in making an evolutionary algorithm successful. We have examined the use of fitness-proportional selection which has originally been proposed to be used in genetic algorithms for population-based evolutionary algorithms. Generalizing the drift analysis technique to capture the drift of the whole population, we have shown that the SGA with polynomial-size populations is not able to optimize even ONEMAX. Moreover, up to logarithmic population size it provably fails to optimize any function with unique optimum. Later on, we have pointed out how to make evolutionary algorithms using fitness-proportional selection successful by changing the fitness landscape due to some scaling function and discussed its impact on the selection pressure.

References

- [1] O. Giel and I. Wegener. Evolutionary algorithms and the maximum matching problem. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS '03)*, pages 415–426. Springer, 2003.
- [2] D. E. Goldberg. *Genetic Algorithms for Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [3] E. Happ, D. Johannsen, C. Klein, and F. Neumann. Rigorous analyses of fitness-proportional selection for optimizing linear functions. In *Proc. of GECCO '08*, pages 953–960. ACM Press, 2008.
- [4] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1st edition, 1975.
- [5] J. Jägersküpper and T. Storch. When the plus strategy outperforms the comma strategy—and when not. In *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007)*, pages 25–32. IEEE Press, 2007.
- [6] T. Jansen, K. A. De Jong, and I. Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13:413–440, 2005.
- [7] P. S. Oliveto, J. He, and X. Yao. Computational complexity analysis of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3):281–293, 2007.
- [8] P. S. Oliveto and C. Witt. Simplified drift analysis for proving lower bounds in evolutionary computation. In *Proc. of Parallel Problem Solving from Nature (PPSN X)*, volume 5199 of *LNCS*, pages 82–91. Springer, 2008.
- [9] I. Wegener and C. Witt. On the optimization of monotone polynomials by simple randomized search heuristics. *Combinatorics, Probability and Computing*, 14:225–247, 2005.
- [10] C. Witt. Runtime analysis of the $(\mu + 1)$ EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.