TECHNISCHE UNIVERSITÄT DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

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

Theoretical Analysis of Diversity Mechanisms for Global Exploration

Tobias Friedrich, Pietro S. Oliveto,
Dirk Sudholt, Carsten Witt

No. CI-237/08

Theoretical Analysis of Diversity Mechanisms
for Global Exploration

Tobias Friedrich

Algorithms and Complexity Group

Max-Planck-Institut für Informatik

Saarbrücken, Germany

Pietro S. Oliveto[1]

School of Computer Science

University of Birmingham

Birmingham, United Kingdom

Dirk Sudholt[2]

Fakultät für Informatik, LS 2

Technische Universität Dortmund

Dortmund, Germany

Carsten Witt[2]

Fakultät für Informatik, LS 2

Technische Universität Dortmund

Dortmund, Germany

Abstract

Maintaining diversity is important for the performance of evolutionary algorithms. Diversity mechanisms can enhance global exploration of the search space and enable crossover to find dissimilar individuals for recombination. We focus on the global exploration capabilities of mutation-based algorithms. Using a simple bimodal test function and rigorous runtime analyses, we compare well-known diversity mechanisms like deterministic crowding, fitness sharing, and others with a plain algorithm without diversification. We show that diversification is necessary for global exploration, but not all mechanisms succeed in finding both optima efficiently.

1   Introduction

The term *diversity* indicates dissimilarities of individuals in evolutionary computation and is considered an important property. A population-based evolutionary algorithm without diversity mechanism is vulnerable to the so-called *hitchhiking effect* where the best individual takes over the whole population before the fitness landscape can be explored properly. When the population becomes redundant, the algorithm basically reduces to a trajectory-based algorithm while still suffering from high computational effort and space requirements for the whole population.

Diversity mechanisms can help the optimization in two ways. On one hand, a diverse population is able to deal with multimodal functions and can explore several hills in the fitness landscape simultaneously. Diversity mechanisms can therefore support global exploration and help to locate several local and global optima. In particular, this behavior is welcome in dynamic environments as the algorithm is more robust w. r. t. changes of the fitness landscape. Moreover, the algorithm can offer several good solutions to the user, a feature desirable in multiobjective optimization. On the other hand, a diverse population gives higher chances to find dissimilar individuals and to create good offspring by recombining different "building blocks". Diversity mechanisms can thus enhance the performance of crossover.

Up to now, the use of diversity mechanisms has been assessed mostly by means of empirical investigations (e. g., [1, 17]). Theoretical runtime analyses involving diversity mechanisms mostly use these mechanisms to enhance the performance of crossover. Jansen and Wegener [9] presented the first proof that crossover can make a difference between polynomial and exponential expected optimization times. They used a very simple diversity mechanism that only shows up as a tie-breaking rule: when there are several individuals with minimal fitness among parents and offspring, the algorithm removes those individuals with a maximal number of genotype duplicates. Nevertheless, this mechanism makes the individuals spread on a certain fitness level such that crossover is able to find suitable parents to recombine. Storch and Wegener [14] presented a similar result for populations of constant size. They used a stronger mechanism that prevents duplicates from entering the population, regardless of their fitness.

The first theoretical runtime analysis considering niching methods was presented by Fischer and Wegener [2] for a fitness function derived from a generalized Ising model on ring graphs. The authors compare the well-known (1+1) EA with a (2+2) GA with fitness sharing. Fitness sharing [10] derates the real fitness of an individual x by a measure related to the similarity of x to all individuals in the population, hence encouraging the algorithm to decrease similarity in the population. Fischer and Wegener prove that their genetic algorithm outperforms the (1+1) EA by a polynomial factor. Sudholt [15] extended this study for the Ising model on binary trees, where the performance gap between GAs and EAs is even larger. While a broad class of $(\mu+\lambda)$ EAs has exponential expected optimization time, a (2+2) GA with fitness sharing finds a global optimum in expected polynomial time.

In all these studies diversity is used to assist crossover. Contrarily, Friedrich, Hebbinghaus, and Neumann [3] focused on diversity mechanisms, as a means to enhance the global exploration of EAs without crossover. Using rigorous runtime analyses, the authors compare a mechanism avoiding genotype duplicates with a strategy avoiding phenotype duplicates to spread individuals on different fitness levels. It is shown for artificial functions that both mechanisms can outperform one another drastically.

Friedrich *et al.* [3] were the first to focus on the use of diversity mechanisms for global exploration, with respect to rigorous runtime analyses. However, their test functions are clearly tailored towards one particular diversity mechanism.

We want to obtain a broader perspective including a broader range of diversity mechanisms. Moreover, we aim at results for a more realistic setting that captures difficulties also arising in practice. Therefore, we compare several well-known diversity mechanisms on the simplest bimodal function that may also appear as part of a real-world problem. If a diversity mechanism is not helpful on a very simple landscape, then we would not hope it would be effective for more complicated functions. Firstly, we rigorously prove that diversity mechanisms are necessary for a simple bimodal function since populations of almost linear size without diversification fail to find both peaks, with high probability. Then we analyze common diversity mechanisms and show that not all of them are effective for avoiding premature convergence even for such a simple landscape. As a result, we hope to get a more objective and more general impression of the capabilities and limitations of common diversity mechanisms.

In the remainder, we first present our bimodal test function in Section 2. Negative results for a plain $(\mu+1)$ EA in Section 3 show that diversification is needed to counteract the hitchhiking effect. In Sections 4 and 5 we investigate the strategies previously analyzed in [3] to avoid genotype or phenotype duplicates, resp. Section 6 deals with the well-known deterministic crowding strategy [10] where offspring directly compete with their associated parents. Fitness sharing, which turns out to be the strongest mechanism, is analyzed in Section 7. We present our conclusions in Section 8.

2 A Simple Bimodal Function

We consider a simple bimodal function called TwoMax that has already been investigated in the context of genetic algorithms by Pelikan and Goldberg [13] and Van Hoyweghen, Goldberg, and Naudts [5]. The function TwoMax is essentially the maximum of OneMax and ZeroMax. Local optima are solutions 0^n and 1^n where the number of zeros or the number of ones, respectively, is maximized. Hence, TwoMax can be seen as a multimodal equivalent of OneMax. The fitness landscape consists of two hills with symmetric slopes, i.e., an unbiased random search heuristic cannot tell in advance which hill is more promising. In contrast to [5,13] we modify the function slightly such that only one hill contains the global optimum, while the other one leads to a local optimum. This is done by simply adding an additional fitness value for 1^n, turning it into a unique global optimum.

Let $|x|_1$ denote the number of 1-bits in x and $|x|_0$ the number of 0-bits in x. Then for $x = x_1 x_2 \ldots x_n$

$$\text{TwoMax}(x) \ := \ \max\{|x|_0, |x|_1\} + \prod_{i=1}^{n} x_i.$$

Figure 1 shows a sketch of TwoMax. Among all search points with more than $n/2$ 1-bits, the fitness increases with the number of ones. Among all search points with less than $n/2$ 1-bits, the fitness increases with the number of zeros.
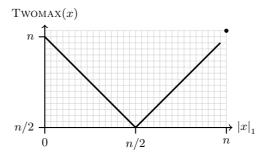
Figure 1: Sketch of TwoMax. The dot indicates the global optimum.

We refer to these sets as *branches* and the algorithms as climbing these two branches of TwoMax.

The TwoMax function may appear in well-known combinatorial optimization problems. For example, the Vertex Cover bipartite graph analyzed in [11] consists of two branches, one leading to a local optimum and the other to the minimum cover. In fact similar proof techniques as those used in this paper have also been applied in the vertex cover analysis of the (μ+1) EA for the bipartite graph. Another function with a similar structure is the Mincut instance analyzed in [16].

Recombining individuals from different branches is likely to yield offspring of very low fitness. Since we are focusing on diversity for global exploration, we refrain from the use of recombination and only consider mutation-based EAs.

3 No Diversity Mechanism

In order to obtain a fair comparison of different diversity mechanisms, we keep one algorithm fixed as much as possible. The basic algorithm, the following (μ+1) EA, has already been investigated by Witt [18].

Alg. 1: (μ+1) EA
Let $t := 0$.
Initialize P_0 with μ individuals chosen uniformly at random.
Repeat
> *Choose $x \in P_t$ uniformly at random.*
> *Create y by flipping each bit in x with probability $1/n$.*
> *Choose $z \in P_t$ with minimal fitness.*
> *If $f(y) \geq f(z)$ then $P_{t+1} = P_t \setminus \{z\} \cup \{y\}$*
> > *else $P_{t+1} = P_t$.*
> *Let $t = t + 1$.*

The (μ+1) EA uses random parent selection and elitist selection for survival. As parents are chosen randomly, the selection pressure is quite low. Nevertheless, the (μ+1) EA is not able to maintain individuals on both branches for

a long time. We now show that if μ is not too large, the individuals on one branch typically get extinct before the top of the branch is reached. Thus, the global optimum is found only with probability close to $1/2$ and the expected optimization time is very large.

Theorem 1. *The (μ+1) EA with no diversity mechanism and $\mu = o(n/\log n)$ does not reach in polynomial time on* TwoMax *the global optimum with probability $1/2 - o(1)$. Its expected optimization time is $\Omega(n^n)$.*

Proof. The probability that during initialization either 0^n or 1^n is created is bounded by $\mu \cdot 2^{-n+1}$, hence exponentially small. In the following, we assume that such an atypical initialization does not happen as this assumption only introduces an error probability of $o(1)$.

Consider the algorithm at a point of time t^* where either 0^n or 1^n is created for the first time. Due to symmetry, the local optimum 0^n is created with probability $1/2$. We now show that with high probability this local optimum takes over the population before the global optimum 1^n is created. Let i be the number of copies of 0^n in the population. From the perspective of extinction, a good event G_i is to increase this number from i to $i + 1$. We have

$$\mathrm{P}(G_i) \geq \frac{i}{\mu} \cdot \left(1 - \frac{1}{n}\right)^n \geq \frac{i}{4\mu}$$

since it suffices to select one out of i copies and to create another copy of 0^n. On the other hand, the bad event B_i is to create 1^n in one generation. This probability is maximized if all $\mu - i$ remaining individuals contain $n - 1$ ones:

$$\mathrm{P}(B_i) \leq \frac{\mu - i}{\mu} \cdot \frac{1}{n} < \frac{1}{n}.$$

Together, the probability that the good event G_i happens before the bad event B_i is

$$\mathrm{P}(G_i \mid G_i \cup B_i) \geq \frac{\mathrm{P}(G_i)}{\mathrm{P}(G_i) + \mathrm{P}(B_i)}. \tag{1}$$

Plugging in bounds for $\mathrm{P}(B_i)$ and $\mathrm{P}(G_i)$ yields the bound

$$\frac{i/(4\mu)}{i/(4\mu) + 1/n} = 1 - \frac{1/n}{i/(4\mu) + 1/n} \geq 1 - \frac{4\mu}{in}.$$

The probability that 0^n takes over the population before the global optimum is reached is therefore bounded by

$$\prod_{i=1}^{\mu} \mathrm{P}(G_i \mid G_i \cup B_i) \geq \prod_{i=1}^{\mu} \left(1 - \frac{4\mu}{in}\right).$$

Using $4\mu/n \leq 1/2$ and $1 - x \geq e^{-2x}$ for $x \leq 1/2$, we obtain

$$\prod_{i=1}^{\mu}\left(1 - \frac{4\mu}{in}\right) \geq \prod_{i=1}^{\mu}\exp\left(-\frac{8\mu}{in}\right) = \exp\left(-\frac{8\mu}{n}\cdot\sum_{i=1}^{\mu}\frac{1}{i}\right)$$
$$\geq \exp(-O((\mu\log\mu)/n))$$
$$\geq 1 - O((\mu\log\mu)/n) = 1 - o(1).$$

Adding up all error probabilities, the first claim follows. If the population consists of copies of 0^n, mutation has to flip all n bits to reach the global optimum. This event has probability n^{-n} and the conditional expected optimization time is n^n. As this situation occurs with probability $1/2 - o(1)$, the unconditional expected optimization time is $\Omega(n^n)$. □

4 No Genotype Duplicates

It has become clear that diversity mechanisms are needed in order to optimize even a simple function such as TwoMax. The simplest way to enforce diversity within the population is not to allow genotype duplicates. The following algorithm has been defined and analyzed by Storch and Wegener [14]. It prevents identical copies from entering the population as a natural way of ensuring diversity. We will, however, show that this mechanism is not powerful enough to explore both branches of TwoMax.

Alg. 2: (μ+1) EA with genotype diversity
Let $t := 0$.
Initialize P_0 with μ individuals chosen uniformly at random.
Repeat
    Choose $x \in P_t$ uniformly at random.
    Create y by flipping each bit in x with probability $1/n$.
    If $y \notin P_t$ then
        Choose $z \in P_t$ with minimal fitness.
        If $f(y) \geq f(z)$ then $P_{t+1} = P_t \setminus \{z\} \cup \{y\}$
                   else $P_{t+1} = P_t$.
    Let $t = t + 1$.

We prove that if the population is not too large, the algorithm can be easily trapped in a local optimum.

Theorem 2. *The (μ+1) EA with genotype diversity and $\mu = o(n^{1/2})$ does not reach in polynomial time on* TwoMax *the global optimum with probability $1/2 - o(1)$. Its expected optimization time is $\Omega(n^{n-1})$.*

Proof. We use a similar way of reasoning as in the proof of Theorem 1. With probability $1/2 - o(1)$, 0^n is the first local or global optimum created at time t^*. Call x *good* (from the perspective of extinction) if $|x|_1 \leq 1$ and *bad* if $|x|_1 \geq n-1$. At time t^* the number of good individuals is at least 1. In the worst case (again

6

from the perspective of extinction) the population at time t^* consists of 0^n and $\mu - 1$ bad individuals with $n - 1$ ones. Provided that the $(\mu+1)$ EA does not flip $n - 2$ bits at once, we now argue that the number of good individuals is monotone unless the unique 0-bit in a bad individual is flipped.

Due to the assumptions on the population only offspring with fitness at least $n - 1$ are accepted, i.e., only good or bad offspring. In order to create a bad offspring, the unique 0-bit has to be flipped since otherwise a clone or an individual with worse fitness is obtained. Hence the number of good individuals can only decrease if a bad individual is chosen as parent and its unique 0-bit is flipped. If there are i good individuals, we denote this event by B_i and have $\mathrm{P}(B_i) = (\mu - i)/\mu \cdot 1/n$.

On the other hand, the number of good individuals is increased from i to $i + 1$ if a good offspring is created and a bad individual is removed from the population. We denote this event by G_i. A good offspring is created with probability at least $1/(3\mu)$ for the following reasons. The point 0^n is selected with probability at least $1/\mu$ and then there are $n - (i - 1) \geq (e/3) \cdot n$ 1-bit-mutations creating good offspring that are not yet contained in the population (provided n is large enough). Along with the fact that a specific 1-bit-mutation has probability $1/n \cdot (1 - 1/n)^{n-1} \geq 1/(en)$, the bound $1/(3\mu)$ follows. After creating such a good offspring, the algorithm removes an individual with fitness $n - 1$ uniformly at random. As there are $i - 1$ good individuals with this fitness and $\mu - i$ bad individuals, the probability to remove a bad individual equals $(\mu - i)/(\mu - 1) \geq (\mu - i)/\mu$. Together,

$$\mathrm{P}(G_i) \; \geq \; \frac{1}{3\mu} \cdot \frac{\mu - i}{\mu} \; = \; \frac{\mu - i}{3\mu^2}.$$

Along with inequality (1), the probability that G_i happens before B_i is at least

$$\frac{\frac{\mu - i}{3\mu^2}}{\frac{\mu - i}{3\mu^2} + \frac{\mu - i}{\mu n}} \; = \; \frac{1}{1 + 3\mu/n} \; = \; 1 - \frac{3\mu/n}{1 + 3\mu/n} \; \geq \; 1 - \frac{3\mu}{n}.$$

The probability that the number of good individuals increases to μ before the global optimum is reached is

$$\prod_{i=1}^{\mu} \mathrm{P}(G_i \mid G_i \cup B_i) \; \geq \; \left(1 - \frac{3\mu}{n}\right)^{\mu} \; \geq \; 1 - \frac{3\mu^2}{n} \; = \; 1 - o(1).$$

Adding up all error probabilities, the first claim follows.

The claim on the expected optimization time follows as the probability to create a global optimum, provided the population contains only search point with at most one 1-bit, is at most $n^{-(n-1)}$. $\qquad\square$

5   No Phenotype Duplicates

Avoiding genotype duplicates does not help much to optimize TWOMAX as individuals from one branch are still allowed to spread on a certain fitness level and

take over the population. A more restrictive mechanism is to avoid phenotype duplicates, i. e., multiple individuals with the same fitness. Such a mechanism has been defined and analyzed by Friedrich *et al.* [3] for plateaus of constant fitness.

The following (μ+1) EA with phenotype diversity avoids that multiple individuals with the same fitness are stored in the population. If at some time step t, a new individual x is created with the same fitness value as a pre-existing one $y \in P_t$, then x replaces y.

Alg. 3: (μ+1) EA with phenotype diversity

Let $t := 0$.
Initialize P_0 with μ individuals chosen uniformly at random.
Repeat
    Choose $x \in P_t$ uniformly at random.
    Create y by flipping each bit in x with probability $1/n$.
    If there exists $z \in P_t$ such that $f(y) = f(z)$
    then $P_{t+1} = P_t \setminus \{z\} \cup \{y\}$,
    else
        Choose $z \in P_t$ with minimal fitness.
        If $f(y) \geq f(z)$ then $P_{t+1} = P_t \setminus \{z\} \cup \{y\}$
                else $P_{t+1} = P_t$.
    Let $t = t + 1$.

From the analysis of Friedrich *et al.* [3] it can be derived that if the population size μ is a constant, then the runtime on a simple plateau is exponential in the problem size n. Only if μ is very close to n the expected runtime is polynomial. In particular, if $\mu = n$ then the same upper bound as that of the (1+1) EA for plateaus of constant fitness [8] can be obtained (i. e., $O(n^3)$). Probably, the phenotype diversity mechanism is not effective on simple plateaus. In the following, by analyzing the mechanism on TWOMAX, we show how also on a simple bimodal landscape, phenotype diversity does not help the (μ+1) EA to avoid getting trapped on a local optimum.

Since the diversity mechanism does not accept multiple individuals with the same fitness value, it does not make sense to have a population size that is greater than the number of different fitness function values. For the TWOMAX function this number is roughly $n/2$.

The following theorem proves that if the population is not too large, then with high probability the individuals climbing one of the two branches will be extinguished before any of them reach the top. Since the two branches of the TWOMAX function are symmetric, this also implies that the global optimum will not be found in polynomial time with probability $1/2 - o(1)$.

Theorem 3. *The (μ+1) EA with phenotype diversity and $\mu = o(n^{1/2})$ does not reach in polynomial time on TWOMAX the global optimum with probability $1/2 - o(1)$. Its expected optimization time is $n^{\Omega(n)}$.*

Proof. We assume that the very unlikely situation that the global and the local optimum are created during initialization does not happen. As proved in previous theorems such probability is exponentially small. We are interested in the first generation T^* where the local optimum or the global one is created. The probability that the local optimum is created equals $1/2$ due to the symmetry of the TwoMax function.

Let individuals with $i < n/2$ zero-bits be called x_i and ones with $i < n/2$ one-bits be called y_i. Because of the phenotype diversity mechanism there may be only one x_i or one y_i in the population at the same time for $i \geq 1$. Consider a situation where the local optimum $y_0 = 0^n$ is reached before the global optimum and note that due to the phenotype diversity y_0 will remain in the population if $\mu \geq 2$. We will show that with probability $1 - o(1)$ the population $\{y_0, \ldots, y_{\mu-1}\}$ is created before the global optimum is found.

Then, if y_i for $i \leq \mu - 1$ is chosen as parent, the minimum number of bits that have to flip in order to create some x_i for $i \leq \mu - 1$ is $n - 2n^{1/2}$. This probability is at most

$$\binom{n}{2n^{1/2}} \left(\frac{1}{n}\right)^{n - 2n^{1/2}} \leq \frac{1}{(n - 2n^{1/2})!} = n^{-\Omega(n)}.$$

The probability that this happens at least once in n^{cn} steps is still exponentially small if $c > 0$ is a small enough constant. This proves the claim on the expected optimization time.

It remains to be proved that the population $\{y_0, \ldots, y_{\mu-1}\}$ is reached with probability $1 - o(1)$. Let m be the smallest i such that $y_{i+1} \notin P_t$. Note that $m \leq \mu - 1$. From the perspective of extinction, the following worst case population

$$P_t = \{y_0, \ldots, y_m, \ x_{m+1}, \ldots, x_{\mu-1}\}$$

maximizes the probability of reaching the global optimum. We are optimistically assuming that the x-individuals are always one fitness level behind each other. It may happen that the next step creates y_{m+1}. Then x_{m+1} is removed from the population and m increases by 1. Such a step is called a *good step*. Note that m is not changed if y_{m+i} is created for $i \geq 2$. For these steps, we pessimistically assume that the current population is still the worst case population. On the other hand, if x_i for $i \leq m$ is created, then y_i is removed and m decreases to $i - 1$. This is referred to as a *bad step*. All other steps lead us back to our worst case population.

We show that with high probability m increases to $\mu - 1$ before a bad step happens. If x_i is chosen as parent, a necessary condition for a bad step is that $i - m$ out of i 0-bits flip. Hence the probability of a bad step is at most

$$\sum_{i=m+1}^{\mu} \frac{1}{\mu} \binom{i}{i-m} \left(\frac{1}{n}\right)^{i-m} = \frac{1}{\mu} \sum_{i=1}^{\mu-m} \binom{i+m}{i} \left(\frac{1}{n}\right)^i$$

$$\leq \frac{1}{\mu} \sum_{i=1}^{\infty} \left(\frac{\mu}{n}\right)^i \leq \frac{1}{\mu} \cdot \frac{\mu}{n - \mu} = \frac{1}{n - \mu} \leq \frac{2}{n},$$

using $\mu = o(n^{1/2}) \leq (1 - e/3)n < n/2$ if n large enough. Hence, the probability of a good step is at least

$$\frac{1}{\mu} \cdot \frac{n-m}{en} \geq \frac{1}{\mu} \cdot \frac{n-\mu}{en} \geq \frac{1}{3\mu}$$

since it suffices to select y_m and to flip exactly one out of $n - m$ bits. So, the probability that a good step happens before a bad step is at least

$$\frac{1/(3\mu)}{1/(3\mu) + 2/n} = 1 - \frac{1}{n/(6\mu) + 1} \geq 1 - \frac{3\mu}{n}.$$

The probability to increase m to $\mu - 1$ by good steps before a bad step happens is bounded by

$$\left(1 - \frac{3\mu}{n}\right)^{\mu} \geq 1 - \frac{3\mu^2}{n} = 1 - o(1),$$

which concludes the proof. $\qquad\square$

Also if the population is larger than $o(n^{1/2})$ with probability $1 - o(1)$ the time to reach the global optimum is exponential in the problem size if the local optimum is found first. Drift analysis will be used to prove the above statement. This proof method was initially introduced by Hajek [6] and afterwards was extended to the run-time analysis of EAs by He and Yao [7]. A general description of the technique can be found in [12]. The following Drift theorem is a useful extension of the general proof method for proving run-time lower bounds that hold with exponentially high probabilities. Its first application can be found in [4].

Theorem 4 (Drift Theorem). *Let $X_0, X_1, X_2, \ldots$ be a Markov process over a set of states S, and $g\colon S \to \mathbf{R}_0^+$ a function that assigns to every state a non-negative real number. Pick two real numbers $a(l)$ and $b(l)$ which depend on a parameter $l \in \mathbf{R}^+$ such that $0 < a(l) < b(l)$ holds and let the random variable T denote the earliest point in time $t \geq 0$ where $g(X_t) \leq a(l)$ holds.*

If there are constants $\lambda > 0$ and $D \geq 1$ and a polynomial $p(l)$ taking only positive values, for which the following four conditions hold

1. $P(g(X_0) \geq b(l)) = 1$

2. $b(l) - a(l) = \Omega(l)$

3. $\forall t \geq 0\colon E[e^{-\lambda(g(X_{t+1}) - g(X_t))} \mid X_t, a(l) < g(X_t) < b(l))] \leq 1 - 1/p(l)]$

4. $\forall t \geq 0\colon E[e^{-\lambda(g(X_{t+1}) - b(l))} \mid X_t, b(l) \leq g(X_t)] \leq D,$

then for all time bounds $B \geq 0$, the following upper bound on probability holds for random variable T

$$P(T \leq B) \leq e^{\lambda(a(l) - b(l))} \cdot B \cdot D \cdot p(l).$$

The following theorem proves exponential expected time for the $(\mu+1)$ EA with phenotype diversity on TWOMAX for any value of μ. We point out again that, given the diversity mechanism, it does not make sense to have a population size of $\mu > n/2$.

Theorem 5. *The $(\mu+1)$ EA with phenotype diversity does not reach in polynomial time on* TWOMAX *the global optimum with probability $1/2 - o(1)$. Its expected optimization time is $e^{\Omega(\sqrt[3]{n})}$.*

Proof. We pick up the line of thought from the proof of Theorem 3. With probability $1 - o(1)$, $\sqrt[3]{n} + 1 = o(n^{1/2})$ individuals $y_0, y_1 \ldots y_{\sqrt[3]{n}}$ are created before the global optimum is found. From the perspective of extinction the worst case population is $P_t = \{y_0, \ldots, y_{\sqrt[3]{n}}, \ x_{\sqrt[3]{n}+1}, \ldots, x_\mu\}$. In the following we will use the drift theorem to prove that then with overwhelming probability it will take exponential time before the global optimum is created. Pessimistically arguing with worst-case populations as in Theorem 3 implies that the process we analyze drifts away from the optimum more slowly compared to the actual one.

Let X_t be the population at time t and let $g(X_t)$ denote the minimal Hamming distance of any individual in the population to the global optimum. Note that this distance corresponds to the number of 0-bits of the closest individual to the optimum and that it equals the value $m+1$ from the proof of Theorem 3. We define $a(l) := 0$ and $b(l) := \sqrt[3]{n} + 1$, that is, $g(X_t) = a(l) = 0$ contains the global optimum, while the population X_t with $g(X_t) = b(l)$ is $\sqrt[3]{n} + 1$ bits away.

With the $\mu - \sqrt[3]{n} - 1$ individuals one fitness level behind each other, $\mathrm{P}(g(X_0) \geq b(l)) = 1$ is satisfied. Also Condition 2 is satisfied since $b(l) - a(l) = \sqrt[3]{n} - 0 = \Omega(\sqrt[3]{n})$.

Conditions 3 and 4 still have to be proved to hold. Let $p_j := \mathrm{P}(g(X_{t+1}) - g(X_t) = j \mid X_t, a(l) < g(X_t) < b(l))$, then we rewrite Condition 3 as

$$\sum_{j=-\sqrt[3]{n}}^{1} p_j \cdot e^{-\lambda j} \ \leq \ 1 - 1/p(l). \tag{2}$$

A move with negative j is heading towards the optimum while one with positive j is drifting away. A move with $j = 1$ occurs if $y_{g(X_t)-1}$ is selected as parent and exactly one out of $n - (g(X_t) - 1) \geq n - \sqrt[3]{n}$ zeros is flipped. Hence

$$p_0 \ \leq \ 1 - p_1 \ \leq \ 1 - \frac{1}{\mu} \cdot \frac{n - \sqrt[3]{n}}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \ \leq \ 1 - \frac{1}{2e\mu}.$$

The probability of a step of length $j \geq 1$ towards the optimum is maximal if $g(X_t) = b(l) = \sqrt[3]{n}$. A necessary condition is to choose $x_{\sqrt[3]{n}+i}$ and to create $x_{\sqrt[3]{n}-j}$ by a step flipping $i+j$ 0-bits for some $i \geq 0$. Since $(i+j)! \geq (i+1)! \cdot j! \geq$

$((i + 1)/e)^{i+1} \cdot j!$, we have

$$p_{-j} \leq \frac{1}{\mu} \sum_{i=0}^{\mu - \sqrt[3]{n}} \binom{i + \sqrt[3]{n}}{i + j} \left(\frac{1}{n}\right)^{i+j}$$

$$\leq \frac{1}{\mu} \sum_{i=0}^{\mu - \sqrt[3]{n}} \left(\frac{i + \sqrt[3]{n}}{n}\right)^{i+j} \cdot \frac{1}{(i+j)!}$$

$$\leq \frac{1}{\mu} \cdot \frac{1}{j!} \sum_{i=0}^{\mu - \sqrt[3]{n}} \left(\frac{i + \sqrt[3]{n}}{n}\right)^{i+1} \cdot \frac{1}{(i+1)!}$$

$$\leq \frac{1}{\mu} \cdot \frac{1}{j!} \sum_{i=0}^{\mu - \sqrt[3]{n}} \left(\frac{e(i + \sqrt[3]{n})}{(i+1)n}\right)^{i+1}$$

$$\leq \frac{1}{\mu} \cdot \frac{1}{j!} \sum_{i=0}^{\mu - \sqrt[3]{n}} \left(\frac{e\sqrt[3]{n}}{n}\right)^{i+1} = \frac{1}{\mu} \cdot \frac{1}{j!} \cdot O(n^{-2/3}).$$

As $\sum_{j=0}^{\infty} e^{\lambda j}/(j!) = \exp(e^{\lambda})$,

$$\sum_{j=1}^{\sqrt[3]{n}} p_{-j} e^{\lambda j} = \frac{1}{\mu} \cdot \sum_{j=1}^{\sqrt[3]{n}} \frac{e^{\lambda j}}{j!} \cdot O(n^{-2/3}) \leq \frac{1}{\mu} \cdot e^{e^{\lambda}} \cdot O(n^{-2/3}).$$

Together, using the trivial bound $p_1 \leq 1$, the left-hand side of Inequality 2 is bounded by

$$\frac{e^{-\lambda}}{\mu} + 1 - \frac{1}{2e\mu} + \frac{1}{\mu} \cdot \exp(e^{\lambda}) \cdot O(n^{-2/3})$$

$$= 1 - \frac{1}{\mu}\left(-e^{-\lambda} + \frac{1}{2e} - \exp(e^{\lambda}) \cdot O(n^{-2/3})\right) \leq 1 - \frac{1}{p(l)}$$

if $\lambda := 2$, $p(l) := 6n$, and n is large enough.

The proof of condition 4 is easier. The following expectation has to be estimated:

$$\forall t \geq 0: \ E[e^{-\lambda(g(X_{t+1}) - b(n))} \mid X_t, b(n) < g(X_t)].$$

This is less or equal to:

$$E[e^{-\lambda(g(X_{t+1}) - g(X_t))} \mid X_t, b(n) < g(X_t)] \leq \sum_{j=-n+(\mu-1)}^{0} r_j \cdot e^{-\lambda j}$$

Here r_j is the probability that the population X_t at time step t such that $b(l) < g(X_t)$ performs a step of length j. The probability of a move going towards the optimum is highest when the process is in position $j = n - (\mu - 1)$. In this case the population is $P_t = \{y_0, y_1 \ldots, y_{\mu-1}\}$. From this position no

moves may occur heading away from the optimum. We bound the sum as follows:

$$\sum_{j=0}^{n-\mu} r_{-j} \cdot e^{\lambda j} \;\leq\; \sum_{j=0}^{\infty} e^{\lambda j} \binom{n}{j} \frac{1}{n^j} \;\leq\; \sum_{j=0}^{\infty} \frac{e^{\lambda j}}{j!} \;=\; \exp(e^{\lambda}) \;<\; D$$

Since the inequality is satisfied for any constant λ and D large enough constant, condition 4 also holds.

Hence, the expected runtime is exponential (that is, $e^{\Omega(\sqrt[3]{n})}$) with overwhelming probability for $B = e^{c \cdot l}$ and c sufficiently small, since $\lambda(a(l) - b(l)) = -\Omega(\sqrt[3]{n})$. $\qquad\square$

6 Deterministic Crowding

In the deterministic crowding mechanism [10], offspring compete directly with their respective parents. In the following algorithm, an offspring replaces its parent if its fitness is at least as good.

Alg. 4: (μ+1) EA with deterministic crowding

Let $t := 0$.
Initialize P_0 with μ individuals chosen uniformly at random.
Repeat
    Choose $x \in P_t$ uniformly at random.
    Create y by flipping each bit in x with probability $1/n$.
    If $f(y) \geq f(x)$ then $P_{t+1} = P_t \setminus \{x\} \cup \{y\}$
                else $P_{t+1} = P_t$.
    Let $t = t + 1$.

The algorithm closely resembles a parallel (1+1) EA since μ individuals explore the landscape independently. However, interactions between the individuals may be obtained by using other operators together with mutation. The crowding mechanism was introduced to guarantee diversity between individuals for the greater effectiveness of the recombination operator [10]. Recently, the mechanism together with crossover has proved to be useful in vertex cover problems by making the difference between polynomial and exponential runtimes for some instances [11]. Here we concentrate on the capabilities of guaranteeing diversity of the mechanism by analyzing the (μ+1) EA with deterministic crowding on the Twomax function. For sufficiently large populations the algorithm can easily reach both local optima.

Theorem 6. *The (μ+1) EA with deterministic crowding and $\mu = \text{poly}(n)$ reaches on Twomax a population consisting of only local or global optima in expected time $O(\mu n \log n)$. In that case the population contains at least one global optimum with probability at least $1 - 2^{-\mu}$.*

Proof. The main observation for the second statement is that the individuals of the population are *independent* due to the crowding mechanism. Due to the symmetry of TWOMAX, the i-th individual in the population reaches the global optimum with probability $1/2$. The probability that at least one individual finds the global optimum is $1 - (1/2)^\mu$.

Let T be the random time until all individuals have reached local or global optima on TWOMAX. It is easier to find a local optimum on TWOMAX than to find a global optimum on ONEMAX, hence we estimate $E(T)$ by the expected global optimization time on ONEMAX. Let $P = \{x_1, \ldots, x_\mu\}$ and $A_{i,j}$ denote the event that the i-th 0-bit in x_j is flipped by a 1-bit-mutation. (We do not consider fixed bit positions since mutation may turn a 1-bit to 0 and a 0-bit to 1, hence exchanging these two bits.) Clearly, all individuals in the population reach the optimum of ONEMAX within t steps if during this period $A_{i,j}$ happens for all $1 \leq j \leq \mu$ and all $1 \leq i \leq |x_j|_0$. In a single generation, we have $P(A_{i,j}) \geq 1/(en\mu)$ (provided $i \leq |x_j|_0$) since x_j is selected for mutation with probability $1/\mu$ and a specific 1-bit-mutation has probability at least $1/(en)$. Let $t := en\mu(\ln \mu + \ln n + \ln 2)$. The probability that $A_{i,j}$ does not happen within t steps is bounded by

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

By the Union bound, the probability that some event $A_{i,j}$ does not happen during t steps is at most $1/2$. We conclude that after t steps all individuals reached the optimum with probability $1/2$. In case this does not happen, we simply repeat the arguments with a new period of t steps. The expected number of periods is at most 2, yielding the bound $E(T) \leq 2t = O(\mu n \log n)$ since $\mu = \text{poly}(n)$. $\qquad\square$

7   Fitness Sharing

Fitness sharing (see, e.g., [10]) derates the real fitness of an individual x by an amount that represents the similarity of x to other individuals in the population. The similarity between x and y is measured by a so-called *sharing function* $\text{sh}(x, y) \in [0, 1]$ where a large value corresponds to large similarities and value 0 implies no similarity. The idea is that if there are several copies of the same individual in the population, these individuals have to share their fitness. As a consequence, selection is likely to remove such clusters and to keep the individuals apart. We denote the *shared fitness* of x in the population P as

$$f(x, P) = \frac{f(x)}{\sum_{y \in P} \text{sh}(x, y)}.$$

The fitness of the population is then defined as

$$f(P) = \sum_{x \in P} f(x, P).$$

It is common practice to use a so-called *sharing distance* σ such that individuals only share fitness if they have distance less than σ. Given some distance function d, a common formulation for the sharing function is

$$\text{sh}(x, y) \;=\; \max\{0, 1 - (d(x,y)/\sigma)^\alpha\}$$

where α is a positive constant that regulates the shape of the sharing function. We use the standard setting $\alpha = 1$ and set the sharing distance to $\sigma = n/2$ as this is the smallest value that still enables us to distinguish between the two branches. As TWOMAX is a function of unitation, we allow the distance function d to depend on the number of ones: $d(x,y) := \big| |x|_1 - |y|_1 \big|$. Such a strategy is known as *phenotypic sharing* [10]. Our precise sharing function is then

$$\text{sh}(x, y) \;=\; \max\left\{ 0, \; 1 - 2\frac{\big| |x|_1 - |y|_1 \big|}{n} \right\}.$$

We now incorporate fitness sharing into the $(\mu{+}1)$ EA. Our goal is to evolve a good population, hence selection works by comparing candidates for next generation's population w.r.t. their $f(P)$-values. This selection strategy has already been analyzed by Sudholt [15].

Alg. 5: $(\mu{+}1)$ EA with fitness sharing

Let $t := 0$.
Initialize P_0 with μ individuals chosen uniformly at random.
Repeat
    Choose $x \in P_t$ uniformly at random.
    Create y by flipping each bit in x with probability $1/n$.
    Let $P_t^* := P_t \cup \{y\}$.
    Choose $z \in P_t^*$ such that $f(P_t^* \setminus \{z\})$ is maximized.
    Let $P_{t+1} = P_t^* \setminus \{z\}$ and $t = t + 1$.

Note that when evaluating $f(P_t^* \setminus \{z\})$ the shared fitness values have to be recomputed for all these populations. However, with the use of dictionaries it suffices to compute $f(y)$ and the sharing values $\text{sh}(x, y)$ for $x \in P_t$ only once.

We now show that the $(\mu{+}1)$ EA with fitness sharing can find both optima on TWOMAX. Imagining all parents and the new offspring on a scale of $|x|_1$, the individuals with the smallest and the largest number of ones have the largest distance to all individuals in the population. Therefore, fitness sharing makes these outer individuals very attractive in terms of shared fitness, hence these individuals are taken over to the next generation. This even holds if an outer individual has the worst fitness in the population.

Lemma 1. *Consider the $(\mu{+}1)$ EA with fitness sharing and $\mu \geq 2$ on* TWOMAX. *Let P_t^* be the enlarged parent population at some point of time t and w.l.o.g. let $P_t^* = \{x_1, \ldots, x_{\mu+1}\}$ with $|x_1|_1 \leq |x_2|_1 \leq \cdots \leq |x_{\mu+1}|_1$. If $|x_\mu|_1 < |x_{\mu+1}|_1$ then $x_{\mu+1} \in P_{t+1}$. Also, if $|x_1|_1 < |x_2|_1$ then $x_1 \in P_{t+1}$.*

Proof. We only prove $x_{\mu+1} \in P_{t+1}$ for $|x_\mu|_1 < |x_{\mu+1}|_1$. The second claim for the case $|x_1|_1 < |x_2|_1$ follows by symmetry. Let $P^- := P_t^* \setminus \{x_{\mu+1}\}$, $P^+ := P_t^* \setminus \{x_\mu\}$ and $P^\cap := P^+ \cap P^-$. We will prove $f(P^-) < f(P^+)$. This implies that P^- does not have maximal population fitness among all possible next populations $P_t^* \setminus \{z\}$ examined in the selection step, hence $x_{\mu+1}$ remains in the population.

We first assume $f(x_\mu) \le f(x_{\mu+1})$. Observe that $\mathrm{sh}(x_\mu, x) > \mathrm{sh}(x_{\mu+1}, x)$ for $x \in P^\cap$ due to the ordering of the x_i and the definition of sh. Hence, $\sum_{y \in P^-} \mathrm{sh}(x_i, y) > \sum_{y \in P^+} \mathrm{sh}(x_i, y)$ for $x \in P^\cap$ and $\sum_{y \in P^-} \mathrm{sh}(x_\mu, y) > \sum_{y \in P^+} \mathrm{sh}(x_{\mu+1}, y)$. Together, we obtain

$$
\begin{aligned}
f(P^-) &= \sum_{x \in P^\cap} \frac{f(x)}{\sum_{y \in P^-} \mathrm{sh}(x,y)} + \frac{f(x_\mu)}{\sum_{y \in P^-} \mathrm{sh}(x_\mu, y)} \\
&< \sum_{x \in P^\cap} \frac{f(x)}{\sum_{y \in P^+} \mathrm{sh}(x,y)} + \frac{f(x_{\mu+1})}{\sum_{y \in P^+} \mathrm{sh}(x_{\mu+1}, y)} \\
&= f(P^+).
\end{aligned}
$$

It remains to show $f(P^-) < f(P^+)$ for the case $f(x_\mu) > f(x_{\mu+1})$. We want to examine the impact of the distance between the two individuals on the fitness and the sharing function. Therefore, we define

$$
\tilde{d} := \min\left\{|x_{\mu+1}|_1, \frac{n}{2}\right\} - |x_\mu|_1.
$$

Note that $\tilde{d} \ge 0$ as $|x_\mu|_1 \le n/2$ follows from $f(x_\mu) > f(x_{\mu+1})$. The following two properties follow immediately from the definition of $\mathrm{sh}(\cdot, \cdot)$ and $f(\cdot)$:

$$
\forall x \in P^\cap: \ \mathrm{sh}(x, x_{\mu+1}) \le \mathrm{sh}(x, x_\mu) - \frac{2\tilde{d}}{n}, \tag{3}
$$

$$
f(x_\mu) - \tilde{d} \le f(x_{\mu+1}). \tag{4}
$$

We will later also prove the following three inequalities:

$$
\forall x \in P^\cap: \frac{f(x, P^-)}{f(x, P^+)} \le 1 - \frac{2\tilde{d}}{n\mu} \tag{5}
$$

$$
\frac{f(x_\mu, P^-)}{f(x_{\mu+1}, P^+)} < 1 + \frac{2\tilde{d}}{n\mu} \tag{6}
$$

$$
f(x_{\mu+1}, P^+) \le \sum_{x \in P^\cap} f(x, P^+) \tag{7}
$$

Using that all terms are positive, inequalities (5), (6), and (7) suffice to prove

our claim:

$$f(P^-) \; = \; f(x_\mu, P^-) + \sum_{x \in P^\cap} f(x, P^-)$$

$$< \; f(x_{\mu+1}, P^+) \cdot \left(1 + \frac{2\tilde{d}}{n\mu}\right) + \sum_{x \in P^\cap} f(x, P^+) \cdot \left(1 - \frac{2\tilde{d}}{n\mu}\right)$$

$$\leq \; f(P^+) - \frac{2\tilde{d}}{n\mu} \left(\sum_{x \in P^\cap} f(x, P^+) - f(x_{\mu+1}, P^-)\right)$$

$$\leq \; f(P^+).$$

It remains to prove the inequalities (5), (6), and (7).

Proof of inequality (5)*:* By inequality (3),

$$\frac{f(x, P^-)}{f(x, P^+)} \; = \; \frac{\sum_{y \in P+} \mathrm{sh}(x,y)}{\sum_{y \in P-} \mathrm{sh}(x,y)}$$

$$\leq \; \frac{\sum_{y \in P^\cap} \mathrm{sh}(x,y) + \mathrm{sh}(x,x_\mu) - 2\tilde{d}/n}{\sum_{y \in P^\cap} \mathrm{sh}(x,y) + \mathrm{sh}(x,x_\mu)}$$

$$\leq \; \frac{\mu - 2\tilde{d}/n}{\mu}$$

using that the sharing function is at most 1. Reducing the last fraction by μ, inequality (5) follows.

Proof of inequality (6)*:* By inequality (4) and $f(x) \geq n/2$ for any $x \in \{0,1\}^n$, we have

$$\frac{f(x_{\mu+1})}{f(x_\mu)} \; \geq \; \frac{f(x_\mu) - \tilde{d}}{f(x_\mu)} \; \geq \; \frac{n/2 - \tilde{d}}{n/2} \; = \; 1 - \frac{2\tilde{d}}{n}.$$

Also, using inequality (3), we obtain

$$\frac{\sum_{y \in P+} \mathrm{sh}(x_{\mu+1}, y)}{\sum_{y \in P-} \mathrm{sh}(x_\mu, y)} \; = \; \frac{1 + \sum_{y \in P^\cap} \mathrm{sh}(x_{\mu+1}, y)}{1 + \sum_{y \in P^\cap} \mathrm{sh}(x_\mu, y)}$$

$$\leq \; \frac{1 + \sum_{y \in P^\cap}(\mathrm{sh}(x_\mu, y) - 2\tilde{d}/n)}{1 + \sum_{y \in P^\cap} \mathrm{sh}(x_\mu, y)}$$

$$\leq \; \frac{1 + (\mu - 1)(1 - 2\tilde{d}/n)}{\mu}. \qquad (8)$$

Taking the last two estimations together,

$$\frac{f(x_\mu, P^-)}{f(x_{\mu+1}, P^+)} \; = \; \frac{\sum_{y \in P+} \mathrm{sh}(x_{\mu+1}, y)}{\sum_{y \in P-} \mathrm{sh}(x_\mu, y)} \cdot \frac{f(x_\mu)}{f(x_{\mu+1})}$$

$$\leq \; \frac{1 + (\mu - 1)(1 - 2\tilde{d}/n)}{\mu(1 - 2\tilde{d}/n)}$$

$$= \; \frac{1 - \frac{2\tilde{d}}{n} + \frac{2\tilde{d}}{n\mu}}{1 - \frac{2\tilde{d}}{n}} \; < \; 1 + \frac{2\tilde{d}}{n\mu},$$

17

which proves inequality (6).

Proof of inequality (7)*:* First consider the right-hand side of inequality (7). This term is minimized if all $x \in P^{\cap}$ equal x_{μ} since then for all individuals in $P^{\cap}$ fitness is minimized and sharing is maximized. Along with $f(x_{\mu}) \geq f(x_{\mu+1})$,

$$\sum_{x \in P^{\cap}} f(x, P^+) \geq \frac{(\mu - 1) \cdot f(x_{\mu})}{\mu - 1 + (1 - 2\tilde{d}/n)} \geq \frac{f(x_{\mu+1})}{1 + \frac{1 - 2\tilde{d}/n}{\mu - 1}}. \tag{9}$$

Note that inequality (8) also implies $\sum_{y \in P^+} \mathrm{sh}(x_{\mu+1}, y) \leq 1 + (\mu - 1)(1 - 2\tilde{d}/n)$. Hence, the left-hand side of inequality (7) can be estimated by

$$f(x_{\mu+1}, P^+) \leq \frac{f(x_{\mu+1})}{1 + (\mu - 1)(1 - 2\tilde{d}/n)}. \tag{10}$$

Comparing the denominators of inequality (9) and inequality (10), inequality (7) follows since $(\mu - 1)^2 \geq 1$. □

Now it is easy to prove an upper bound on TWOMAX. To the best of our knowledge, the following theorem provides the first runtime analysis of an EA with fitness sharing for population sizes greater than 2.

Theorem 7. *The (μ+1) EA with fitness sharing and $\mu \geq 2$ reaches on* TWOMAX *a population containing both optima in expected optimization time* $O(\mu n \log n)$.

Proof. For a population P, we consider the following characteristic values as potential functions: $m_0(P)$ denotes the maximum number of zeros and $m_1(P)$ the maximum number of ones for the individuals in P. We are interested in the expected time until both potentials become n.

According to Lemma 1, both potentials are non-decreasing. If $m_0(P) = k$ then we wait for an individual with k zeros to be chosen and for the number of zeros to be increased. The expected time for this to happen is bounded from above by $O(\mu \cdot n/(n - k))$. Hence, the expected time until the m_0-potential reaches its maximum value n is $O(\mu n \log n)$. A symmetrical statement holds for the m_1-potential, hence the expected time until both optima are found is $O(\mu n \log n)$. □

8   Conclusions

We have examined the behavior of different diversity mechanisms on a fitness landscape consisting of two hills with symmetric slopes. We rigorously proved that without any diversification the (μ+1) EA is unable to climb up both hills (Theorem 1). This still holds if we avoid genotype or phenotype duplicates. On the other hand, stronger diversity mechanisms like fitness sharing and deterministic crowding allow the (μ+1) EA to find both optima of our test function

Twomax with high probability. For fitness sharing we could prove this for arbitrary populations of size $\mu \geq 2$. For deterministic crowding, a slightly larger population is needed.

The work has also revealed important open problems. Theorems 1 and 2 apply only to sublinear population sizes. While experimental studies indicate similar results for larger populations, a theoretical analysis is difficult. It seems that a more thorough understanding of the long-term dynamics of genetic drift is required to strengthen the theorems.

References

[1] N. Chaiyaratana, T. Piroonratana, and N. Sangkawelert. Effects of diversity control in single-objective and multi-objective genetic algorithms. *Journal of Heuristics*, 13(1):1–34, 2007.

[2] S. Fischer and I. Wegener. The one-dimensional Ising model: Mutation versus recombination. *Theoretical Computer Science*, 344(2–3):208–225, 2005.

[3] T. Friedrich, N. Hebbinghaus, and F. Neumann. Rigorous analyses of simple diversity mechanisms. In *Proc. of GECCO '07*, pages 1219–1225, London, 2007. ACM Press.

[4] O. Giel and I. Wegener. Evolutionary algorithms and the maximum matching problem. In *Proc. of STACS '03*, pages 415–426, London, UK, 2003. Springer.

[5] D. E. Goldberg, C. Van Hoyweghen, and B. Naudts. From twomax to the Ising model: Easy and hard symmetrical problems. In *Proc. of GECCO '02*, pages 626–633. Morgan Kaufmann, 2002.

[6] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, 14:502–525, 1982.

[7] J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, 2001.

[8] T. Jansen and I. Wegener. Evolutionary algorithms: How to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Transactions on Evolutionary Computation*, 5(6):589–599, 2001.

[9] T. Jansen and I. Wegener. Real royal road functions: where crossover provably is essential. *Discrete Applied Mathematics*, 149(1-3):111–125, 2005.

[10] S. W. Mahfoud. Niching methods. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C6.1:1–4. Institute of Physics Publishing and Oxford University Press, Bristol, New York, 1997.

[11] P. Oliveto, J. He, and X. Yao. Population-based evolutionary algorithms for the vertex cover problem. Technical Report CSR-07-11, University of Birmingham, School of Computer Science, December 2007.

[12] 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.

[13] M. Pelikan and D. E. Goldberg. Genetic algorithms, clustering, and the breaking of symmetry. In *Proc. of PPSN VI*, pages 385–394, London, UK, 2000. Springer.

[14] T. Storch and I. Wegener. Real royal road functions for constant population size. *Theoretical Computer Science*, 320:123–134, 2004.

[15] D. Sudholt. Crossover is provably essential for the Ising model on trees. In *Proc. of GECCO '05*, pages 1161–1167, Washington DC, 2005. ACM Press.

[16] D. Sudholt. Memetic algorithms with variable-depth search to overcome local optima. Technical Report CI-238/08, Collaborative Research Center 531, Technische Universität Dortmund, 2008.

[17] R. K. Ursem. Diversity-guided evolutionary algorithms. In *Proc. of PPSN-VII*, pages 462–471. Springer, 2002.

[18] C. Witt. Runtime analysis of the $(\mu+1)$ EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.