Proceedings Paper:

Covantes Osuna, E. and Sudholt, D. orcid.org/0000-0001-6020-1646 (2017) Analysis of the Clearing Diversity-Preserving Mechanism. In: FOGA '17 Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms. 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA '17), 12-15 Jan 2017, Copenhagen, Denmark. Association for Computing Machinery . ISBN 978-1-4503-4651-1

https://doi.org/10.1145/3040718.3040731

Analysis of the Clearing Diversity-Preserving Mechanism

Edgar Covantes Osuna
University of Sheffield
Sheffield, S1 4DP,
United Kingdom
ecovantes1@sheffield.ac.uk

Dirk Sudholt
University of Sheffield
Sheffield, S1 4DP,
United Kingdom
d.sudholt@sheffield.ac.uk

ABSTRACT

Clearing is a niching method inspired by the principle of assigning the available resources among a subpopulation to a single individual. The clearing procedure supplies these resources only to the best individual of each subpopulation: the winner. So far, its analysis has been focused on experimental approaches that have shown that clearing is a powerful diversity mechanism. We use empirical analysis to highlight some of the characteristics that makes it a useful mechanism and runtime analysis to explain how and why it is a powerful method. We prove that a (μ+1) EA with large enough population size and a phenotypic distance function always succeeds in optimising all functions of unitation for small niches in polynomial time, while a genotypic distance function requires exponential time. Finally, we prove that a (μ+1) EA with phenotypic and genotypic distances is able to find both optima in TWOMAX for large niches in polynomial expected time.

Keywords

Clearing, diversity-preserving mechanisms, evolutionary algorithm, runtime analysis

1. INTRODUCTION

One of the major difficulties in a population-based evolutionary algorithm (EA) is the premature convergence toward a sub-optimal individual before the fitness landscape is explored properly. Diversity-preserving mechanisms provide the ability to visit many and/or different unexplored regions of the search space and generate solutions that differ in various significant ways from those seen before [7, 8, 12].

A diverse population can deal with multimodal functions and can explore several hills in the fitness landscape simultaneously, so they can therefore support global exploration and help to locate several local and global optima. The algorithm can offer several good solutions to the user, a feature desirable in multiobjective optimisation. Also, it provides higher chances to find dissimilar individuals and to create good offspring with the possibility of enhancing the performance of other procedures such as crossover [5].

Most analyses and comparisons made between diversity-preserving mechanisms are assessed by means of empirical investigations [2, 19] or theoretical runtime analyses [3, 4, 6, 9, 13, 14]. There are examples where empirical investigations are used to support theoretical runtime analyses and close the gap between both fields [5, 15]. Both approaches are important to understand how these mechanisms impact the EA runtime and if they enhance the search for obtaining good individuals. These different expectations imply where EAs and which diversity-preserving mechanism should be used and, perhaps even more important, where they should not be used.

In this sense, we use empirical investigations to get insights into the behaviour of the clearing diversity mechanism and we use theoretical runtime analyses to provide foundations of the behaviour of this mechanism. *Clearing* is a niching method inspired by the principle of sharing limited resources within subpopulations of individuals characterised by some similarities. Instead of evenly sharing the available resources among the individuals of a subpopulation, the clearing procedure supplies these resources only to the best individual of each subpopulation: the winner. The winner takes all rather than sharing resources with the other individuals of the same niche (subspace suitable for a small group of the same or similar type of individuals) as is done with fitness sharing [16].

Like in fitness sharing, the clearing algorithm uses a dissimilarity measure given by a threshold called *clearing radius* σ (or sharing radius in the context of fitness sharing) between individuals to determine if they belong to the same subpopulation or not.

The basic idea is to preserve the fitness of the individual that has the best fitness called dominant individual, while it resets the fitness of all the other individuals of the same subpopulation to zero[1]. With such a mechanism, two approaches can be considered. For a given population the set of winners is unique. The winner and all the individuals that it dominates are then fictitiously removed from the population. Then we proceed in the same way with the new population which is then obtained. Thus, the list of all the winners is produced after a certain number of steps.

ACM ISBN .
DOI:

---

[1] We tacitly assume that all fitness values are larger than 0 for simplicity. In case of a fitness function f with negative fitness values we can change clearing to reset fitness to $f_{\min} - 1$, where $f_{\min}$ is the minimum fitness value of f, such that all reset individuals are worse than any other individuals.

On the other hand, the population can be dominated by several winners. It is also possible to generalise the clearing algorithm by accepting several winners chosen among the *niche capacity* κ (best individuals of each niche defined as the maximum number of winners that a niche can accept). Thus, choosing niching capacities between one and the population size offers intermediate situations between the maximum clearing ($\kappa = 1$) and a standard EA ($\kappa \geq \mu$).

Empirical investigations made in [16, 18] mention that clearing surpasses all other niching methods because of its ability to produce a great quantity of new individuals by randomly recombining elements of different niches, controlling this production by resetting the fitness of the poor individuals in each different niche. Furthermore, an elitist strategy prevents the rejection of the best individuals.

As in past research, we test the clearing diversity mechanism to confirm if the mechanism is able to provide good solutions by means of experiments, and we include theoretical runtime analysis to prove how and why an EA is able to obtain good solutions depending on how the population size, σ, κ, and the dissimilarity measure are chosen.

In the remainder of this paper, we first present the algorithmic approach, functions of unitation, including a bimodal test function of unitation called TWOMAX, and the dissimilarity measures used in Section 2. Section 3 contains the experimental approach and results that gave rise to the theoretical analysis presented in Sections 4 and 5 for small and large niches, respectively. In Section 4 we show how the clearing mechanism is able to solve, for small niches and the right distance function, all functions of unitation, and in Section 5 we show how clearing solves TWOMAX with the most natural distance function: Hamming distance. We present our conclusions in Section 6, where we mention how well the empirical results match with our theoretical results, giving additional insight into the dynamic behaviour of the algorithm.

2. PRELIMINARIES

We focus our analysis on the simple EA with a finite population called (μ+1) EA. Our aim is to develop rigorous runtime bounds of (μ+1) EA with the clearing diversity mechanism. We want to study how diversity helps to escape local optima. The basic (μ+1) EA has already been investigated in [20]. We incorporate the clearing method into the basic algorithm, resulting in Algorithm 1.

The idea behind Algorithm 1 is: once a population with μ individuals is generated, an individual y is selected and changed according to mutation. A temporary population P_t^* is created from population P_t and the individual y, then the fitness of each individual in P_t^* is updated according to the clearing procedure shown in Algorithm 2. Each individual is compared with the winner(s) of each niche in order to check if it belongs to a certain niche or not and to check if its a winner or if it is cleared. Here $d(x_i, x_j)$ is any dissimilarity measure (distance function) between two individuals x_i and x_j of population P. Finally, we keep control of the *niche capacity* defined by κ.

For the sake of clarity, the replacement policy will be the one defined in [20]: the individuals with best fitness are selected (set of winners) and individuals coming from the new generation are preferred if their fitness values are at least as good as the current ones (novelty is rewarded).

Algorithm 1 (μ+1) EA with clearing

```
Let $t := 0$ and initialise $P_0$ with $\mu$ individuals chosen uniformly at random.
while optimum not found do
    Choose $x \in P_t$ uniformly at random.
    Create $y$ by flipping each bit in $x$ independently with probability $1/n$.
    Let $P_t^* := P_t \cup \{y\}$.
    Update $f(P_t^*)$ with the clearing procedure.
    Choose $z \in P_t^*$ with worst fitness uniformly at random.
    if $f(y) \geq f(z)$ then
        Let $P_{t+1} = P_t^* \setminus \{z\}$.
    else
        Let $P_{t+1} = P_t^* \setminus \{y\}$.
    end if
    $t := t + 1$.
end while
```

Algorithm 2 Clearing

```
Sort $P$ according to fitness of individuals by decreasing values.
for $i = 0$ to $|P|-1$ do
    if $f(x_i) > 0$ then
        $nbwinners := 1$
        for $j = i + 1$ to $|P|-1$ do
            if $f(x_j) > 0$ and $d(x_i, x_j) < \sigma$ then
                if $nbwinners < \kappa$ then
                    $nbwinners := nbwinners + 1$
                else
                    $f(x_j) := 0$
                end if
            end if
        end for
    end if
end for
```

We consider functions of unitation—functions defined over the number of 1-bits contained in a string—for the analysis of small niches and a particular bimodal function of unitation called TWOMAX$(x) := \max\{\sum_{i=1}^{n} x_i, n - \sum_{i=1}^{n} x_i\}$ (see Figure 1) for the analysis of large niches, respectively. The function TWOMAX consists of two different symmetric branches ZEROMAX and ONEMAX with 0^n and 1^n as global optima, respectively, and has already been investigated for the analysis of the effectiveness of diversity-preserving mechanisms [5, 14].

Since we aim at analysing the global exploration capabilities of Algorithm 1, we analyse the expected time until both optima have been reached. TWOMAX is an ideal benchmark function for clearing as it is simply structured, hence facilitating a theoretical analysis, and it is hard for evolutionary algorithms to find both optima as they have the maximum possible Hamming distance. Its choice further allows comparisons with previous approaches such as avoiding genotype or phenotype duplicates [5], deterministic crowding [5], and fitness sharing [5, 14].

Finally, as dissimilarity measures, we have considered two distances, genotypic (Hamming distance) and phenotypic (usually defined as Euclidean distance between two phenotypes). As TWOMAX is a function of unitation, we have

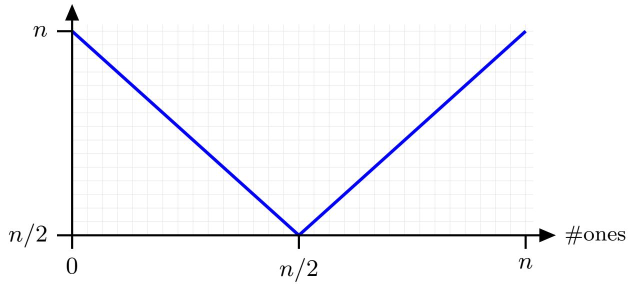


Figure 1: Sketch of the function TWOMAX.

adopted the same approach as in [5, 14], allowing the distance function d to depend on the number of ones: $d(x, y) := ||x|_1 - |y|_1|$.

3. EXPERIMENTS AND THEORY

The experimental approach is focused on the analysis of Algorithm 1 and its behaviour when the parameters σ, κ, and μ are changed and how these parameters can be set. The main objective is to find out general behaviours in order to provide foundations for our theoretical analysis.

In [5, 14] it has been proved that fitness sharing and deterministic crowding allow (μ+1) EA to find both optima for TWOMAX with high probability. We are interested in observing if clearing has the same capacity, so we consider exponentially increasing population sizes $\mu = 2, 4, 8, \ldots, 1024$ for just one size of $n = 30$ (but a theoretical analysis that holds for all n) and perform 100 runs with different settings of parameters σ and κ, so for this experimental framework, we have defined $\sigma = \{1, 2, \sqrt{n}, n/2\}$, $\kappa = \{1, \sqrt{\mu}, \mu/2, (\mu/2 + \sqrt{\mu}), \mu\}$ with phenotypic distance.

Since we are interested in proving how good/bad clearing is, we define the following outcomes and stopping criteria for each run. *Success*, both branches of TWOMAX have been reached, i. e., the run is stopped if the population contains both 0^n and 1^n in the population. *Failure*, when the population only consists of copies of the same genotype or when a run was stopped manually when after 2000 generations stagnation was detected.

3.1 Empirical Analysis and Results

Before starting to define the results, it is better to define overall behaviours, then to focus on specific results in order to understand how the mentioned parameters work together.

Regarding the *niche capacity* κ, it is mentioned in [16] that while the value of κ approaches the size of the population, the clearing effect vanishes and the search becomes a standard EA. This effect is verified in the present experimental approach. With $\kappa \geq \mu/2$ the capability of the method to explore both branches of TWOMAX is reduced. With a small population $\mu \leq 64$ and $\sigma \leq 2$, one branch takes over, removing the individuals on the other branch achieving at least 0.84 success rate. The only way to compensate this is to increase σ between $\sqrt{n}$ and $n/2$ in order to let more individuals participate in the niche, with $\mu \leq 32$ to achieve 0.80 success rate. A reduced niching capacity seems to have a better effect exploring both branches. The best cases were $1 \leq \kappa \leq \sqrt{\mu}$ for maintaining a set of winners, avoiding takeover or extinction.

For small values of $\sigma = \{1, 2\}$ and $\kappa = 1$, with sufficiently many individuals, $\mu = (n/2 + 1) \cdot \kappa$, every individual can creates its own niche, and since only one individual is allowed to be the winner, the individuals are spread in the search space reaching both optima with 1.0 success and 0.0 failure rate. In this scenario, since we are allowing sufficiently many individuals in the population, individuals can be initialised in both branches, reaching their respective peak as shown in Figure 2 (in this case we only show the behaviour of the population with $\mu = \{8, 16, 32\}$, higher values for μ have the same behaviour).

In this scenario we are making use of small differences between individuals rather than using problem-specific knowledge, such as the minimum distance, that allows us to discriminate between the two branches or optima. Instead, it seems that it is the population size that provides enough pressure to solve it. Also, we use the size of the population to have individuals on both branches or occupy all niches as we will show in Section 4.1. In this scenario, using genotypic distance (μ+1) EA fails with 1.0 rate because of this metric, since we have defined a small *clearing radius* individuals with the same phenotype will result in a large Hamming distance, creating winners with the same fitness (as will be proved in Section 4.2).

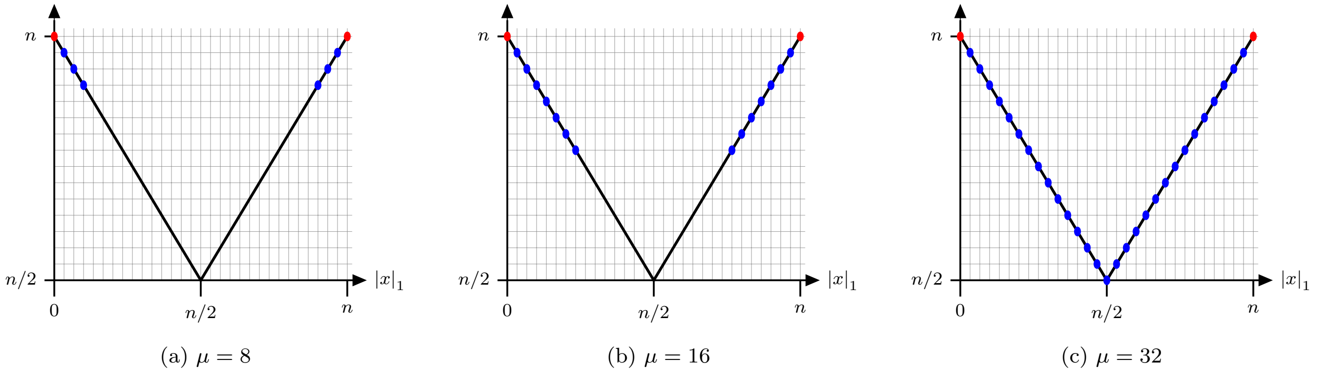
Second, the *clearing radius* σ, defining a small *clearing radius* $\sigma = \{1, 2\}$ with a small *niche capacity* $\kappa < \sqrt{\mu}$ and $\mu \leq 4$ can create takeover or extinction of a certain branch with 1.0 failure rate because one branch may evolve faster. In order to avoid this, it is necessary to increase μ. With $\mu \geq 8$ and a small κ, every niche can have a reduced number of winners, cleared individuals are eliminated as soon as new and better individuals are created (with their respective niches). Also, as we increase μ the spread of the individuals is such that individuals in one branch can reach the other branch as a result of this behaviour with a success rate of 1.0.

For $\sigma = \{\sqrt{n}, n/2\}$ and $\kappa \leq \mu/2$ the behaviour of the algorithm is the opposite. In the case of larger niches it is possible to divide the search space in fewer niches with more winners in each niche. Here the individuals have the opportunity to move, change inside the niche, reach other niches allowing the movement between branches, reaching the opposite optimum with success rate of at least 0.97.

With $\sigma = \sqrt{n}$, $1 \leq \kappa \leq \mu/2$ and $\mu \geq 8$, the method was able to reach both optima with at least 0.97 success rate and 0.03 failure rate. In Figure 3 the effect of κ can be seen with sufficiently many individuals. With restrictive niche capacities (Figure 3a), the population is scattered in the search space while this capacity is reduced as we allow more individuals to be part of each niche (Figure 3c). This behaviour can be generalised and is more evident for larger values of μ.

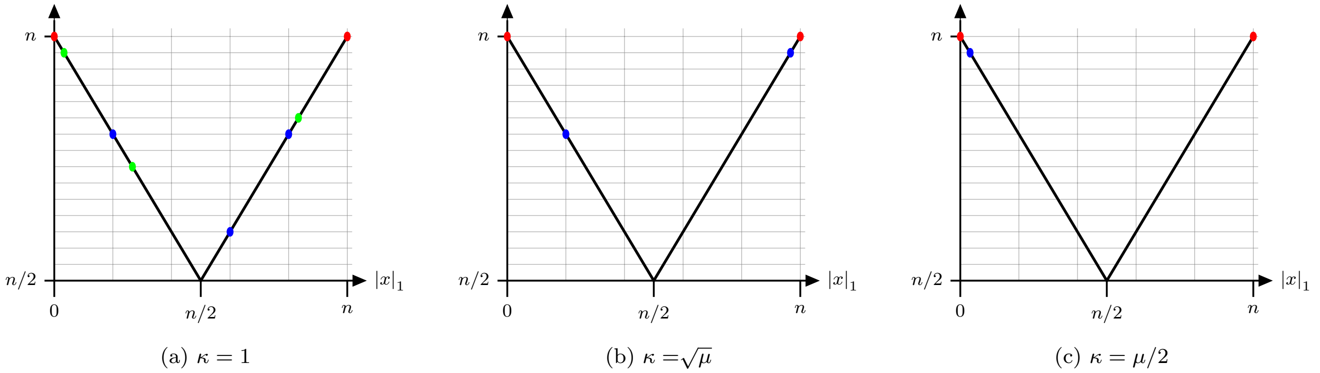
With $\mu \geq 8$ it is possible to overcome takeover and extinction if σ and κ are chosen appropriately. For example, if $\kappa > \sqrt{\mu}$ and $\sigma = n/2$ it will be necessary to provide $\mu \geq 64$ in order to compensate the takeover of a niche in a certain branch due to the number of winners allowed by κ and the large set up of σ and finally achieve 1.0 success rate.

Finally, for $\sigma = n/2$, $\kappa = 1$, and sufficiently large μ, the algorithm was able to reach both optima with 1.0 success rate. As shown in Figure 4, the search space is divided in 2 niches by σ. Even if all the individuals are initialised in one branch it is possible to climb down. Once there is a winner



(a) $\mu = 8$ (b) $\mu = 16$ (c) $\mu = 32$

Figure 2: Snapshot of a typical population at the time both optima were reached, showing the spread of individuals in branches of TWOMAX for $n = 30$, $\sigma = 1$ and $\kappa = 1$. Where the red (extreme) points represent optimal individuals, blue points represent niche winners. The rows on the grid represents the fitness value of an individual and its position on TWOMAX and the columns represent the partitioned search space (niches) created by the parameter σ.



(a) $\kappa = 1$ (b) $\kappa = \sqrt{\mu}$ (c) $\kappa = \mu/2$

Figure 3: Snapshot of a typical population at the time both optima were reached, showing the spread of individuals in branches of TWOMAX for $n = 30$, $\sigma = \sqrt{n}$ and $\mu = 8$. Where the red (extreme) points represent optimal individuals, blue points represent niche winners, and the green points represent cleared individuals. The rows on the grid represents the fitness value of an individual and its position on TWOMAX and the columns represent the partitioned search space (niches) created by the parameter σ.

in the other branch, this individual will climb up until it reaches the opposite optimum that includes the creation of individuals in the branch as proved in our theoretical analysis for large niches in Section 5.

The last two scenarios have the property of dividing the search space in niches in which the individuals are able to spread, move, climb down a branch; this behaviour allows those individuals to reach different niches, until the opposite branch and optima is reached. In this sense, extreme points always survive, a desirable property in this method.

4. SMALL NICHES

In this section we prove that (μ+1) EA with phenotypic clearing and a small niche capacity is not only able to achieve both optima of TWOMAX but is also able to optimise all functions of unitation with a large enough population, while genotypic clearing fails in achieving such a task.

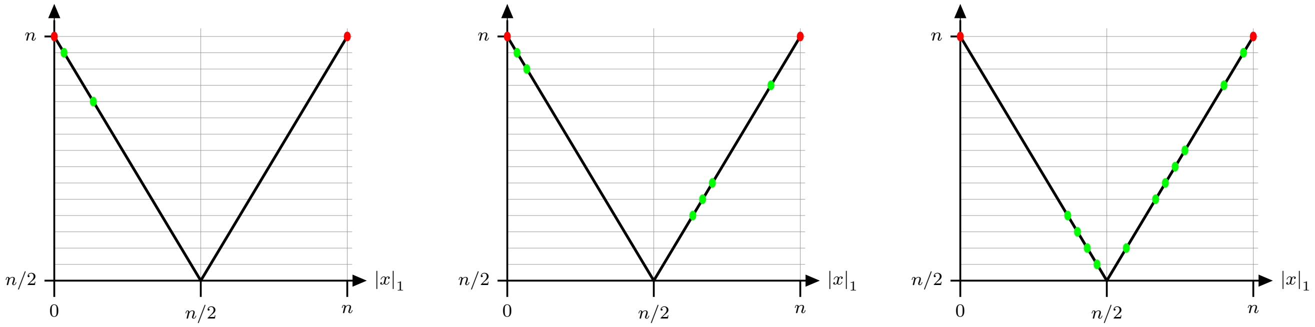
4.1 Phenotypic Clearing

First it is necessary to define a very important property of clearing, which is its capacity of preventing the rejection of the best individuals in (μ+1) EA, and once μ is defined large enough, clearing and the population size pressure will always optimise any function of unitation.

Lemma 4.1 *Consider* (μ+1) EA *with phenotypic clearing and* $\sigma = 1$, $\mu \geq (n+1)\cdot\kappa$ *on any fitness function. Then, winners are never removed from the population, i.e., if* $x \in P_t$ *is a winner then* $x \in P_{t+1}$.

Proof. After the first evaluation with clearing, individuals dominated by other individuals are cleared and the dominant individuals are declared as winners. Cleared individuals are removed from the population when new winners are created and occupy new niches. Once an individual becomes a winner, it can only be removed if the size of the population is not large enough to maintain it, as the worst winner is removed if a new winner reaches a new better niche. Since there are at most $n+1$ niches, each having at most κ winners, if $\mu \geq (n+1)\cdot\kappa$ then there must be a cleared individual amongst the $\mu+1$ parents and offspring considered for deletion at the end of the generation. Thus, a cleared individual will be deleted, so winners cannot be removed from the population. □

The behaviour described above means, that with the defined parameters and sufficiently large μ to occupy all the



(a) $\mu = 8$ (b) $\mu = 16$ (c) $\mu = 32$

Figure 4: Snapshot of a typical population at the time both optima were reached, showing the spread of individuals in branches of TWOMAX for $n = 30$, $\sigma = n/2$ and $\kappa = 1$. Where the red (extreme) points represent optimal individuals, blue points represent niche winners, and the green points represent cleared individuals. The rows on the grid represents the fitness value of an individual and its position on TWOMAX and the columns represent the partitioned search space (niches) created by the parameter σ.

niches, we have enough conditions for the furthest individuals (individuals with the minimum and maximum number of ones in the population) to reach the opposite edges.

Lemma 4.2 *Let f be a function of unitation and $\sigma = 1$, $\mu \geq (n+1) \cdot \kappa$. Then, the expected time for finding the search points 0^n and 1^n with* (μ+1) EA *with phenotypic clearing on f is $O(\mu n \log n)$.*

Proof. Now that we know that a winner cannot be removed from the population by Lemma 4.1, it is just a matter of finding the expected time until 0^n and 1^n are found. Because of the elitist approach of (μ+1) EA, winners will never be replaced if we assume a large enough population size, the winners with the minimum and maximum number of ones in the population will never be removed, we can apply a level-based argument as follows. First, we will focus on estimating the time until the 1^n individual is found (by symmetry, the same analysis apply for the 0^n individual). If the current farthest niche is i, it has a probability of being selected at least of $1/\mu$. In order to create a niche $j > i$, it is just necessary that one of the $n - i$ zeroes is flipped into 1-bit and the other bits remains unchanged. Each bit flip has a probability of being changed (mutated) of $1/n$ and the remaining bits remains unchanged is $(1 - 1/n)^{n-1}$. Hence, the probability of creating niche j from i with $j > i$ is bounded as follows

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

The expected time for increasing the best niche i is hence at most $(\mu e n)/(n - i)$ and the expected time for finding 1^n is at most

$$\sum_{i=0}^{n-1} \frac{\mu e n}{n-i} = \mu e n \sum_{i=1}^{n} \frac{1}{i} \leq \mu e n \ln n = O(\mu n \log n).$$

Adding the same time for finding 0^n proves the claim. ☐

Once the search points 0^n and 1^n have been found, we can focus on the time required for the algorithm until all intermediate niches are discovered.

Lemma 4.3 *Let f be any function of unitation, $\sigma = 1$, $\mu \geq (n+1) \cdot \kappa$, and assume that the search points 0^n and 1^n are contained in the population. Then, the expected time until all niches are found with* (μ+1) EA *with phenotypic clearing on f is $O(\mu n)$.*

Proof. According to Lemma 4.1 and the elitist approach of (μ+1) EA, winners will never be replaced if we assume a large enough population size and by assumption we already have found both search points 0^n and 1^n. First, we will focus on estimating the time until all niches with $i \geq n/2$ are found (by symmetry, the same analysis applies to $< n/2$ niches). Now, let us divide the search space into $n/2$, if there is an empty niche with $i \geq n/2$ ones, then there must be niche i such that $i \geq n/2$ and $i + 1$ exists. So the probability of selecting this $i + 1$ niche is at least $1/\mu$, and since it is just necessary to flip one of at least $n/2$ 0-bits with probability $1/n$, we have a probability of at least $1/2$ to do so, and a probability of leaving the remaining bits untouched of $(1 - 1/n)^{n-1} \geq 1/e$, all together the probability is bounded from below by $1/(2\mu e)$. Using the level-based argument used before, the expected time to occupy all niches $\geq n/2$ is bounded by

$$\sum_{i=n/2}^{n-1} \frac{2\mu e}{1} \leq 2\mu e n = O(\mu n). \qquad \square$$

Theorem 4.4 *Let f be a function of unitation and $\sigma = 1$, $\mu \geq (n+1) \cdot \kappa$. Then, the expected optimisation time of* (μ+1) EA *with phenotype clearing on f is $O(\mu n \log n)$.*

Proof. Now that we have defined and proved all conditions where the algorithm is able to maintain every winner in the population (Lemma 4.1), to find the extreme search points (Lemma 4.2) and intermediate niches (Lemma 4.3) of the function f, we can conclude that the total time required to optimise the function of unitation f is $O(\mu n \log n)$. ☐

4.2 Genotypic Clearing

In the case of genotypic clearing, (μ+1) EA behaves like the diversity-preserving mechanism called *no genotype duplicates* already analysed in [5]. (μ+1) EA with no genotype duplicates rejects the new offspring if the genotype is already contained in the population. The same happens for (μ+1) EA with genotypic clearing and $\sigma = 1$ if the population is initialised with μ mutually different genotypes (which happens with probability at least $1 - \binom{\mu}{2}2^{-n}$). In other words, conditional on the population being initialised with mutually different search points, both algorithms are identical. In [5, Theorem 2], it was proved that (μ+1) EA with no genotype duplicates and $\mu = o(n^{1/2})$ is not powerful enough to explore the landscape and can be easily trapped in one optimum of TWOMAX. Adapting [5, Theorem 2] to the goal of finding both optima and noting that $\binom{\mu}{2}2^{-n} = o(1)$ for the considered μ yields the following.

Corollary 4.5 *The probability that* (μ+1) EA *with genotypic clearing,* $\sigma = 1$ *and* $\mu = o(n^{1/2})$ *finds both optima on* TWOMAX *in time* n^{n-2} *is at most* $o(1)$*. The expected time for finding both optima is* $\Omega(n^{n-1})$*.*

As mentioned before, the use of a proper distance is really important in the context of clearing. In our case, we use phenotypic distance for functions of unitation, which has been proved to provide more significant information at the time it is required to define small differences (in our case small niches) among individuals in a population, so the use of that knowledge can be taken into consideration at the time the algorithm is set up. Otherwise, if there is no more knowledge related to the specifics of the problem, genotypic clearing can be used but with larger niches as shown in the following section.

5. LARGE NICHES

While small niches work with phenotypic clearing, Theorem 4.5 showed that with genotypic clearing small niches are ineffective. This makes sense as for phenotypic clearing with $\sigma = 1$ a niche with i ones covers $\binom{n}{i}$ search points, whereas a niche in genotypic clearing with $\sigma = 1$ only covers one search point. In this section we turn our attention to larger niches, where, according to our empirical observations from Section 3, cleared search points are likely to spread, move, and climb down a branch.

We first present general insights into these population dynamics with clearing. These results capture the behaviour of the population in the presence of only one winning genotype x^* (of which there may be κ copies). We estimate the time until in this situation the population evolves a search point of Hamming distance d from said winner, for any $d \le \sigma$, or for another winner to emerge (for example, in case an individual of better fitness than x^* is found).

These time bounds are very general as they are independent of the fitness function. This is possible since, assuming the winners are fixed at x^*, all other search points within the clearing radius receive a fitness of 0 and hence are subject to a random walk.

We demonstrate the usefulness of our general method by an application to TWOMAX with a clearing radius of $\sigma = n/2$, where all winners are copies of either 0^n or 1^n. The results hold both for genotypic clearing and phenotypic clearing as the phenotypic distance of any point x to 0^n (1^n, resp.) equals the Hamming distance of x to 0^n (1^n, resp.).

5.1 Population Dynamics with Clearing

We assume that the population contains only one winner genotype x^*, of which there are κ copies. For any given integer $0 \le d \le \sigma$, we analyse the time for the population to reach a search point of Hamming distance at least d from x^*, or for a winner different from x^* to emerge.

To this end, we will study a potential function φ that measures the dynamics of the population. Let

$$\varphi(P_t) = \sum_{x \in P_t} H(x, x^*)$$

be the sum of all Hamming distances of individuals in the population to the winner x^*. The following lemma shows how the potential develops in expectation.

Lemma 5.1 *Let* P_t *be the current population of* (μ+1) EA *with genotypic clearing on any fitness function such that the only winners are* κ *copies of* x^* *and* $H(x, x^*) < \sigma$ *for all* $x \in P_t$*. Then the expected change of the potential is*

$$\mathrm{E}(\varphi(P_{t+1}) - \varphi(P_t) \mid P_t) = 1 - \frac{\varphi(P_t)}{\mu}\left(\frac{2}{n} + \frac{\kappa - 1}{\mu + 1 - \kappa}\right)$$

unless a winner different from x^* *is created.*

Before proving the lemma, let us make sense of this formula. Ignore the term $\frac{\kappa-1}{\mu+1-\kappa}$ for the moment and consider the formula $1 - \frac{\varphi(P_t)}{\mu} \cdot \frac{2}{n}$. Note that $\varphi(P_t)/\mu$ is the average distance to the winner in P_t. If the population has spread such that is has reached an average distance of $n/2$ then the expected change would be $1 - \frac{\varphi(P_t)}{\mu} \cdot \frac{2}{n} = 1 - \frac{n}{2} \cdot \frac{2}{n} = 0$. Moreover, a smaller average distance will give a positive drift (expected value in the decrease of the distance after a single function evaluation) and an average distance larger than $n/2$ will give a negative drift. This makes sense as a search point performing an independent random walk will attain an equilibrium state around Hamming distance $n/2$ from x^*.

The term $\frac{\kappa-1}{\mu+1-\kappa}$ reflects the fact that losers in the population do not evolve in complete isolation. The population always contains κ copies of x^* that may create offspring and may prevent the population from venturing far away from x^*. In other words, there is a constant influx of search points descending from winners x^*. As the term $\frac{\kappa-1}{\mu+1-\kappa}$ indicates, this effect grows with κ, but (as we will see later) it can be mitigated by setting the population size μ sufficiently large.

Proof of Lemma 5.1. If an individual $x \in P_t$ is selected as parent, the expected distance of its mutant to x^* is

$$\begin{aligned} &H(x, x^*) + \frac{n - H(x, x^*)}{n} - \frac{H(x, x^*)}{n} \\ &= H(x, x^*) + 1 - \frac{2H(x, x^*)}{n}. \end{aligned}$$

Hence after a uniform parent selection and mutation, the expected distance in the offspring is

$$\sum_{x \in P_t} \frac{1}{\mu} \cdot \left(H(x, x^*) + 1 - \frac{2H(x, x^*)}{n}\right) = \frac{\varphi(P_t)}{\mu} + 1 - \frac{2\varphi(P_t)}{\mu n}.$$

After mutation and clearing procedure, there are $\mu + 1$ individuals in P_t with κ copies of x^*. As all $\mu + 1 - \kappa$ non-winner individuals in P_t have fitness 0, one of these will be selected uniformly at random for deletion. The expected distance to x^* in the deleted individual is

$$\begin{aligned}
&\sum_{x\,\in P_t\setminus\{x^*\}} \frac{1}{\mu+1-\kappa}\cdot H(x,x^*)\\
&= \sum_{x\in P_t} \frac{1}{\mu+1-\kappa}\cdot H(x,x^*)\\
&= \frac{\varphi(P_t)}{\mu+1-\kappa}.
\end{aligned}$$

Together, the expected change of the potential is

$$\mathrm{E}(\varphi(P_{t+1})-\varphi(P_t)\mid P_t) = \frac{\varphi(P_t)}{\mu}+1-\frac{2\varphi(P_t)}{\mu n}-\frac{\varphi(P_t)}{\mu+1-\kappa}.$$

Using that

$$\begin{aligned}
\frac{\varphi(P_t)}{\mu}-\frac{\varphi(P_t)}{\mu+1-\kappa} &= \frac{(\mu+1-\kappa)\varphi(P_t)}{\mu(\mu+1-\kappa)}-\frac{\mu\varphi(P_t)}{\mu(\mu+1-\kappa)}\\
&= -\frac{(\kappa-1)\varphi(P_t)}{\mu(\mu+1-\kappa)}
\end{aligned}$$

the above simplifies to

$$\begin{aligned}
\mathrm{E}(\varphi(P_{t+1})-\varphi(P_t)\mid P_t) &= 1-\frac{2\varphi(P_t)}{\mu n}-\frac{(\kappa-1)\varphi(P_t)}{\mu(\mu+1-\kappa)}\\
&= 1-\frac{\varphi(P_t)}{\mu}\left(\frac{2}{n}+\frac{\kappa-1}{\mu+1-\kappa}\right).
\end{aligned}$$

□

The potential allows us to conclude when the population has reached a search point of distance at least d from x^*. The following lemma gives a sufficient condition.

Lemma 5.2 *If P_t contains κ copies of x^* and $\varphi(P_t) > (\mu-\kappa)(d-1)$ then P_t must contain at least one individual x with $H(x,x^*) \geq d$.*

Proof. There are at most $\mu-\kappa$ individuals different from x^*. By the pigeon-hole principle, at least one of them must have at least distance d from x^*. □

In order to bound the time for reaching a high potential given in Lemma 5.2, we will use the following drift theorem, a straightforward extension of the variable drift theorem [11] towards reaching any state smaller than some threshold a. It can be derived with simple adaptations to the proof in [17].

Theorem 5.3 (Generalised variable drift theorem) *Consider a stochastic process $X_0, X_1, \dots$ on $\mathbb{N}_0$. Suppose there is a monotonic increasing function $h\colon \mathbb{R}^+ \to \mathbb{R}^+$ such that the function $1/h(x)$ is integrable on $[1,m]$, and with*

$$\mathrm{E}(X_t - X_{t+1}\mid X_t = k) \geq h(k)$$

for all $k \in \{a,\dots,m\}$. Then the expected first hitting time of any state from $\{0,\dots,a-1\}$ for $a\in\mathbb{N}$ is at most

$$\frac{a}{h(a)}+\int_a^m \frac{1}{h(x)}\,\mathrm{d}x.$$

The following lemma now gives an upper bound on the first hitting time (the random variable that denotes the first point in time to reach a certain point) of a search point with distance at least d to the winner x^*.

Lemma 5.4 *Let P_t be the current population of (μ+1) EA with genotypic clearing and $\sigma \leq n/2$ on any fitness function such that P_t contains κ copies of a unique winner x^* and $H(x,x^*) < d$ for all $x \in P_t$. For any $0 \leq d \leq \sigma$, if $\mu \geq \kappa\cdot\frac{dn-2d+2}{n-2d+2}$ then, the expected time until a search point x with $H(x,x^*) \geq d$ is found, or a winner different from x^* is created, is $O(\mu n \log \mu)$.*

Proof. We pessimistically assume that no other winner is created and estimate the first hitting time of a search point with distance at least d. As φ can only increase by at most n in one step, $h_{\max} := (\mu-\kappa)(d-1)+n$ is an upper bound on the maximum potential that can be achieved in the generation where a distance of d is reached or exceeded for the first time.

In order to apply drift analysis, we define a distance function that describes how close the algorithm is to reaching a population where a distance d was reached. We consider the random walk induced by $h_{\max}-\varphi(P_t)$, stopped as soon as a Hamming distance of at least d from x^* is reached. Due to our definition of $h_{\max}$, the random walk only attains values in $\mathbb{N}_0$ as required by the variable drift theorem.

By Lemma 5.1, abbreviating $\alpha := \frac{1}{\mu}\left(\frac{2}{n}+\frac{\kappa-1}{\mu+1-\kappa}\right)$, provided $h(P_t) > 0$, h decreases in expectation by

$$1-\alpha\varphi(P_t) = 1-\alpha h_{\max}+\alpha h(P_t).$$

By definition of h and Lemma 5.2, the population reaches a distance of at least d once the distance $h_{\max}-\varphi(P_t)$ has dropped below n. Using the generalised variable drift theorem, the expected time till this happens is at most

$$\frac{n}{1-\alpha h_{\max}+\alpha n}+\int_n^{h_{\max}} \frac{1}{1-\alpha h_{\max}+\alpha x}\,\mathrm{d}x$$

Using $\int \frac{1}{ax+b}\,\mathrm{d}x = \frac{1}{a}\ln|ax+b|$ [1, Equation 3.3.15], we get

$$\begin{aligned}
&\frac{n}{1-\alpha h_{\max}+\alpha n}+\left[\frac{1}{\alpha}\ln(1-\alpha h_{\max}+\alpha x)\right]_n^{h_{\max}}\\
&= \frac{n}{1-\alpha h_{\max}+\alpha n}+\frac{1}{\alpha}\cdot(\ln(1)-\ln(1-\alpha h_{\max}+\alpha n))\\
&= \frac{n}{1-\alpha h_{\max}+\alpha n}+\frac{1}{\alpha}\ln((1-\alpha h_{\max}+\alpha n)^{-1}).
\end{aligned}$$

We now bound the term $1-\alpha h_{\max}+\alpha n$ from below as follows.

$$\begin{aligned}
&1-\alpha h_{\max}+\alpha n\\
&= 1-(\mu-\kappa)(d-1)\cdot\frac{1}{\mu}\left(\frac{2}{n}+\frac{\kappa-1}{\mu+1-\kappa}\right)\\
&\geq 1-(\mu-\kappa)(d-1)\cdot\frac{1}{\mu}\left(\frac{2}{n}+\frac{\kappa}{\mu-\kappa}\right)\\
&= 1-\frac{2(\mu-\kappa)(d-1)+\kappa(d-1)n}{\mu n}\\
&= \frac{\mu n-2\mu d+2\kappa d-\kappa dn+2\mu-2\kappa+\kappa n}{\mu n}\\
&= \frac{\kappa}{\mu}+\frac{n-2d+2}{n}-\frac{\kappa dn-2\kappa d+2\kappa}{\mu n}\\
&\geq \frac{\kappa}{\mu}+\frac{n-2d+2}{n}-\frac{n-2d+2}{n}\\
&= \frac{\kappa}{\mu}
\end{aligned}$$

where in the penultimate step we used the assumption $\mu \geq \kappa\cdot\frac{dn-2d+2}{n-2d+2}$. Along with $\alpha \geq 2/(\mu n)$, the expected time bound simplifies to

$$\frac{n}{1-\alpha h_{\max}+\alpha n}+\frac{1}{\alpha}\ln((1-\alpha h_{\max}+\alpha n)^{-1}) \leq \frac{n}{\kappa/\mu}+\frac{\mu n}{2}\ln(\mu/\kappa)=O(\mu n\log\mu). \qquad \square$$

The minimum threshold for μ, $\kappa\cdot\frac{dn-2d+2}{n-2d+2}$, contains a factor of κ. The reason is that the fraction of winners in the population needs to be small enough to allow the population to escape from the vicinity of x^*. The population size hence needs to grow proportionally to the number of winners κ the population is allowed to store.

Note that the restriction $d \leq \sigma \leq n/2$ is necessary in Lemma 5.4. Individuals evolving within the clearing radius, but at a distance larger than $n/2$ to x^* will be driven back towards x^*. If d is significantly larger than $n/2$, we conjecture that the expected time for reaching a distance of at least d from x^* becomes exponential in n.

5.2 An Upper Bound for TwoMax

It is now easy to apply Lemma 5.4 in order to achieve a running time bound on TWOMAX. Putting $d=\sigma=n/2$, the condition on μ simplifies to

$$\mu \geq \kappa\cdot\frac{dn-2d+2}{n-2d+2}=\kappa\cdot\frac{n^2/2-n+2}{2}$$

which is implied by $\mu \geq \kappa n^2/4$. Lemma 5.4 then implies the following. Recall that for $x^* \in \{0^n, 1^n\}$, genotypic distances $H(x, x^*)$ equal phenotypic distances, hence the result applies to both genotypic and phenotypic clearing.

Corollary 5.5 *Consider* (μ+1) EA *with genotypic or phenotypic clearing,* $\kappa \in \mathbb{N}, \mu \geq \kappa n^2/4$ *and* $\sigma = n/2$ *on* TWOMAX *with a population containing* κ *copies of* 0^n *(*1^n*). Then the expected time until a search point with at least (at most)* $n/2$ *ones is found is* $O(\mu n\log\mu)$.

Theorem 5.6 *The expected time for* (μ+1) EA *with genotypic or phenotypic clearing,* $\mu \geq \kappa n^2/4$, $\mu \leq \mathrm{poly(n)}$ *and* $\sigma = n/2$ *finding both optima on* TWOMAX *is* $O(\mu n \log n)$.

Proof. We first estimate the time to reach one optimum, 0^n or 1^n. The population is elitist as it always contains a winner with the best-so-far fitness. Hence we can apply a fitness-level argument as follows. If the current best fitness is i, it can be increased by selecting an individual with fitness i (probability at least $1/\mu$) and flipping only one of $n-i$ bits with the minority value (probability at least $(n-i)/(en)$). The expected time for increasing the best fitness i is hence at most $\mu\cdot en/(n-i)$ and the expected time for finding some optimum $x^* \in \{0^n, 1^n\}$ is at most

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

In order to apply Corollary 5.5, we need to have κ copies of x^* in the population. While this isn't the case, a generation picking x^* as parent and not flipping any bits creates another winner x^* that will remain in the population. If there are j copies of x^*, the probability to create another winner is at least $j/\mu\cdot(1-1/n)^n \geq j/(4\mu)$ (using $n \geq 2$). Hence the time until the population contains κ copies of x^* is at most

$$\sum_{j=1}^{\kappa}\frac{4\mu}{j}=O(\mu\log\kappa)=O(\mu\log n)$$

as $\kappa \leq \mu \leq \mathrm{poly}(n)$.

By Corollary 5.5, the expected time till a search point on the opposite branch is created is $O(\mu n\log\mu) = O(\mu n\log n)$. Since the best individual on the opposite branch is a winner in its own niche, it will never be removed. This allows the population to climb this branch as well. Repeating the arguments from the first paragraph of this proof, the expected time till the second optimum is found is at most $e\mu n\ln n$.

Adding up all expected times proves the claim. $\square$

Note that, in contrast to previous analyses of fitness sharing [5, 14], the above analysis does not make use of the specific fitness values of TWOMAX. The main argument of how to escape from one local optimum only depends on the size of its basin of attraction. Our results therefore easily extend to more general function classes that can be optimised by leaving a basin of attraction of width at most $n/2$ (e. g. variants of TWOMAX with different slopes as defined in Theorem 1 of [10], or asymmetric variants with a suboptimal branch having a smaller basin of attraction).

One limitation of Theorem 5.6 is the steep requirement on the population size: $\mu \geq \kappa n^2/4$. Experiments suggest that smaller population sizes are effective as well, so the quadratic dependence on n could be an artefact of our approach. The condition on μ was chosen to ensure a positive drift of the potential for all populations that haven't reached distance d yet, including the most pessimistic scenario of all losers having distance $d-1$ to x^*. Such a scenario is unlikely: experiments suggest that the population tends to spread out, covering a broad range of distances (see Figure 4). With such a spread, a distance of d can be reached with a much smaller potential than that indicated by Lemma 5.2. For such populations, a smaller μ might still guarantee a positive drift. We conjecture that (μ+1) EA is still efficient on TWOMAX if $\mu = O(n)$. However, proving this may require new arguments on the distribution of losers inside the population.

6. CONCLUSIONS

The present empirical and theoretical investigation has shown that clearing possesses desirable and powerful characteristics. We have used these empirical investigations to get an insight into the behaviour of this diversity-preserving mechanism and to rigorously prove its ability to explore the landscape in two cases, small and large niches.

In the case of small niches, we have proved that clearing can exhaustively explore the landscape when the proper distance and parameters like *clearing radius*, *niche capacity* and population size are set. Also, we have proved that clearing is powerful enough to optimise all functions of unitation. In the case of large niches, clearing has been proved to be as strong as other diversity-preserving mechanisms like deterministic crowding and fitness sharing since it is able to find both optima of the test function TWOMAX.

Our theoretical results have also shown that the present analysis can be extended to more general function classes. Also, further theoretical analysis is necessary related to the dynamics of the population (including the distribution of the losers inside the population) since the experiments suggest that smaller population sizes are effective in the case of large niches for TWOMAX.

7. ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers of the pre-conference reviews for their many valuable suggestions which improved the paper. And to the Consejo Nacional de Ciencia y Tecnología — CONACYT (the Mexican National Council for Science and Technology) for the financial support under the grant no. 409151 and registration no. 264342. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 618091 (SAGE).

8. REFERENCES

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, ninth Dover printing, tenth GPO printing edition, 1964.

[2] 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.

[3] B. Doerr, W. Gao, and F. Neumann. Runtime Analysis of Evolutionary Diversity Maximization for OneMinMax. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pages 557–564. ACM, 2016.

[4] T. Friedrich, N. Hebbinghaus, and F. Neumann. Rigorous Analyses of Simple Diversity Mechanisms. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, pages 1219–1225. ACM, 2007.

[5] T. Friedrich, P. S. Oliveto, D. Sudholt, and C. Witt. Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation*, 17(4):455–476, 2009.

[6] W. Gao and F. Neumann. Runtime analysis for maximizing population diversity in single-objective optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 777–784. ACM, 2014.

[7] M. Gendreau and J.-Y. Potvin. Tabu Search. In *Handbook of Metaheuristics*, pages 41–59. Springer US, 2010.

[8] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

[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] T. Jansen and C. Zarges. Example landscapes to support analysis of multimodal optimisation. In *14th International Conference on Parallel Problem Solving from Nature (PPSN XIV)*, pages 792–802. Springer International Publishing, 2016.

[11] D. Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik, 2010.

[12] M. Lozano and C. García-Martínez. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research*, 37(3):481–497, 2010.

[13] P. S. Oliveto and D. Sudholt. On the Runtime Analysis of Stochastic Ageing Mechanisms. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 113–120. ACM, 2014.

[14] P. S. Oliveto, D. Sudholt, and C. Zarges. On the runtime analysis of fitness sharing mechanisms. In *13th International Conference on Parallel Problem Solving from Nature (PPSN 2014)*, volume 8672 of *LNCS*, pages 932–941. Springer, 2014.

[15] P. S. Oliveto and C. Zarges. Analysis of diversity mechanisms for optimisation in dynamic environments with low frequencies of change. *Theoretical Computer Science*, 561, Part A:37–56, 2015.

[16] A. Pétrowski. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798–803, 1996.

[17] J. E. Rowe and D. Sudholt. The choice of the offspring population size in the (1,λ) evolutionary algorithm. *Theoretical Computer Science*, 545:20–38, 2014.

[18] B. Sareni and L. Krahenbuhl. Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, 1998.

[19] R. K. Ursem. Diversity-Guided Evolutionary Algorithms. In *Parallel Problem Solving from Nature – PPSN VII*, pages 462–471. Springer Berlin Heidelberg, 2002.

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