

Runtime Analysis of Evolutionary Diversity Maximization for OneMinMax

Benjamin Doerr
Laboratoire d'Informatique
École Polytechnique
Paris-Saclay
France

Wanru Gao
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, Australia

Frank Neumann
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, Australia

ABSTRACT

Diversity mechanisms are key to the working behaviour of evolutionary multi-objective algorithms. With this paper, we contribute to the theoretical understanding of such mechanisms by means of rigorous runtime analysis. We consider the OneMinMax problem for which it has been shown in [11] that a standard benchmark algorithm called $(\mu + 1)$ -SIBEA is not able to obtain a population with optimal hypervolume distribution in expected polynomial time if the population size is relatively small. We investigate the same setting as in [11] and show that $(\mu + 1)$ -SIBEA is able to achieve a good approximation of the optimal hypervolume distribution very efficiently. Furthermore, we study OneMinMax in the context of search-based diversity optimization and examine the time until $(\mu + 1)$ -SIBEA with a search-based diversity mechanism has obtained a population of maximal diversity covering the whole Pareto front.

Categories and Subject Descriptors

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

Keywords

Evolutionary multi-objective optimization, Diversity, Runtime analysis, Theory

1. INTRODUCTION

Most probably one of the greatest success stories of evolutionary computation lies in the area of evolutionary multi-objective optimization [6, 7]. Evolutionary algorithms are frequently applied to various multi-objective problems in important areas such as renewable energy [13] and water network distribution [17]. Using an evolutionary algorithm for a given multi-objective problem, the population of an EA is evolved into a set of solutions which represents the trade-offs according to the given objective functions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '16, July 20–24, 2016, Denver, Colorado, USA.

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4206-3/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2908812.2908827>

In contrast to high profile applications of evolutionary multi-objective optimization, the theoretical foundations lack far behind their practical success. The main reason for this is that population-based evolutionary algorithms are hard to analyze. For multi-objective problems this especially holds if a population is used which is not able to cover the whole Pareto front. The key part of an evolutionary algorithm for multi-objective optimization is the selection step which decides which individuals are transferred in the next generation. Almost all selection methods follow the principle of Pareto dominance in an explicit or implicit way. Algorithms such as NSGA-II, SPEA2, and IBEA basically differ in the way they differentiate between incomparable solutions.

With this paper, we contribute to the theoretical understanding of evolutionary multi-objective optimization and study the runtime until they have achieved important goals. The area of runtime analysis has become a major contributor to the theory of evolutionary computation over the last 20 years (see the books [3, 10] for comprehensive presentations). Theoretical runtime results on the hypervolume-based algorithms are rather rare. Results have been obtained for the μ -distribution maximizing the hypervolume indicator [1, 2] and the behaviour of simple hypervolume-based algorithms for simple benchmark functions [5, 11]. We first consider a simple hypervolume-based algorithm called $(\mu + 1)$ -SIBEA, which is inspired by the famous SMS-EMOA [4]. For the OneMinMax problem, it has been shown in [11] that $(\mu + 1)$ -SIBEA is not able to achieve a population having maximal hypervolume in the case of $\mu = \sqrt{n}$ when starting with a worst-case initial population. In this paper we consider the same setting and show that $(\mu + 1)$ -SIBEA is able to obtain a good approximation of the optimal hypervolume in expected polynomial time.

After having obtained this result, we turn our attention to another recent development in evolutionary multi-objective optimization, namely decision space diversity [12, 15, 16]. Here the goal is to obtain a set of Pareto optimal solutions that differ according to the underlying search space. Such a set of solutions can be very valuable to decision makers who are able to judge solutions in an intuitive way, but without quantifying their knowledge in terms of an objective function. Having a diverse set of solutions according to the components of a solution therefore gives the decision maker more options of implementing a good solution in different ways. Initial studies on the runtime behaviour for search space diversity optimization have been obtained in [9].

Again, we consider OneMinMax and analyze the runtime of diversity optimization techniques for evolutionary multi-

Algorithm 1: $(\mu + 1)$ -SIBEA.

Start with an initial population P consisting of μ elements from S .

repeat forever

 Select x from P uniformly at random;

$x' \leftarrow \text{mutate}(x)$;

$\hat{P} \leftarrow P \cup \{x'\}$;

 Let z be a randomly chosen individual with

$c_H(z, \hat{P}) = \min_{x \in \hat{P}} c_H(x, \hat{P})$;

$P \leftarrow \hat{P} \setminus \{z\}$;

objective optimization. We consider the case $\mu = n + 1$, i.e. the population can cover the whole Pareto front and investigate how a population of maximum search space diversity covering the whole front can be obtained by $(\mu + 1)$ -SIBEA incorporating a search space diversity mechanism. We point out that although the problem is very simple, there are non optimal populations where the diversity can not be improved by 1-bit flips. Considering 2-bit flips, we show that a population of maximal diversity is obtained in expected time $O(n^3 \log n)$.

The paper is structured as follows. In Section 2, we introduce the algorithm and the setup that is subject to our investigations. Section 3 shows our approximation results for the hypervolume indicator and $\mu \leq \sqrt{n}$. Our results for maximizing search space diversity are presented in Section 4 and finally we finish with some concluding remarks.

2. PRELIMINARIES

We start by introducing some basic concepts regarding hypervolume maximization and search space diversity optimization using evolutionary algorithms in the following.

We consider the search space $S = \{0, 1\}^n$, i.e. candidate solutions are bitstrings of length n . In multi-objective optimization, we consider a vector-valued fitness function $f : S \rightarrow \mathbb{R}^m$ where $m \geq 2$ is the number of objectives. In this paper, we assume that all objective functions should be maximized. The fitness of a search point $x \in S$ is given by the vector $f(x) = (f_1(x), \dots, f_m(x))$. We define $f(x) \geq f(x')$ iff $f_i(x) \geq f_i(x')$ for all $i \in \{1, \dots, m\}$. In this case, we say that the objective vector $f(x)$ dominates the objective vector $f(x')$. The set of non-dominated objective vectors is called the Pareto front and the classical goal in multi-objective optimization is to obtain for each objective vector of the Pareto front a corresponding solution. As the Pareto front for most problems is to large evolutionary multi-objective algorithms evolve a set of solutions that covers the Pareto front in a good way.

Throughout this paper, we consider the problem

$$\text{OneMinMax}(x) := (\|x\|_0, \|x\|_1)$$

where the number of zeros ($\|x\|_0$) and ones ($\|x\|_1$) has to be maximized at the same time. The problem has the property that all search points are on the Pareto front and our goal is to study how evolutionary multi-objective algorithms can obtain diverse sets of solutions with respect to the search- and objective space.

2.1 Hypervolume-Based Algorithm

The hypervolume indicator measures a set of elements in \mathbb{R}^m (corresponding to images of elements in S) with the volume of the dominated portion of the objective space. In particular, given a reference point $r \in \mathbb{R}^m$, the hypervolume indicator is defined on a set $P \subset S$ as

$$I_H(P) = \lambda \left(\bigcup_{x \in P} [f_1(x), r_1] \times [f_2(x), r_2] \times \dots \times [f_m(x), r_m] \right)$$

where $\lambda(S)$ denotes the Lebesgue measure of a set S and $[f_1(a), r_1] \times [f_2(a), r_2] \times \dots \times [f_m(a), r_m]$ is the orthotope with $f(a)$ and r in opposite corners.

We define the contribution of an element $x \in P$ to the hypervolume of a set of elements P as

$$c_H(x, P) = I_H(P \setminus \{x\}) - I_H(P)$$

We analyse how the $(\mu + 1)$ -SIBEA (see Algorithm 1) optimizes the hypervolume indicator for OneMinMax. The algorithm starts with a set P of μ solutions and produces in each iteration from a randomly chosen individual $x \in P$ one offspring x' by mutation resulting in a population $\hat{P} = P \cup \{x'\}$. The mutation operator considered throughout this paper is standard-bit mutation which flips each bit of the parent individual x with probability $1/n$. In order to obtain the population of the new generation an individual $z \in \hat{P}$ with minimal hypervolume contribution is discarded.

We analyze how well $(\mu + 1)$ -SIBEA approximates the Pareto front when $\mu \ll n$. To measure runtime, we count the expected number of iterations of the repeat-loop in Algorithm 1 (called expected time) until the algorithm has achieved a population of desired quality. Previous work by Nyugen, Sutton and Neumann [11] showed that finding the absolutely best approximation can take long, so that no advantage over other algorithms computing the full front was visible. We show in Sections 3 that relatively good approximations, still having a $1 - o(1)$ fraction of the maximum possible hypervolume, can be computed highly efficiently.

Since our approximations will be very good (they all lose only a lower order term on the best-possible approximation), and since the real point of comparison is not the best-possible approximation achievable with μ point, but rather the true front, it will be more convenient to talk about the discrepancy to the ideal solution rather than the hypervolume achieved by our solutions. Assume that the multi-objective optimization problem under consideration has a Pareto front F , which gives rise to a hypervolume of $I_H(F)$. Then for any set of feasible solutions P , we define its (*hypervolume*-) *discrepancy* to be $\text{disc}(P) := I_H(F) - I_H(P)$. Note that whenever $\mu < |F|$, each set $|P|$ with $|P| = \mu$ has a positive discrepancy.

2.2 Search Space Diversity Optimization

To study search space diversity optimization for OneMinMax we consider a population size that is able to cover the whole Pareto front, i.e. $\mu \geq n + 1$. We analyze $(\mu + 1)$ -SIBEA with a search-space diversity mechanism and study the time until it has produced a population that is diverse with respect to the underlying search space.

There are many ways to measure the difference between different individuals. As discussed in [9], the diversity measurement should have the three properties of twinning, monotonicity in varieties and monotonicity in distance.

Since pseudo-Boolean functions are defined on bit-strings, we use Hamming distance

$$H(x, y) = \sum_{i=1}^n |x_i - y_i|,$$

where $x_i, y_i \in \{0, 1\}$, to evaluate the difference between two individuals. The diversity of a set of solutions P is defined as the sum of Hamming distance between each pair of individuals in P . Note that in general P can be a multi-set which may include duplicates. In order to meet the twinning property [14, 15], duplicates are removed when computing the diversity of a (multi-)set P based on the Hamming distance.

DEFINITION 1. For a given population P , the population diversity is defined as $D(P) = \sum_{\{x, y\} \in \hat{P} \times \hat{P}} H(x, y)$, where \hat{P} is the set with all distinct solutions in P .

The diversity optimization is conducted until population covers the whole Pareto-front. The contribution of solution x to the population diversity is defined as

$$c_D(x, P) = D(P) - D(P \setminus \{x\}).$$

Taken both the population diversity and hypervolume indicator into consideration, the contribution of an individual is defined as

$$c(x, P) = (c_H(x, P), c_D(x, P)).$$

For two individuals $x, y \in P$, we define $c(x, P) < c(y, P)$ if $c_H(x, P) < c_H(y, P)$ or $c_H(x, P) = c_H(y, P) \wedge c_D(x, P) < c_D(y, P)$, which indicates y is better than x in quality. And we also define $c(x, P) \leq c(y, P)$ iff $c_H(x, P) \leq c_H(y, P) \wedge c_D(x, P) \leq c_D(y, P)$.

In order to obtain a population which is optimal in both hypervolume indicator and population diversity, we combine the classical $(\mu + 1)$ -SIBEA with the contribution defined above. The $(\mu + 1)$ -SIBEA with solution diversity optimization is defined as $(\mu + 1)$ -SIBEA_D. The whole process of $(\mu + 1)$ -SIBEA_D is given in Algorithm 2.

When considering $(\mu + 1)$ -SIBEA_D, we focus on the aspect of maximizing search space diversity. The selecting process involves the hypervolume contribution as the premier component. It has been shown in [11] that $(\mu + 1)$ -SIBEA computes for each Pareto optimal objective vector a corresponding search point, i. e. covers the whole Pareto front, in time $O(\mu n \log n)$, if $\mu \geq n + 1$. For our investigations regarding search space diversity, we consider population sizes $\mu = n + 1$. As maximizing the hypervolume is premier goal in $(\mu + 1)$ -SIBEA_D, a population containing for each Pareto optimal objective vector, is obtained in time $O(\mu n \log n)$ following the analysis in [11]. We will work under the assumption that such a population has already been obtained and are interested in the expected time until such a population has maximal search space diversity.

We study our algorithm in terms of the number of fitness evaluations until it has produced a population P that has the optimal hypervolume indicator as well as the maximal diversity $D(P)$. The *expected optimization time* refers to the expected number of fitness evaluations to reach this goal. We represent the population in a $\mu \times n$ matrix where each individual is a row. This allows us to point out when a population has maximal diversity.

Algorithm 2: $(\mu + 1)$ -SIBEA_D

Start with an initial population P consisting of μ elements from S .

repeat forever

Select x from P uniformly at random;

$x' \leftarrow \text{mutate}(x)$;

$\hat{P} \leftarrow P \cup \{x'\}$;

Let z be a randomly chosen individual with

$c(z, \hat{P}) = \min_{x \in \hat{P}} c(x, \hat{P})$;

$P \leftarrow \hat{P} \setminus \{z\}$;

3. APPROXIMATING THE PARETO FRONT

In this section, we study Algorithm 1 for $\mu \leq \sqrt{n}$. We show that $(\mu + 1)$ -SIBEA can compute approximations to the Pareto front with discrepancy of this optimal order. Note that the constant factor gap remaining to the absolute optimal is not a real problem. By running the $(\mu + 1)$ -SIBEA with a larger μ value (larger by a suitable constant factor, leading to a constant factor increase of the runtime), we can much easier reduce the discrepancy than by trying to find the optimal approximation for a fixed μ . The main result of this section is as follows.

THEOREM 1. Let $\mu \leq \sqrt{n}$. Then $(\mu + 1)$ -SIBEA in an expected number of $O(\mu n \log n)$ iterations computes a set of μ points having hypervolume-discrepancy at most $2n^2/\mu$.

Note that for OneMinMax any search point x lies on the Pareto front F and we have $\|x\|_0 + \|x\|_1 = n$ for any search point. The position $f(x)$ of any point in the objective space is determined by its first coordinate $f_1(x) = \|x\|_0$. Therefore, we can uniquely describe the position $f(P)$ of any set P of μ points in the objective space via the length of the intervals formed by consecutive points. More precisely, let $P = \{x^{(1)}, \dots, x^{(\mu)}\}$ be a set of μ search points. Assume that P is sorted increasingly with respect to the first objective, that is, for all $i \in [2.. \mu]$ we have $f_1(x^{(i-1)}) \leq f_1(x^{(i)})$. Define the length of the interval formed by these two search points on the front by $\ell_i := f_1(x^{(i)}) - f_1(x^{(i-1)})$. Define further the artificial interval lengths $\ell_1 = f_1(x^{(1)})$ and $\ell_{\mu+1} = n - f_1(x^{(\mu)})$. Then $f_1(x^{(i)}) = \sum_{j=1}^i \ell_j$ for all $i \in [1.. \mu]$, hence the position of P in the objective space is indeed fully described by the $\ell = (\ell_1, \dots, \ell_{\mu+1})$.

From ℓ we can easily compute the discrepancy.

LEMMA 2. Let P be a population, then its discrepancy is $\text{disc}(P) = \sum_{i=1}^{\mu+1} \frac{\ell_i^2}{2} + (-r_1)\ell_1 + (-r_2)\ell_{\mu+1} - n/2$.

PROOF. We compute

$$I_H(F) = (-r_1 \cdot (n - r_2)) + (-r_2 \cdot (n - r_1)) + \frac{n^2}{2} - \frac{n}{2}$$

and

$$I_H(P) = (-r_1 \cdot (n - r_2 - \ell_1)) + (-r_2 \cdot (n - r_1 - \ell_{\mu+1})) + \frac{n^2}{2} - \sum_{i=1}^{\mu+1} \frac{\ell_i^2}{2}.$$

Therefore, we have

$$\begin{aligned} \text{disc}(P) &= I_H(F) - I_H(P) \\ &= \sum_{i=1}^{\mu+1} \frac{\ell_i^2}{2} + (-r_1)\ell_1 + (-r_2)\ell_{\mu+1} - n/2. \end{aligned}$$

□

The following lemma shows that $(\mu + 1)$ -SIBEA is always able to achieve a good progress in terms of reducing the discrepancy as long as the discrepancy is at least $2n^2/\mu$. For simplicity, we assume $r = (0, 0)$ for now and generalize to other reference points afterwards.

LEMMA 3. *Let $r = (0, 0)$ and P be the current population with $\text{disc}(P) \geq 2n^2/\mu$. Then $E(\text{disc}(P')) \leq (1 - \frac{1}{2e\mu n}) \text{disc}(P)$ holds for the population P' of the next generation.*

PROOF. We first observe that the discrepancy, since it basically is the negative of the hypervolume, cannot increase during a run of the $(\mu + 1)$ -SIBEA. Therefore, we can show an upper bound for $\text{disc}(P')$ by regarding suitable events that decrease the discrepancy, ignoring all others. The events we regard is that the point $x^{(i)}$ moves one step into the direction of the larger adjacent interval, given that this exists.

More precisely, let $i \in [1..\mu]$. If $\ell_i < \ell_{i+1}$, let E_i be the event that the $(\mu + 1)$ -SIBEA picks $x^{(i)}$ for variation and flips in $x^{(i)}$ exactly one bit and this flips from one to zero. The probability for this event is $\Pr(E_i) = (1/\mu)(1 - 1/n)^{n-1}((n - \|x^{(i)}\|_0)/n) \leq \frac{1}{e\mu n}(n - \|x^{(i)}\|_0)$. If this new individual replaces its parent, then the new population P' is identical to P except that $\ell'_i = \ell_i + 1$ and $\ell'_{i+1} = \ell_{i+1} - 1$. Consequently,

$$\begin{aligned} \text{disc}(P') &= \text{disc}(P) + \frac{1}{2}((\ell_i + 1)^2 - \ell_i^2 + (\ell_{i+1} - 1)^2 - \ell_{i+1}^2) \\ &= \text{disc}(P) - (\ell_{i+1} - \ell_i - 1) \end{aligned}$$

by Lemma 2. Since we assumed $\ell_i < \ell_{i+1}$, such a step does not increase discrepancy and is therefore accepted. If the new individual does not replace its parent, then this is for the reason that removing another individual leads to an at least as small discrepancy. Consequently, the new (and accepted) population P' again fulfills $\text{disc}(P') \leq \text{disc}(P) - (\ell_{i+1} - \ell_i - 1)$.

Similarly, if $\ell_i > \ell_{i+1}$, let E_i be the event that the $(\mu + 1)$ -SIBEA picks $x^{(i)}$ for variation and flips in $x^{(i)}$ exactly one bit and this flips from zero to one. The probability for this event is $\Pr(E_i) = (1/\mu)(1 - 1/n)^{n-1}(\|x^{(i)}\|_0/n) \geq \frac{1}{e\mu n}\|x^{(i)}\|_0$, and it results in a population P' with $\text{disc}(P') \leq \text{disc}(P) - (\ell_i - \ell_{i+1} - 1)$.

Based on the above discussion, let us define $\delta_i = (n - \|x^{(i)}\|_0)(\ell_{i+1} - \ell_i - 1)$, if $\ell_i < \ell_{i+1}$, and $\delta_i = \|x^{(i)}\|_0(\ell_i - \ell_{i+1} - 1)$, if $\ell_i > \ell_{i+1}$. If $\ell_i = \ell_{i+1}$, let $\delta_i = 0$. Then

$$E(\text{disc}(P')) \leq \text{disc}(P) - \frac{1}{e\mu n} \sum_{i=1}^{\mu} \delta_i.$$

Note that $\|x^{(i)}\|_0 = \sum_{j=1}^i \ell_j$, so the δ_i can be expressed solely via the ℓ_j . Therefore, let us define $\delta = \delta(\ell_1, \dots, \ell_{\mu+1}) := \sum_{i=1}^{\mu} \delta_i$, where the δ_i are defined as at the beginning of this paragraph.

We now argue that δ is smallest when the ℓ_i are sorted decreasingly. To this aim, consider a fixed sequence of $\ell = (\ell_1, \dots, \ell_{\mu+1})$. Let k be maximal such that $\ell_j \geq \ell_{j+1}$ for

all $j \in [1..k - 1]$. In other words, k is minimal subject to $\ell_k < \ell_{k+1}$. Let b be maximal such that $\ell_{k+1} > \ell_b$. Define $\ell' = (\ell_1, \dots, \ell_{b-1}, \ell_{k+1}, \ell_b, \dots, \ell_k, \ell_{k+2}, \dots, \ell_{\mu+1})$. In other words, ℓ' is obtained from sorting the first $k + 1$ entries of ℓ , which means removing the $(k + 1)$ st entry and inserting it between the $(b - 1)$ st and b th entry. We show that $\delta' = \delta(\ell')$ is not larger than $\delta = \delta(\ell)$. We use the shorthands $(\dots)_0 := \max\{\dots, 0\}$, $L_i := \sum_{j=1}^i \ell_j$, and $L'_i := \sum_{j=1}^i \ell'_j$. We start with the case the $\ell_{k+2} \leq \ell_k$ (and thus also $\ell_{k+2} < \ell_{k+1}$). Then

$$\begin{aligned} \delta - \delta' &= \sum_{i=b-1}^{k+1} (\delta_i - \delta'_i) \\ &= L_{b-1}((\ell_{b-1} - \ell_b - 1)_0 - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\ &\quad - \delta'_b \\ &\quad + \sum_{i=b}^{k-1} (L_i - L'_{i+1})(\ell_i - \ell_{i+1} - 1)_0 \\ &\quad + \delta_k \\ &\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 - (\ell_k - \ell_{k+2} - 1)_0) \\ &= L_{b-1}(\ell_{b-1} - \ell_b - 1 - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\ &\quad - (L_{b-1} + \ell_{k+1})(\ell_{k+1} - \ell_b - 1) \\ &\quad - \sum_{i=b}^{k-1} \ell_{k+1}(\ell_i - \ell_{i+1} - 1)_0 \\ &\quad + (n - L_k)(\ell_{k+1} - \ell_k - 1) \\ &\quad L_{k+1}(\ell_{k+1} - \ell_{k+2} - 1 - (\ell_k - \ell_{k+2} - 1)_0) \\ &= L_{b-1}(\ell_{b-1} - \ell_{k+1} - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\ &\quad - \ell_{k+1}(\ell_{k+1} - \ell_b - 1) \\ &\quad - \sum_{i=b}^{k-1} \ell_{k+1}(\ell_i - \ell_{i+1} - 1)_0 \\ &\quad + n(\ell_{k+1} - \ell_k - 1) \\ &\quad + L_k(\ell_{k+1} - \ell_{k+2} - 1 - (\ell_k - \ell_{k+2} - 1)_0) \\ &\quad - (\ell_{k+1} - \ell_k - 1) \\ &\quad + \ell_{k+1}(\ell_{k+1} - \ell_{k+2} - 1 - (\ell_k - \ell_{k+2} - 1)_0). \end{aligned} \tag{1} \tag{2} \tag{3} \tag{4} \tag{5} \tag{6} \tag{7} \tag{8} \tag{9} \tag{10} \tag{11}$$

The terms (7), (9), and (10)+(11) are easily seen to be non-negative. (8) can be estimated by $-\sum_{i=b}^{k-1} \ell_{k+1}(\ell_i - \ell_{i+1} - 1)_0 \geq -\sum_{i=b}^{k-1} \ell_{k+1}(\ell_i - \ell_{i+1}) \geq -\ell_{k+1}(\ell_b - \ell_k)$. Consequently,

$$\begin{aligned} \delta - \delta' &\geq \ell_{k+1}(-(\ell_{k+1} - \ell_b - 1) - (\ell_b - \ell_k) + \ell_{k+1} \\ &\quad - \ell_{k+2} - 1 - (\ell_k - \ell_{k+2} - 1)_0) \\ &= \ell_{k+1}(\ell_k - \ell_{k+2} - (\ell_k - \ell_{k+2} - 1)_0) \geq 0. \end{aligned}$$

We continue with the case that $\ell_{k+2} > \ell_{k+1}$, implying $\ell_{k+2} > \ell_k$. We start the same computation as above, but have to replace (4) and (6) by $(n - L_{k+1})(\ell_{k+2} - \ell_{k+1} - 1)_0 - (\ell_{k+2} - \ell_k - 1)_0 = (n - L_{k+1})(\ell_k - \ell_{k+1})$. Continuing with this from

(5) on, we obtain

$$\begin{aligned}
\delta - \delta' &= L_{b-1}(\ell_{b-1} - \ell_{k+1} - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\
&\quad - \ell_{k+1}(\ell_{k+1} - \ell_b - 1) \\
&\quad - \sum_{i=b}^{k-1} \ell_{k+1}(\ell_i - \ell_{i+1} - 1)_0 \\
&\quad + n + L_k(\ell_k - \ell_{k+1} + 1) - L_{k+1}(\ell_k - \ell_{k+1}) \\
&\geq 0 \\
&\quad - \ell_{k+1}(\ell_{k+1} - \ell_b - 1) \\
&\quad - \ell_{k+1}(\ell_b - \ell_k) \\
&\quad + n + \ell_{k+1}(\ell_{k+1} - \ell_k - 1) = n > 0.
\end{aligned}$$

Let us finally consider the case $\ell_k < \ell_{k+2} \leq \ell_{k+1}$. Again, nothing changes except (4), which now is

$$\begin{aligned}
\delta_{k+1} - \delta'_{k+1} &= L_{k+1}(\ell_{k+1} - \ell_{k+2} - 1)_0 - (n - L_{k+1})(\ell_{k+2} - \ell_k - 1)_0 \\
&= -n(\ell_{k+2} - \ell_k - 1) + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 \\
&\quad + \ell_{k+2} - \ell_k - 1).
\end{aligned}$$

Together with (3), this becomes

$$\begin{aligned}
\delta_k + \delta_{k+1} - \delta'_{k+1} &= (n - L_k)(\ell_{k+1} - \ell_k - 1) - n(\ell_{k+2} - \ell_k - 1) \\
&\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 + \ell_{k+2} - \ell_k - 1) \\
&= n(\ell_{k+1} - \ell_{k+2}) - L_k \ell_{k+1} + \ell_k(L_k - L_{k+1}) \\
&\quad + L_k - L_{k+1} + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 + \ell_{k+2}) \\
&= n(\ell_{k+1} - \ell_{k+2}) - L_k \ell_{k+1} + \ell_k \ell_{k+1} - \ell_{k+1} \\
&\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 + \ell_{k+2}).
\end{aligned}$$

Adding (1) to (2), we obtain

$$\begin{aligned}
\delta - \delta' &= L_{b-1}(\ell_{b-1} - \ell_{k+1} - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\
&\quad - \ell_{k+1}(\ell_{k+1} - \ell_b - 1) \\
&\quad - \sum_{i=b}^{k-1} \ell_{k+1}(\ell_i - \ell_{i+1} - 1)_0 \\
&\quad + n(\ell_{k+1} - \ell_{k+2}) - L_k \ell_{k+1} + \ell_k \ell_{k+1} - \ell_{k+1} \\
&\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 + \ell_{k+2}) \\
&= L_{b-1}(\ell_{b-1} - \ell_{k+1} - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\
&\quad + \ell_{k+1}(-\ell_{k+1} + \ell_b + 1 - \ell_b + \ell_k - L_k - \ell_k - 1) \\
&\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 + \ell_{k+2}) \\
&\quad + n(\ell_{k+1} - \ell_{k+2}) \\
&= L_{b-1}(\ell_{b-1} - \ell_{k+1} - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\
&\quad - L_{k+1} \ell_{k+1} \\
&\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0 + \ell_{k+2}) \\
&\quad + n(\ell_{k+1} - \ell_{k+2}) \\
&= L_{b-1}(\ell_{b-1} - \ell_{k+1} - (\ell_{b-1} - \ell_{k+1} - 1)_0) \\
&\quad + (n - L_{k+1})(\ell_{k+1} - \ell_{k+2}) \\
&\quad + L_{k+1}((\ell_{k+1} - \ell_{k+2} - 1)_0) \geq 0.
\end{aligned}$$

Let us return to our population P at the beginning of the round regarded. We recall that its position in the objective space is uniquely described by the sequence of the lengths $\ell_1, \dots, \ell_{\mu+1}$ of intervals formed by the individuals on the front. Let $\ell'_1, \dots, \ell'_{\mu+1}$ be a decreasing sorting of $\ell_1, \dots, \ell_{\mu+1}$, that is, we have $\ell'_i \geq \ell'_{i+1}$ for all $i \in [1.. \mu]$ and the multisets

formed by the two sequences are identical. Note that ℓ' can be obtained from ℓ by a sequence of operations of the type "remove the first element offending the decreasing order and insert it to the left of the first element smaller than it. Above we showed that any such operation does not increase the δ -value of the sequence. Consequently, we have $\delta(\ell') \leq \delta(\ell)$. Consequently, we can estimate the drift from population P as follows.

$$\begin{aligned}
E(\text{disc}(P')) &\leq \text{disc}(P) - \frac{\delta}{e\mu n} \\
&\leq \text{disc}(P) - \frac{\delta'}{e\mu n}.
\end{aligned}$$

Note that when $r = (0, 0)$ as assumed, $\text{disc}(P)$ only depends on the set of interval lengths, so changing the order of the intervals does not change the discrepancy. Therefore, we can simply assume that P already gives rise to a sorted sequence of interval lengths. We compute

$$\delta = \sum_{i=1}^{\mu} L_i(\ell_i - \ell_{i+1} - 1)_0 \geq \sum_{i=1}^{\mu} L_i(\ell_i - \ell_{i+1}) - \sum_{i=1}^{\mu} L_i$$

and

$$\begin{aligned}
\sum_{i=1}^{\mu} L_i(\ell_i - \ell_{i+1}) &= \sum_{i=1}^{\mu} \left(\sum_{j=1}^i \ell_j \right) (\ell_i - \ell_{i+1}) \\
&= \sum_{j=1}^{\mu} \ell_j \sum_{i=j}^{\mu} (\ell_i - \ell_{i+1}) \\
&= \sum_{j=1}^{\mu} \ell_j (\ell_j - \ell_{\mu}) \\
&\geq \sum_{j=1}^{\mu} \ell_j^2 - n\ell_{\mu}.
\end{aligned}$$

Since ℓ_{μ} is the smallest interval length, it is not larger than the average interval length, that is, we have $\ell_{\mu} \leq n/\mu$. Together with the trivial estimate $L_i \leq n$, we finally have

$$\begin{aligned}
E(\text{disc}(P')) &\leq \text{disc}(P) - \frac{\delta}{e\mu n} \\
&\leq \text{disc}(P) - \frac{1}{e\mu n} \left(\sum_{j=1}^{\mu} \ell_j^2 - n\ell_{\mu} - \sum_{i=1}^{\mu} L_i \right) \\
&\leq \text{disc}(P) - \frac{1}{e\mu n} \left(\sum_{j=1}^{\mu} \ell_j^2 - n(\mu + n/\mu) \right) \\
&\leq \text{disc}(P) - \frac{1}{e\mu n} \left(2 \text{disc}(P) + n - n(\mu + n/\mu) \right)
\end{aligned}$$

If $\mu \leq \sqrt{n}$ and $\text{disc}(P) \geq 2n^2/\mu$, then this yields $E(\text{disc}(P')) \leq (1 - \frac{1}{2e\mu n}) \text{disc}(P)$ as desired. \square

PROOF OF THEOREM 1. To not obscure the main proof arguments by additional technicalities, we only give the proof for the case that $r = (0, 0)$, that is, that the reference point lies in the origin. There is no doubt that all other cases can be handled in generally the same way, however, with more tedious calculations. Since in [11] mainly the case $r = (-n^2, -n^2)$ was regarded, we add a reduction from this case to the case $r = (0, 0)$ at the end of this proof.

We use Lemma 3 together with the multiplicative drift theorem [8]. Since Lemma 3 gives a bound for the drift only for a part of the discrepancy ranges, but the drift theorem requires such a drift in the whole range, we need to artificially prolong the process as follows.

Consider a run of the $(\mu + 1)$ -SIBEA. Let X_0 be the discrepancy of the initial population. Assume that this is at least $2n^2/\mu$ (otherwise we are done). For $t > 0$, define X_t as follows. If the discrepancies of all populations after iterations $0, \dots, t$ were at least $2n^2/\mu$, then let X_t be the discrepancy after round t . Otherwise let $X_t := X_{t-1}$ with probability $(1 - \frac{1}{2e\mu n})$ and $X_t = 0$ else. This defines a random process in the nonnegative integers satisfying $E(X_t) \leq (1 - \frac{1}{2e\mu n})E(X_{t-1})$ for all $t \geq 1$. Also, $X_0 \leq n^2$ by definition of the discrepancy. By the multiplicative drift theorem, it takes at most an expected time of $2e\mu n(1 + \ln(n^2))$ iterations until $X_t = 0$. Consequently, the expected first time for the discrepancy to be at most $2n^2/\mu$, is at most $2e\mu n(1 + \ln(n^2)) = O(\mu n \log n)$.

We now add a quick argument why the above is also sufficient for the case that the reference point is $r = (-n^2, -n^2)$. In this case, it was shown in [11] that the maximal OneMax value in the population never decreases and that the minimal OneMax value in the population never increases. Based on this, it was shown that after an expected number of $O(\mu n \log n)$ iterations, the population contains the extreme points $(0, \dots, 0)$ and $(1, \dots, 1)$. From that point on, knowing that the population continues to contain at least one copy of each extreme point, the discrepancy of all future populations is independent of the reference point (apply Lemma 2 with $\ell_1 = \ell_{\mu+1} = 0$). Consequently, after an initial segment of $O(\mu n \log n)$ iterations, our proof above applies, and we obtain the same $O(\mu n \log n)$ bound for the expected time complexity. \square

4. SEARCH SPACE DIVERSITY OPTIMIZATION

We now turn to population sizes $\mu = n + 1$ and investigate how evolutionary algorithms can optimize search space diversity under the condition that for each Pareto optimal objective vector at least one search point is contained in the population.

The following lemma shows crucial properties of a population maximized in population diversity.

LEMMA 4. *Let $\mu = n + k \leq 2^n$, where $k \geq 1$. If the population P fulfils all of the following properties:*

1. *For each Pareto optimal objective vector v , there is an $s \in P$ with $f(s) = v$.*
2. *There are no duplicated individuals in P .*
3. *Each column of the matrix representing P has either $\lfloor \mu/2 \rfloor$ or $\lceil \mu/2 \rceil$ 1-bits.*

then P is optimal for OneMinMax in population diversity.

PROOF. According to the definition of OneMinMax, there are $(n+1)$ different points in the Pareto-front. Since $\mu \geq n+1$, the individuals in P have to cover the entire Pareto-front in order to be optimal in the population diversity.

Let P be a population of size μ containing no duplicate and P' be the population obtained from P by replacing at least one of its individuals x by a duplicate of the other

$(\mu - 1)$ individuals. According to the monotonicity in varieties property of diversity measurement and Definition 1, we have $D(P) > D(P')$ as $\hat{P} = \hat{P}' \cup \{x\}$. This implies that no population containing duplicates can be optimal if $\mu \leq 2^n$.

Let matrix M represent a population P that does not contain any duplicates. We show that P has maximal diversity among all populations containing no duplicates if it contains $\lfloor \mu/2 \rfloor$ or $\lceil \mu/2 \rceil$ 0-bits in each column.

The contribution of each column has no influence on any other column. Hence, the population diversity equals to the sum of the diversity contribution of every column in the matrix. The contribution to population diversity of each column can be written as $m_i(\mu - m_i)$, where m_i represents the number of 1's in the i th column and the overall population diversity of P is given by

$$\sum_{i=1}^n m_i(\mu - m_i).$$

The quadratic continuous function $g(x) = x(\mu - x)$ has the global maximum value of $\mu^2/4$ when $x = \mu/2$. This implies that the maximum is attained for $x = \lfloor \mu/2 \rfloor$ and $x = \lceil \mu/2 \rceil$ when restricting the inputs of g to integers. Hence, P has maximal diversity if it contains $\mu/2$ 1-bits in each column if μ is even. In the case that μ is odd, P has maximal diversity if each column has either $\lfloor \mu/2 \rfloor$ or $\lceil \mu/2 \rceil$ 1-bits. \square

Our proof for optimizing search space diversity will rely on 2-bit flips. We first show that there are non-optimal populations where $(\mu + 1)$ -SIBEA_D is not able to achieve any progress when restricted to 1-bit flips. Lemma 4 suggests that the population with maximum diversity and full coverage of the Pareto-front should have balanced number of 1-bits and 0-bits if $\mu = n + 1$ is even. We will see in the proof for $(\mu + 1)$ -SIBEA_D that this is exactly the case. Although 1-bit flip can improve the population diversity in most cases, there are some situations where there does not exist 1-bit flip that can increase the population diversity.

Some examples are included as Figure 1. The populations shown in the example are almost balanced in all columns in numbers of 1-bits and 0-bits but there is no 1-bit flip which can improve the population diversity to optimality. In the first population in Figure 1, there are only two columns which are not balanced in the numbers of 0-bits and 1-bits, which are the 1st and 3rd column. Either increasing the number of 1-bits or decreasing the number of 0-bits will improve the population diversity. On the contrary, touching the other columns will decrease the population diversity. Since the offspring after 1-bit flip can only replace the individual with the same objectives in order to keep the coverage of the Pareto-front, the change caused by a 1-bit flip depends on the Hamming distance between the selected individual and its neighbours in the objective space. The change to the population diversity caused by a 1-bit flip on individual z can be represented as,

$$c(z) = S_- - S_+ - \frac{1}{2}(H(z, z') + 1),$$

where S_- and S_+ denotes the total number of 1-bits in the columns has one 1-bit decreased and increased respectively. And $H(z, z')$ represents the Hamming distance between the original individual z and the neighbour z' which got replaced by the offspring.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	0	0	0	1	0	1	1	0	0
0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	4	5	4	4	4	4	4	4	4	5	4	4	4	3	4	4	4	5	4	4	4	4	3

Figure 1: The 8×7 matrices represent some populations for which there is no 1-bit flip can improve the population diversity. The last rows report the numbers of 1-bits in corresponding columns.

For the example in Figure 1, in order to increase the contribution to the population diversity, an offspring should fulfil the requirement of $c(z) > 0$, which means the columns except for 1st and 3rd ones should all remain balanced and the balance rate of these two columns should be increased. It is impossible to improve the population diversity since there is no offspring that is able to increase the contribution of these two certain columns without decreasing the contribution of the other columns. Flipping the first 0-bit in the all 0 bitstring or the third 1-bit in all 1 bitstring can obtain offspring which has the same contribution to the population diversity which is acceptable by the algorithm. However that event will lead to another population with the same population diversity and there are still these two mutations do not decrease the population diversity, which is the same situation as previous. Therefore no further improvement can be achieved by 1-bit flip for this population.

Since flipping one bit cannot guarantee the population diversity to be maximized, we focus on 2-bit flip to fulfil the task.

LEMMA 5. *If $\mu = n + 1$ and population diversity is not maximal, then there always exists at least one 2-bit flip in an individual to improve the population diversity.*

PROOF. By construction of the Algorithm 2, when $\mu = n + 1$, there should exist exactly one individual in the population which refers to each point in the Pareto-front, as proved in Lemma 4. The event that selecting one individual and flipping a 1-bit and a 0-bit of it results in an offspring with the same objective value as its parent. The offspring can only replace its parent and this replacement only happens when the offspring has a larger contribution to the population diversity.

As proved in Lemma 4, in a matrix representing a population which does not have optimal population diversity, there must exist two columns that the number of 0-bits in one column is greater than that of the other. Let the number of 0-bits in these two columns be s_1 and s_2 , where $s_1 > s_2$ and both s_1 and s_2 are integers. The overall contribution of the two columns to the population diversity is $s_1 \cdot (\mu - s_1) + s_2 \cdot (\mu - s_2)$. Since $s_1 < s_2$, there must exist at least one row where there is 0-bit and 1-bit in corresponding columns. Flipping the certain two bits does not affect the contribution of other columns to the population diversity. The overall contribution after the event should be

$$(s_1 - 1)(\mu - s_1 + 1) + (s_2 + 1)(\mu - s_2 - 1).$$

Therefore, the change of contribution is $2(s_1 - s_2) - 1$. Since s_1 and s_2 are integers, $s_1 - s_2 \geq 1$. Hence, we get $2(s_1 - s_2) - 1 \geq 1$.

Since the offspring is only compared with its parent, it is impossible to introduce change to the other columns except for these two columns. Therefore, there must exist at least one two-bit flip that should increase the population diversity. \square

We can now prove our main result on search space diversity maximization for OneMinMax and show that the $(\mu + 1)$ -SIBEA_D obtains an optimal population in expected time $O(n^3 \log n)$.

THEOREM 6. *Let $\mu = n + 1$, the expected optimization time of $(\mu + 1)$ -SIBEA_D on OneMinMax is upper bounded by $O(n^3 \log n)$.*

PROOF. The algorithm $(\mu + 1)$ -SIBEA obtains a population of maximum hypervolume if $\mu \geq n + 1$ in expected time $O(\mu n \log n)$ as shown in [11]. We assume that a population of maximal hypervolume has already been obtained and investigate how search space diversity is optimized. The Multiplicative Drift Theorem [8] is used to prove the expected runtime bound.

Define $X^{(t)} = D_{OPT} - D(P)$ and $X^{(t+1)} = D_{OPT} - D(P')$, where D_{OPT} denotes the maximum value of the population diversity and P' represents the population in the next generation of P .

Assume at time t there are k 2-bit flips that can improve the population diversity to optimality no matter in what order these k 2-bit flips happen. Such set of events exist for all populations which are not maximized in population diversity if the 2 bits selected is a 1-bit from the columns with more than average number of 1-bits and a 0-bit from the columns with less than average number of 1-bits. The average number of 1-bits refers to $(n + 1)/2$ when n is odd and $n/2$ when n is even. According to Lemma 5, such 2-bit flip always exists before the population diversity is optimized. As long as the certain columns are not balanced in number of 1-bits and 0-bits, the 2-bit flip can improve the population diversity.

According to the algorithm, flipping these certain bits of an individual does not affect the other individuals in the population. The numbers of 1-bits remain the same except for the two columns, therefore among the other $(k - 1)$ 2-bit flips, the 2-bit flips involving bits in the other $(n - 2)$ columns are still available. Since the k 2-bit flips are selected to improve the population diversity to optimality, if there exist a 2-bit flip involves the two columns, the number of 1-bits of the column should still be unbalanced after the previous event. According to Lemma 5, the other 2-bit flips can improving the contribution to population diversity of the columns and then the flips are acceptable. Then the order of the k 2-bit flips does not affect the improvement.

Hence, the k 2-bit flips can be done in any order and result in a population with maximized population diversity as assumption.

The probability for an individual to be selected and two certain bits flipped is $\frac{1}{\mu} \cdot \frac{1}{n} \cdot \frac{1}{n} \cdot (1 - \frac{1}{n})^{n-2} \geq \frac{1}{e\mu n^2}$. The probability for one of the k 2-bit flip happen is at least $k \cdot \frac{1}{e\mu n^2}$. The average expected improvement by the k 2-bit flips is

$$\frac{D_{OPT} - D(P)}{k} = \frac{X^{(t)}}{k}.$$

Then the drift can be represented as

$$E[X^{(t)} - X^{(t+1)}] \geq k \frac{1}{e\mu n^2} \cdot \frac{X^{(t)}}{k} = \frac{X^{(t)}}{e\mu n^2}.$$

In the worst case, the population in the beginning is with the most unbalance rate. It is clear that $s = X^{(t)} \leq D_{OPT}$. The maximum population diversity when $\mu = n + 1$ is $\mu \cdot (\frac{n+1}{2})^2 = \frac{(n+1)^3}{4}$. Therefore we can get $s_0 \leq \frac{(n+1)^3}{4}$ and $s_{min} = 1$.

According to the Theorem 3 in [8], the expected runtime for maximizing the population diversity on OneMinMax is

$$E[T] \leq e\mu n^2 (1 + \ln(s_0/s_{min})) = O(n^3 \log n).$$

This completes the proof. \square

5. CONCLUSIONS

With this paper, we have contributed to the theoretical understanding of diversity mechanisms in evolutionary multi-objective optimization by means of rigorous runtime analysis. We have studied a baseline algorithm called $(\mu + 1)$ -SIBEA for the problem OneMinMax. Complementing the investigations of who have shown that for a population size of $\mu = \sqrt{n}$ where $(\mu + 1)$ -SIBEA is not able to achieve the optimal hypervolume, we have shown that the algorithm is able to achieve a good approximation of the optimal hypervolume in time $O(\mu n \log n)$ for all $\mu \leq \sqrt{n}$. For a population size of $\mu = n + 1$, we have investigated $(\mu + 1)$ -SIBEA in connection with a search space diversity mechanism and shown that the algorithm obtains a population of maximal search space diversity covering the whole Pareto front in expected time $O(n^3 \log n)$.

Acknowledgements

Frank Neumann was supported by Australian Research Council grants DP140103400, DP160102401.

References

- [1] A. Auger, J. Bader, and D. Brockhoff. Theoretically investigating optimal μ -distributions for the hypervolume indicator: First results for three objectives. In *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference*, pages 586–596, 2010.
- [2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal μ -distributions and the choice of the reference point. In *Foundations of Genetic Algorithms, 10th ACM SIGEVO International Workshop, FOGA 2009*, pages 87–102, 2009.
- [3] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2011.
- [4] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653 – 1669, 2007.
- [5] D. Brockhoff, T. Friedrich, and F. Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature - PPSN X, 10th International Conference*, pages 651–660, 2008.
- [6] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.
- [7] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.
- [8] B. Doerr, D. Johannsen, and C. Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012.
- [9] W. Gao and F. Neumann. Runtime analysis for maximizing population diversity in single-objective optimization. In *Genetic and Evolutionary Computation Conference, GECCO 2014*, pages 777–784, 2014.
- [10] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.
- [11] A. Q. Nguyen, A. M. Sutton, and F. Neumann. Population size matters: Rigorous runtime results for maximizing the hypervolume indicator. *Theoretical Computer Science*, 561, Part A:24 – 36, 2015. Genetic and Evolutionary Computation.
- [12] O. M. Shir, M. Preuss, B. Naujoks, and M. T. M. Emmerich. Enhancing decision space diversity in evolutionary multiobjective algorithms. In *Evolutionary Multi-Criterion Optimization, 5th International Conference, EMO 2009*, pages 95–109, 2009.
- [13] R. Tran, J. Wu, C. Denison, T. Ackling, M. Wagner, and F. Neumann. Fast and effective multi-objective optimisation of wind turbine placement. In *Genetic and Evolutionary Computation Conference, GECCO 2013*, pages 1381–1388, 2013.
- [14] T. Ulrich, J. Bader, and L. Thiele. Defining and optimizing indicator-based diversity measures in multiobjective search. In *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference*, pages 707–717, 2010.
- [15] T. Ulrich, J. Bader, and E. Zitzler. Integrating decision space diversity into hypervolume-based multiobjective search. In *Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings*, pages 455–462, 2010.
- [16] T. Ulrich and L. Thiele. Maximizing population diversity in single-objective optimization. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011*, pages 641–648, 2011.
- [17] F. Zheng, A. R. Simpson, and A. C. Zecchin. Improving the efficiency of multi-objective evolutionary algorithms through decomposition: An application to water distribution network design. *Environmental Modelling and Software*, 69:240–252, 2015.