Runtime Analysis for the NSGA-II: Proving, Quantifying, and Explaining the Inefficiency for Many Objectives

Weijie Zheng, Member, IEEE, and Benjamin Doerr

Abstract—The nondominated sorting genetic algorithm II (NSGA-II) is one of the most prominent algorithms to solve multiobjective optimization problems. Despite numerous successful applications, several studies have shown that the NSGA-II is less effective for larger numbers of objectives. In this work, we use mathematical runtime analyses to rigorously demonstrate and quantify this phenomenon. We show that even on the simple $m$-objective generalization of the discrete OneMinMax benchmark, where every solution is Pareto optimal, the NSGA-II also with large population sizes cannot compute the full Pareto front (objective vectors of all Pareto optima) in subexponential time when the number of objectives is at least three. The reason for this unexpected behavior lies in the fact that in the computation of the crowding distance, the different objectives are regarded independently. This is not a problem for two objectives, where any sorting of a pairwise incomparable set of solutions according to one objective is also such a sorting according to the other objective (in the inverse order).

Index Terms—Many-objective optimization, nondominated sorting genetic algorithm II (NSGA-II), runtime analysis, theory of computing.

I. INTRODUCTION

MANY real-world optimization problems have several, usually conflicting objectives. In this situation, it is not possible to simply compute one good solution. Instead, a common solution concept for such multiobjective problems is to compute a representative set of Pareto solutions (solutions which are not strictly dominated by others) and let a human decision maker choose one of these. The two dominant approaches to such problems are mathematical programming and evolutionary algorithms (EAs). The latter profits from the fact that evolutionary algorithms naturally work with sets of solutions (“populations”).

The most prominent multiobjective EA (MOEA) with many successful applications in various domains is the nondominated sorting genetic algorithm II (NSGA-II) [1], see [2] or the more than 40 000 citations on Google scholar. Despite numerous positive results on the NSGA-II, several studies have observed that the NSGA-II is less effective when the number of objectives grows. From their experimental comparison of three MOEAs, Khare et al. [3] reported that the NSGA-II increasingly suffers in converging to the global Pareto front when the number of objectives increases from 2 to 8. This early study is a pure performance comparison, so no explanations for the different behaviors were sought. Also, it should be noted that the other algorithms regarded also suffered in different ways from growing numbers of objectives. Purshouse and Fleming [4] aimed for further empirical observations. They studied the range of suitable configurations of the NSGA-II that can result in good solutions. That the suitable ranges shrink along with the increasing number of objectives, indicates the increasing difficulty for the NSGA-II for more objectives. They also observed some factors that might be relevant for the poor performance of the NSGA-II, like that when the population evolves, the proportion of nondominated solutions in the population rapidly increases to 100%, and that in each iteration, only a low proportion of the newly generated solutions dominates the current solutions. These observations are regarded as one of the key challenges for many-objective optimizations, see the surveys [5], [6].

Since apparently there is a lack of understanding of the performance of the NSGA-II for many-objective problems, we try to approach this research question via a mathematical runtime analysis [7], see also [8], [9], [10], [11]. Such analyses are an integral part of the theory of heuristic search. While often restricted to simple algorithmic settings, this alternative approach has led to several deep and very reliable (namely, mathematically proven) results in the past, see, e.g., [12], [13], [14], and [15]. Also, often the proofs also reveal the reason why a certain phenomenon can be observed.

To this aim, we conduct a mathematical runtime analysis of the NSGA-II on the $m$-objective version of the classic bi-objective ONEINMAX benchmark. This pseudo-Boolean (that is, defined on bit-strings of length $n$) benchmark is very simple in several respects, for example, any solution is Pareto...
optimal and the objectives are all equivalent to the OneMax benchmark, which is generally considered as the easiest single-objective pseudo-Boolean benchmark [16], [17], [18]. For the bi-objective OneMinMax problem, a good performance of the NSGA-II has been proven recently [19], [20], [21].

Our main (proven) result is that—despite the simplicity of the problem and in drastic contrast to the bi-objective setting—for all numbers \( m \geq 3 \) of objectives the NSGA-II also with large population sizes cannot find the full Pareto front faster than in exponential time. Even worse, we prove that for an exponential time, the population of the NSGA-II will miss a constant fraction of the Pareto front. Our experiments confirm this finding in a very clear manner.

We also give an explanation for the drastic change of behavior between two and three objectives, namely, that in the definition of the crowding distance, the different objectives are regarded independently (we make this point much more precise in the body of this article when all necessary notation is introduced). This is not a problem for two objectives, because for a set of pairwise nondominated solutions a sorting with respect to one objective automatically is a sorting with respect to the other objective (in the opposite order). Hence, here in fact the two objectives automatically are not treated independently. From three objectives on, such a correlation between the objectives does not exist, and this can lead to the problems made precise in this work.

This understanding has two implications. On the negative side, it appears very likely that the difficulties observed for the OneMinMax benchmark will also occur for many other optimization problems, including continuous optimization problems. On the positive side, this understanding suggests to search for an alternative crowding distance measure that does not treat the objectives independently.

We note that given the known difficulties of the NSGA-II with more objectives, several alternatives to the crowding distance have been proposed. For example, Deb and Jain [22] proposed the NSGA-III, where the crowding distance is computed by a system of reference points. In their SMS-EMOA, Beume et al. [23] replaced the crowding distance by the hypervolume. The hypervolume measure was also considered as the second sorting criterion in the multiobjective CMA-ES [24]. It was also used in hybrid with the crowding distance [25]. Instead of removing points with the smallest crowding distances in the NSGA-II, the diversity preservation way via clustering from SPEA [26] was used to select the survived individuals [27], [28]. However, none of these alternatives is as accepted in practice as the NSGA-II. For this reason, we hope that our work can not only spur the development of superior algorithms but also motivate practitioners to try moving from the NSGA-II to these more modern algorithms. We note that very recently [29] a runtime analysis of the NSGA-III appeared that showed that the difficulties proven here for the NSGA-II do not arise with the NSGA-III.

We further note that several promising variations of the crowding distance have been proposed, e.g., [30] and [31], which again are not used a lot in practice. For the latter [31], we easily observe that the NSGA-II with this variant of the crowding distance can effectively optimize the three-objective OneMinMax benchmark. However, since this variant only differs from the traditional crowding distance when objective values occur multiple times in the population, we also immediately see that this variant displays the same unfavorable behavior on Example 1 as the traditional NSGA-II.

II. PREVIOUS WORK

For reasons of brevity, we will not discuss the practical and empirical works on the NSGA-II beyond what we did in the introduction. For the theoretical side, we note that mathematical analyses have always accompanied the design and analysis of evolutionary algorithms. The first (mathematical) runtime analyses of MOEAs have appeared in the early 2000s [32], [33], [34]. They and many of the subsequent works regarded synthetic algorithms like the Semo or global SEMO (GSEMO). While these are much simpler than the algorithms used in practice, they are still close enough to these to admit useful conclusions. It took some time until more realistic MOEAs, such as the hypervolume-based simple indicator-based EA (SIBEA) [35] or the decomposition-based MOEA/D [36], were regarded under the runtime analysis paradigm, and only very recently the first runtime analysis of the NSGA-II was presented [19], [37].

Besides demonstrating that runtime analyses are feasible for this kind of complex algorithms, this work proved that the NSGA-II with population size \( N \) computes the full Pareto front of the biobjective OneMinMax benchmark defined on bit strings of length \( n \) in expected time \( O(Nn \log n) \) if \( N \geq 4(n + 1) \). For the LeadingOnesTrailingZeroses benchmark, a bound of \( O(Nn^2) \) was shown. For the reasonable choice \( N = \Theta(n) \) of the population size, these runtimes agree with previous results for the Semo and Gsemo algorithms. Using a population size strictly larger than the size \( n + 1 \) of the Pareto front is necessary: for \( N = n + 1 \), the NSGA-II will not find the Pareto front for an exponential time, and moreover, miss a constant fraction of it. For smaller population sizes, however, the NSGA-II can still compute good approximations of the Pareto front as shown in [21]. These two works regard a simplified version of the NSGA-II that does not employ crossover. In [20], for the first time a runtime analysis for the NSGA-II with crossover is conducted, however, the runtime bounds proven are not lower than those in [19]. Also, this work lowers the required population size to \( 2(n + 1) \) by assuming that two objectives are sorted inversely. Further, runtime improvements are obtained from a novel selection mechanism. Reference [38] is the first work to conduct a runtime analysis on a multimodal benchmark, the OneJumpZeroJump benchmark [39]. The first lower bounds, matching the previously shown upper bounds for OneMinMax and OneJumpZeroJump, were shown in [40]. That crossover can lead to speed-ups, was proven in [41] and [42]. A first runtime analysis for the NSGA-II on a combinatorial optimization problem, was conducted in [43]. We note that all these works regard problems consisting of two objectives only.

There are some runtime analyses for many-objective problems, however mostly for the very simplistic Semo algorithm.
Already the journal version [44, Sec. V] of the first MOEA runtime analysis [32] contains a many-objective runtime analysis, namely, a proof, that the SEMO covers the Pareto front of mCOCZ and mLOTZ (which are the \( m \)-objective analogues of the classic COUNTING_ONES, COUNTING_ZEROS, and LEADING_ONES, TRAILING_ZEROS benchmarks) with problem size \( n \) and objective number \( m \geq 4 \) in an expected number of \( O(n^{m+1}) \) function evaluations. Some of these bounds were improved in [45], namely, to \( O(n^m) \) for the SEMO on mCOCZ with \( m > 4 \) and to \( O(n^3 \log n) \) for the special case \( m = 4 \). As often in the runtime analysis of MOEAs, the complicated population dynamic prevented the proofs of any interesting lower bounds. In the only other runtime analysis for many-objective problems, Huang et al. [46] analyzed how the MOEA/D optimizes the benchmarks mCOCZ and mLOTZ. As the MOEA/D decomposes the multiobjective problem into several single-objective subproblems and solves these in a co-evolutionary way, this framework is fundamentally different from most MOEAs and in particular the NSGA-II, so we do not discuss these results in more detail.

III. PRELIMINARIES

In this section, we give brief introductions to many-objective optimization and to two kinds of MOEAs (NSGA-II series and SEMO series).

A. Many-Objective Optimization

We call optimization problems with more than two objectives \emph{many-objective optimization} problems. In this article, we consider the maximization of a pseudo-Boolean many-objective optimization problem with \( m \geq 3 \) objectives and problem size of \( n \), \( f = (f_1, \ldots, f_m) : \{0, 1\}^n \rightarrow \mathbb{R}^m \). Dominance is used to compare the quality of solutions. We say that \( x \) weakly dominates \( y \), denoted as \( x \succeq y \), if and only if \( f_i(x) \geq f_i(y) \) for all \( i = 1, \ldots, m \), and we say that \( x \) (strictly) dominates \( y \), denoted as \( x \succ y \), if and only if \( x \succeq y \) and there exists \( i \in \{1, \ldots, m\} \) such that \( f_i(x) > f_i(y) \). We call a solution \( x \) Pareto optimal if there is no solution that strictly dominates it. The function value of the Pareto optimal \( x \) is called a \emph{Pareto front point}. All Pareto front points form the \emph{Pareto front}, denoted by \( F^* \) in this article. We call a Pareto front point \( v \in F^* \) covered by a (multi-)set of solutions \( P \) if there exists \( x \in P \) such that \( v = f(x) \).

In this article, the \emph{aim} is to compute a set of solutions \( P \) that covers the Pareto front, that is, such that for any Pareto front point \( v \in F^* \) there exists \( x \in P \) such that \( v = f(x) \). This aim is widely used in runtime analysis works [44], [45]. We note that this aim is different from obtaining all Pareto optima (the entire efficient set), the target in multimodal optimization [47], [48], [49].

B. Algorithms

For MOEAs, an individual is a solution, that is, a bit-string \( x \in \{0, 1\}^n \), and a population is a multiset of individuals. The \emph{NSGA-II} algorithm [1] has a fixed population size \( N \). After the random initialization, in each iteration (also called “generation”), the offspring population \( Q_i \) with size \( N \) is generated from the current population (parent population) \( P_i \) via parent selection and usually crossover and mutation. We call the parent selection strategy \emph{fair selection} if each individual in the parent population is selected once as a parent to generate an offspring individual. Other parent selection strategies include binary tournament selection, stochastic tournament selection [20], etc. To apply crossover, the NSGA-II selects \( N \) parents and divides them into \( N/2 \) pairs. In [1], with probability 0.1, the parents go directly to the mutation, and otherwise, the algorithm uses the following \emph{one-point crossover}. Independently in each pair, it uniformly at random picks a crossover position \( i \in \{1, \ldots, n\} \), and two parents exchange their first \( i \) bits to form two new parents for the mutation. For the mutation, the \emph{one-bit mutation} operator generates an offspring by flipping one (uniformly at random picked) bit of its parent, and the \emph{standard bitwise mutation} operator generates an offspring by flipping each bit independently with probability \( 1/n \).

After the offspring population is generated, \( N \) individuals are kept among from the combined parent and offspring population \( R_i = P_i \cup Q_i \) of size \( 2N \). The NSGA-II first uses the procedure fast-nondominated-sort() from [1] to partition the population into fronts \( F_1, F_2, \ldots \) where \( F_i \) consists of the nondominated individuals in \( R_i \setminus \bigcup_{j<i} F_j \). Let \( i^* = \min\{i \geq 1 | \bigcup_{j=1}^i F_j \geq N \} \). Then all individuals in \( F_i \) with \( i < i^* \) are kept in the next parent population. For the individuals in the critical \( F_{i^*} \), it computes their crowding distance (see Algorithm 1) and keeps the \( N - |\bigcup_{j<i^*} F_j| \) individuals with largest crowding distance (breaking ties randomly). This original survival selection has some drawbacks and a modified (sequential) one was proposed [30] and proven to be more efficient for the approximation for the bi-objective \emph{ONE_MINMAX} problem [21]. Algorithm 2 describes the procedure of the NSGA-II with the original survival selection and also with the sequential survival selection.

The SEMO and GSEMO (see Algorithm 3) are MOEAs predominantly analyzed in the evolutionary computation theory community. Different from the NSGA-II, they do not work with a fixed population size, but keep any solution until a better one (in the domination sense) is found. They start with a single random solution. Each iteration, one offspring is generated from mutating a random member of the population. It

---

**Algorithm 1** Computation of the Crowding Distance \( cDis(S) \)

**Input:** \( S = \{S_1, \ldots, S_{|S|}\} \), a set of individuals

**Output:** \( cDis(S) = (cDis(S_1), \ldots, cDis(S_{|S|})) \), where \( cDis(S_i) \) is the crowding distance for \( S_i \)

1. \( cDis(S) = (0, \ldots, 0) \)
2. for each objective \( f_i \) do
   3. Sort \( S \) in order of descending \( f_i \) value: \( S_{i,1}, \ldots, S_{i,|S|} \)
   4. \( cDis(S_{i,1}) = +\infty \), \( cDis(S_{i,|S|}) = +\infty \)
5. for \( j = 2, \ldots, |S| - 1 \) do
   6. \( cDis(S_{i,j}) = cDis(S_{i,j}) + \frac{f_i(S_{i,j-1}) - f_i(S_{i,j+1})}{f_i(S_{i,1}) - f_i(S_{i,|S|})} \)
7. end for
8. end for

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Algorithm 2 NSGA-II
1: Uniformly at random generate the initial population \( P_0 = \{x_1, x_2, \ldots, x_N\} \) with \( x_i \in \{0, 1\}^n \), \( i = 1, 2, \ldots, N \).
2: for \( t = 0, 1, 2, \ldots \) do
3: Generate the offspring population \( Q_t \) with size \( N \).
4: Use fast-non-dominated-sort() in [1] to divide \( R_t \) into fronts \( F_1, F_2, \ldots \).
5: Find \( i^* \geq 1 \) such that \( |\bigcup_{i=1}^{i^*-1} F_i| < N \) and \( |\bigcup_{i=1}^{i^*} F_i| \geq N \).
6: Use Algorithm 1 to separately calculate the crowding distance of each individual in \( F_1, \ldots, F_r \).
7: if Original Survival Selection [1] then
8: Let \( \tilde{F}_r \) be the \( N - |\bigcup_{i=1}^{i^*-1} F_i| \) individuals in \( F_r \) with largest crowding distance, chosen at random in case of a tie.
9: \( P_{t+1} = (\bigcup_{i=1}^{i^*-1} F_i) \cup \tilde{F}_r \).
10: else if Sequential Survival Selection [21], [30] then
11: while \( |\bigcup_{i=1}^{i^*} F_i| \neq N \) do
12: Let \( x \) be the individual with the smallest crowding distance in \( F_r \), chosen at random in case of a tie.
13: \( F_r = F_r \setminus \{x\} \).
14: Update the crowding distance of all neighbors of \( x \).
15: end while
16: end if
17: end for

Algorithm 3 SEMO/GSEMO
1: Uniformly at random generate an individual (solution) \( x \in \{0, 1\}^n \), and the initial population \( P_0 = \{x\} \).
2: for \( t = 0, 1, 2, \ldots \) do
3: Uniformly at random select an individual \( x \) from \( P_t \).
4: Generate the offspring \( x' \) by one-bit mutation for the SEMO, or by standard bit-wise mutation for the GSEMO.
5: if there is \( y \in P_t \) such that \( x' \geq y \) then
6: \( P_{t+1} = \{y \in P_t \mid x' \not\geq y\} \cup \{x'\} \).
7: end if
8: end for

is added to the population if it is not strictly dominated by a member of the population, and once it is added, the individuals that are weakly dominated by it will be removed. The only difference between SEMO and GSEMO is that SEMO uses one-bit mutation and GSEMO uses standard bitwise mutation.

IV. INEFFECTIVENESS FOR FOUR AND MORE OBJECTIVES

In this section, we prove our main result that the NSGA-II cannot effectively compute the Pareto front of the \( m \)ONE\( m \)IN\( m \)AX problem (except for the case \( m = 2 \)). Since \( m \) is necessarily even in the definition of \( m \)ONE\( m \)IN\( m \)AX below, we discuss separately in the subsequent section an example of a three-objective \( m \)ONE\( m \)IN\( m \)AX problem that also cannot be solved effectively by the NSGA-II. This shows that the bi-objective setting is structurally different and, in a sense, an exceptional case.

We start by giving the formal definition of the \( m \)-objective version of the \( m \)ONE\( m \)IN\( m \)AX problem, then give an informal description of the reasons for the difficulties of the NSGA-II in \( m \)-objective optimization, and finally prove our main result that the NSGA-II selection with very high-probability loses a constant fraction of the Pareto front (when the population size is larger than the Pareto front size by a constant factor, which can be arbitrarily large). This result then implies that the NSGA-II takes at least a time exponential in the Pareto front size to compute the full Pareto front.

A. \( m \)ONE\( m \)IN\( m \)AX Benchmark

We now define an \( m \)-objective version of the bi-objective \( m \)ONE\( m \)IN\( m \)AX benchmark. The \( m \)ONE\( m \)IN\( m \)AX benchmark, first proposed in [50], is arguably the easiest multiobjective benchmark. Different from the still simple COUNTINGONESCOUNTINGZEROES and LEADINGONESTRAILINGZEROES benchmarks, it has the property that any solution is Pareto optimal. We now define an \( m \)-objective version of it maintaining this property. Recalling the widely used aim in the runtime analysis discussed in Section III-A, this article profits from focussing on the process of covering the Pareto front. Despite the simplicity of the many-objective \( m \)ONE\( m \)IN\( m \)AX problem, in this section, we will show that the NSGA-II cannot efficiently optimize it.

To define an \( m \)-objective version of this benchmark, we proceed in the same fashion that was used in [44] to define the many-objective versions of the COUNTINGONESCOUNTINGZEROES and LEADINGONESTRAILINGZEROES benchmarks, that is, for an even number \( m \), we split the \( n \) binary decision variables into \( m/2 \) blocks of size \( n/(m/2) \) and define a bi-objective \( m \)ONE\( m \)IN\( m \)AX problem on each block.

Definition 1 (\( m \)ONE\( m \)IN\( m \)AX): Let the number \( m \in \mathbb{N} \) of objectives be even and let the problem size \( n \) be a multiple of \( m/2 \). Let \( n' = 2n/m \in \mathbb{N} \). The \( m \)-objective function \( m \)ONE\( m \)IN\( m \)AX\( = (f_1, \ldots, f_m): \{0, 1\}^n \rightarrow \mathbb{R}^m \) is defined by

\[
f_i(x) = \begin{cases} \sum_{j=1}^{n'} (1 - x_{j+((i-1)n'/2)}) & \text{if } i \text{ is odd} \\ \sum_{j=1}^{n'} x_{j+((i-2)n'/2)} & \text{else} \end{cases}
\]

for all \( x = (x_1, \ldots, x_n) \in \{0, 1\}^n \).

We note that, as for the bi-objective \( m \)ONE\( m \)IN\( m \)AX benchmark, any \( x \in \{0, 1\}^n \) is a Pareto optimum of \( m \)ONE\( m \)IN\( m \)AX. The Pareto front thus is \( F^* = \{(i_1, n'-i_1, \ldots, i_m/2, n'-i_m/2) \mid i_1, \ldots, i_m/2 \in [0, n']\} \). Its size is \( M := |F^*| = (n + 1)^{m/2} = (2n/m + 1)^{m/2} \). Note that the Pareto front size is only polynomial in the problem size \( n \) when \( m \) is a constant. Naturally, the set of all Pareto optima (efficient set) is exponential in \( n \).

B. Deficiencies of the Many-Objective Crowding Distance

By definition, the crowding distance of an individual \( x \) in a set of pairwise nondominated individuals \( S \) is computed as the sum of the crowding distance contributions of \( x \) in each objective. This is convenient as it allows to compute the crowding distance by solving \( m \) sorting problems. Also, this appears
intuitively a good measure of the importance of a solution, at least when regarding bi-objective illustrations, such as [1, Fig. 1]. Unfortunately, and this is a main insight from this work, this intuition is correct only for two objectives, and for larger numbers the independent treatment of the objectives allows for very undesirable results.

Example 1: Let us give a simple example. Let \( f: [0, 1]^{400} \to [0, 200]^4 \) be the 4-objective mONEMINMAX problem with problem size \( n = 400 \) and \( m = 4 \). Let \( S \) be a set of 5 individuals such that

\[
\{ (99, 101, 0, 200), (101, 99, 0, 200), (0, 200, 99, 101), (0, 200, 101, 99), (100, 100, 100, 100) \}.
\]

Let \( x \in S \) be the individual with \( f(x) = (100, 100, 100, 100) \). One would hope that \( x \) has a large crowding distance since it is relatively isolated in many respects. For example, \( x \) is the only point in the half-plane \( \{ (a, b, c, d) \in \mathbb{R}^4 \mid a + c > 101 \} \), which contains roughly \((7/8) \cdot 0 \ldots 200)^4\). Also, with an \( L_1 \) distance of more than 200 the point \( f(x) \) is far from all other points, which in contrast have an \( L_1 \) distance of only 2 from one other point in \( f(S) \). Unfortunately, this hope does not come true. The crowding distance of \( x \) is \( 4 \cdot (2/101) = (8/101) \leq 0.08 \). In contrast, all other points in each objective have a crowding distance contribution of at least \( (99/101) \geq 0.98 \). Hence, from the crowding distance perspective, \( x \) appears as the by far least important point, where in contrast it is obvious that one could easily omit one of \( (99, 101, 0, 200) \) and \( (101, 99, 0, 200) \) from \( S \) without noticeably reducing the diversity of \( S \).

The reason for this undesired result is that the computation of the crowding distance is based on an independent consideration of the objectives. This allows that points far away from a solution \( x \) have a huge influence on its crowding distance. For example, the points \( (99, 101, 0, 200) \) and \( (101, 99, 0, 200) \) are both far from \( (100, 100, 100, 100) \) in the objective space, yet they are the reason for a small crowding distance contribution of the first and second objective to the crowding distance of \( (100, 100, 100, 100) \). Likewise, the points \( (0, 200, 99, 101) \) and \( (0, 200, 101, 99) \) are far from \( (100, 100, 100, 100) \), but cause a small crowding distance contribution of the third and fourth objective.

We note that this problem does not occur for two objectives. The reason is that there a sorting of a domination-free set with respect to one objective automatically is a sorting with respect to the other objective (in inverse order). Consequently (when assuming distinct objective values), the crowding distance of an individual \( x \) is determined only by its two unique neighbors, which are the same for both objectives. Hence, it cannot happen here that “points far away” have an influence on the crowding distance of \( x \).

We also note that the above example also holds for continuous problems. For example, these points are in the critical front (thus are mutually nondominated) for solving a continuous problem, and the point \( x \) in the less crowded regions only has a small crowding distance.

C. Exponential Lower Bound for \( m \geq 4 \)

We described above in some detail what is the true reason for the difficulties we observe in this section, and we did so because we feel that it aids the reader’s understanding and might ease finding a remedy to this problem.

Fortunately, for our mathematical proofs, we can resort to a simpler, elementary-algebra argument, which will imply that there is only a very limited number of individuals with positive crowding distance. We note that a similar argument was used in [19] in a positive manner (in the analysis of biobjective problems). There, the existence of a sufficient number of individuals with crowding distance zero was exploited to argue that these individuals will be removed first, and consequently, the (at least one) individual with positive crowding distance per Pareto front will survive the selection process.

**Lemma 1:** Let \( m \in \mathbb{N} \) with \( m \geq 4 \). Let \( S \) be a set of pairwise nondominated individuals in \( \{0, 1\}^n \). Assume that we compute the crowding distance \( c_{\text{Dis}}(S) \) with respect to the objective function \( m \text{ONEMINMAX} \) via Algorithm 1. Then at most \( 4n + 2m \) individuals in \( S \) have a positive crowding distance.

**Proof:** Consider the sorted list \( S_1, \ldots, S_{|S|} \) (step 8 in Algorithm 1) for calculating the component of the crowding distance w.r.t. a certain objective \( f_i \). Let \( V = f_i(S) \) and \( V = \{ v_1, \ldots, v_{|V|} \} \) such that \( v_1 > v_2 > \cdots > v_{|V|} \). For \( j \in [1, \ldots, |V|] \), let \( H_j = \{ s \in S \mid f_i(s) = v_j \} \). Since \( S \) is sorted according to increasing \( f_i \) value, there are \( a, b \in [1, \ldots, |S|] \) with \( a \leq b \) such that \( H_j = \{ S_{a,} \ldots, S_{b} \} \). By definition of the crowding distance, the at most two individuals \( S_{a,b} \) cause a small crowding distance contribution from the \( i \)-th objective. Since \( |V| \leq n' + 1 \) and there are \( m \) objectives, we know that there are at most \( 2m(n' + 1) = 2m(2n/m + 1) = 4n + 2m \) individuals with positive crowding distance.

We note that the result above only refers to the computation of the crowding distance. It is thus independent of other components of the NSGA-II like the mutation operator, the possible use of crossover, and the selection rules. Also, due to its simplicity, it can easily be transformed to apply to other objective functions with a limited number of values in each objective, e.g., the knapsack problem, for which several many-objective works exist (see [51] and the references therein). In its most general form, it would state that if \( f \) is an \( m \)-objective problem and \( v_i \) is the number of objective values of the \( i \)-th objective, then at most

\[
2 \sum_{i=1}^{m} v_i
\]

individuals can have a positive crowding distance. We omit the details and continue with the heart of this article, an exponential lower bound of the NSGA-II on \( m \text{ONEMINMAX} \) for \( m \geq 4 \).

**Theorem 1:** Let \( m \in \mathbb{Z}_{>4} \) and \( a > 1 \) be constants. Consider the NSGA-II with an arbitrary way to generate the offspring population and with either the original survival selection or the sequential survival selection regarded in [21] and [30]. Assume that this algorithm optimizes the \( m \text{ONEMINMAX} \)
benchmark with problem size $n$, using a population size of at most $N \leq aM$, where $M = (2n/m + 1)^{m/2}$ is the size of the Pareto front of $m$OneMax. Then with probability at least $1 - T \exp(-\Omega(M))$, for the first $T$ iterations at least a constant fraction of Pareto front is not covered by $P_t$. In particular, the time to compute the full Pareto front is exponential in $n^{m/2}$ both in expectation and with high probability.

Proof: Since we aim at an asymptotic results, we can assume that $n$ is at least $10m$ and large enough to ensure that $a \leq M/8$. Let $R_t$ be the combined parent and offspring population in iteration $t$. Let $M'$ denote the size of the part of the Pareto that is front covered by $R_t$, that is, $M' = |f(R_t)|$ when viewing $f(R_t)$ as set, not as multiset. Let $\gamma$ be a constant in $[4/5, 1)$. If $M' < \gamma M$, then with probability one, the next generation will cover at most $M' < \gamma M$ Pareto front points as the survival selection cannot increase the coverage of the Pareto front.

Now consider the case $M' \geq \gamma M$. We note that there are at most $|R_t|/(4\gamma^2)$ Pareto front points with more than $4\gamma^2$ corresponding individuals in $R_t$. However, there would be more than $|R_t|/(4\gamma^2)(4\gamma^2) = |R_t|$ individuals in $R_t$. Let $U$ denote the set of Pareto front points that have at most $4\gamma^2$ corresponding individuals in $R_t$, and let $U' = \{u \in U \mid \forall x \in R_t : f(x) = u \implies \text{cDis}(x) = 0\}$. Then, using $R_t = 2N \leq 2aM$, we have

$$|U| \geq M' - \frac{|R_t|}{4\gamma^2} \geq \gamma M - \frac{2aN}{4\gamma^2} = \frac{\gamma}{2} M.$$  

Let $A$ denote the number of individuals in $R_t$ with positive crowding distance. From Lemma 1 we know that $A \leq 4n + 2m$. Thus, we have

$$|U'| \geq |U| - A \geq \frac{\gamma}{2} M - (4n + 2m) \geq \frac{\gamma}{4} M$$

where the last inequality uses $4n + 2m \leq (\gamma/4)(2n/m + 1)^{m/2}$ for $n$ sufficiently large. Besides, our assumptions, most notably $n \geq 10m$, imply that $N \geq A$. More detailedly, we compute

$$N = \frac{1}{2}|R_t| \geq \frac{1}{2}M' \geq \frac{1}{2}M = \frac{\gamma}{2} \left(\frac{2n}{m} + 1\right)^{m/2} \geq \frac{3}{2} \left(1 + \frac{2n}{m}(\frac{\gamma}{4} - 1)\right) \left(\frac{2n}{m} + 1\right) \geq \frac{1}{2} \left(1 + \frac{2n}{m}\right)^{m/2} \geq 4n + 2m \geq A$$

where the second line uses $\gamma \geq 4/5$ and $(1 + x)^y \geq 1 + xy$ for any $x \geq -1$ and $y \geq 1$, the third line uses $m \geq 4$, and the last line uses $n \geq 10m$. Thus, the number of the individuals with zero crowding distance $2N - A$ is greater than or equal to $N$, the number of individuals to be removed in the survival selection. Since all individuals in $R_t$ are Pareto optimal and thus $F_t = F_1$, the original survival selection will remove $N$ individuals out of the $2N - A$ individuals with zero crowding distance uniformly at random. For the sequential survival selection, it is not difficult to see that removing an individual with zero crowding distance does not change the crowding distances of all other individuals. Consequently, the selection process is identical to the original survival selection.

We thus analyze the effect of randomly removing $N$ individuals with zero crowding distance. To ease the argument, we instead consider the process where we $N$ times independently and uniformly at random pick an individual from the $2N - A$ individuals with replacement and then remove these individuals. Obviously, the number of uncovered values in $U'$ after this process, denoted by $Y$, is stochastically dominated by that number, denoted by $X$, after the original process. For a certain value in $U'$ with $b$ corresponding individuals in $R_t$, we know that its probability to be uncovered is at least

$$\begin{align*}
\left(\frac{N}{b}\right) b! \left(\frac{1}{2N-A} \right)^b \left(1 - \frac{b}{2N-A}\right)^{N-b} & = \left(\frac{N}{b}\right) b! \left(\frac{1}{2N-A} \right)^b \left(1 - \frac{b}{2N-A}\right)^{(2N-A) -1} \left(\frac{2N-A}{2N-A}\right)^{N-b} \\
\geq \left(\frac{N}{b}\right) b! \left(\frac{1}{2N-A} \right)^b \exp \left(-\frac{(N-b)b}{2N-A} - b\right) & \geq \left(\frac{N}{b}\right) b! \left(\frac{1}{2N-A} \right)^b \frac{1}{e^b} \frac{1}{e^b} \frac{1}{(2N-A)^b} \\
\geq \left(\frac{N}{b}\right) b! \left(\frac{1}{2N-A} \right)^b \frac{1}{e^b} \frac{1}{e^b} \frac{1}{(2N-A)^b} & \geq \frac{N}{b} b! \left(\frac{1}{2N-A} \right)^b \frac{1}{e^b} \frac{1}{e^b} \frac{1}{(2N-A)^b} := p
\end{align*}$$

where the second and the last inequality use $2N - A \geq N$, and the last inequality also uses $N - 4\gamma^2 + 1 > 0$ since $N \geq M/2 \geq \gamma M/2 \geq 4\gamma^2$ from $a \leq M/8$. Obviously, $p = \Theta(1)$.

We have just seen that $E[Y] \geq p|U'| \geq (1/4)p\gamma M$. We note that $Y$ is functionally dependent on the independent $N$ picks, and that each pick will influence $Y$ by at most $1$. Hence, we apply the method of bounded differences [52] and obtain that

$$\Pr[Y \leq \frac{1}{8} p\gamma M] \leq \Pr\left[Y \leq \frac{1}{2} E[Y]\right] \leq \exp(-\Omega(E[Y])) = \exp(-\Omega(M)).$$

Therefore, with probability at least $1 - \exp(-\Omega(M))$, at least $(1/8)p\gamma M$ objective values of the Pareto front are not covered after the survival selection.

Theorem 1 shows that for any constant $m \geq 4$, that is, for the polynomial Pareto front size $M$, this exponential lower bound already exists. We note that unlike for Lemma 1, the proof of the result above does not immediately extend to the $m$COCZ and $m$LOTZ benchmarks. The reason is that for these benchmarks, not all individuals are Pareto optimal. Consequently, the nondominated sorting of $R_t$ may contain several fronts, and thus the lower-priority fronts could prevent Pareto optimal individuals from entering into the selection competition. While the proof does not immediately extend, we do believe that the intrinsic problems of the crowding distance for more than two objectives persist on these benchmarks and forbid an effective optimization. Our informal arguments are twofold. On the one hand, the problem that there is only a small number of individuals with positive crowding distance applies to all fronts of the nondominated sorting. So very likely, there is a front in which the survival decisions are taken mostly at random, and this should lead to problems, possibly even to the population not approaching the Pareto front at all. On the other hand, if the population comes close to covering the Pareto front, then the combined parent and offspring population is likely
to have more than $N$ Pareto optimal individuals. In this case, the same difficulties as exploited in the proof above arise. As said, we speculate that the NSGA-II has difficulties to optimize $m$COZ and $m$LOTZ, but we leave the formal proof of such a statement as future work.

D. Also the Combined Parent and Offspring Population Does Not Cover the Pareto Front

Our main result in the previous section was that the population $P_t$ of the NSGA-II for an at least exponential time does not cover the full Pareto front. One could be optimistic that this is better for the combined parent and offspring population $R_t$, which, with its size only by a factor of two larger, then would be an interesting output of the algorithm. We now show that this is not true and that also $R_t$ misses a constant fraction of the Pareto front for a long time.

Since this result naturally depends on how precisely the offspring are generated, we regard in the following one particular setting, namely, that each parent, creates exactly one offspring via one-bit mutation. We are very optimistic that all other mutation-based variants of the NSGA-II regarded in previous runtime analyses works would admit the same result. For reasons of brevity, we do not go into details. We are also optimistic that the following result would remain true for crossover-based variants of the NSGA-II [20], but with the little understanding we currently have on how crossover works in the NSGA-II, this is clearly only a speculation. The main argument of our rigorous proof below, and the reason for our optimism that similar results hold for other variants of the NSGA-II, is that a fair proportion of the Pareto front points not covered by $P_t$ only have a constant probability of being generated in $Q_t$. Hence in expectation, $Q_t$ will miss a constant fraction of these, and with the large amount of independent randomness, this expectation can be turned into a statement true with probability $1 - \exp(-M)$. This argument is similar to the proof of Lemma 10 of the extended version of [19] on the arxiv preprint server, however, some adaptations were necessary to cope with the larger population size, the larger number of objectives, and the different structure of the Pareto front. Note that the lower bound in Theorem 1 stems from the survival selection regardless of the generation of the offspring, and the following theorem indicates the pessimistic aspect even before the survival selection.

Theorem 2: Let $m \in \mathbb{Z}_{\geq 0}$ and $a > 1$ be constants. Consider using the NSGA-II with population size $N \leq aM = a(2n/m + 1)^{\frac{3}{2}}$, fair selection (every parent creates one offspring), and one-bit mutation to optimize $m$ONEINMAX with problem size $n$. Let $\varepsilon \in (0, 1)$ be a sufficiently small constant. Assume that in some iteration $t$ the population $P_t$ covers at most a $(1 - \varepsilon)$ fraction of the Pareto front. Then with probability at least $1 - \exp(-2M)$ the combined parent and offspring population $R_t$ will cover at most a fraction of $1 - (1 - \varepsilon)\exp(-[20am(5m - \varepsilon)]/[\varepsilon(5m^2 - 4(5m - \varepsilon))])$ of the Pareto front.

To ease the presentation, we use the following natural definition of a neighbor.

Definition 2 (Neighbors): Let $F^* \subseteq [0..n]^m$ be the Pareto front for $m$ONEINMAX. For any two Pareto front points $v = (v_1, n' - v_1, \ldots, v_m/2, n' - v_m/2)$ and $v' = (v'_1, n' - v'_1, \ldots, v'_m/2, n' - v'_m/2)$ in $F^*$, we call $v$ and $v'$ neighbors, denoted by $v \sim v'$, if there exists an $i \in [1..m/2]$ such that $|v_i - v'_i| = 1$ and $v_j = v'_j$ for all $j \neq i$.

We now prove Theorem 2.

Proof of Theorem 2: We first show that there are at most $(\varepsilon/5)M$ Pareto front points whose neighbors in total have at least $[5am/\varepsilon]$ corresponding individuals in $P_t$. Formally, for any Pareto front point $v \in F^*$, let $P_v = \{x \in P_t \mid f(x) \sim v\}$ be the individuals whose function values are neighbors of $v$. Let $\Delta = [5am/\varepsilon] - 1$ and let $F' = \{v \in F \mid |P_v| \geq \Delta + 1\}$. Then we show $|F'| \leq amM/(\Delta + 1)$ (and thus $|F'| \leq \frac{5}{2}M$ by the definition of $\Delta$). If this was not true, then, since each individual is in $P_v$ for at most $m$ different points $v$, we have $m|P_i| \geq \sum_{v \in F'}|P_v|$ and thus $|P_i| \geq \frac{1}{m}(\Delta + 1)|F'| > \frac{1}{m}(\Delta + 1)\frac{am}{\Delta + 1}M = aM$ which contradicts $N \leq aM$.

Let $U$ be the set of uncovered Pareto fronts for $P_t$. We assumed $|U| \geq \varepsilon M$. Let $U_1$ be the subset of $U$ defined by $U_1 = \{v = (v_1, n' - v_1, \ldots, v_m/2, n' - v_m/2) \in F \setminus F': |v_i| \leq \left\lfloor \frac{2}{5m^\varepsilon}(n' + 1) \right\rfloor, n' - \left\lfloor \frac{2}{5m^\varepsilon}(n' + 1) \right\rfloor \leq i \leq n \}$. Then

$|U_1| \geq |U| - 2\frac{2}{5m^\varepsilon}(n' + 1)\frac{M}{m\varepsilon + 1} \geq \frac{2}{5m^\varepsilon}M \geq \frac{2}{5}\varepsilon M$.

In the following, we will show that at most a constant fraction (less than one) of $U_1$ can be generated in the offspring population (and thus in $R_t$) both with high probability and in expectation.

For any $v = (v_1, n' - v_1, \ldots, v_m/2, n' - v_m/2) \in U_1$, the probability that a parent from $P_v$ generates $v$ is at most

$$\max_{1 \leq i \leq m/2} \left\{ \frac{v_i + 1}{n} - \frac{n'}{n} \right\} \leq \frac{n'}{n} - \left\lfloor \frac{2}{5m^\varepsilon}(n' + 1) \right\rfloor + 1 \leq \frac{n'}{n} - \left( \frac{2}{5m^\varepsilon}n' - 1 \right) + 1$$

$$\leq \left(1 - \frac{2\varepsilon}{5m^\varepsilon}\right) \frac{n'}{n} + \frac{2}{m} = \left(1 - \frac{2\varepsilon}{5m^\varepsilon}\right) \frac{2}{m} + \frac{2}{n}$$

$$\leq \left(1 - \frac{4(5m - \varepsilon)}{5m^2}\right) \frac{2}{5m} \frac{4(5m - \varepsilon)}{5m^2}$$

where the last inequality uses $n \geq m$. From the definition of $U_1$, we know that there are at most $\Delta$ individuals in $P_t$ that can generate $v$. Thus, the probability that $v$ is not generated from $P_t$, that is, remains uncovered in $R_t$, is at least

$$\left(1 - \frac{4(5m - \varepsilon)}{5m^2}\right)^{\Delta} \geq \left(1 - \frac{4(5m - \varepsilon)}{5m^2}\right)^{\frac{5am}{\varepsilon}}$$

$$= \left(1 - \frac{4(5m - \varepsilon)}{5m^2}\right)^{\frac{5am}{\varepsilon}} \left(1 - \frac{4(5m - \varepsilon)}{5m^2}\right)^{\frac{4(5m - \varepsilon)}{5m^2 - 4(5m - \varepsilon)}}$$

$$\geq \exp\left(-\frac{20am(5m - \varepsilon)}{\varepsilon(5m^2 - 4(5m - \varepsilon))}\right) = p$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where the first inequality uses $\Delta \leq 5am/\varepsilon$. We note that $p = \Theta(1)$ for $m = \Theta(1)$.

Let $Y$ denote the number of uncovered points for $R_t$. Then $E[Y] \geq p|U_1| \geq (2/5)\varepsilon pM$. Note that $Y$ is determined by $N$ independent random choices in the mutation for each parent, and each choice of the mutation can reduce $Y$ by at most $1$ and will not increase $Y$. Hence, from the bounded difference method [53, Th. 1.10.27], we have $\Pr \{ Y \leq (1/5)\varepsilon pM \} \leq \exp(-\Omega(M))$. 

E. Synthetic MOEAs

For reasons of completeness, we now briefly analyze the performance of the SEMO and GSEMO on the $m$ONEMINMAX benchmark. Not surprisingly, the fact that they keep a solution for every nondominated objective value in the population avoids the problems seen for the NSGA-II above. Since there is little evidence that these two algorithms perform well in practice, this observation is more of an academic interest, though.

Our analysis follows roughly the lines of the proof of the result that the SEMO finds the Pareto front of COUNTINGONESCOUNTINGZERORES in time $O(n^{m+1})$ in [44]. A slightly tighter bound of $O(n^m)$ and, for $m = 4$, of $O(n^2 \log n)$, was shown via a very general (and thus slightly technical) method in [45]. We are optimistic that this method can be applied to our problem as well and then gives comparable bounds, but since 1) we only aim at showing that the SEMO and GSEMO do not have the difficulties proven for the NSGA-II in our main result and 2) it is not at all clear how tight any of these bounds is (recall that the Pareto front has size $(2n/m)^{m/2}$ only), we content ourselves with a simple proof of the weaker result.

For any point $u \in F^*$ with some neighbor covered by the population $P$, the probability to generate such point in one generation is at least $1/(n|P|)$ for the SEMO and at least $(1/P)^{1/(1 - |1/n|)^{m-1}}(1/n) \geq 1/(1/\varepsilon n|P|)$ for the GSEMO. Since all covered Pareto front points will be covered, we know that the expected number of iterations to cover $u$ is $O(n|P|)$. Since there are $M$ Pareto front points, we know that $|P| \leq M$ during the process. With $M$ Pareto front points to cover, the expected number of iterations to cover the whole Pareto front is at most $O(nm^2) = O(n(2n/m + 1)^m)$. For a clear comparison, we put this simple result into the following theorem. We do not believe this result to be tight, but it suffices to show that the SEMO/GSEMO does not suffer critically from the larger number of objectives, different from the NSGA-II.

Theorem 3: Consider using the SEMO/GSEMO to optimize the $m$ONEMINMAX benchmark with problem size $n$. Then the Pareto front is covered in an expected number of $O(n(2n/m + 1)^m)$ function evaluations.

V. INEFFECTIVENESS FOR THREE OBJECTIVES

Since the classic way of obtaining many-objective versions of bi-objective benchmarks only gives benchmarks for even numbers of objective [44], in Section IV we could only prove that the NSGA-II is ineffective from four objectives on. Since small numbers of objectives are common, it is an interesting question at what point precisely the ineffectiveness starts. For this reason, we now define a three-objective version of $m$ONEMINMAX and prove that the NSGA-II cannot optimize it efficiently as well. Here, the lower degree of symmetry of the $3$-ONEMINMAX problem will need some adjustment compared to the previous section, but the broad line of argument will be the same.

A. 3-ONEMINMAX

We define a three-objective $ONEMINMAX$ benchmark as follows.

Definition 3: Let $n \in \mathbb{N}$ be even. The three-objective function $3$-ONEMINMAX $= (f_1, f_2, f_3) : \{0, 1\}^n \to \mathbb{R}^3$ is defined by

$$f_1(x) = \sum_{i=1}^{n} (1 - x_i) \quad f_2(x) = \sum_{i=1}^{n/2} x_i \quad f_3(x) = \sum_{i=n/2+1}^{n} x_i$$

for all $x = (x_1, \ldots, x_n) \in \{0, 1\}^n$.

We immediately see that the first objective counts the number of zeros in the whole bit string, whereas the second and third objectives count the numbers of ones in the first and second half of it. As for the $m$ONEMINMAX function defined for even $m$ in Section IV, any $x \in \{0, 1\}^n$ is a Pareto optimum of $3$-ONEMINMAX. The Pareto front thus is $F^* = \{ (n - v_2 - v_3) | v_2, v_3 \in \{0..n\}' \}$, where we write $n' := n/2$, and it has the size $M := |F^*| = (n/2)^2$. We note that it would be easy to extend this definition to any odd number $m$ of objectives, but we do not see a huge interest in this.

B. Exponential Lower Bound for 3-ONEMINMAX

We now follow similar arguments in Section IV-B. In the discussion after Lemma 1, a general bound of $2 \sum_{i=1}^{n} v_i$ (where $v_i$ is the number of objective values of the $i$th objective) holds for the number of individuals with positive crowding distance. This value is $2(n + n/2 + n/2 + 1) = 4n + 6$ for the 3-ONEMINMAX benchmark.

Lemma 2: Consider a set of pairwise nondominated individuals $S$ for the NSGA-II solving 3-ONEMINMAX with problem size $n$. Then the crowding distance $\text{cDis}(S)$ computed via Algorithm 1 will have at most $4n + 6$ positive values.

Now we establish a result analogous to Theorem 1. Let $A \leq 4n + 6$. For a constant $\gamma \in [4/5, 1)$, let $N \geq (\gamma/2)(n/2 + 1)^2$. Since

$$\frac{\gamma}{2}(n/2 + 1)^2 \geq \frac{2}{5}(n/2 + 1)^2 \geq \frac{1}{10}n^2 + \frac{2}{5}n + \frac{2}{5} \geq 4n + 16 + \frac{2}{5} \geq 4n + 6$$

for $n \geq 40$, we have $N \geq A$. Using this in the place of (2) in the proof of Theorem 1, and replacing Lemma 1 by Lemma 2 there as well, we obtain a proof of the following theorem.
Theorem 4: Let $a > 1$ be a constant. Consider the NSGA-II with an arbitrary way to generate the offspring population and with either the original survival selection or the sequential survival selection regarded in [21] and [30]. Assume that this algorithm optimizes the 3-ONE_MIN_MAX benchmark with problem size $n$, using a population size of at most $N \leq aM$, where $M = (n/2 + 1)^2$ is the size of the Pareto front of 3-ONE_MIN_MAX. Then with at least probability $1 - T \exp(-\Omega(M))$, for the first $T$ iterations at least a constant fraction of Pareto front is not covered by $P_t$. In particular, the time to compute the full Pareto front is exponential in $n^2$ both in expectation and with high probability.

C. Also the Combined Parent and Offspring Population Does Not Cover the Pareto Front

In a similar manner as in Theorem 2, we observe also for 3-ONE_MIN_MAX that also the combined parent and offspring population does not cover the full Pareto front. While similar, Theorem 5 cannot simply be obtained from changing $m$ to $3$ in Theorem 5. Some adjustments are needed due to the different structure of the Pareto front.

Theorem 5: Let $a \geq 1$ be any constant. Consider using the NSGA-II with population size $N \leq aM$, fair selection, and one-bit mutation to optimize 3-ONE_MIN_MAX with problem size $n$. Let $\epsilon \in (0,1)$ be a sufficiently small constant. Assume that in some iteration $t$ the population $P_t$ covers at most a $(1 - \epsilon)$ fraction of the Pareto front. Then with probability $1 - \exp(-\Omega(M))$, the combined parent and offspring population $R_t$ covers at most a fraction of $1 - (3/5)\epsilon \exp(-((20a)/(20 - \epsilon)))$ of the Pareto front.

Proof: For any Pareto front point $v \in F^*$, let $P_v = \{x \in P_t | f(x) \sim v\}$ be the individuals whose function values are neighbors of $v$. Let $\Delta = [(20a)/\epsilon] - 1$ and let $F = \{v \in F | |P_v| \geq \Delta + 1\}$. Then we show $|F^*| \leq [(4a)/(\Delta + 1)]M$ (and thus $|F^*| \leq (4\epsilon/5)M$ by the definition of $\Delta$). If this was not true, then, since each individual is in $P_t$, for at most 4 different points $v$, we have $|P_v| \geq \sum_{v \in F^*} |P_v|$ and thus

$$|P_v| \geq \frac{1}{4}((\Delta + 1)|F^*|) \geq \frac{1}{4}(\Delta + 1)\frac{4a}{\Delta + 1}M = aM$$

which contradicts $N \leq aM$.

Let $U$ be the set of uncovered Pareto fronts for $P_t$. We assumed $|U| \geq \epsilon M$. Let $U_1$ be defined by

$$U_1 = \{(n - v_1 - v_2, v_1, v_2) \in U \setminus F^* | v_1, v_2 \in \left[\left[\frac{\epsilon}{10}(n' + 1)\right], \left[\frac{\epsilon}{10}(n' + 1)\right]\right]\}.$$ 

Then

$$|U_1| \geq |U| - 2\frac{\epsilon}{10}(n' + 1)\frac{M}{(\sigma + 1)} - \frac{1}{4}\epsilon M \geq \frac{3}{2}\epsilon M.$$ 

For any $v = (n - v_1 - v_2, v_1, v_2) \in U_1$, the probability that a parent from $P_v$ generates $v$ is at most

$$\max\left\{\left\lfloor \frac{v_1 + 1}{n'}\right\rfloor, \left\lfloor \frac{v - v_1 + 1}{n'}\right\rfloor\right\} \leq \frac{n'}{n} - \left\lfloor \frac{\epsilon}{10}(n' + 1)\right\rfloor + 1 \leq \frac{n'}{n} - \left\lfloor \frac{\epsilon}{10}(n' + 1)\right\rfloor + 1 \leq \frac{n'}{n} - \left\lfloor \frac{\epsilon}{10}(n' + 1)\right\rfloor + 1 \leq \frac{1}{2}\left(1 - \frac{\epsilon}{10}\right) + \frac{2}{n} \leq 1 - \frac{\epsilon}{20}$$

where the last inequality uses $n \geq 4$. From the definition of $U_1$, we know that there are at most $\Delta$ individuals in $P_t$ that can generate $v$. Thus, the probability that $v$ is not generated from $P_t$, that is, remains uncovered in $R_t$, is at least

$$(1 - \frac{\epsilon}{20})^\Delta \geq (1 - \frac{\epsilon}{20})^{20a/\epsilon} = (1 - \frac{\epsilon}{20})^{\frac{20}{\epsilon}} \frac{20}{20 - \epsilon} \geq \exp\left(-\frac{20a}{20 - \epsilon}\right) := p.$$ 

Let $Y$ denote the number of uncovered points for $R_t$. Then $E[Y] \geq p|U_1| \geq (2/5)\epsilon pM$. Note that $Y$ is determined by $N$ independent random choices in the mutation for each parent, and each choice of the mutation can reduce $Y$ by at most 1 and will not increase $Y$. Hence, from the bounded difference method [53, Th. 1.10.27], we have $Pr[Y \leq (1/5)pM] \leq \exp(-\Omega(M))$.

D. Synthetic MOEAs

With the same arguments as for Theorem 3, we easily obtain the following result that the SEMO/GSEMO easily optimizes also 3-ONE_MIN_MAX.

Theorem 6: Consider using the SEMO/GSEMO to optimize the 3-ONE_MIN_MAX benchmark with problem size $n$. Then the Pareto front is covered in an expected number of $O(n^3)$ function evaluations.

VI. EXPERIMENTS

In this section, we present a few experimental results illustrating and complementing our theoretical results.

We concentrate on the 4-ONE_MIN_MAX problem since it is more symmetric than the 3-ONE_MIN_MAX problem, but still allows for good visualizations. We note that the two objectives counting zeroes are fully determined by the two corresponding objectives that count ones. Consequently, displaying only these two objectives allows us to present the full information in a 2-D graph. We use the problem size $n = 40$, which gives a Pareto front of reasonable size $M = 441$, but is small enough to keep the plots readable.

Since we understand much better the NSGA-II without crossover, we use the NSGA-II with mutation as the only variation operator. We use fair selection for reproduction (each parent creates one offspring), standard bitwise mutation with mutation rate $(1/n)$ as mutation operator, and we select the next parent population as in the original NSGA-II paper [1]. Since any solution of the mONE_MIN_MAX problem is Pareto optimal, all solutions of the combined parent and offspring population lie on the first front and thus the selection is solely based on the crowding distance, breaking ties randomly. We note that the original definition of the NSGA-II also used the crowding distance together with tournament selection for the selection for reproduction. Since our theoretical results showed that almost all individuals have the same crowding distance (of zero), we did not see much advantage for this approach here and preferred fair selection as a mean to reduce the variance. Theorem 1 showed difficulties of the NSGA-II for any population size that is a constant factor larger than the Pareto front size $M$, so we used the population sizes $4M, 16M, 64M$, 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
and 256M. Noting that the largest of these population sizes, 256M = 112,896 is quite large and we conduct runs with 1000 iterations, we only conducted ten independent repetitions of all NSGA-II runs. Since the variances observed were always relatively small, this appeared sufficient.

For a first impression of the optimization behavior of the NSGA-II, we display in Fig. 1 the coverage of the Pareto front in the first 1000 iterations of exemplary runs. As can be seen in all four plots, the coverage of the Pareto front after some initial gains quickly reaches a stagnation point and then oscillates around this point. This fits well to our theoretical results.

Fig. 1. Size of the part of the Pareto front covered by the parent population $P_t$ and the combined parent and offspring population $R_t$ in the first 1000 iterations of exemplary runs of the NSGA-II on the 4-ONE MINMAX problem with problem size $n = 40$.

Fig. 2. Size of the Pareto front covered by $P_t$ and $R_t$ (mean value ± standard deviation) in the 1000th iteration for the NSGA-II (fair selection, standard bitwise mutation) with population size $N = 4M, 16M, 64M, 256M$ on mONE MINMAX with problem size $n = 40$ and four objectives, where $M = 441$ is the Pareto front size (ten independent runs).
see that the coverage of the Pareto front slightly improves with growing population size. We determined the average coverage after 1000 iterations in ten runs (Fig. 2). The slow increase of the coverage shows clearly that increasing the population size gives only very small improvements. From the roughly linear increase (w.r.t. to the log-scale population size) one could speculate that a population size exponential in the Pareto front size is necessary to cover the whole Pareto front in 1000 iterations.

In contrast, as predicted by Theorem 3, the GSEMO algorithm computes the Pareto front very efficiently. In 30 independent runs, the GSEMO found the full Pareto front on average in $8.84 \times 10^4$ iterations (standard deviation $2.97 \times 10^3$). We recall that the GSEMO evaluates one individual per iteration. Hence, these $8.84 \times 10^4$ fitness evaluations are significantly less than, e.g., the 1000 · $M = 1.764 \times 10^6$ fitness evaluations leading to no coverage of the Pareto front in the first chart of Fig. 1.

To gain some insight into what happens in a single iteration, in Fig. 3 we display (in the objective space, only regarding the particular objective function) the combined parent and offspring population after 1000 iterations, the individuals with positive crowding distance, and the selected next parent population. The sizes of these populations naturally fit to the previous experiments, but what is new is the shape of the populations, which covers a relatively compact region around the center of the Pareto front. This is an insight that was not detected in our mathematical analysis. We believe that this fact strongly relies on the particular objective function, in contrast to the key argument in the proof of Theorem 1, the negative effect of the random selection of the individuals with crowding distance zero, which depended very little on the particular objective function. For these reasons, we did not try to capture this observation via mathematical means. We suspect that with some effort this would be possible, and it might be a way to prove that even much larger population sizes are not sufficient to find the Pareto front of the mONEMINMAX problem. We also see that the individuals with positive crowding distance appear randomly distributed, which fits again to our mathematical intuition.

To check that a similar behavior is observed also by the 3-ONEMINMAX benchmark, we conducted one small experiment on problem size $n = 40$, with a population size of $N = 16M$, over ten independent runs. It showed that the size of the Pareto front covered by the combined parent and offspring population $R$ in the 1000th iteration has a mean value of 291 (standard deviation of 7.36). This is again significantly smaller than the Pareto front size of (again) 441, and indicates that also in experiments there is no huge difference between the cases $m = 3$ and $m = 4$.

Overall, these experimental results fit well to our mathematical results and exhibit a few interesting details which our proof could not detect.

### VII. Conclusion

In this work, we conducted the first mathematical runtime analysis of the NSGA-II on many-objective problems. It confirms in a very strong manner the empirical knowledge that the NSGA-II is less effective for larger numbers of objectives: already for the very simple mONEMINMAX benchmark, for which any solution is Pareto optimal, the population of the NSGA-II will miss a constant fraction of the Pareto front for at least exponential time. In this result, the population size can be any constant multiple of the Pareto front size.

Besides this quantification of the ineffectiveness, our work also exhibits the reason for these difficulties: since the crowding distance is computed by regarding the different objectives independently, solutions that are far away may still cause a solution to have a very low-crowding distance, disfavoring it in the selection stage. This effect can be so strong that almost all solutions have a crowding distance of zero. In this case, essentially a random set of $N$ individuals will be removed in the selection stage, which makes it very probable that desired solutions are lost.

The obvious question arising from this work is whether there is a definition for the crowding distance which (ideally provably) avoids this短coming, but which can still be computed efficiently. A second interesting task for future research would be to conduct similar mathematical runtime analyses for other common MOEAs. This work has shown that the NSGA-II suffers drastically from three objectives on, but the synthetic GSEMO algorithm does not (ignoring a polynomial increase of the runtime, which cannot be avoided simply because also the Pareto front increases in size). This raises the question to what extent other MOEAs provably suffer from larger numbers of objectives. Empirical work like [3] has shown that larger numbers of objectives are a challenge for many MOEAs, but that they suffer from these in a different manner. Hence, a mathematical analysis and asymptotic quantification of these observations would be very interesting.

We also note that this article does not discuss the convergence to the Pareto front point as each solution is Pareto optimal for mONEMINMAX. Some work like [54] pointed out the divergent behavior of the NSGA-II and other MOEAs. How heavily these behaviors influence the convergence ability of the NSGA-II is still unknown. It would be an interesting future work.

### References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.

[2] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms: A survey of the state of the art,” Swarm Evol. Comput., vol. 1, pp. 32–49, Mar. 2011.

[3] V. Khare, X. Yao, and K. Deb, “Performance scaling of multi-objective evolutionary algorithms,” in Proc. Int. Conf. Evol. Multi-Criterion Optim., 2003, pp. 376–390.

[4] R. C. Purshouse and P. J. Fleming, “On the evolutionary optimization of many conflicting objectives,” IEEE Trans. Evol. Comput., vol. 11, no. 6, pp. 770–784, Dec. 2007.

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization: A short review,” in Proc. IEEE Congr. Evol. Comput., 2008, pp. 2419–2426.

[6] B. Li, J. Li, K. Tang, and X. Yao, “Many-objective evolutionary algorithms: A survey,” ACM Comput. Surveys, vol. 48, no. 1, pp. 1–35, 2015.

[7] S. Droste, T. Jansen, and I. Wegener, “On the analysis of the (1+1) evolutionary algorithm,” Theor. Comput. Sci., vol. 276, pp. 1–2, pp. 51–81, 2002.
D. Corus, P. S. Oliveto, and D. Yazdani, “Fast immune system-inspired algorithms,” in Proc. Int. Joint Conf. Artif. Intell., 2022, pp. 1227–1233.

D. S. Oliveto, P. Husbands, and S. W. Whiteson, “Fast evolutionary algorithms for the multimodal problem,” IEEE Trans. Evol. Comput., vol. 25, no. 5, pp. 956–970, Oct. 2021.

D. Ang, A. V. Eremeev, and P. K. Lehre, “Escaping local optima with non-elitist evolutionary algorithms,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 1275–1283.

F. Neumann and C. Witt, “Runtime analysis of single- and multi-objective evolutionary algorithms for chance constrained optimization problems with normally distributed random variables,” in Proc. Int. Joint Conf. Artif. Intell., 2022, pp. 4800–4806.

B. Dörer, D. Johannsen, and C. Winzen, “Multiplicative drift analysis,” Algorithmica, vol. 64, pp. 673–697, Dec. 2012.

D. Sudholt, “A new method for lower bounds on the running time of evolutionary algorithms,” IEEE Trans. Evol. Comput., vol. 17, no. 3, pp. 418–435, Jun. 2013.

D. C. Dang, A. Opris, B. Salehi, and D. Sudholt, “A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation,” in Proc. Conf. Artif. Intell., 2023, pp. 12390–12398.

B. Doerr and Z. Qu, “Runtime analysis for the NSGA-II: Provable speed-ups from crossover,” in Proc. Conf. Artif. Intell., 2023, pp. 12399–12407.

C. Igel, N. Hansen, and S. Roth, “Covariance matrix adaptation for multi-objective optimisation,” Evol. Comput., vol. 15, no. 1, pp. 1–28, 2007.

Y. Wang, M. Emmerich, A. Deutz, and T. Bücker, “Diversity-indicator based multi-objective evolutionary algorithm: DI-MOEA,” in Proc. Int. Conf. Evol. Multi-Criterion Optim., 2019, pp. 346–358.

E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 257–271, Nov. 1999.

K. Deb, M. Mohan, and S. Mishra, “A fast multi-objective evolutionary algorithm for identifying well-spread Pareto-optimal solutions,” Kanpur Genet. Algorithms Lab., Kanpur, India, KanGAL Rep. 2003002, 2003.

V. L. Vachhani, V. K. Dabhi, and H. B. Prajapat, “Improving NSGA-II for solving multi objective function optimization problems,” in Proc. Int. Conf. Comput. Commun. Tech., 2016, pp. 1–6.

S. Wiether and B. Doerr, “A mathematical runtime analysis of the non-dominated sorting genetic algorithm III (NSGA-III),” in Proc. Int. Joint Conf. Artif. Intell., 2023, pp. 5657–5663.

S. Kukkonen and K. Deb, “Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems,” in Proc. Conf. Evol. Comput. 2006, pp. 1179–1186.

F. Fortin and M. Parizeau, “Revisiting the NSGA-II crowding-distance computation,” in Proc. Genet. Evol. Comput. Conf., 2013, pp. 623–630.

M. Laumanns, L. Thiele, E. Zitzler, E. Weizl, and K. Deb, “Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem,” in Parallel Problem Solving From Nature, Heidelberg, Germany: Springer, 2002, pp. 44–53.

O. Giel, “Expected runtimes of a simple multi-objective evolutionary algorithm,” in Proc. Congr. Evol. Comput., 2003, pp. 1918–1925.

D. Thierens, “Convergence time analysis for the multi-objective counting ones problem,” in Proc. Evol. Multi-Criterion Optim., 2003, pp. 355–364.

D. Brockhoff, T. Friedrich, and F. Neumann, “Analyzing hypervolume indicator based algorithms,” in Parallel Problem Solving From Nature, Heidelberg, Germany: Springer, 2008, pp. 651–660.

Y.-L. Li, Y.-R. Zhou, Z.-H. Zhan, and J. Zhang, “A primary theoretical study on decomposition-based multiobjective evolutionary algorithms,” IEEE Trans. Evol. Comput., vol. 20, no. 4, pp. 563–576, Aug. 2016.

W. Zheng and B. Doerr, “Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II),” Artif. Intell., vol. 325, Dec. 2023, Art. no. 104016. [Online]. Available: https://doi.org/10.1016/j.artint.2023.104016

B. Doerr and Z. Qu, “A first runtime analysis of the NSGA-II on a multimodal problem,” IEEE Trans. Evol. Comput., vol. 27, no. 5, pp. 1288–1297, Oct. 2023. [Online]. Available: https://doi.org/10.1109/TEVC.2023.3250552

W. Zheng and B. Doerr, “Theoretical analyses of multiobjective evolutionary algorithms on multimodal objectives,” in Evolutionary Computation, Cambridge, MA, USA: MIT Press, 2023. [Online]. Available: https://doi.org/10.1162/evco_a_00328

B. Doerr and Z. Qu, “From understanding the population dynamics of the NSGA-II to the first proven lower bounds,” in Proc. Conf. Evol. Comput., 2023, pp. 12408–12416.

C.-C. Dang, A. Opiris, B. Salehi, and D. Sudholt, “A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation,” in Proc. Conf. Artif. Intell., 2023, pp. 12390–12398.

B. Doerr and Z. Qu, “Runtime analysis for the NSGA-II: Provable speed-ups from crossover,” in Proc. Conf. Artif. Intell., 2023, pp. 12399–12407.

S. Cerf, B. Doerr, B. Hebras, J. Kahane, and S. Wietherg, “The first proven performance guarantees for the non-dominated sorting genetic algorithm II (NSGA-II) on a combinatorial optimization problem,” in Proc. Int. Joint Conf. Artif. Intell., 2023, pp. 5522–5530.

M. Laumanns, L. Thiele, and E. Zitzler, “Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions,” IEEE Trans. Evol. Comput., vol. 8, no. 2, pp. 170–182, Apr. 2004.

C. Igel, C. Qian, and K. Tang, “A general approach to running time analysis of multi-objective evolutionary algorithms,” in Proc. Int. Joint Conf. Artif. Intell., 2018, pp. 1405–1411.

Z. Huang, Y. Zhou, C. Luo, and Q. Lin, “A runtime analysis of typical decomposition approaches in MOEA/D framework for many-objective optimization problems,” in Proc. Int. Joint Conf. Artif. Intell., 2021, pp. 1682–1688.

M. Preuss, Multimodal Optimization by Means of Evolutionary Algorithms. Cham, Switzerland: Springer, 2015.

J. J. Liang, C. Yue, and B. Qu, “Multimodal multi-objective optimization: A preliminary study,” in Proc. Congr. Evol. Comput., 2016, pp. 2454–2461.

R. Tanabe and H. Ishibuchi, “A review of evolutionary multimodal multi-objective optimization,” IEEE Trans. Evol. Comput., vol. 24, no. 1, pp. 193–200, Feb. 2020.

O. Giel and P. K. Lehre, “On the effect of populations in evolutionary multi-objective optimisation,” Evol. Comput., vol. 18, no. 3, pp. 335–356, 2010.

H. Ishibuchi, Y. Tanigaki, H. Masuda, and Y. Noyima, “Distance-based analysis of crossover operators for many-objective knapsack problems,” in Parallel Problem Solving From Nature, Cham, Switzerland: Springer, 2014, pp. 600–610.

C. McDiarmid, “On the method of bounded differences,” in Surveys in Combinatorics. Cambridge, U.K.: Cambridge Univ. Press, 1989, pp. 48–118.

B. Doerr, “Probabilistic tools for the analysis of randomized optimization heuristics,” in Theory of Evolutionary Computation: Recent Developments in Discrete Optimization, B. Doerr and F. Neumann, Eds. Cham, Switzerland: Springer, 2020, pp. 1–87. [Online]. Available: https://arxiv.org/abs/1801.06733.

M. Emmerich, A. Deutz, R. Li, and J. Kruisselbrink, “Getting lost or getting trapped: On the effect of moves to incompressible points in multiobjective hillclimbing,” in Proc. Genet. Evol. Comput. Comput. GECCO Workshop, 2010, pp. 1963–1966.
Weijie Zheng (Member, IEEE) received the bachelor’s degree in mathematics and applied mathematics from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in July 2013, and the Doctoral degree in computer science and technology from Tsinghua University, Beijing, China, in October 2018.

He is currently an Assistant Professor with the School of Computer Science and Technology, International Research Institute for Artificial Intelligence, Harbin Institute of Technology (Shenzhen), Shenzhen, China. His current research majorly focuses on the theoretical analysis and design of evolutionary algorithms, such as NSGA-II, binary differential evolution, and estimation-of-distribution algorithms.

Benjamin Doerr received the Diploma, Ph.D., and Habilitation degrees in mathematics from the Christian-Albrechts-Universität zu Kiel, Kiel, Germany, in 1998, 2000, and 2005, respectively.

He is a Full Professor (professeur de classe exceptionnelle) with the École Polytechnique, Palaiseau, France. From 2005 to 2013, he was a Senior Researcher (tenured) with the Max Planck Institute for Informatics, Saarbrücken, Germany, and an Adjunct Professor with Saarland University, Saarbrücken. His research interests include mathematical analyses of problem-specific algorithms and randomized search heuristics as well as the development of new algorithms based on such analyses.

Prof. Doerr is an Editor of several scientific journals, among them, Artificial Intelligence, Journal of Complexity, and Natural Computing.