



HAL
open science

A Mathematical Runtime Analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III)

Simon Wietheger, Benjamin Doerr

► **To cite this version:**

Simon Wietheger, Benjamin Doerr. A Mathematical Runtime Analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). 32nd INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE IJCAI 2023, Aug 2023, Macao, Macau SAR China. pp.5657-5665, 10.24963/ijcai.2023/628 . hal-04104770v2

HAL Id: hal-04104770

<https://hal.science/hal-04104770v2>

Submitted on 6 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Mathematical Runtime Analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III)*

Simon Wietheger¹, Benjamin Doerr²

¹Hasso Plattner Institute, University of Potsdam, Germany.‡

²Laboratoire d’Informatique (LIX), CNRS, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France

firstname.lastname@student.hpi.de, lastname@lix.polytechnique.fr

Abstract

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is the most prominent multi-objective evolutionary algorithm for real-world applications. While it performs evidently well on bi-objective optimization problems, empirical studies suggest that it is less effective when applied to problems with more than two objectives. A recent mathematical runtime analysis confirmed this observation by proving the NSGA-II for an exponential number of iterations misses a constant factor of the Pareto front of the simple 3-objective ONEMINMAX problem.

In this work, we provide the first mathematical runtime analysis of the NSGA-III, a refinement of the NSGA-II aimed at better handling more than two objectives. We prove that the NSGA-III with sufficiently many reference points – a small constant factor more than the size of the Pareto front, as suggested for this algorithm – computes the complete Pareto front of the 3-objective ONEMINMAX benchmark in an expected number of $O(n \log n)$ iterations. This result holds for all population sizes (that are at least the size of the Pareto front). It shows a drastic advantage of the NSGA-III over the NSGA-II on this benchmark. The mathematical arguments used here and in the previous work on the NSGA-II suggest that similar findings are likely for other benchmarks with three or more objectives.

1 Introduction

Many practical applications require to optimize for multiple, conflicting objectives. Such tasks can be tackled by population-based algorithms, whose population eventually represents a set of Pareto solutions, which cannot strictly be dominated by any other solution. Thereby, they represent multiple useful trade-offs between the objectives and allow the user to choose among these according to their personal preferences. Indeed, evolutionary algorithms (EAs), or, more

precisely, multi-objective evolutionary algorithms (MOEAs), have been successfully applied to many real-world problems [Zhou *et al.*, 2011]. Among these, Zhou *et al.* [Zhou *et al.*, 2011] identify the non-dominated sorting genetic algorithm (NSGA-II) [Deb *et al.*, 2002] as the most prominent one.

Both empirical evaluations [Khare *et al.*, 2003; Purshouse and Fleming, 2007] and recent mathematical runtime analyses (see the previous works section) confirm the strong results of the NSGA-II on bi-objective benchmarks. The performance on problems with 3 or more objectives, however, is not as well understood. Empirical studies, for example [Khare *et al.*, 2003], suggest that the NSGA-II struggles with such problems. A recent mathematical runtime analysis [Zheng and Doerr, 2022b] shows that the NSGA-II regularly loses desirable solutions when optimizing the 3-objective 3-OMM problem, and consequently, cannot find its Pareto front (the set of Pareto optimal solution values) in sub-exponential time. As a remedy, Deb and Jain [Deb and Jain, 2014] proposed a modified version of the NSGA-II, called NSGA-III. It replaces the crowding distance, a measure which the NSGA-II uses in addition to the dominance relation to determine which individuals are taken in the next generation, by a procedure involving reference points in the solution space. Their evaluations on benchmarks with 3 to 15 objectives show that the NSGA-III is suitable for more than 2 objectives.

These empirical insights are, however, not yet backed with a theoretical understanding. In order to fill this gap, we mathematically analyze the runtime of the NSGA-III on the 3-OMM problem. We show that by employing sufficiently many reference points (a small constant factor more than the size of the Pareto front, as suggested for this algorithm) and a population at least of the size of the Pareto front, $N \geq (\frac{n}{2} + 1)^2$, once a solution for a point on the Pareto front is found, the population will always contain such a solution. This is a notable difference to the NSGA-II [Zheng and Doerr, 2022b] and enables us to prove that after an expected number of $O(n \log n)$ iterations the NSGA-III (for all future iterations) has a population that covers the Pareto front. Overall, this result indicates, in a rigorous manner, that the selection mechanism of the NSGA-III has significant advantages over the one of the NSGA-II. Possibly, our result also indicates that more algorithm users should switch from the NSGA-II, still the dominant algorithm in practice, to the NSGA-III. We note that the latter has as additional parameter

* Author-generated version. Full version at <https://arxiv.org/abs/2211.08202>.

‡ Work done while visiting École Polytechnique, France.

the number of reference points, but the general recommendation to use by a small factor more reference points than the size of the Pareto front (or, in the case of approximate solutions, the size of the desired solution set) renders it easy to choose this parameter. We note that our results support this parameter choice, our proven guarantees also hold from the point on when the number of reference points is a small constant factor larger than the Pareto front. We note that using more reference points does not significantly increase the runtime (not at all when counting fitness evaluations and only moderately when counting wall-clock time), so in any case the choice of this parameter appears not too critical.

2 Previous Work

For the sake of brevity, we do not further discuss empirical and practical works here. Since the beginning of the century, mathematical runtime analyses [Auger and Doerr, 2011; Neumann and Witt, 2010; Jansen, 2013; Zhou *et al.*, 2019; Doerr and Neumann, 2020] have been employed also to multi-objective randomized search heuristics [Laumanns *et al.*, 2002; Giel, 2003; Thierens, 2003]. At first, research focused on analyzing simple, synthetic algorithms like the SEMO and the global SEMO (GSEMO). Though in practical applications, usually more sophisticated algorithms are used, these analyses still led to useful insights. Later, the runtimes of more realistic algorithms have been studied mathematically [Brockhoff *et al.*, 2008; Nguyen *et al.*, 2015; Doerr *et al.*, 2016; Li *et al.*, 2016; Huang *et al.*, 2019; Huang and Zhou, 2020; Bian *et al.*, 2023]. Only recently, first mathematical runtime analyses of the NSGA-II on bi-objective benchmarks have appeared. The first one of these proves a running time of $O(Nn \log n)$ function evaluations on the ONEMINMAX benchmark and of $O(Nn^2)$ on the LOTZ (LEADINGONESTRILINGZEROES) benchmark, when employing a population of $N \geq 4(n + 1)$ individuals [Zheng *et al.*, 2022]. A central observation in their proof is that this population size suffices to ensure that, once a solution for a point on the Pareto front is sampled, the population will always contain a solution with this objective value. Employing a population size that exactly matches the size of the Pareto front does not suffice, as then, for an exponential time, the NSGA-II will miss a constant fraction of the Pareto front. Nevertheless, a smaller population is still able to find good approximations of the Pareto front [Zheng and Doerr, 2022a]. Further, by assuming the objectives to be sorted identically, the required size of the population was reduced to $2(n + 1)$ [Bian and Qian, 2022]. The same work studies the NSGA-II when employing crossover, but does not improve the running time bounds of [Zheng *et al.*, 2022]. Also, it introduces a novel selection mechanism, improving the running time on the LOTZ benchmark to $O(n^2)$. Recently, the NSGA-II was studied on a multimodal benchmark [Doerr and Qu, 2023a]. Very recently, also lower bounds were proven [Doerr and Qu, 2023b], examples for the usefulness of crossover were found [Doerr and Qu, 2023c; Dang *et al.*, 2023], and a runtime analysis on a combinatorial optimization problem appeared [Cerf *et al.*, 2023]. All these results only cover bi-objective benchmarks.

The only mathematical runtime analysis of the NSGA-II on a benchmark consisting of more than two objectives gave a disillusioning result [Zheng and Doerr, 2022b]. When run on the simple ONEMINMAX benchmark, a multi-objective version of the classic ONEMAX benchmark, and the number of objectives is some constant $m \geq 3$, the combined parent and offspring population can contain only $O(n)$ solutions with positive crowding distance. All other solutions have a crowding distance of zero, hence the selection between them is fully at random. Since this benchmark has a Pareto front size of order $n^{\lceil m/2 \rceil}$, at least such a population size is necessary when trying to compute the Pareto front. With these (necessary) parameters, almost all selection decisions are random, which easily implies that regularly Pareto optimal solution values are lost from the population. This easy argument suggests that the difficulties proven in that work are not restricted to the ONEMINMAX benchmark, but are likely to appear for many problems having at least three objectives.

For the simple SEMO, more results exist on benchmarks with more than two objectives. We describe them briefly to ease the comparison, but note that the very different main working principle – keeping all non-dominated solutions – makes the SEMO somewhat special and not always very practical. The first bounds on the expected number of function evaluations when optimizing the m -objective variants of COUNTINGONESCOUNTINGZEROES and LEADINGONESTRILINGZEROES, m COCZ and m LOTZ, are in $O(n^{m+1})$, for bit strings of length n [Laumanns *et al.*, 2004b]. For m COCZ, the bound was later improved to $O(n^m)$, if $m > 4$, and $O(n^3 \log n)$, if $m = 4$ [Bian *et al.*, 2018]. Further, the MOEA/D, an algorithm that decomposes a multi-objective problem into multiple single-objective problems and solves them in a co-evolutionary manner, has been studied on m COCZ and m LOTZ [Huang *et al.*, 2021]. As this approach drastically differs from the NSGA-II and NSGA-III, we do not discuss these results in detail.

We note that all these results consider synthetic problems composed of unimodal objectives. This is not surprising given that a multi-objective analogue of the multimodal JUMP benchmark was proposed only recently [Doerr and Zheng, 2021] and given that there are only very sporadic runtime analyses for bi-objective combinatorial optimization problems (not counting those from multi-objectivizing single-objective problems), e.g., [Laumanns *et al.*, 2004a; Kumar and Banerjee, 2006; Neumann, 2007; Horoba, 2009; Neumann and Witt, 2022; Cerf *et al.*, 2023]

3 Preliminaries

We now define the required notation for multi-objective optimization as well as the considered objective functions and give an introduction to the NSGA-III.

3.1 Multi-objective Optimization

For $m \in \mathbb{N}$, an m -objective function f is a tuple (f_1, \dots, f_m) , where $f_i: \Omega \rightarrow \mathbb{R}$ for some search space Ω . For all $x \in \Omega$, we define $f(x) = (f_1(x), \dots, f_m(x))$. Other than in single-objective optimization, there is usually no solution that minimizes all m objective functions simultaneously. For two solutions x, y , we say that x dominates y and write $x \preceq y$ if and

only if $f_j(x) \leq f_j(y)$ for all $1 \leq j \leq m$. If additionally there is a j_0 such that $f_{j_0}(x) < f_{j_0}(y)$, we say that x *strictly dominates* y , denoted by $x \prec y$. A solution is *Pareto-optimal* if it is not strictly dominated by any other solution. We refer to the set of objective values of Pareto-optimal solutions as the *Pareto front*. In our analyses, we analyze the number of function evaluations until the population covers the Pareto front, i.e., until for each value p on the Pareto front the population contains a solution x with $f(x) = p$. For a vector $v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$, we denote its length by

$$|v| = \sqrt{v_1^2 + v_2^2 + v_3^2}.$$

3.2 3-OMM Benchmark

We are interested in studying the NSGA-III on a 3-objective function. The ONEMINMAX function, first proposed by [Giel and Lehre, 2010], translates the well-established ONEMAX benchmark into a bi-objective setting. It is defined as ONEMINMAX: $\{0, 1\}^n \rightarrow \mathbb{N} \times \mathbb{N}$ by

$$\begin{aligned} \text{ONEMINMAX}(x) &= (\text{ZEROMAX}(x), \text{ONEMAX}(x)) \\ &= \left(n - \sum_{i=1}^n x_i, \sum_{i=1}^n x_i \right) \end{aligned}$$

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

The following 3-objective version 3-OMM (for 3-ONEMINMAX) was proposed in [Zheng and Doerr, 2022b]. For even n , define 3-OMM: $\{0, 1\}^n \rightarrow \mathbb{N}^3$ by

$$3\text{-OMM}(x) = \left(n - \sum_{i=1}^n x_i, \sum_{i=1}^{n/2} x_i, \sum_{i=n/2+1}^n x_i \right)$$

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

3.3 NSGA-III

The main structure of the NSGA-III [Deb and Jain, 2014] is identical to the one of the NSGA-II [Deb *et al.*, 2002]. It is initialized with a random population of size N . In each iteration, the user applies mutation and/or crossover operators to generate an offspring population of size N . As the NSGA framework is an MOEA with a fixed population size, out of this total of $2N$ individuals, N have to be selected for the next iteration.

Because non-dominated solutions are to be preferred, the following ranking scheme is used to set the dominance relation as the predominant criterion for the survival of individuals. Individuals that are not strictly dominated by any other individual in the population obtain rank 1. Recursively, the other ranks are defined. Each individual that has not yet been ranked and is only strictly dominated by individuals of rank $1, \dots, k-1$ is assigned rank k . Clearly, an individual is more interesting the lower its rank is. Let F_i denote the set of individuals with rank i and let i^* be minimal such that $\sum_{i=1}^{i^*} |F_i| \geq N$. All individuals with rank at most $i^* - 1$ survive into the next generation. Further, $0 < k \leq N$ individuals of rank i^* have to be selected for the next generation such that the new population is again of size N , and the next iteration can begin. The only difference between the NSGA-II and the

Algorithm 1: NSGA-II and NSGA-III

- 1 Let the initial population P_0 be composed of N individuals chosen independently and uniformly at random from $\{0, 1\}^n$.
 - 2 **for** $t = 0, 1, 2, \dots$ **do**
 - 3 Generate offspring population Q_t with size N
 - 4 Use fast-non-dominated-sort() from Deb *et al.* [Deb *et al.*, 2002] to divide $R_t = P_t \cup Q_t$ into F_1, F_2, \dots
 - 5 Find $i^* \geq 1$ such that $\sum_{i=1}^{i^*-1} |F_i| < N$ and $\sum_{i=1}^{i^*} |F_i| \geq N$
 - 6 $Z_t = \bigcup_{i=1}^{i^*-1} F_i$
 - 7 Select $\tilde{F}_{i^*} \subseteq F_{i^*}$ such that $|Z_t \cup \tilde{F}_{i^*}| = N$ (use crowding-distance for NSGA-II and Algorithm 3 for NSGA-III)
 - 8 $P_{t+1} = Z_t \cup \tilde{F}_{i^*}$
-

NSGA-III is the procedure of selecting the k individuals of rank i^* . While the NSGA-II employs crowding-distance, the NSGA-III uses reference points, typically distributed in some structured manner on the normalized hyperplane, in order to select a diverse population. For the whole framework, see Algorithm 1. Note that whenever we refer to sets of individuals, we actually refer to multi-sets as each solution might be represented multiple times in the population.

In order to select individuals from the critical rank i^* , the NSGA-III normalizes the objective functions and associates each individual with a reference point.

Regarding the normalization step, we do not consider the procedure as given in [Deb and Jain, 2014] but the improved and more detailed normalization given in [Blank *et al.*, 2019] by one of the two original authors among others. Consider any iteration. Let \hat{z}_j^* be the minimum observed value in the j th objective over all generations including the current offspring. We use \hat{z}_j^* to estimate the ideal point. Further, for each objective j we compute an extreme point in that objective by using an achievement scalarization function. Consider the hyperplane spanned by these points. The intercepts of this hyper plane with the coordinate axes give the Nadir point estimate \hat{z}^{nad} . In case that H is not well-defined by the extreme points or if an intercept is either smaller than a given positive threshold ϵ_{nad} or larger than the highest observed value over all generations in that objective, \hat{z}^{nad} is instead defined by the maximum value in each objective of individuals in the first non-dominated front. Last, for each objective in which the Nadir point estimate is smaller than the ideal point estimate plus the threshold ϵ_{nad} , the maximum value in that objective over all current non-dominated fronts is used for the Nadir point estimate in that objective instead. The normalized objective functions f_j^n are now defined as

$$f_j^n(x) = \frac{f_j(x) - \hat{z}_j^*}{\hat{z}_j^{\text{nad}} - \hat{z}_j^*} \quad (1)$$

for $j \in \{1, \dots, M\}$. Algorithm 2 formalizes the normalization procedure, for a more detailed description see [Blank *et*

al., 2019]. We note that Blank et al. span the hyperplane H by the extreme points *after* subtracting the ideal point estimate \hat{z}^* , while in the interest of a clear notation we span H by the original extreme points. This leads some individual lines slightly differing from [Blank et al., 2019], though the described algorithm is identical.

Algorithm 2: Normalization as given by [Blank et al., 2019]

Input : $f = (f_1, \dots, f_M)$: objective function
 $z^* \in \mathbb{R}^M$: observed min. in each objective
 $z^w \in \mathbb{R}^M$: observed max. in each objective
 $E \subseteq \mathbb{R}^M$: extreme points of previous iteration, initially $\{\infty\}^M$

- 1 **for** $j = 1$ **to** M **do**
- 2 $\hat{z}_j^* = \min\{z_j^*, \min_{z \in Z} f_j(z)\}$
- 3 Determine an extreme point $e^{(j)}$ in the j th objective from $Z \cup E$ using an achievement scalarization function
- 4 valid = False
- 5 **if** $e^{(1)}, \dots, e^{(M)}$ are linearly independent **then**
- 6 valid = True
- 7 Let H be the hyperplane spanned by $e^{(1)}, \dots, e^{(M)}$
- 8 **for** $j = 1$ **to** M **do**
- 9 I_j = the intercept of H with the j th objective axis
- 10 **if** $I_j < \epsilon_{\text{nad}}$ or $I_j > z_j^w$ **then**
- 11 $\hat{z}_j^{\text{nad}} = I_j$
- 12 **else**
- 13 valid = False
- 14 **break**
- 15 **if** valid = False **then**
- 16 **for** $j' = 1$ **to** M **do** $\hat{z}_{j'}^{\text{nad}} = \max_{x \in F_1} f_{j'}(x)$;
- 17 **for** $j = 1$ **to** M **do**
- 18 **if** $\hat{z}_j^{\text{nad}} < \hat{z}_j^* + \epsilon_{\text{nad}}$ **then**
 $\hat{z}_j^{\text{nad}} = \max_{x \in F_1 \cup \dots \cup F_k} f_j(x)$;
- 19 Define $f_j^n(x) = (f_j(x) - \hat{z}_j^*) / (\hat{z}_j^{\text{nad}} - \hat{z}_j^*) \quad \forall x \in \{0, 1\}^n, j \in \{1, \dots, M\}$

After the normalization, each individual of rank at most i^* is associated with its closest reference point with respect to the normalized objectives. More precisely, an individual x is associated to the reference point $\text{rp}(x)$ that minimizes the angle between the point vectors of x and $\text{rp}(x)$. That is, $\text{rp}(x)$ is the reference point such that the distance between x and the line passing through the origin and $\text{rp}(x)$ is minimal. Then, one iterates through the reference points, always selecting the one with the fewest associated individuals that are already selected for the next generation. Ties are resolved randomly. If the reference point only has associated individuals that are already selected, it is skipped. Otherwise, among the not yet selected individuals the one closest to the reference point (with

respect to the normalized objective function) is selected for the next generation. Once more, ties are resolved randomly. If the next generation already contains an individual that is associated with the reference point, other measures than the distance to the reference point can be considered. The selection terminates as soon as the required number of individuals is reached. This procedure is formalized in Algorithm 3.

Algorithm 3: Selection based on a set R of reference points when maximizing the function f

Input : Z_t : the multi-set of already selected individuals
 F_{i^*} : the multi-set of individuals to choose from

- 1 $f^n = \text{NORMALIZE}(f, Z = Z_t \cup F_{i^*})$ using Algorithm 2
- 2 Associate each individual $x \in Z_t \cup F_{i^*}$ to the reference point $\text{rp}(x)$
- 3 For each reference point $r \in R$, let ρ_r denote the number of (already selected) individuals in Z_t associated with r
- 4 $R' = R, \tilde{F}_{i^*} = \emptyset$
- 5 **while** True **do**
- 6 Let $r_{\min} \in R'$ be such that $\rho_{r_{\min}}$ is minimal (break ties randomly)
- 7 Let $x_{r_{\min}} \in F_{i^*} \setminus \tilde{F}_{i^*}$ be the individual that is associated with r_{\min} and minimizes the distance between $f^n(x_{r_{\min}})$ and r_{\min} (break ties randomly)¹
- 8 **if** $x_{r_{\min}}$ exists **then**
- 9 $\tilde{F}_{i^*} = \tilde{F}_{i^*} \cup \{x_{r_{\min}}\}$
- 10 $\rho_{r_{\min}} = \rho_{r_{\min}} + 1$
- 11 **if** $|S_t| + |\tilde{F}_{i^*}| = N$ **then**
- 12 **break all and return** \tilde{F}_{i^*}
- 13 **else**
- 14 $R' = R' \setminus \{r\}$

For our analyses, we assume that the NSGA-III employs a set of structured reference points in the normalized hyperplane as proposed by Deb and Jain [Deb and Jain, 2014]. In the case of 3 objectives, this corresponds to a set of points in the triangle spanned by $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, and $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$. Divide the lines between two pairs of these points into p divisions of equal length. Consider the lines that pass through the start and end points of all divisions and are orthogonal to the respective side. Every intersection of these lines marks a reference point, see Figure 1. By [Deb and Jain, 2014, Equation 3], this creates $\binom{3+p-1}{p} = \binom{p+2}{2}$ reference points. Observe that these reference points partition the non-negative domain of the spanned triangle in regular hexagonal Voronoi cells.

¹If $\rho_{r_{\min}} > 0$, $x_{r_{\min}}$ can be selected in any other diversity-preserving manner from the associated individuals.

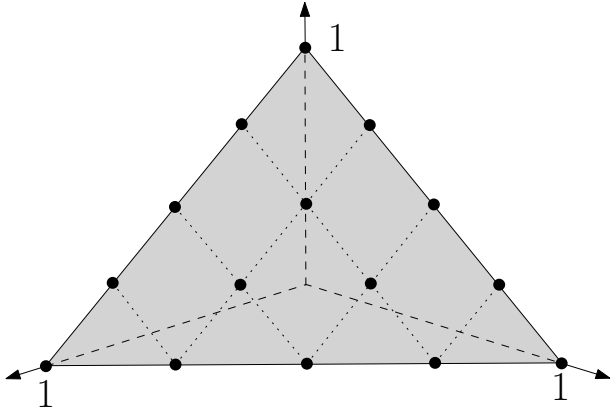


Figure 1: Structured set of reference points for 3 objectives with $p = 4$ divisions based on [Deb and Jain, 2014, Figure 1]).

4 How the Reference Point Mechanism Avoids Losing Solution Values

Before analyzing the optimization time of the NSGA-III on 3-OMM, we show that, by employing sufficiently many reference points, once the population covers a point on the Pareto front, it is covered for all future iterations. To this end, we first analyze how the normalization shifts the points on the Pareto front to then conclude that every reference point is associated with at most one point of the Pareto front. With this, we argue that already sampled points on the Pareto front are never lost again. Our analysis assumes that the population is non-degenerated, i.e., that for each objective the maximum value over all generations is larger than the minimum value over all generations. However, w.h.p. this holds starting in the first iteration. We note that for 3-OMM no individual strictly dominates another. Thus, all individuals are in the first rank and we analyze on the way ties inside that rank are broken, focusing on the only difference between the NSGA-II and the NSGA-III.

Lemma 1. *Assume that for each objective value observed in a previous iteration, the current population contains at least one individual with that objective value. Let for each objective i , z_i^{\min} and z_i^{\max} be the minimum and maximum value in the current population. Then, each objective i is normalized as $f_i^n(x) = \frac{f_i(x) - z_i^{\min}}{z_i^{\max} - z_i^{\min}}$.*

Proof. Note that due to the assumption in the statement the z_i^{\min} and z_i^{\max} are the extreme values not only for the current population but also among all previous generations. Thus, $\begin{pmatrix} z_1^{\min} \\ z_2^{\min} \\ z_3^{\min} \end{pmatrix}$ is exactly the ideal point estimate \hat{z}^* . Further, $\begin{pmatrix} z_1^{\max} \\ z_2^{\max} \\ z_3^{\max} \end{pmatrix}$ is the Nadir-point estimate \hat{z}^{nad} . This estimate is defined by the intercepts of the hyperplane spanned by the extreme points with the objective axes though its objective values might be set to the maximum of the respective objective among all individuals or all non-dominated individuals in the current generation. As for 3-OMM, all individuals are non-dominated, in the latter case the Nadir point estimate in each objective j is exactly z_j^{\max} . The first case only occurs if the extreme points describe a well-defined hyperplane and the intercepts of that

hyperplane are at most the maximum observed value in the respective objective over all generations. Note that all possible objective values for 3-OMM lay in the same plane $E : v_1 + v_2 + v_3 = n$. The intercept of this plane with each objective is n . As for 3-OMM, the second and third objective never take any value larger than $n/2$, this case never occurs. Thus, by Equation 1, $f_i^n(x) = (f_i(x) - z_i^{\min}) / (z_i^{\max} - z_i^{\min})$. \square

Lemma 2. *By employing $p \geq 21n$ divisions along each objective, all individuals that are associated with the same reference point have the same objective value.*

Proof. Due to space constraints, we only give the proof idea and the less technical arguments. The complete proof is to be found in the full version.

Each individual x is associated to the reference point r that minimizes the distance between $f^n(x)$ and the line between the origin and r . Equivalently, x is associated to the reference point r that minimizes the angle between the point vectors r and $f^n(x)$. To show the statement, we first upper bound the angle between normalized objective values and their nearest reference point and then lower bound the angle between two different normalized objective values to show that for $p \geq 21n$ the latter is more than twice as large as the former. Thereby, individuals with different objective values are never associated with the same reference point because if they were then the angle between their normalized objective values would be at most the sum of their angles to that reference point.

To upper bound the angle between $f^n(x)$ and r , first note that when scaling $f^n(x)$ to $t = a \cdot f^n(x)$ such that t lies in the non-negative domain of the reference point plane ($\sum_{i=1}^3 t_i = 1$), the angle between t and r is the same as between $f^n(x)$ and r . When employing p divisions, the reference points partition the non-negative domain of the reference point plane into equilateral triangles with side length $\sqrt{2}/p$. As t lies in any of these triangles, there is a reference point r such that $|t - r| \leq \sqrt{2}/p$.

Consider the point $z = (t + r)/2$ exactly in between t and r and $\Delta = t - z$. Then, $t = z + \Delta$ and $r = z - \Delta$. Thus, the angle between the point vectors t and r is at most $\cos^{-1} \left(\frac{(z + \Delta) \circ (z - \Delta)}{|(z + \Delta)| \cdot |(z - \Delta)|} \right)$. As $|z|^2 \geq 1/9$ because $\sum_{i=1}^3 z_i = 1$ and $|\Delta| \leq \sqrt{2}/p$, we have

$$\begin{aligned} & \frac{(z + \Delta) \circ (z - \Delta)}{|(z + \Delta)| \cdot |(z - \Delta)|} \\ &= \frac{\sum_{i=1}^3 (z_i + \Delta_i)(z_i - \Delta_i)}{\sqrt{\sum_{i=1}^3 (z_i + \Delta_i)^2} \sqrt{\sum_{i=1}^3 (z_i - \Delta_i)^2}} \\ &\geq \frac{\sum_{i=1}^3 z_i^2 - \Delta_i^2}{\sum_{i=1}^3 z_i^2} = \frac{|z|^2 - |\Delta|^2}{|z|^2} = 1 - \frac{|\Delta|^2}{|z|^2} \geq 1 - \frac{18}{p^2}. \end{aligned}$$

Since the function \cos^{-1} is decreasing, this yields that the angle between the vectors is at most $\cos^{-1}(1 - 18/p^2)$.

By similar arguments, we bound the angle between the normalized objective value vectors between any pair of individuals in the supplementary material and show that by employing $p \geq 21n$ divisions for the reference points, the angle between

the normalized objective values of any pair of individuals is more than twice as large as the angle between any individual and its closest reference point, when considering their respective vectors from the origin. Thus, no two individuals with different objective values are associated with the same reference point. \square

We note that $21n$ divisions correspond to $\binom{21n+2}{2} \in \mathcal{O}(n^2)$ divisions, so the number of reference points differs by the size of the Pareto front by a constant factor.

Our bound of $p \geq 21n$ is likely not tight in order to guarantee unique associations, as suggested by our experiments in Section 6. Nevertheless, it is only off by a constant factor to the actual bound. This holds as we require more than $n/\sqrt{2}$ divisions to create $(n/2 + 1)^2$ reference points, one for each possible objective value. If we do not have that many reference points, there are at least two different objective values associated with the same reference point and when selecting individuals from that reference point by chance no individuals of one of the objective values might survive.

Using Lemma 2, we now show that the population does not lose once sampled objective values.

Lemma 3. *Consider the NSGA-III optimizing a multi-objective function f with a population of size N . Let F^* be the Pareto front of f . Assume that in each iteration the number of objective values of non-strictly dominated individuals is at most N and that all individuals associated with the same reference point have the same objective value. Then, once the population contains a solution x with $f(x) \in F^*$, the population will contain a solution for x' such that $f(x') = f(x)$ in all future iterations.*

Proof. Consider any iteration with a population that contains such an x' . After the recombination and mutation step, the complete population of old solutions and offspring contains $2N$ solutions. Let F_1 denote the subset of solutions that are not strictly dominated. Then, $x' \in F_1$. If $|F_1| \leq N$, all individuals in F_1 including x' survive into the next generation. Otherwise, the objective functions are normalized with respect to F_1 and the individuals in F_1 are associated with a reference point each. By our assumptions, there are now at most N reference points with at least one associated individual. Thus, at least one individual is selected from each reference point with non-empty association set. In particular, one of the individuals associated with the same reference point as x' survives. By our assumption, it has the same objective value as x' . \square

5 Runtime of the NSGA-III on 3-OMM

We are now able to give a first upper bound on the expected optimization time of the NSGA-III on 3-OMM.

Theorem 4. *Consider the NSGA-III with population size $N \geq (\frac{n}{2} + 1)^2$ and $p \geq 21n$ divisions along each objective optimizing 3-OMM. Assume the reproduction step to be such that each individual in the population has at least a chance c^{-1} to create an offspring via standard mutation. Then, w.h.p., the population covers the Pareto front after $4ecn \ln(n)$ iterations, which corresponds to $O(cn^3 \log(n))$ evaluations of the fitness function.*

Proof. We upper bound the probability that after $4ecn \ln(n)$ iterations not all objective values on the Pareto front have been sampled. To this end, we first give an upper bound on the probability that any specific objective value (a, b) has not been sampled after $4ecn \ln(n)$ iterations. In each iteration i , let $d_i = \min_{s \in \text{Population}} |f_1(s) - a| + |f_2(s) - b|$ be the distance of (a, b) to the closest individual in the population, i.e., the minimum number of bit flips required to turn this individual into one with objective value (a, b) . By Lemma 3 the population will never lose a sampled objective value, so d_i never decreases. Further, for all $1 \leq \ell \leq n$, define the geometrically distributed random variable X_ℓ as the number of iterations i with $d_i = \ell$. Then, the number of iterations until (a, b) is covered is $X = \sum_{\ell=1}^n X_\ell$. Observe that X_ℓ has a success probability of at least $p_i = \frac{i}{ecn}$ by choosing the closest individual for mutation ($\frac{1}{c}$), flipping any of the at least i bits that take it closer ($\frac{i}{n}$), and not flipping any other bit ($\frac{1}{e}$). By [Doerr, 2020, Theorem 1.10.35], $\Pr[X \geq (1 + 3)ecn \ln(n)] \leq n^{-3}$, i.e., the probability that the population does not cover (a, b) after $4ecn \ln(n)$ iterations is at most n^{-3} . Hence, by the union bound, the probability that all $(\frac{n}{2} + 1)^2$ objective values are sampled after $4ecn \ln(n)$ iterations is at least

$$1 - \left(\frac{n}{2} + 1\right)^2 \cdot n^{-3} \geq 1 - \frac{1}{n}.$$

Thus, w.h.p. $4ec \ln(n)$ iterations suffice to cover the complete Pareto front. Each of these iterations employs $(\frac{n}{2} + 1)^2$ fitness evaluations, so w.h.p. a total of $O(cn^3 \ln(n))$ fitness evaluations are required. \square

6 Experimental Evaluation

We support our theoretical findings with empirical evaluations of the NSGA-III on the 3-OMM benchmark. To this end, we employ the DEAP library [Fortin *et al.*, 2012], which provides a framework for benchmarking evolutionary algorithms and holds an implementation of the NSGA-III and NSGA-II.² All experiments on the NSGA-III are conducted with a population size of $N = (n/2 + 1)^2$ (the size of the complete Pareto front) and $p = 4.65n$ divisions on the reference point plane. While Theorem 4 requires $p \geq 21n$, the theoretical proven bound is most likely not tight. In all our experiments we included a control routine to check whether any points on the Pareto front were lost during the process. For $p = 4.65n$, no point was ever lost, suggesting a much smaller bound than $p \geq 21n$. In the reproduction step, we apply standard bit mutation (flipping each bit independently with probability $1/n$) on each individual.

Figure 2 shows the coverage of the Pareto front of 3-OMM in runs of the NSGA-III and the NSGA-II. While all 3 runs

²See https://github.com/SimonWiet/experiments_nsga3 for code and results of the experiments. We slightly modified DEAP's implementation of the NSGA-II to avoid some unwanted bias on the order of individuals with same objective values or crowding distance, which lead to significantly different performances on the 3-OMM benchmark. We aimed at staying as close to the intuitive interpretation of the original algorithm as possible. Details are discussed in the repository.

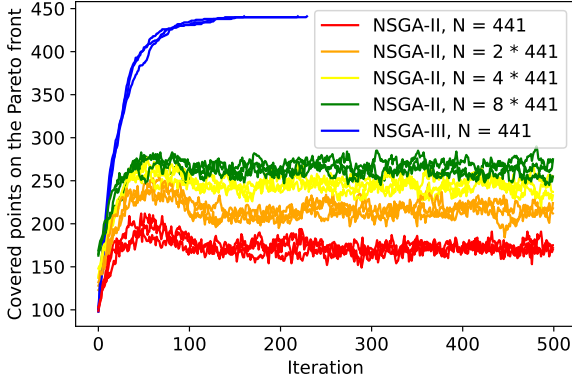


Figure 2: The number of covered points on the Pareto front per iteration of the NSGA-III and the NSGA-II, not using crossover, with population size 1, 2, 4, and 8 times the size of the Pareto front (441) when optimizing the 3-OMM benchmark. Three independent runs are depicted for each algorithm/population size. The lines for the NSGA-III (blue) stop when it has covered the complete Pareto front.

of the NSGA-III (with a population size equal to the size of the Pareto front) find the complete Pareto front in less than 300 iterations, the NSGA-II shows a real progress only in the first few iterations and then stagnates at some fraction of the Pareto front covered (with some variance). Increasing the population size mildly increases this stagnation level, but even with a population of 8 times the size of the Pareto front, the NSGA-III never has even 300 out of the 441 elements in the Pareto front. The figure shows that the effect of doubling the population size of the NSGA-II reduces with increasing population size, which suggests that a truly large population size would be needed in order to find the complete Pareto front in polynomial time.

The figure illustrates well that the advantage of the NSGA-III over the NSGA-II lies in its property of never losing Pareto dominant solutions, which is reflected in the non-decreasing curve of the NSGA-III in the figure as opposed to the oscillating curves of the NSGA-II.

Further, we run the NSGA-III on 3-OMM using standard bit mutation and uniform crossover (for a pair of parent bit strings, at each bit position there is a chance of 1/2 of swapping the respective bits between the parents). We randomly partitioned the population in pairs of 2 bit strings and for each pair tested a chance of 1/2 or 9/10 to perform a crossover on this pair before applying standard mutation.

Figure 3 shows the number of iterations it took the NSGA-III to find the complete Pareto front of the 3-OMM benchmark for the different crossover rates and over different values of the problem size n . The data shows quite clearly that a crossover rate of 1/2 moderately increases the runtime and that a crossover rate of 0.9 significantly increases the runtime. This suggests that the main progress is due to the mutation operator, justifying our focus on standard bit mutation in our analysis.

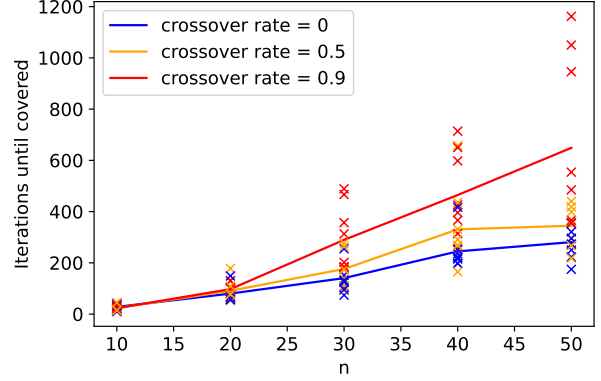


Figure 3: The number of iterations required to cover the complete Pareto front for multiple runs of the NSGA-III. Shown are 8 runs for each of the different values of problem size $n \in \{10, 20, \dots, 50\}$ and the different crossover variants (mutation only, and crossover rates of 1/2 and 9/10).

7 Conclusion

This first mathematical runtime analysis of the NSGA-III, we proved that this algorithm with suitable parameters never loses a Pareto optimal solution value when optimizing the 3-OMM problem. This is in drastic contrast to the NSGA-II, which provably also with large populations regularly suffers from such losses and consequently cannot find (or approximate beyond a certain constant factor) the Pareto front of 3-OMM in sub-exponential time [Zheng and Doerr, 2022b]. From this positive property of the NSGA-III, we derive that it finds the full Pareto front of 3-OMM in an expected number of $O(n \log n)$ iterations. Our experimental results confirm the drastically different behavior of the algorithms.

From the mathematical proofs we are optimistic that the key property of not losing desired solution values extends to broader classes of problems. In fact, our main argument was that the angle (with the origin) between any two different normalized solutions is large enough to prevent that both are associated with the same reference point. We have not used any properties of the optimization process there, but only the structure of the set of objective values of the problem. We are thus optimistic that similar properties hold for other optimization problems, say with integral objectives of a bounded range.

This first rigorous result on the NSGA-III clearly is not enough to derive reliable advice on which MOEAs to prefer, but we believe that it should motivate more users of the NSGA-II to also try the NSGA-III. If only part of the drastic differences proven here extend to broader classes of problems, this switch is definitely justified.

Acknowledgments

This work was supported by a public grant as part of the Investissements d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH and a fellowship via the International Exchange Program of École Polytechnique.

References

- [Auger and Doerr, 2011] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics*. World Scientific Publishing, 2011.
- [Bian and Qian, 2022] Chao Bian and Chao Qian. Better running time of the non-dominated sorting genetic algorithm II (NSGA-II) by using stochastic tournament selection. In *Parallel Problem Solving From Nature, PPSN 2022*, pages 428–441. Springer, 2022.
- [Bian *et al.*, 2018] Chao Bian, Chao Qian, and Ke Tang. A general approach to running time analysis of multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 1405–1411. IJCAI, 2018.
- [Bian *et al.*, 2023] Chao Bian, Yawen Zhou, Miqing Li, and Chao Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, 2023. To appear.
- [Blank *et al.*, 2019] Julian Blank, Kalyanmoy Deb, and Proteeek Chandan Roy. Investigating the normalization procedure of nsga-iii. In Kalyanmoy Deb, Erik Goodman, Carlos A. Coello Coello, Kathrin Klamroth, Kaisa Miettinen, Sanaz Mostaghim, and Patrick Reed, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, page 229–240, Cham, 2019. Springer International Publishing.
- [Brockhoff *et al.*, 2008] Dimo Brockhoff, Tobias Friedrich, and Frank Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature, PPSN 2008*, pages 651–660. Springer, 2008.
- [Cerf *et al.*, 2023] Sacha Cerf, Benjamin Doerr, Benjamin Hebras, Jakob Kahane, and Simon Wietheger. The first proven performance guarantees for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) on a combinatorial optimization problem. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*. ijcai.org, 2023. To appear.
- [Dang *et al.*, 2023] Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Conference on Artificial Intelligence, AAI 2023*. AAAI Press, 2023. To appear.
- [Deb and Jain, 2014] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18:577–601, 2014.
- [Deb *et al.*, 2002] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
- [Doerr and Neumann, 2020] Benjamin Doerr and Frank Neumann, editors. *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*. Springer, 2020. Also available at http://www.lix.polytechnique.fr/Labo/Benjamin.Doerr/doerr_neumann_book.html.
- [Doerr and Qu, 2023a] Benjamin Doerr and Zhongdi Qu. A first runtime analysis of the NSGA-II on a multimodal problem. *Transactions on Evolutionary Computation*, 2023. <https://doi.org/10.1109/TEVC.2023.3250552>.
- [Doerr and Qu, 2023b] Benjamin Doerr and Zhongdi Qu. From understanding the population dynamics of the NSGA-II to the first proven lower bounds. In *Conference on Artificial Intelligence, AAI 2023*. AAAI Press, 2023. To appear.
- [Doerr and Qu, 2023c] Benjamin Doerr and Zhongdi Qu. Runtime analysis for the NSGA-II: Provable speed-ups from crossover. In *Conference on Artificial Intelligence, AAI 2023*. AAAI Press, 2023. To appear.
- [Doerr and Zheng, 2021] Benjamin Doerr and Weijie Zheng. Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In *Conference on Artificial Intelligence, AAI 2021*, pages 12293–12301. AAAI Press, 2021.
- [Doerr *et al.*, 2016] Benjamin Doerr, Wanru Gao, and Frank Neumann. Runtime analysis of evolutionary diversity maximization for OneMinMax. In *Genetic and Evolutionary Computation Conference, GECCO 2016*, pages 557–564. ACM, 2016.
- [Doerr, 2020] Benjamin Doerr. Probabilistic tools for the analysis of randomized optimization heuristics. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 1–87. Springer, 2020. Also available at <https://arxiv.org/abs/1801.06733>.
- [Fortin *et al.*, 2012] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.
- [Giel and Lehre, 2010] Oliver Giel and Per Kristian Lehre. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18:335–356, 2010.
- [Giel, 2003] Oliver Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Congress on Evolutionary Computation, CEC 2003*, pages 1918–1925. IEEE, 2003.
- [Horoba, 2009] Christian Horoba. Analysis of a simple evolutionary algorithm for the multiobjective shortest path problem. In *Foundations of Genetic Algorithms, FOGA 2009*, pages 113–120. ACM, 2009.
- [Huang and Zhou, 2020] Zhengxin Huang and Yuren Zhou. Runtime analysis of somatic contiguous hypermutation operators in MOEA/D framework. In *Conference on Artificial Intelligence, AAI 2020*, pages 2359–2366. AAAI Press, 2020.
- [Huang *et al.*, 2019] Zhengxin Huang, Yuren Zhou, Zefeng Chen, and Xiaoyu He. Running time analysis of MOEA/D

- with crossover on discrete optimization problem. In *Conference on Artificial Intelligence, AAAI 2019*, pages 2296–2303. AAAI Press, 2019.
- [Huang *et al.*, 2021] Zhengxin Huang, Yuren Zhou, Chuan Luo, and Qingwei Lin. A runtime analysis of typical decomposition approaches in MOEA/D framework for many-objective optimization problems. In *International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 1682–1688, 2021.
- [Jansen, 2013] Thomas Jansen. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Springer, 2013.
- [Khare *et al.*, 2003] Vineet Khare, Xin Yao, and Kalyanmoy Deb. Performance scaling of multi-objective evolutionary algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Lothar Thiele, and Kalyanmoy Deb, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, page 376–390, Berlin, Heidelberg, 2003. Springer.
- [Kumar and Banerjee, 2006] Rajeev Kumar and Nilanjan Banerjee. Analysis of a multiobjective evolutionary algorithm on the 0-1 knapsack problem. *Theoretical Computer Science*, 358:104–120, 2006.
- [Laumanns *et al.*, 2002] Marco Laumanns, Lothar Thiele, Eckart Zitzler, Emo Welzl, and Kalyanmoy Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving from Nature, PPSN 2002*, pages 44–53. Springer, 2002.
- [Laumanns *et al.*, 2004a] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing*, 3:37–51, 2004.
- [Laumanns *et al.*, 2004b] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8:170–182, 2004.
- [Li *et al.*, 2016] Yuan-Long Li, Yu-Ren Zhou, Zhi-Hui Zhan, and Jun Zhang. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20:563–576, 2016.
- [Neumann and Witt, 2010] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer, 2010.
- [Neumann and Witt, 2022] Frank Neumann and Carsten Witt. Runtime analysis of single- and multi-objective evolutionary algorithms for chance constrained optimization problems with normally distributed random variables. In *International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 4800–4806. ijcai.org, 2022.
- [Neumann, 2007] Frank Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181:1620–1629, 2007.
- [Nguyen *et al.*, 2015] Anh Quang Nguyen, Andrew M. Sutton, and Frank Neumann. Population size matters: rigorous runtime results for maximizing the hypervolume indicator. *Theoretical Computer Science*, 561:24–36, 2015.
- [Purshouse and Fleming, 2007] Robin C. Purshouse and Peter J. Fleming. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11:770–784, 2007.
- [Thierens, 2003] Dirk Thierens. Convergence time analysis for the multi-objective counting ones problem. In *Evolutionary Multi-Criterion Optimization, EMO 2003*, pages 355–364. Springer, 2003.
- [Zheng and Doerr, 2022a] Weijie Zheng and Benjamin Doerr. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, pages 611–619. ACM, 2022.
- [Zheng and Doerr, 2022b] Weijie Zheng and Benjamin Doerr. Runtime analysis for the NSGA-II: proving, quantifying, and explaining the inefficiency for three or more objectives. *CoRR*, abs/2211.13084, 2022.
- [Zheng *et al.*, 2022] Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Conference on Artificial Intelligence, AAAI 2022*, pages 10408–10416. AAAI Press, 2022.
- [Zhou *et al.*, 2011] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagarathnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1:32–49, 2011.
- [Zhou *et al.*, 2019] Zhi-Hua Zhou, Yang Yu, and Chao Qian. *Evolutionary Learning: Advances in Theories and Algorithms*. Springer, 2019.