Near-Tight Runtime Guarantees for Many-Objective Evolutionary Algorithms

Simon Wietheger[1] and Benjamin Doerr[2]

[1] Algorithms and Complexity Group, Technische Universität Wien, Vienna, Austria
swietheger@ac.tuwien.ac.at
[2] Laboratoire d'Informatique (LIX), CNRS, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France doerr@lix.polytechnique.fr

Abstract. Despite significant progress in the field of mathematical runtime analysis of multi-objective evolutionary algorithms (MOEAs), the performance of MOEAs on discrete many-objective problems is little understood. In particular, the few existing bounds for the SEMO, global SEMO, and SMS-EMOA algorithms on classic benchmarks are all roughly quadratic in the size of the Pareto front.

In this work, we prove near-tight runtime guarantees for these three algorithms on the four most common benchmark problems OneMinMax, CountingOnesCountingZeros, LeadingOnesTrailingZeros, and OneJumpZeroJump, and this for arbitrary numbers of objectives. Our bounds depend only linearly on the Pareto front size, showing that these MOEAs on these benchmarks cope much better with many objectives than what previous works suggested. Our bounds are tight apart from small polynomial factors in the number of objectives and length of bitstrings. This is the first time that such tight bounds are proven for many-objective uses of these MOEAs. Further, we show that our bounds also transfer to the more practically motivated NSGA-III algorithm.

Keywords: evolutionary multi-objective optimization · runtime analysis · SEMO · NSGA · theory.

1 Introduction

Evolutionary algorithms such as the SMS-EMOA, NSGA-II, or MOEA/D are among the most successful approaches to tackle optimization problems with several conflicting objectives [6,29]. Despite the challenges stemming from the often complex population dynamics in multi-objective evolutionary algorithms (MOEAs), the analysis of MOEAs via theoretical means has made considerable progress in the last twenty years. Starting with simplistic algorithms such as the *simple evolutionary multi-objective optimizer (SEMO)* [18] and the global SEMO [13], this line of research has now reached the maturity to deal with state-of-the-art algorithms such as the MOEA/D, NSGA-II, NSGA-III, and SMS-EMOA [19,24,22,4].

However, this progress so far was mostly restricted to the analysis of MOEAs on bi-objective optimization problems. In particular, the few existing mathematical runtime analyses for the SEMO, global SEMO, and SMS-EMOA [17,3,27] for problems with general number m of objectives show runtime guarantees roughly quadratic in the size of the Pareto front; the recent work [25] proves that the NSGA-II cannot optimize the OneMinMax problem in time better than exponential in the size of the Pareto front when the number of objectives is three or more. These results could give the impression that MOEAs have significant difficulties in dealing with larger numbers of objectives, clearly beyond the mere increase of the size of the Pareto front with increasing numbers of objectives.

In this work, we revisit this twenty years old question and prove significantly stronger performance guarantees, which give a different impression. More specifically, we analyze the runtimes of the SEMO, global SEMO, SMS-EMOA, and NSGA-III on the OneMinMax (OMM), CountingOnesCountingZeros (COCZ), LeadingOnesTrailingZeros (LOTZ), and OneJumpZeroJump (OJZJ) problems for a general (even) number m of objectives. We prove runtime guarantees showing that these algorithms compute the Pareto fronts of these problems in an expected time (number of function evaluations) that is linear in the size of the largest set incomparable solutions (apart from small polynomial factors in the number m of objectives and the bitstring length n). For all benchmarks except LOTZ, this number coincides with the size of the Pareto front. Since naturally the size of the Pareto front is a lower bound for these runtimes, these guarantees are tight apart from the small factors. In the bi-objective case ($m = 2$) our results match the state-of-the-art bounds (apart from constant factors).

Together with the parallel and independent work on the NSGA-III by [21], these are the first that tight runtime guarantees for these MOEAs for general numbers of objectives, and they improve significantly over the previous results with their quadratic dependence on the Pareto front size.

We are optimistic that our methods can also be applied to other MOEAs and discuss some sufficient conditions for transferring the obtained bounds. In particular, as both the SMS-EMOA and NSGA-III fulfill these conditions, we believe that our arguments can also be applied to variants of the NSGA-II that do not suffer from the problems detected in [25] and to the $(\mu + 1)$ SIBEA [5].

2 Previous Work

The mathematical runtime analysis of randomized search heuristics is an active area of research for more than 30 years now, see [20,1,16,30,10]. Since around 20 years ago, also the runtime of multi-objective evolutionary algorithms (MOEAs) has been analyzed with mathematical means. Starting with simple toy algorithms like the *simple evolutionary multi-objective optimizer (SEMO)* [18] or the *global SEMO (GSEMO)* [13], the field has steadily progressed and is now able to also predict the runtimes of state-of-the-art algorithms such as the MOEA/D [19], NSGA-II [28], NSGA-III [22], and the SMS-EMOA [4].

Looking closer at the results obtained, we note that the vast majority of the runtime analyses of MOEAs consider only bi-objective problems. The sporadic results regarding more than two objectives appear less mature, and the runtime guarantees are far from the (mostly trivial) existing lower bounds.

One natural reason for the additional difficulty of runtime analyses for many-objective problems, visible from comparing the proofs of results for two and for more objectives, is the richer structure of the Pareto front. In bi-objective problems, the Pareto front has a one-dimensional structure. Hence the typical runtime analysis first estimates the time to find some solution on the Pareto front and then regards how the MOEA progresses along the Pareto front in the only two directions available. For problems with more objectives, the Pareto front is higher-dimensional, and hence there are many search trajectories from the first solution on the Pareto front to a particular solution.

The main runtime results for more than two objectives are the following. Already the journal version [17, Section V] of the first runtime analysis work on MOEAs [18] contains two many-objective runtime results, namely proofs that the SEMO computes the Pareto front of mCOCZ and mLOTZ (which are the m-objective analogues of the classic COCZ and LOTZ benchmarks) with problem size n and even number $m \geq 4$ of objectives in an expected number of $O(n^{m+1})$ function evaluations. While the result for mLOTZ naturally extends the $O(n^3)$ bound for the bi-objective LOTZ problem, the same is not true for mCOCZ, where the bi-objective runtime guarantee is $O(n^2 \log n)$. Considerably later, the bounds for mCOCZ were slightly improved in [3], namely to $O(n^m)$ for $m > 4$ and to $O(n^3 \log n)$ for the special case $m = 4$. As often in the runtime analysis of MOEAs, the complicated population dynamics prevented the proof of any interesting lower bounds, so only the trivial bound $\Omega(n^{m/2}\Theta(m)^{-m/2})$, which is the size of the Pareto front for both problems, is known.

Huang, Zhou, Luo and Lin [15] analyzed how the MOEA/D [23] optimizes the benchmarks mCOCZ and mLOTZ. As the MOEA/D decomposes the multi-objective problem into several single-objective subproblems and solves these in a co-evolutionary way, this framework is fundamentally different from the MOEAs regarded in this work, so we do not discuss these results in more detail.

Surprisingly, the NSGA-II [8] has enormous difficulties with discrete many-objective problems. Zheng and Doerr [25] showed that this prominent algorithm with any population size that is linear in the Pareto front size cannot optimize the OMM problem in polynomial time (in expectation) when the number of objectives is three or more. The proof of this result suggests that this is an intrinsic problem of the crowding distance, and that similar negative results hold for the other benchmark problems in this work.

The NSGA-III [7] might cope better with more objectives, however, this was proven only for the 3-objective OMM problem, where a runtime guarantee of $O(Nn \log n)$ function evaluations was shown in [22] (when the population size N is at least the size $(\frac{n}{2} + 1)^2$ of the Pareto front). In a very recent parallel and independent work, [21] crucially generalized the property of the NSGA-III used in the analyses of the 3-objective OMM problem to arbitrary benchmarks and numbers of objectives. They used this property to prove the runtime of the NSGA-III (for a proper choice of reference points and population size μ) to be $O(\mu n \log n)$ on mOMM and mCOCZ, and $O(\mu n^2)$ on mLOTZ, for arbitrary but constant and even m. Our results on the NSGA-III heavily rely on their proven property and extend their results to arbitrary, non-constant numbers of objectives.

Very recently [27], the runtime of the GSEMO and the SMS-EMOA [2] on the OJZJ_k problem for arbitrary (even) numbers m of objective was shown to be $O(M^2 n^k)$ and $O(\mu M n^k)$, respectively, where $M = (2\frac{n}{m} - 2k + 3)^{m/2}$ is the size of the Pareto front of this problem and $\mu \geq M$ denotes the size of the population of the SMS-EMOA.

We note that all bounds discussed in this section except for the ones of the NSGA-III are quadratic in the Pareto front size, times some small polynomial in m and n. As our results will show, this quadratic dependence is not necessary and merely stems from the difficulty to analyze many-objective MOEAs.

3 Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$ and for $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. Whenever we are speaking of how close one bitstring is to another, we are referring to their Hamming distance. Many of our proofs rely on two well-known bounds in probability theory, which we give here for completeness and reference in our proofs by name. First, a *union bound* states for some finite set of events $\{E_1, \ldots, E_\ell\}$ that the probability of any of the events happening is at most the sum over the individual probabilities, that is, $\Pr[\bigcup_{i\in[\ell]} E_i] \leq \sum_{i\in[\ell]} \Pr[E_i]$. Second, we employ a variant of the *Chernoff bound*. Let $X_1, \ldots, X_\ell$ be independent random variables with values in $\{0, 1\}$ and let $X = \sum_{i\in[\ell]} X_i$. Then for any $0 < \delta < 1$ we have

$$\Pr[X \leq (1-\delta)E[X]] \leq \exp\bigl(-\tfrac{1}{2}\delta^2 E[X]\bigr).$$

In particular, $\Pr[X \leq \frac{1}{2}E[X]] \leq \exp(-\frac{1}{8}E[X])$.

3.1 Multi-Objective Optimization

Let $m \in \mathbb{N}$. An m-objective function f is a tuple $(f_1, \ldots, f_m)$ such that $f_i \colon \Omega \to \mathbb{R}$ for some search space Ω, for all $i \in [m]$. We define the objective value of $x \in \Omega$ to be $f(x) = (f_1(x), \ldots, f_m(x))$. There is usually no solution that maximizes all m objective functions at the same time. For $x, y \in \Omega$ we write $x \succeq y$ if and only if $f_i(x) \geq f_i(y)$ for all $i \in [m]$ and say that x *dominates* y. If additionally $f_j(x) > f_j(y)$ for some $j \in [m]$, we say that x *strictly dominates* y and write $x \succ y$. A solution $x \in \Omega$ is Pareto-optimal if it is not strictly dominated by any other solution. The *Pareto front* is the set of objective values of Pareto-optimal solutions. Given an algorithm and an objective function, we are interested in the number of function evaluations until the population covers the Pareto front, that is, until for all values p on the Pareto front there is a solution x in the population such that $f(x) = p$. We note that all algorithms analyzed in this work in each iteration create one new individual and thus only require one function evaluation per iteration. Thus we simply analyze the number of iterations until the Pareto front is covered and remark here that the initial population also has to be evaluated (1 evaluation for the SEMO and GSEMO, μ iterations for the SMS-EMOA with population size μ).

All the objective functions we consider are defined on the search space of bitstrings of length n for $n \in \mathbb{N}$. For $m = 2m'$ objectives, they are obtained by partitioning individuals into m' blocks of size $b = \frac{n}{m'}$ and applying a bi-objective function to each block. For some individual $x \in \{0,1\}^n$ and $i \in [m']$, we define x^i to be the ith block of x, that is the substring from $x_{b(i-1)+1}$ to x_{bi}. We define $|x^i|_1 = \sum_{j=(i-1)b+1}^{ib} x_j$ (and $|x^i|_0 = b - |x^i|_1$) to denote the number of 1-bits (and 0-bits) in the ith block.

3.2 Benchmarks

We evaluate the algorithms on four established multi-objective benchmarks.

mOneMinMax (mOMM). The objective function mOMM translates the well-established OneMax benchmark in a setting with $m = 2m'$ objectives for some $m' \in \mathbb{N}$. Intuitively, the bitstring is divided into m' equally sized blocks that each contribute two objectives, the number of 1-bits and the number of 0-bits in that block. The bi-objective case of $m' = 1$ was proposed by [14] and later generalized to arbitrary m' [25]. Let $b, m' \in \mathbb{N}$ and $n = bm'$. For all $x \in \{0,1\}^n$, define $m\text{OMM}(x) = (f_1(x), \ldots, f_m(x))$ where for all $i \in [m']$

$$f_{2i}(x) = |x^i|_1 \quad \text{and} \quad f_{2i-1}(x) = b - |x^i|_1.$$

The benchmark mOMM is special in the sense that each of the $S_m^{\text{OMM}} := \left(\frac{n}{m'} + 1\right)^{m'}$ possible objective values lies on the Pareto front.

mCountingOnesCountingZeros (mCOCZ). The COCZ benchmark and its multi-objective variant mCOCZ [17] are closely related to OMM and mOMM. However, the objectives cooperate on the first half of the bitstring and only the second half is evaluated just like for mOMM. Formally, let $m' \in \mathbb{N}$, $b \in 4\mathbb{N}$, $n = 2bm'$, and $m = 2m'$. Then for all $x \in \{0,1\}^n$, define $\text{mCOCZ}(x) = (f_1(x), \ldots, f_m(x))$ where for all $i \in [m']$

$$f_{2i}(x) = \sum_{j=1}^{bm'} x_j + \sum_{j=ib+1}^{ib} x_{bm'+j} \quad \text{and} \quad f_{2i-1}(x) = \sum_{j=1}^{bm'} x_j + \sum_{j=ib+1}^{ib} 1 - x_{bm'+j}.$$

Observe that $S_m^{\text{COCZ}} = (\frac{n}{2m'} + 1)^{m'}$ is the size of Pareto front for the mCOCZ problem and also the maximum size of any set of pairwise non-dominating individuals.

mLeadingOnesTrailingZeros (mLOTZ). The objective function mLOTZ is the many objective variant of the bi-objective LOTZ benchmark [17]. Intuitively, it has two objectives per block, one being the number 1-bits up to the first 0-bit and the other being the number of 0-bits behind the last 1-bit. Formally, let $b, m' \in \mathbb{N}$, $n = bm'$, and $m = 2m'$. Then for all $x \in \{0,1\}^n$, define $m\text{LOTZ}(x) = (f_1(x), \ldots, f_m(x))$ where for all $i \in [m']$

$$f_{2i}(x) = \sum_{j=(i-1)b+1}^{ib} \prod_{j'=1}^{j} x_{j'} \quad \text{and} \quad f_{2i-1}(x) = \sum_{j=(i-1)b+1}^{ib} \prod_{j'=j}^{ib} 1 - x_{j'}.$$

Observe that $\bar{S}_m^{\text{LOTZ}} = S_m^{\text{OMM}} = (\frac{n}{m'}+1)^{m'}$ is the size of Pareto front for the mLOTZ problem. However, the size S_m^{LOTZ} of the largest incomparable set is asymptotically tightly bounded by $S_m^{\text{LOTZ}} \le (\frac{n}{m'}+1)^{2m'-1}$ [21] and thus almost quadratic in $\bar{S}_m^{\text{LOTZ}}$.

mOneJumpZeroJump$_k$ (mOJZJ$_k$). The objective function $m\text{OJZJ}_k$ is the recently introduced many objective variant [27] of the bi-objective OJZJ benchmark [11]. Intuitively, it has two objectives per block, one for the number of 1-bits and one for the number of 0-bits. However, it creates a fitness valley with decreasing objective value if the number of 0-bits or 1-bits is at least 1 and at most k, where k is a parameter of the benchmark. Formally, let $b, m' \in \mathbb{N}$, $n = bm'$ and $m = 2m'$. Then for all $x \in \{0,1\}^n$, define $m\text{OJZJ}_k(x) = (f_1(x), \ldots, f_m(x))$ where for all $i \in [m']$

$$f_{2i}(x) = \text{Jump}_k(x^i) \quad \text{and} \quad f_{2i-1}(x) = \text{ZeroJump}_k(x^i) \quad \text{with}$$
$$\text{Jump}_k(x) = \begin{cases} |x|_1 + k, & \text{if } |x|_1 \le b-k \text{ or } |x|_1 = b; \\ b - |x|_1, & \text{else;} \end{cases}$$
$$\text{ZeroJump}_k(x) = \begin{cases} |x|_0 + k, & \text{if } |x|_0 \le b-k \text{ or } |x|_0 = b; \\ b - |x|_0, & \text{else.} \end{cases}$$

We assume $2 \le k \le \frac{n}{2m'}$. The 2-objective OJZJ benchmark with jumps of size k has a Pareto front of size $n - 2k + 3$ [11]. The same term is the maximum size of any set of pairwise non-dominating individuals. Thus, the Pareto front of $m\text{OJZJ}_k$, which corresponds to OJZJ in m' individual blocks of size $\frac{n}{m'}$, is of size $S_{m,k}^{\text{OJZJ}} = \left(\frac{n}{m'} - 2k + 3\right)^{m'}$. The same term bounds the size of any set of pairwise non-dominating individuals for $m\text{OJZJ}_k$.

3.3 SEMO and GSEMO

The SEMO and GSEMO start the first generation with a single, random individual in the population. In each iteration, they uniformly at random choose an individual from the current population and mutate it to a new solution x'. Now they remove all solutions from the population that are dominated by x' and add x' to the population if and only if it is not strictly dominated by a solution in the population, see Algorithm 1. This way, the population stores exactly one individual for each encountered objective value that is not strictly dominated by any other encountered objective value. Observe that by evaluating

Algorithm 1: (Global) SEMO

Input : objective function $f = (f_1, \ldots, f_m)$,
length of bitstrings n with $\frac{n}{m} \in \mathbb{N}$

Generate $x_0 \in \{0,1\}^n$ uniformly at random and let $P_0 \coloneqq \{x_0\}$
for $t = 1, 2, \ldots$ **do**
  Select x from P_{t-1} uniformly at random and let $x' \coloneqq (x)$
  $P_t \coloneqq \{p \in P_{t-1} \mid x' \not\succeq p\}$
  if *there is no* $p \in P_t$ *such that* $p \succ x'$ **then**
    $P_t \coloneqq P_t \cup \{x'\}$

the objective function only once after a new individual is created and storing the value for future comparisons, the number of evaluations is exactly the number of generations (plus 1 for the initial individual). The only difference between the SEMO and the GSEMO is the mutation step. While the SEMO uniformly at random selects one bit of the parent and flips that bit to create an offspring, the GSEMO independently flips each bit of the parent with some probability p. Here we assume the conventional mutation rate $p = \frac{1}{n}$.

4 Mathematical Analyses of GSEMO

We start our contribution by giving upper bounds on the optimization time of the GSEMO on the four benchmarks. Afterward, in Section 5, these results are transferred to other MOEAs as well. To start our analysis, we observe that for these benchmarks the size of any set of incomparable solutions is at most $S_m^{\mathrm{OMM}}, S_m^{\mathrm{COCZ}}, S_m^{\mathrm{LOTZ}}$, or $S_{m,k}^{\mathrm{OJZJ}}$, respectively. Consequently, these are upper bounds on the population size in any iteration of the GSEMO.

4.1 mOneMinMax

If a bitstring has a_i bits of value 1 in the ith block for all $i \in [m']$, we call it an $(a_1, a_2, \ldots, a_{m'})$-bitstring. We abbreviate the notation of vectors of the Pareto front from $(n-a_1, a_1, n-a_2, a_2, \ldots, n-a_{m'}, a_{m'})$ to $(a_1, a_2, \ldots, a_{m'})$. Define the set C_m to contain all bitstrings for which each of the m' blocks consists either only of 1-bits or only of 0-bits ("corners"). Observe that $|C_m| = 2^{m'}$ and that each bitstring in C_m is the unique individual of the respective objective value. For the upper bound, we separately bound the time until all individuals in C_m are in the population, see Lemma 1, and then analyze how from this point on all other individuals are generated, see Lemma 2. The following theorem combines these results to bound the total optimization time.

Theorem 1. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mOMM. Let T denote the number of iterations until the population matches the complete Pareto front and let t be*

$$\left(\frac{\ln(2)m'+2}{\ln(n)} + 16\frac{{m'}^2+2m'}{n} + 2\right) eS_m^{\mathrm{OMM}} n\ln(n) + 1.$$

Then $T \le t$ with high probability and $E[T] \le (1-\frac{1}{n})^{-2}t$.

Proof. Let t_1 and t_2 be the optimization times in Lemmas 1 and 2. We have $t \ge t_1 + \lceil t_2 \rceil$ by observing $\ln(n) \ge \ln(m')$ and $\ln(n)+1 \ge \ln(n+1) \ge \ln\left(\frac{n}{m'}+1\right)$. Thus, $T \le t$ with a high probability of at least $(1-\frac{1}{n})^2$. We employ a simple restart argument to obtain an upper bound on the expected value of T. Each sequence of t iterations fails to cover the Pareto front with probability at most $1-(1-\frac{1}{n})^2$. Due to the convergence of the geometric series we have

$$E[T] \le \sum_{i=0}^{\infty} \left(1 - \left(1-\frac{1}{n}\right)^2\right)^i t = \left(1-\frac{1}{n}\right)^{-2} t. \qquad \square$$

Lemma 1. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mOMM and let T denote the number of iterations until the population contains C_m. Then*

$$T \le \left(\ln(2)\frac{m'}{\ln(n)} + 2\right) eS_m^{\mathrm{OMM}} n\ln(n)$$

with probability at least $1-\frac{1}{n}$.

Proof. We first prove a tail bound for the time T_C until the population contains the corner bitstring 1^n. By the symmetry of mOMM, this bound applies to all other elements in C_m as well. Hence, applying a union bound over the tail bounds for the individual elements in C_m yields a bound on the time until all elements are covered.

Let x be a member of the population with maximum number of 1-bits. Let $i = n - |x|_1$ be the number of 0-bits of x. Then the probability of sampling an individual that has $i-1$ many 0-bits in the next iteration is at least $\frac{1}{S_m^{\mathrm{OMM}}} \cdot i \cdot \frac{1}{n} \cdot (1-\frac{1}{n})^{n-1} \ge \frac{i}{enS_m^{\mathrm{OMM}}} = p_i$, by choosing x as parent, flipping any one of its 0-bits, and not flipping any other bit. Hence, $E[T_C] \le \sum_{i=1}^{n} \frac{1}{p_i}$.

For $1 \le i \le n$, let X_i be independent geometric random variables, each with success probability p_i, and let $X = \sum_{i=1}^{n} X_i$. Then X stochastically dominates T_C, and thus a tail bound for X also applies to T_C. By Theorem 1.10.35 in [9], a tail bound for sums of geometric random variables with harmonic success probabilities, for all $\delta \ge 0$ we have

$$\Pr[T_C \ge (1+\delta)eS_m^{\mathrm{OMM}} n\ln(n)] \le \Pr[X \ge (1+\delta)eS_m^{\mathrm{OMM}} n\ln(n)] \le n^{-\delta}.$$

By the symmetry of the problem and operators, the bound also holds for all other elements in C_m. Let $\delta = m'\log_n(2) + 1$. A union bound helps to give a tail bound on the time T until the population contains all elements in C_m. The probability that $T \leq (m' \log_n(2) + 2)eS_m^{\mathrm{OMM}} n \ln(n)$ is at least

$$\begin{aligned} 1 - |C_m| \cdot \Pr[T_C \geq (m' \log_n(2) + 2)eS_m^{\mathrm{OMM}} n \ln(n)] \\ \geq 1 - 2^{m'} n^{-m' \log_n(2) - 1} = 1 - \tfrac{1}{n}. \end{aligned}$$

We note that this applies for arbitrary starting configurations, as all that we assumed about the initial population was that it is non-empty. □

Lemma 2. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mOMM starting with a population that contains at least all individuals in C_m. Let T denote the number of iterations until the population matches the complete Pareto front and let*

$$t = \max\left\{1, 8\frac{m'(m' \ln\left(\frac{n}{m'} + 1\right) + \ln(m') + \ln(n))}{n}\right\} \cdot \ 2eS_m^{\mathrm{OMM}} n.$$

Then $T \leq \lceil t \rceil$ with probability at least $1 - \frac{1}{n}$.

Proof. Consider any objective value $v = (a_1, a_2, \ldots, a_{m'})$ on the Pareto front. Let $c_0 \in C_m$ be any closest corner to an $(a_1, a_2, \ldots, a_{m'})$-bitstring. We bound the time until an $(a_1, a_2, \ldots, a_{m'})$-bitstring is generated by bounding the time until a marked individual c becomes an $(a_1, a_2, \ldots, a_{m'})$-bitstring. Let initially $c = c_0$. Whenever the individual c creates an offspring c' by flipping exactly one bit and c' is closer to any $(a_1, a_2, \ldots, a_{m'})$-bitstring than c, we update c to be c'. We also replace c by c' whenever an individual c' with the same objective value replaces c in the population. The time until the population contains an $(a_1, a_2, \ldots, a_{m'})$-bitstring is at most the time until c is an $(a_1, a_2, \ldots, a_{m'})$-bitstring. We first bound the probability that after t iterations there are exactly a_i bits of value 1 in the ith block of c, for any fixed $1 \leq i \leq m'$. By symmetry, suppose without loss of generality that $a_i \leq \frac{n}{2m'}$ and the ith block of c_0 to be $0^{n/m'}$. All c we will encounter have between 0 and a_i bits with value 1. The probability of creating an offspring of c in the next iteration that flips one of the at least $\frac{n}{2m'}$ bits of value 0 in the ith block and no other bit is at least

$$\frac{1}{S_m^{\mathrm{OMM}}} \cdot \frac{n}{2m'} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{2em'S_m^{\mathrm{OMM}}} =: p.$$

After $a_i \leq \frac{n}{2m'}$ such iterations, c contains the correct number of 1-bits in the ith block. Thus, for the time T_i until the ith block of c is correct we have $E[T_i] \leq \frac{a_i}{p}$. For $j \in [\lceil t \rceil]$, let X_j be independent random variables, each with a Bernoulli distribution with success probability p. Let $X = \sum_{j=1}^{\lceil t \rceil} X_j$. Then, due to stochastic domination, $\Pr[T_i > \lceil t \rceil] \leq \Pr[X < a_i] \leq \Pr[X \leq a_i]$. By observing $E[X] = p\lceil t \rceil \geq \frac{n}{m'}$ we have

$$\Pr[X \leq a_i] \leq \Pr\left[X \leq \frac{n}{2m'}\right] \leq \Pr\left[X \leq \frac{1}{2}E[X]\right].$$

Applying a Chernoff bound yields

$$\Pr[T_i > \lceil t \rceil] \le \exp\Big(-\frac{1}{8}E[X]\Big) \quad \le \exp\Big(-\ln(m') - m'\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big).$$

Using a union bound over all blocks gives that any fixed objective value on the Pareto front is not sampled in t iterations with probability at most

$$\exp\Big(-\ln(m') - m'\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big)m' = \exp\Big(-m'\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big).$$

Let E denote the event that after $\lceil t \rceil$ iterations there is still an objective value $(a_1, a_2, \dots, a_{m'})$ such that the respective individual c does not contain the correct number 1-bits in any block. By applying a union bound we have

$$\Pr[E] \le \exp\Big(-m'\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big)S_m^{\mathrm{OMM}} = \exp(-\ln(n)) = \frac{1}{n}$$

by observing $S_m^{\mathrm{OMM}} = \exp(m'\ln(\frac{n}{m'}+1))$. □

4.2 *m*CountingOnesCountingZeros

Due to the similarity between mOMM and mCOCZ, our previous proofs can be adapted to also work for mCOCZ. We first show that with probability at least $1-\frac{1}{n}$ the population after $2eS_m^{\mathrm{COCZ}}n\ln(n)$ iterations contains an individual such that the cooperative, first half is maximized. From this point on, we employ the same ideas as for Theorem 1 by only considering individuals with maximum cooperative part. More details are placed in the supplementary material.

Theorem 2. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mCOCZ. Let T denote the number of iterations until the population matches the complete Pareto front and let t be*

$$\left(\frac{\ln(2)m'+2}{\ln(n)} + 16\frac{m'^2+2m'}{n} + 4\right)eS_m^{\mathrm{COCZ}}n\ln(n) + 1.$$

Then $T \le t$ with high probability and $E[T] \le (1-\frac{1}{n})^{-3}t$.

Comparing the bounds for mOMM and mCOCZ on bitstrings of the same length, we note that the bound for mCOCZ is smaller than the one on mOMM as $S_m^{\mathrm{COCZ}} \approx 2^{-m/2}S_m^{\mathrm{OMM}}$.

4.3 *m*LeadingOnesTrailingZeros

In contrast to mOMM and mCOCZ, for mLOTZ the probability to transform a solution of a certain objective value into one where one of the values changed by 1 is less depending on the objective value itself. In fact, this probability is at least $\frac{1}{eS_m^{\mathrm{LOTZ}}n}$ for all objective values. Employing a similar strategy as in the proof of

Lemma 2, but starting from the initial individual gives the following Theorem 3. This strategy is not compromised by the fact that some intermediate solutions might vanish from the population when strictly dominated. In that case, the dominating solutions actually are at least as close to a desired solution as the dominated one. Details are given in the supplementary material.

Theorem 3. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mLOTZ. Let T denote the number of iterations until the population matches the complete Pareto front and let*

$$t = \max\left\{1, \frac{4{m'}^2 \ln\left(\frac{n}{m'}+1\right) + 8m' \ln(n)}{n}\right\} 2eS_m^{\mathrm{LOTZ}} \frac{n^2}{m'}.$$

Then $T \le \lceil t \rceil$ with high probability and $E[T] \le (1-\frac{1}{n})^{-1}\lceil t \rceil$.

While, unlike our other bounds for the GSEMO, this result does not improve over the existing $O(n^{2m'-1})$ bound [17], we note that our bound applies to all choices for the numbers of objectives while the previous one assumed it to be arbitrary but constant.

4.4 mOneJumpZeroJump$_k$

We define the set $K_{m,k} = \{(a_1, \ldots, a_{m'}) \mid a_i \in \{k, \frac{n}{m'} - k\}$ for all $i \in [m']\}$ to contain all objective values of individuals that in each block have either exactly k bits of value 0 or exactly k bits of value 1. Further, we define the set $C_{m,k} = \{(a_1, \ldots, a_{m'}) \mid a_i \in \{0, k, \frac{n}{m'} - k, \frac{n}{m'}\}$ for all $i \in [m']\}$ to contain all objective values of individuals that in each block have either only 1-bits, only 0-bits, exactly k bits of value 0, or exactly k bits of value 1.

For the $m\mathrm{OJZJ}_k$ benchmark, we separately consider three phases: the time until $K_{m,k}$ is covered, the time until C_m is covered, and the time until the remaining Pareto front is covered. While the first and third phase roughly compare to Lemmas 1 and 2, the second phase dominates the running time. There, progress is made by jumping over the valley of low fitness, which for any fixed block takes time in $eS_{m,k}^{\mathrm{OJZJ}} n^k$ as the only way is to simultaneously flip the k bits. The bounds we obtain for the $m\mathrm{OJZJ}_k$ benchmark are only applicable if $m' \ge 2$. For the case $m' = 1$, we thus refer to previous results in the literature, which show that the expected number of iterations until the GSEMO solves $2\mathrm{OJZJ}_k$ is at most $eS_{2,k}^{\mathrm{OJZJ}}(\frac{3}{2}n^k + 2n\ln(\lceil \frac{n}{2} \rceil) + 3)$ [26]. Details on the proofs are placed in the supplementary material.

Theorem 4. *Let $m' \in \mathbb{N}_{\ge 2}$ and $m = 2m'$. Consider the GSEMO optimizing $m\mathrm{OJZJ}_k$. Let T denote the number of iterations until the population matches the complete Pareto front and let*

$$t = \left(\frac{\ln(4)m' + \ln(n)}{\ln(m')} + 1\right) 3e\ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k.$$

Then $T \le t$ with high probability . Further,

$$E[T] \le \left(1 - \frac{1}{m'}\right)^{-1} \left(\frac{\ln(4)m'}{\ln(m')} + 2\right) 3e\ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k.$$

5 Runtime Results for the SEMO, SMS-EMOA, and NSGA-III

We started our mathematical runtime analysis of many-objective MOEAs with an analysis of the GSEMO, the most prominent MOEA in theoretical works. We are very optimistic that our general methods apply to many other MOEAs as well. We discuss sufficient conditions to extend our results to other MOEAs and demonstrate this on the SEMO, an algorithm prominent in theory, as well as the SMS-EMOA and NSGA-III, algorithms often used in practical applications.

Note that all our proofs for upper bounds for the GSEMO only rely on three properties of the algorithm:

1. Once a solution x is generated, all future populations contain a solution y such that $y \succeq x$.
2. The chance to select an individual from the population for mutation is at least $\frac{1}{S}$, where S is typically an upper bound on the size of sets of incomparable solutions. The bounds on the runtime will include a factor of S.
3. The employed mutation operator is bitwise mutation with a chance of $\frac{1}{n}$ to flip a bit. Other mutation operators are possible but might affect the bounds and ability to solve some benchmarks at all, as we discuss for the SEMO.

While the latter two properties enable progress to be made, the first one is responsible for not loosing already made progress, that is, not loosing any desired solutions and intermediate solutions on the path to a desired solution.

5.1 SEMO

The only difference between the SEMO and GSEMO is the mutation step. While the GSEMO flips each bit independently with probability $\frac{1}{n}$, the SEMO uniformly at random selects any bit and flips it. This makes it impossible for the SEMO to solve $m\mathrm{OJZJ}_k$, as argued for the bi-objective case [11]. By the same reasoning, the SEMO cannot solve $m\mathrm{OJZJ}_k$ for any m. Nevertheless, the SEMO is able to solve all other benchmarks discussed in this work. In fact, we note that in all proofs, except for the ones concerning $m\mathrm{OJZJ}_k$, our arguments exclusively build on improvements by flipping exactly one bit. Thus, not only do all proven results immediately transfer, but actually improve by a factor e, that previously accounted for the probability of no other bit than the desired one flipping.

Theorem 5. *Consider the SMS-EMOA optimizing mOMM, mCOCZ, or mLOTZ. Then the respective bounds for the GSEMO as given by Theorems 1, 2, and 3, divided by e, also hold for the SEMO.*

5.2 SMS-EMOA

The SMS-EMOA works with a population of fixed size μ. Similar to the (G)SEMO, it produces one offspring solution in each generation. We consider the SMS-EMOA using bit-wise mutation just like employed by the GSEMO. While the (G)SEMO

Algorithm 2: SMS-EMOA with population size μ

Input : objective function $f = (f_1, \dots, f_m)$,
length of bitstrings n with $\frac{n}{m} \in \mathbb{N}$

Let $P_0 \coloneqq \{x_1, \dots, x_\mu\}$, where each $x_i \in \{0,1\}^n$ is generated independently and uniformly at random

for $t = 1, 2, \dots$ **do**

  Select x from P_{t-1} uniformly at random and let $x' \coloneqq (x)$

  Divide $R_t \coloneqq P_t \cup \{x'\}$ into fronts $F_1, \dots, F_{i^*}$, by fast-non-dominated-sort() [8].

  Pick $z' \in \arg\min_{z \in F_{i^*}} \Delta_r(z, F_{i^*})$ uniformly at random and let $P_t \coloneqq R_t \setminus z'$

only relies on the (strict) domination relation to select the surviving solutions for the next generation, the SMS-EMOA sorts solutions into fronts $F_1, \dots, F_{i^*}$, where each front contains all pairwise not strictly dominating solutions that are not yet represented in an earlier front. Then one element of F_{i^*} is removed to reduce the population size back to μ, namely one with smallest hypervolume contribution. For some set S and reference point r, the hypervolume of S is $\mathrm{HV}_r(S) = \mathcal{L}\big(\bigcup_{u \in S}\{h \in \mathbb{R}^m \mid r \leq h \leq f(u)\}\big)$, where $\mathcal{L}$ is the Lebesque measure. The hypervolume contribution of an individual of an individual $x \in F$ is $\Delta_r(x, F) = \mathrm{HV}_r(F) - \mathrm{HV}_r(F \setminus \{x\})$. Since we only regard maximization problems with non-negative objective values, as common, we use the reference point $r = (-1, \dots, -1)$. Algorithm 2 states the SMS-EMOA in pseudocode.

Our results for the GSEMO transfer to the SMS-EMOA if μ is at least as large as the Pareto front (and thereby the largest set of incomparable solutions, at least for the discussed benchmarks). The central observation is that then, once the population of the SMS-EMOA contains an individual x, all future generations will contain at least one individual x' such that $f(x') \succeq f(x)$ [27, Lemma 4], yielding the first of the three above mentioned properties. For the second and third one, we note that the chance to select an individual for mutation is simply $\frac{1}{\mu}$ and bitwise mutation is applied. This yields the results as for the GSEMO though μ replaces the factor S in all runtime bounds.

Theorem 6. *Consider the SMS-EMOA optimizing one of mOMM, mCOCZ, mLOTZ, or mOJZJ$_k$, let S denote the size of the largest incomparable set of solutions, and let the population size be $\mu \geq S$. Then the respective bounds for the GSEMO given by Theorems 1-4, multiplied by $\frac{\mu}{S}$, also hold for the SMS-EMOA.*

5.3 NSGA-II and NSGA-III

We only briefly summarize the definitions of the NSGA-II and NSGA-III and refer to [22] for a more detailed description. Both algorithms work with a population of fixed size μ. Each iteration, every individual produces an offspring, here we assume by bitwise mutation. The combined population of size 2μ is then sorted into ranks, where each rank contains all solutions that are only dominated by solutions with lower rank. The new generation of μ individuals is selected by choosing solutions with lower rank first. As a tiebreaker, the NSGA-II employs crowding distance while the NSGA-III uses reference points in the solution space.

As shown in [25] and discussed in the introduction, the NSGA-II struggles for $m > 2$ objectives. Our proofs transferred to the SMS-EMOA as with a large enough population, no non-dominated objective value is ever lost. This however does not hold true for the NSGA-II, as proven for mOMM in [25].

In contrast, the NSGA-III preserves non-dominated solutions for reasonable choices of the population size and reference points [21]. Each individual is mutated with probability $\frac{1}{S} = 1$ in each generation, eliminating the factor S from the running time. However, every generation requires μ fitness evaluations.

Theorem 7. *Consider the NSGA-III optimizing mOMM, mCOCZ,mLOTZ, or mOJZJ$_k$, let S denote the size of the largest incomparable set of solutions and $f_{\max}$ denote the largest fitness value over all solutions and objectives. Assume $\mu \geq S$ and the NSGA-III to employ a set of reference points $\mathcal{R}_p$ as defined in [21] with $p \geq 2m^{3/2} f_{\max}$. Then the respective bounds for the GSEMO as given by Theorems 1-4 multiplied by $\frac{\mu}{S}$, also hold for the NSGA-III.*

6 Conclusion

In this work, we revisited the problem of proving performance guarantees for MOEAs dealing with more than two objectives. In the first major progress after the initial work on this question twenty years ago [17], we proved runtime guarantees for three classic algorithms on four classic benchmarks that are all (except for mLOTZ) linear in the size of the Pareto front (apart from small factors polynomial in n and m), in contrast to the previous bounds, which were all quadratic in the size of the Pareto front. Our results thus suggest that MOEAs cope much better with many-objective problems than known. In fact, our work hints at that the performance loss observed with increasing number of objectives in experiments is caused by the increasing size of the Pareto front rather than by particular algorithmic difficulties of many-objective optimization.

In the supplementary material, we provide a short empirical comparison of the SEMO, GSEMO, SMS-EMOA, and NSGA-II on the discussed benchmarks. We find that, fitting to our intuition and our runtime guarantees, the SEMO is faster than the GSEMO except on mOJZJ$_k$ (which it cannot optimize). The optimization times of the GSEMO and SMS-EMOA seem roughly comparable suggesting that the fixed population size of the SMS-EMOA does neither hinder nor improve much over the optimization process as transferred from the GSEMO.

Acknowledgments. Simon Wietheger acknowledges the support by the Austrian Science Fund (FWF, START Project Y1329). Benjamin Doerr was supported by a public grant as part of the Investissements d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Auger, A., Doerr, B. (eds.): Theory of Randomized Search Heuristics. World Scientific Publishing (2011)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research **181**, 1653–1669 (2007)
3. Bian, C., Qian, C., Tang, K.: A general approach to running time analysis of multi-objective evolutionary algorithms. In: International Joint Conference on Artificial Intelligence, IJCAI 2018. pp. 1405–1411. IJCAI (2018)
4. Bian, C., Zhou, Y., Li, M., Qian, C.: Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In: International Joint Conference on Artificial Intelligence, IJCAI 2023. pp. 5513–5521. ijcai.org (2023)
5. Brockhoff, D., Friedrich, T., Neumann, F.: Analyzing hypervolume indicator based algorithms. In: Parallel Problem Solving from Nature, PPSN 2008. pp. 651–660. Springer (2008)
6. Coello, C.A.C., Lamont, G.B., van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, 2nd edn. (2007)
7. Deb, K., Jain, H.: 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)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**, 182–197 (2002)
9. Doerr, B.: Probabilistic tools for the analysis of randomized optimization heuristics. In: Doerr, B., Neumann, F. (eds.) Theory of Evolutionary Computation: Recent Developments in Discrete Optimization, pp. 1–87. Springer (2020), also available at `https://arxiv.org/abs/1801.06733`
10. Doerr, B., Neumann, F. (eds.): Theory of Evolutionary Computation—Recent Developments in Discrete Optimization. Springer (2020)
11. Doerr, B., Zheng, W.: Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In: Conference on Artificial Intelligence, AAAI 2021. pp. 12293–12301. AAAI Press (2021)
12. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research **13**, 2171–2175 (jul 2012)
13. Giel, O.: Expected runtimes of a simple multi-objective evolutionary algorithm. In: Congress on Evolutionary Computation, CEC 2003. pp. 1918–1925. IEEE (2003)
14. Giel, O., Lehre, P.K.: On the effect of populations in evolutionary multi-objective optimisation. Evolutionary Computation **18**, 335–356 (2010)
15. Huang, Z., Zhou, Y., Luo, C., Lin, Q.: 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. pp. 1682–1688 (2021)
16. Jansen, T.: Analyzing Evolutionary Algorithms – The Computer Science Perspective. Springer (2013)
17. Laumanns, M., Thiele, L., Zitzler, E.: Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. IEEE Transactions on Evolutionary Computation **8**, 170–182 (2004)

18. Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., Deb, K.: Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In: Parallel Problem Solving from Nature, PPSN 2002. pp. 44–53. Springer (2002)
19. Li, Y.L., Zhou, Y.R., Zhan, Z.H., Zhang, J.: A primary theoretical study on decomposition-based multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation **20**, 563–576 (2016)
20. Neumann, F., Witt, C.: Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity. Springer (2010)
21. Opris, A., Dang, D.C., Neumann, F., Sudholt, D.: Runtime analyses of NSGA-III on many-objective problems. In: Genetic and Evolutionary Computation Conference, GECCO 2024. ACM (2024), to appear
22. Wietheger, S., Doerr, B.: A mathematical runtime analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). In: International Joint Conference on Artificial Intelligence, IJCAI 2023. pp. 5657–5665. ijcai.org (2023)
23. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**, 712–731 (2007)
24. Zheng, W., Doerr, B.: Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II). Artificial Intelligence **325**, 104016 (2023)
25. Zheng, W., Doerr, B.: Runtime analysis for the NSGA-II: proving, quantifying, and explaining the inefficiency for many objectives. IEEE Transactions on Evolutionary Computation (2023), in press, `https://doi.org/10.1109/TEVC.2023.3320278`
26. Zheng, W., Doerr, B.: Theoretical analyses of multiobjective evolutionary algorithms on multimodal objectives. Evolutionary Computation (2023). `https://doi.org/10.1162/evco_a_00328`, `https://doi.org/10.1162/evco_a_00328`, in Press, `https://doi.org/10.1162/evco_a_00328`
27. Zheng, W., Doerr, B.: Runtime analysis of the SMS-EMOA for many-objective optimization. In: Conference on Artificial Intelligence, AAAI 2024. AAAI Press (2024)
28. Zheng, W., Liu, Y., Doerr, B.: A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In: Conference on Artificial Intelligence, AAAI 2022. pp. 10408–10416. AAAI Press (2022)
29. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation **1**, 32–49 (2011)
30. Zhou, Z.H., Yu, Y., Qian, C.: Evolutionary Learning: Advances in Theories and Algorithms. Springer (2019)

A Supplementary Material

Here we provide the missing proofs of the results in Section 4 and details on our experiments that were omitted from the main part due to space constraints.

A.1 Missing Proofs for Section 4

Our analysis for mCOCZ first bounds the time until an individual is generated that optimizes the cooperative part and then reuses the results for mOMM.

Theorem 4. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mCOCZ. Let T denote the number of iterations until the population matches the complete Pareto front and let t be*

$$\left(\frac{\ln(2)m' + 2}{\ln(n)} + 16\frac{{m'}^2 + 2m'}{n} + 4\right) eS_m^{\mathrm{COCZ}} n\ln(n) + 1.$$

Then $T \le t$ with high probability and $E[T] \le (1 - \frac{1}{n})^{-3} t$.

Proof. We first consider the time until T_0 until an individual is generated that optimizes the cooperative part. Recall that b is the number of bits in each of the m' blocks. For an individual x, we define $\mathrm{half}(x) = \sum_{i=1}^{m'} \sum_{j=(i-1)b+1}^{ib-b/2} 1 - x_j$ to be the number of zeros in the cooperative parts. For a population P, let $\mathrm{half}(P) = \min_{x \in P} \mathrm{half}(x)$. Then T_0 is the time until $\mathrm{half}(P) = 0$. Observe that $x \succeq y$ implies $\mathrm{half}(x) \le \mathrm{half}(y)$. Thus, $\mathrm{half}(P)$ is non-increasing over the generations. Fix any generation with population P and let $\mathrm{half}(P) = i$, which is witnessed by some $x \in P$ with $\mathrm{half}(x) = i$. Then with probability at least

$$\frac{1}{S_m^{\mathrm{COCZ}}} \cdot \frac{i}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \ge \frac{i}{eS_m^{\mathrm{COCZ}} n} =: p_i$$

this iteration produces a (non-dominated) offspring x' with $\mathrm{half}(x') = i - 1$, namely be selecting x as parent for mutation and by then only flipping one of the i bits of value 0 in the cooperative parts. As $\mathrm{half}(P)$ is non-increasing, after at most $\frac{n}{2}$ such iterations we have $\mathrm{half}(P) = 0$.

For $i \in [\frac{n}{2}]$, let X_i be independent geometric random variables, each with success probability p_i, and let $X = \sum_{i=1}^{n/2} X_i$. Then X stochastically dominates T_0. By applying Theorem 1.10.35 in [9] we have

$$\Pr[X \ge 2eS_m^{\mathrm{COCZ}} n\ln(n)] \le \frac{1}{n},$$

that is, with probability at least $1 - \frac{1}{n}$ we have $T_0 \le 2eS_m^{\mathrm{COCZ}} n\ln(n)$. Observe that a solution x such that $\mathrm{half}(x) = 0$ is never strictly dominated by any other solution. Thus we can employ the same proofs as used for Theorem 1 by only considering progress made on individuals x such that $\mathrm{half}(x) = 0$. This gives that the next

$$\left(\frac{\ln(2)m' + 2}{\ln(n)} + 16\frac{{m'}^2 + 2m'}{n} + 2\right) eS_m^{\mathrm{COCZ}} n \ln(n) + 1$$

iterations cover the complete Pareto front with with probability at least $(1 - \frac{1}{n})^2$. Combining the two bounds gives that the GSEMO solves mCOCZ after t iterations with probability at least $(1 - \frac{1}{n})^3$, proving the first part of the statement. We note that this applies for arbitrary starting configurations, as all that we assumed about the initial population was that it is non-empty.

We employ a simple restart argument to obtain an upper bound on the expected value of T. As each sequence of t iterations fails to cover the Pareto front with probability at most $1 - (1 - \frac{1}{n})^3$ and due to the convergence of the geometric series we have

$$E[T] \leq \sum_{i=0}^{\infty} \left(1 - \left(1 - \frac{1}{n}\right)^3\right)^i t = \left(1 - \frac{1}{n}\right)^{-3} t. \qquad \square$$

In the analysis of the mLOTZ benchmark, we examine how in parallel some initial individual is step by step transformed into a solution for each objective value on the Pareto front.

Theorem 5. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing mLOTZ. Let T denote the number of iterations until the population matches the complete Pareto front and let*

$$t = \max\left\{1, \frac{8{m'}^2 \ln\left(\frac{n}{m'} + 1\right) + 8m' \ln(n)}{n}\right\} 2eS_m^{\mathrm{LOTZ}} \frac{n^2}{m'}.$$

Then $T \leq \lceil t \rceil$ with high probability and $E[T] \leq (1 - \frac{1}{n})^{-1} \lceil t \rceil$.

Proof. Let x_0 be an individual in the initial population. Consider any objective value $v = (a_1, \frac{n}{m'} - a_1, a_2, \frac{n}{m'} - a_2, \ldots, a_{m'}, \frac{n}{m'} - a_{m'})$ on the Pareto front and let s_v be the corresponding bitstring. We bound the time until s_v is sampled by bounding the time until a marked individual c becomes s_v. Let initially $c = x_0$. Whenever the individual c creates an offspring c' by flipping exactly one bit and c' is closer to v in any objective we update c to be c'. We also replace c by c' if c is removed from the population because it is strictly dominated or replaced by some bitstring c'. The time until the population contains s_v is dominated by the time until $c = s_v$. For any fixed $i \in [m']$, let T_i denote the number of iterations until the ith block of individual c has exactly a_i leading 1-bits and $\frac{n}{m'} - a_i$ trailing 0-bits. We first bound $\Pr[T_i \leq \lceil t \rceil]$.

Let the distance of c to s_v in block i be defined as

$$d_c = \max\{0, a_i - \mathrm{LO}_i(c)\} + \max\{0, \frac{n}{m'} - a_i - \mathrm{TZ}_i(c)\},$$

where $\mathrm{LO}_i(c)$ and $\mathrm{TZ}_i(c)$ denote the numbers of leading 1-bits and trailing 0-bits in the ith block of c, respectively. Observe that $c = s_v$ if and only if $d_c = 0$, that $d_c \le \frac{n}{m'}$, and that if c is replaced by an individual c' then $c' \succeq c$ and we have $d_{c'} \le d_c$. Thus, we have $c = s_v$ after at most $\frac{n}{m'}$ iterations that flip either the first 0-bit or the last 1-bit in the ith block, depending on c and v, and not flip any other bit, and thereby decrease d_c by at least 1. Each iteration has probability at least

$$\frac{1}{S_m^{\mathrm{LOTZ}}} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \ge \frac{1}{e S_m^{\mathrm{LOTZ}} n} =: p$$

to yield such an improvement, namely by selecting c as parent for mutation and by then flipping only the designated bit. For $j \in [\lceil t \rceil]$, let X_j be independent random variables, each with a Bernoulli distribution with success probability p. Let $X = \sum_{j=1}^{\lceil t \rceil} X_j$. Then, due to stochastic domination,

$$\Pr[T_i > \lceil t \rceil] \le \Pr\Big[X < \frac{n}{m'}\Big] \le \Pr\Big[X \le \frac{n}{m'}\Big].$$

By observing that $E[X] = p\lceil t \rceil \ge 2\frac{n}{m'}$ we have

$$\Pr\Big[X \le \frac{n}{m'}\Big] \le \Pr\Big[X \le \frac{1}{2}E[X]\Big].$$

Applying a Chernoff bound and noting that $\ln(m') < \ln(n)$ yields

$$\begin{aligned}
&\Pr[T_i > \lceil t \rceil] \le \exp\Big(-\frac{1}{8}E[X]\Big) \\
&\le \exp\Big(-\ln(m') - (2m'-1)\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big).
\end{aligned}$$

Using a union bound over all m' blocks gives that any fixed objective value on the Pareto front is not sampled in t iterations with probability at most

$$\begin{aligned}
m' \cdot \exp\Big(-\ln(m') - (2m'-1)\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big)& \\
= \exp\Big(-(2m'-1)\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big)&.
\end{aligned}$$

Let E denote the event that after $\lceil t \rceil$ iterations there is still any objective value v on the Pareto front such that the respective individual c does not have the correct number of leading 1-bits and trailing 0-bits in any block. By applying a union bound over all objective values we have

$$\Pr[E] \le S_m^{\mathrm{LOTZ}} \cdot \exp\Big(-(2m'-1)\ln\Big(\frac{n}{m'}+1\Big) - \ln(n)\Big) = \exp(-\ln(n)) = \frac{1}{n}.$$

by observing $S_m^{\mathrm{LOTZ}} \le \exp((2m'-1)\ln(\frac{n}{m'}+1))$. Thus, with high probability E does not happen. We note that this applies for arbitrary starting configurations, as all that we assumed about the initial population was that it is non-empty.

We employ a simple restart argument to obtain an upper bound on the expected value of T. Each sequence of $\lceil t \rceil$ iterations fails to cover the Pareto front with probability at most $\frac{1}{n}$. Due to the convergence of the geometric series we have

$$E[T] \le \sum_{i=0}^{\infty} \left(\frac{1}{n}\right)^i \lceil t \rceil = \left(1 - \frac{1}{n}\right)^{-1} \lceil t \rceil. \qquad \square$$

Next, we consider the mOJZJ$_k$ benchmark. We note that our bounds are only applicable if $m' \ge 2$. For the case $m' = 1$, we thus refer to previous results in the literature, which show that the expected number of iterations until the GSEMO solves 2OJZJ$_k$ is at most $eS_{2,k}^{\mathrm{OJZJ}}(\frac{3}{2}n^k + 2n\ln(\lceil \frac{n}{2} \rceil) + 3)$ [26].

Theorem 6. *Let $m' \in \mathbb{N}_{\ge 2}$ and $m = 2m'$. Consider the GSEMO optimizing mOJZJ$_k$. Let T denote the number of iterations until the population matches the complete Pareto front and let*

$$t = \left(\frac{\ln(4)m' + \ln(n)}{\ln(m')} + 1\right) 3e\ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k.$$

Then $T \le t$ with high probability . Further,

$$E[T] \le \left(1 - \frac{1}{m'}\right)^{-1} \left(\frac{\ln(4)m'}{\ln(m')} + 2\right) 3e\ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k.$$

Proof. Let t_1, t_2 and t_3 be bounds on the optimization times of the three phases given by Lemmas A1-A3, respectively. Observe $t_1 \le t_2, \lceil t_3 \rceil$, thus $t_1 + t_2 + \lceil t_3 \rceil \le 3t_2 = t$ and $T \le t$ with a high probability of at least $(1 - \frac{1}{n})^3$.

For the expected value, we employ the same simple restart argument as before and obtain that the expected time for the first and third phase are at most

$$\left(1 - \frac{1}{n}\right)^{-1} \left(\ln(2)\frac{m'}{\ln(n)} + 2\right) eS_{m,k}^{\mathrm{OJZJ}} (n-k)\ln(n-k)$$

and

$$\Big\lceil \left(1 - \frac{1}{n}\right)^{-1} \max\left\{2\left(\frac{n}{2m'} - k\right), 8\ln(m') + 8m'\ln\left(\frac{n}{m'} - 2k + 3\right) + 8\ln(n)\right\} \cdot 2em' S_{m,k}^{\mathrm{OJZJ}} \Big\rceil,$$

respectively. This holds as the arguments in the lemmas can be applied repeatedly, as all they assume about the initial population was that it is non-empty (for the first phase) and cover objective values generated in the previous phases (for the third phase). The expected time of the second phase is upper bounded by

$$\left(1 - \frac{1}{m'}\right)^{-1} \left(\frac{\ln(4)m'}{\ln(m')} + 2\right) e\ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k,$$

see Lemma A2. Once more, the bound on the second phase is the largest one, so by multiplying it by 3 we obtain

$$E[T] \le \left(1 - \frac{1}{m'}\right)^{-1} \left(\frac{\ln(4)m'}{\ln(m')} + 2\right) 3e \ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k. \qquad \square$$

The analysis of the first phase is comparable to the one of the first phase on mOMM in Lemma 2. However, extra arguments are required as the sets C_m for mOMM and $K_{m,k}$ for $m\mathrm{OJZJ}_k$ are not the same and, more importantly, a careful definition of progress is required to ensure that strictly dominated intermediate solutions in the valley of low fitness do not inflict any problems. We solve this by analyzing progress on a distance measure that is non-increasing when replacing a solution by a dominating one.

Lemma A1. *Let $m' \in \mathbb{N}$ and $m = 2m'$. Consider the GSEMO optimizing $m\mathrm{OJZJ}_k$. Let T denote the number of iterations until the population covers $K_{m,k}$ and let*

$$t = \left(\ln(2)\frac{m'}{\ln(n)} + 2\right) e S_{m,k}^{\mathrm{OJZJ}} (n-k)\ln(n-k).$$

Then $T \le t$ with probability at least $1 - \frac{1}{n}$.

Proof. Observe that by the symmetry of the problem and the operators, the expected bound to cover any fixed objective value in $K_{m,k}$ is the same for all elements in $K_{m,k}$. Thus, we first give a tail bound until the population contains an individual with objective value $(k,\dots,k) \in K_{m,k}$, that is, a bitstring such that the number of 1-bits in each block is exactly k. We then use this bound to give a tail bound on the time until the population contains all individuals in $K_{m,k}$.

Let x_0 be any bitstring in the initial population. We bound the time until a $(k,\dots,k)$-bitstring is sampled by bounding the time until a marked individual c becomes a $(k,\dots,k)$-bitstring. Let initially $c = x_0$. Whenever the individual c creates an offspring c' by flipping exactly one bit and c' is closer to any $(k,\dots,k)$-bitstring than c, we update c to be c'. We also replace c by c' whenever an individual c' replaces c in the population, which happens if $c' \succ c$ or c and c' have the same objective value. The time until the population contains a $(k,\dots,k)$-bitstring is at most the time until c is a $(k,\dots,k)$-bitstring.

Let $|c^i|_1$ denote the number of 1-bits in the ith block of c. Note that once $|c^i|_1 = k$ for some $i \in [m']$, the ith block of c will change no more. We first give a tail bound on the time until $|c^i|_1 = k$ for any fixed $i \in [m']$. We define $d_c = |c^i|_1 - k$ if $|c^i|_1 \ge k$ and otherwise $d_c = \frac{n}{m'} - |c^i|_1 - k$. Observe $0 \le d_c \le \frac{n}{m'} - k$ and if $d_c = 0$ then $|c^i|_1 = k$.

There are three possibilities on how an individual c' replaces c. First, by creating c' from c by flipping a single bit and decreasing the distance to a $(k,\dots,k)$-bitstring. Then $d_{c'} \le d_c - 1$ if the flipped bit is in the ith block and otherwise $d_{c'} = d_c$. Second, c might be replaced by an individual with the same objective value in the ith block, giving $d_{c'} = d_c$. Third, c' might strictly dominate c in the ith block. Then either $|c^i|_1 < k$ and $|c^i|_1 < |c'^i|_1 \leq \frac{n}{m'} - k$ or $|c^i|_1 > \frac{n}{m'} - k$ and $k \leq |c'^i|_1 < |c^i|_1$. In both cases $d'_c < d_c$. Thus, after at most $\frac{n}{m'} - k$ iterations of the first type that flip a bit in the ith block we have $d_c = 0$. The probability than an iteration yields such progress by selecting c for mutation, flipping one of the at least $d_c + k$ bits that bring us closer to k 1-bits and no other bit is at least

$$\frac{1}{S^{\text{OJZJ}}_{m,k}} \cdot (d_c + k) \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{d_c + k}{eS^{\text{OJZJ}}_{m,k} n} \geq \frac{d_c}{eS^{\text{OJZJ}}_{m,k}(n-k)} = p_{d_c}.$$

by observing that $d_c + k \leq n$ implies $d_c n \leq d_c n - d_c k + kn - k^2$ so $\frac{d_c}{n-k} \leq \frac{d_c+k}{n}$.

For $j \in [\frac{n}{m'} - k]$, let X_j be independent geometric random variables, each with success probability p_{d_c}, and let $X = \sum_{d_c=1}^{n/m'-k} X_j$. Then X stochastically dominates T_C, and thus a tail bound for X also applies to T_C. By using $m' \geq 1$ and once more applying Theorem 1.10.35 in [9] we have

$$\Pr[X \geq (1+\delta) e S^{\text{OJZJ}}_{m,k} (n-k) \ln(n-k)] \leq n^{-\delta}$$

for all $\delta \geq 0$. For $\delta = \ln(2)\frac{m'}{\ln(n)} + 1$ we obtain that no $(k, \ldots, k)$-bitstring is sampled after t iterations with probability at most $n^{-\ln(2)\frac{m'}{\ln(n)} - 1}$.

Let E denote the event that after t iterations there is still an objective value in $K_{m,k}$ such that the population does not contain a corresponding individual. By applying a union bound we have

$$\Pr[E] \leq |K_{m,k}| \cdot n^{-\ln(2)\frac{m'}{\ln(n)} - 1} = \frac{1}{n}$$

by observing $|K_{m,k}| = 2^{m'} = n^{\ln(2)m'/\ln(n)}$. We note that this applies for arbitrary starting configurations, as all that we assumed about the initial population was that it is non-empty. $\square$

The second phase dominates the optimization time. Here, the fitness valleys have to be crossed in every block, requiring k simultaneous bitflips each. To prepare for the final phase, here we bound the time until solutions for all combinations of which blocks are right before or across the fitness valley are generated.

Lemma A2. *Let $m' \in \mathbb{N}_{\geq 2}$ and $m = 2m'$. Consider the GSEMO optimizing mOJZJ$_k$ starting with a population that contains at least one individual of each objective value in $K_{m,k}$. Let T denote the number of iterations until the population covers $C_{m,k}$ and let*

$$t = \left(\frac{\ln(4)m' + \ln(n)}{\ln(m')} + 1\right) e \ln(m') S^{\text{OJZJ}}_{m,k} n^k.$$

Then $T \leq t$ with probability at least $1 - \frac{1}{n}$. Further,

$$E[T] \leq \left(1 - \frac{1}{m'}\right)^{-1} \left(\frac{\ln(4)m'}{\ln(m')} + 2\right) e \ln(m') S^{\text{OJZJ}}_{m,k} n^k.$$

Proof. Consider any objective value $v = (a_1, a_2, \ldots, a_{m'}) \in C_{m,k}$. Let c_0 be any individual with objective value in $K_{m,k}$ that is closest to any $(a_1, a_2, \ldots, a_{m'})$-bitstring. We bound the time until an $(a_1, a_2, \ldots, a_{m'})$-bitstring is sampled by bounding the time until a marked individual c becomes an $(a_1, a_2, \ldots, a_{m'})$-bitstring. Let initially $c = c_0$. Whenever the individual c creates an offspring c' that is closer to any $(a_1, a_2, \ldots, a_{m'})$-bitstring than c by flipping the remaining k bits in any block and not flipping any other bit, we update c to be c'. We also replace c by c' whenever an individual c' replaces c in the population. This only happens if c and c' have the same objective value as all c have objective values on the Pareto front. The time T_v until the population contains a $(a_1, a_2, \ldots, a_{m'})$-bitstring is at most the time until c is a $(a_1, a_2, \ldots, a_{m'})$-bitstring. Suppose c differs from being an $(a_1, a_2, \ldots, a_{m'})$-bitstring in i blocks. Each iteration has a probability of at least

$$\frac{1}{S_{m,k}^{\mathrm{OJZJ}}} \cdot i \cdot \left(\frac{1}{n}\right)^k \cdot \left(1 - \frac{1}{n}\right)^{n-k} \geq \frac{i}{e S_{m,k}^{\mathrm{OJZJ}} n^k} = p_i$$

to select c for mutation and complete any of the i incomplete blocks by flipping the remaining k bits and no other bit.

Let X_i be independent geometric random variable for all $i \in [m']$, each with success probability p_i, and let $X = \sum_{i=1}^{m'} X_i$. Then X stochastically dominates T_v, and thus a tail bound for X also applies to T_v. By once more applying Theorem 1.10.35 in [9], we have

$$\Pr\big[X \geq (1+\delta) e \ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k\big] \leq {m'}^{-\delta}. \tag{1}$$

By choosing $\delta = \frac{\ln(4)m' + \ln(n)}{\ln(m')}$, the probability to not sample an $(a_1, a_2, \ldots, a_{m'})$-bitstring in the next t iterations is at most

$${m'}^{-\ln(4)\frac{m'}{\ln(m')} - \frac{\ln(n)}{\ln(m')}} = n^{-\ln(4)\frac{m'}{\ln(n)} - 1}.$$

Thus, for all choices of m', the probability to not generate a solution with value v after t generations is at most $n^{-\ln(4)m'/\ln(n) - 1}$.

Let E denote the event that after t iterations there is still an objective value $v \in C_{m,k}$ such that the population does not contain a corresponding individual. By applying a union bound we have

$$\Pr[E] \leq |C_{m,k}| \cdot n^{-\ln(4)\frac{m'}{\ln(n)} - 1} = \frac{1}{n}$$

by observing $|C_{m,k}| = 4^{m'} = n^{\ln(4)m'/\ln(n)}$.

For the expected value, consider Equation 1 with $\delta = \ln(4)\frac{m'}{\ln(m')} + 1$ and recall that $m' \geq 2$ This gives a tail bound for the runtime of

$$t' := \left(\frac{\ln(4)m'}{\ln(m')} + 2\right) e \ln(m') S_{m,k}^{\mathrm{OJZJ}} n^k.$$

Let E' denote the event that after t' iterations there is still an objective value $v \in C_{m,k}$ such that the population does not contain a corresponding individual. By applying a union bound and observing $|C_{m,k}| = {m'}^{\ln(4)m'/\ln(m')}$ we have

$$\Pr[E'] \leq |C_{m,k}| \cdot {m'}^{-\ln(4)\frac{m'}{\ln(m')}-1} = \frac{1}{m'}.$$

We employ a simple restart argument. Each sequence of t' iterations fails to cover the Pareto front with probability at most $\frac{1}{m'}$. Due to the convergence of the geometric series we have

$$E[T] \leq \sum_{i=0}^{\infty} \left(\frac{1}{m'}\right)^i t' = \left(1 - \frac{1}{m'}\right)^{-1} t'. \qquad \square$$

We observe that t has an additional summand of $\frac{\ln(n)}{\ln(m')}$ in the parenthesis that is not present in the bound of the expected value. This additional summand is necessary to obtain the tail bound, as can be seen best when considering a constant value for m'. Then, a constant number of events (jumps over the fitness valley) have to happen, each with an estimated waiting time of $eS_{m,k}^{\mathrm{OJZJ}} n^k$. In order to obtain that all these events happen with high probability, that is, probability $1 - O(1/n)$, in a given time, this time has to depend on n to some degree, and this is the additional $\frac{\ln(n)}{\ln(m')}$ term. For the expected value this does not matter, as the expected time until all events happen is bounded by the sum over the (constantly many) individual expected waiting times.

The third phase accounts for generating solutions for the remaining part of the Pareto front. We consider every missing solution and note that for each block, it lies either in the inner Pareto front or beyond one of the two fitness valleys. Thanks to the second phase, for each possible combination there is already a solution in the current population. Thus no more jumps or walks through the fitness valley are required and all that remains is a walk on the inner Pareto front to generate the exact objective value. Apart from these changed preconditions, the analysis is comparable to the process of covering the remaining Pareto front of mOMM in Lemma 3.

Lemma A3. *Let $m' \in \mathbb{N}_{\geq 2}$ and $m = 2m'$. Consider the GSEMO optimizing mOJZJ$_k$ starting with a population that contains at least one individual of each objective value in $C_{m,k}$. Let T denote the number of iterations until the population covers the complete Pareto front and let*

$$t = \max\left\{2\left(\frac{n}{2m'} - k\right), 8\ln(m') + 8m'\ln\left(\frac{n}{m'} - 2k + 3\right) + 8\ln(n)\right\} \cdot 2em'S_{m,k}^{\mathrm{OJZJ}}.$$

Then $T \leq \lceil t \rceil$ with probability at least $1 - \frac{1}{n}$.

Proof. Consider any objective value $v = (a_1, a_2, \ldots, a_{m'})$ on the Pareto front that is not in $C_{m,k}$. Let c_0 be any individual in the population with objective value in $C_{m,k}$ that is closest to any $(a_1, a_2, \ldots, a_{m'})$-bitstring. We bound the time until an $(a_1, a_2, \ldots, a_{m'})$-bitstring is sampled by bounding the time until a marked individual c becomes an $(a_1, a_2, \ldots, a_{m'})$-bitstring. Let initially $c = c_0$. Whenever the individual c creates an offspring c' by flipping exactly one bit and c' is closer to any $(a_1, a_2, \ldots, a_{m'})$-bitstring than c, we update c to be c'. We also replace c by c' whenever an individual c' replaces c in the population. This only happens if c and c' have the same objective value as all c have objective values on the Pareto front. The time until the population contains an $(a_1, a_2, \ldots, a_{m'})$-bitstring is at most the time until c is an $(a_1, a_2, \ldots, a_{m'})$-bitstring.

Due to the symmetry of the problem, we can without loss of generality for all $i[m']$ assume $a_i \leq \frac{n}{2m'}$ and c_0 to have no bits of value 1 if $a_i = 0$ and k bits of value 1, otherwise. We first give a tail bound on the time until c has a_i bits of value 1 in the ith block for any fixed $1 \leq i \leq m'$. The probability of increasing the number of 1-bits in block i in any iteration by selecting c for mutation, flipping one of the at least $\frac{n}{2m'}$ bits of value 0 in the ith block, and not flipping any other bit is at least

$$\frac{1}{S_{m,k}^{\mathrm{OJZJ}}} \cdot \frac{n}{2m'} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{2em' S_{m,k}^{\mathrm{OJZJ}}} = p.$$

After at most $a_i - k \leq \frac{n}{m'} - k$ such iterations, c contains the correct number of 1-bits in the ith block. Thus, for the time T_i until the ith block of c is correct we have $E[T_i] \leq (a_i - k) \cdot \frac{1}{p}$. For $1 \leq j \leq \lceil t \rceil$, let X_j be independent random variables, each with a Bernoulli distribution with success probability p. Let $X = \sum_{j=1}^{\lceil t \rceil} X_j$. Then $\Pr[T_i > \lceil t \rceil] \leq \Pr[X < a_i - k] \leq \Pr[X \leq a_i - k]$. By observing $E[X] = p\lceil t \rceil \geq 2(\frac{n}{2m'} - k)$ we have

$$\Pr[X \leq a_i - k] \leq \Pr\Big[X \leq \frac{n}{2m'} - k\Big] \leq \Pr\Big[X \leq \frac{1}{2}E[X]\Big].$$

Applying a Chernoff bound yields

$$\Pr[T_i > \lceil t \rceil] \leq \exp\Big(-\frac{1}{8}E[X]\Big) \leq \exp\Big(-\ln(m') - m' \ln\Big(\frac{n}{m'} - 2k + 3\Big) - \ln(n)\Big).$$

Using a union bound over all blocks gives that any fixed objective value on the Pareto front is not sampled in $\lceil t \rceil$ iterations with probability at most

$$\begin{aligned} m' \cdot \exp\Big(-\ln(m') - m' \ln\Big(\frac{n}{m'} - 2k + 3\Big) - \ln(n)\Big) \\ = \exp\Big(-m' \ln\Big(\frac{n}{m'} - 2k + 3\Big) - \ln(n)\Big). \end{aligned}$$

Let E denote the event that after $\lceil t \rceil$ iterations there is still an objective value in the Pareto front that does not have a respective individual in the population. By applying a union bound we have

$$\Pr[E] \leq S_{m,k}^{\mathrm{OJZJ}} \cdot \exp\Big(-m' \ln\Big(\frac{n}{m'} - 2k + 3\Big) - \ln(n)\Big) = \exp(-\ln(n)) = \frac{1}{n}$$

by observing

$$S_{m,k}^{\mathrm{OJZJ}} = (\frac{n}{m'} - 2k + 3)^{m'} = \exp(m' \ln(\frac{n}{m'} - 2k + 3)). \qquad \square$$

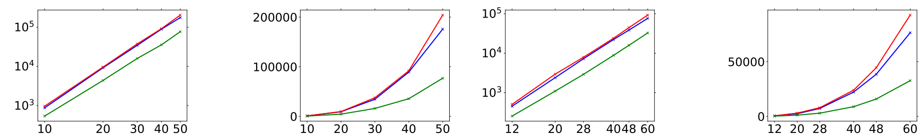
A.2 Experimental Evaluation

We augment our theoretical findings with a short empirical comparison of the SEMO, GSEMO, SMS-EMOA, and NSGA-II on the discussed benchmarks. This is interesting both because we have no proven lower bounds that match our upper bounds (note that due to the complex population dynamics, very few such lower bounds exist for MOEAs, and none for three or more objectives) and because our asymptotic bounds hide constant factors and lower-order terms. [3]

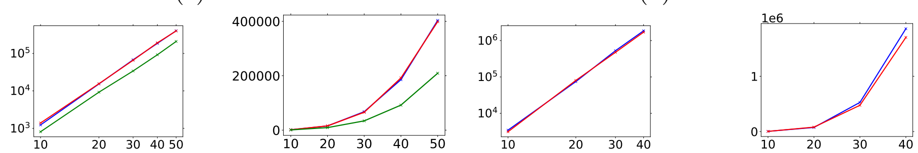
Figure 1 visualizes the optimization times (number of iterations until the complete Pareto front is represented in the population) for different problem sizes n, for the discussed algorithms and benchmarks. Fitting to our intuition and our runtime guarantees, the SEMO is faster than the GSEMO on all benchmarks except $m\mathrm{OJZJ}_k$ (which it cannot optimize). Since the SEMO cannot optimize problems that require to flip more than one bit, we would nevertheless not recommend to use it in practical applications, but leave it in its role as an easy-to-analyze algorithm for theoretical studies. The optimization times of the GSEMO and SMS-EMOA seem roughly comparable, suggesting that the fixed population size of the SMS-EMOA does neither hinder nor improve much over the optimization process as transferred from the GSEMO.

As the expected runtime and space consumption increases exponentially in m, we limited our experiments to the case to $m = 4$. Nevertheless, already for $m = 4$, the evaluated algorithms outperform the NSGA-II, which in 3 independent runs on each benchmark did not succeed to cover more than 91% of the Pareto front in the first 10^6 iterations, for a population of 4 times the size of the Pareto front, $k = 2$ in case of $m\mathrm{OJZJ}_k$, and $n = 20$ (except $n = 40$ for COCZ for a Pareto front of comparable size). For comparison, each run of each other algorithm on each benchmark took less than $2 \cdot 10^5$ iterations to cover the complete Pareto front for $n = 20$.

[3] See `https://github.com/SimonWiet/MOEA_evaluation` for code and results of the experiments. While we use our own implementation of the SEMO, GSEMO, and SMS-EMOA, we employ the DEAP library [12] for the NSGA-II. We note that due the reasons discussed in [22], we employ a slightly modified version of their NSGA-II implementation to stay as close to the intuitive interpretation of the NSGA-II as possible.



(a) 4OMM (b) 4COCZ



(c) 4LOTZ (d) 4OJZJ_2

Fig. 1: Evaluation of the SEMO (green), GSEMO (blue), and SMS-EMOA with a population of the size of the Pareto front (red) on the 4-objective versions of the four benchmarks discussed in this work. The y-axis marks the number of iterations (note that we have one function evaluation per iteration for all algorithms) until the complete Pareto front is covered. The right plots show a regular scale, the left plots apply a logarithmic scale on both axes. Problem sizes $n \in \{10, 20, 30, 40, 50, 60\}$ are on the x-axis ($n \in \{12, 20, 28, 40, 48, 60\}$ for 4COCZ since the problem size here has to be a multiple of 4, no data points for $n > 40$ on 4OJZJ_2 due to the high runtime, no data for the SEMO on 4OJZJ_2 due to its inability to solve this problem as discussed in Section 5.1).
The displayed values are the means over 30 independent runs. To increase the readability, we did not display the standard deviation, but we note that for none of the data points it exceeded 66% of the respective mean (and often it was much smaller).