Runtime Analysis of the SMS-EMOA for Many-Objective Optimization

Weijie Zheng[1], **Benjamin Doerr**[2*]

[1] School of Computer Science and Technology, International Research Institute for Artificial Intelligence, Harbin Institute of Technology, Shenzhen, China

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

zhengweijie@hit.edu.cn, lastname@lix.polytechnique.fr

Abstract

The widely used multiobjective optimizer NSGA-II was recently proven to have considerable difficulties in many-objective optimization. In contrast, experimental results in the literature show a good performance of the SMS-EMOA, which can be seen as a steady-state NSGA-II that uses the hypervolume contribution instead of the crowding distance as the second selection criterion.

This paper conducts the first rigorous runtime analysis of the SMS-EMOA for many-objective optimization. To this aim, we first propose a many-objective counterpart, the m-objective mOJZJ problem, of the bi-objective OJZJ benchmark, which is the first many-objective multimodal benchmark used in a mathematical runtime analysis. We prove that SMS-EMOA computes the full Pareto front of this benchmark in an expected number of $O(M^2n^k)$ iterations, where n denotes the problem size (length of the bit-string representation), k the gap size (a difficulty parameter of the problem), and $M = (2n/m - 2k + 3)^{m/2}$ the size of the Pareto front. This result together with the existing negative result on the original NSGA-II shows that in principle, the general approach of the NSGA-II is suitable for many-objective optimization, but the crowding distance as tie-breaker has deficiencies.

We obtain three additional insights on the SMS-EMOA. Different from a recent result for the bi-objective OJZJ benchmark, the stochastic population update often does not help for mOJZJ. It results in a $1/\Theta(\min\{Mk^{1/2}/2^{k/2}, 1\})$ speed-up, which is $\Theta(1)$ for large m such as $m > k$. On the positive side, we prove that heavy-tailed mutation still results in a speed-up of order $k^{0.5+k-\beta}$. Finally, we conduct the first runtime analyses of the SMS-EMOA on the bi-objective OneMinMax and LOTZ benchmarks and show that it has a performance comparable to the GSEMO and the NSGA-II.

Introduction

The NSGA-II is the most widely-applied multiobjective evolutionary algorithm (MOEA). Non-dominated sorting and crowding distance are its two major features differentiating it from basic MOEAs such as the GSEMO or the $(\mu + 1)$ SIBEA. Zheng, Liu, and Doerr (2022) conducted the first runtime analysis of the NSGA-II (see (Zheng and Doerr 2023a) for the journal version). This work quickly inspired many interesting follow-up results in bi-objective optimization (Zheng and Doerr 2022; Bian and Qian 2022; Doerr and Qu 2023a,b,c; Dang et al. 2023a,b; Cerf et al. 2023). In contrast to these positive results for two objectives, Zheng and Doerr (2023b) proved that for $m \geq 3$ objectives the NSGA-II needs at least exponential time (in expectation and with high probability) to cover the full Pareto front of the m-objective OneMinMax benchmark, a simple many-objective version of the basic OneMax problem where all search points are Pareto optimal. They claimed that the main reason for this low efficiency is the independent computation of the crowding distance in each objective.

A very recent work showed that the NSGA-III, a successor algorithm of the NSGA-II aimed at better coping with many objectives, can efficiently solve the 3-objective OneMinMax problem (Wietheger and Doerr 2023). Since apparently practitioners much prefer the NSGA-II (more than 17,000 citations on Google scholar only in the last five years) over the NSGA-III (5,035 citations since its publication in 2013), it remains an interesting question whether there are variants of the NSGA-II which better cope with many objectives.

With the SMS-EMOA, an interesting variant of the NSGA-II was proposed by Beume, Naujoks, and Emmerich (2007). This algorithm is a steady-state variant of the NSGA-II (that is, in each iteration only a single offspring is generated and possibly integrated into the population) that further replaces the crowding distance as secondary selection criterion with the classic hypervolume contribution. Many empirical works (see the almost 2,000 papers citing (Beume, Naujoks, and Emmerich 2007)) confirmed the good performance of the SMS-EMOA for many-objective optimization. The first mathematical runtime analysis of the SMS-EMOA was conducted very recently by Bian et al. (2023), who proved that its expected runtime on the bi-objective OJZJ problem is $O(n^{k+1})$. They also proposed a stochastic population update mechanism and proved that it has the often superior runtime of $O(n^{k+1}\min\{1, n/2^{k/4}\})$. Zheng et al. (2024) proved that the SMS-EMOA has an expected runtime of $O(n^4)$ on the bi-objective DLTB problem.

Our Contributions: This paper conducts the first mathematical runtime analysis of the SMS-EMOA for more than two objectives. We first define the mOJZJ benchmark,

*Corresponding author.

an m-objective counterpart of the bi-objective OJZJ problem (Zheng and Doerr 2023d), which is the problem analyzed in the first runtime analysis for the SMS-EMOA (Bian et al. 2023). We note that the mOJZJ problem is the first multimodal many-objective benchmark proposed for a theoretical analysis, to the best of our knowledge. We prove that the SMS-EMOA covers the full Pareto front of this benchmark in an expected number of $O(M^2 n^k)$ iterations, where n is the problems size, k the gap size (a difficulty parameter of the problem), m the number of objectives, and $M = (2n/m - 2k + 3)^{m/2}$ the size of the Pareto front.

We recall that the original NSGA-II needs at least exponential time to optimize the mONEMINMAX problem, which is a special case of mOJZJ with gap size $k = 1$. Since the SMS-EMOA employs non-dominated sorting, but replaces the crowding distance in the original NSGA-II by the hypervolume contribution, our result in a similar fashion as the analysis of the NSGA-III in (Wietheger and Doerr 2023) suggests that the general approach of the NSGA-II is suitable for many-objective optimization and that it is only the crowding distance as tie-breaker which is not appropriate for more than two objectives.

We then analyze whether the better performance of the SMS-EMOA on the bi-objective OJZJ problem achieved via a new stochastic population update (Bian et al. 2023) extends to the m-objective mOJZJ problem. Unfortunately, we shall observe that only a speed-up of order $1/\Theta(\min\{Mk^{1/2}/2^{k/2}, 1\})$ is obtained, which is $\Theta(1)$ when m is large, e.g., $m > k$.

On the positive side, we show that the advantage of heavy-tailed mutation is preserved. We analyze the SMS-EMOA with heavy-tailed mutation on the mOJZJ benchmark and prove that a speed-up of order $k^{0.5+k-\beta}$ is achieved. This is the same speed-up as observed for single-objective and bi-objective JUMP problems (Doerr et al. 2017; Zheng and Doerr 2023d). We note that this is the first theoretical work to support the usefulness of heavy-tailed mutation in many-objective optimization.

Finally, since so far the performance of the SMS-EMOA was only analyzed on the bi-objective OJZJ problem (Bian et al. 2023), we conduct mathematical runtime analyses of the SMS-EMOA also on the two most prominent bi-objective benchmarks ONEMINMAX and LOTZ. We prove that the SMS-EMOA finds the Pareto fronts of these benchmarks in an expected number of iterations of at most $2e(n+1)n(\ln n+1)$ for ONEMINMAX and at most $2en^2(n+1)$ for LOTZ. These are the same asymptotic runtimes (in terms of fitness evaluations) as known for the GSEMO and the NSGA-II.

Preliminaries

In many-objective optimization, one tries to find good solutions for a problem containing several, but at least three objectives. This work considers *pseudo-Boolean maximization*, hence our problem is described by a function $f = (f_1, \dots, f_m) : \{0,1\}^n \to \mathbb{R}^m$. Here $m > 2$ is the number of objectives and $n \in \mathbb{N}$ is called the *problem size*. As in bi-objective optimization (that is, the case $m = 2$), usually not all solutions are comparable. We say that a solution $x \in \{0,1\}^n$ *dominates* a solution y, denoted as $x \succ y$, if $f_i(x) \geq f_i(y)$ for all $i \in \{1, \dots, m\}$, and at least one of these inequalities is strict. For a given many-objective function f, we say that $x \in \{0,1\}^n$ is a *Pareto optimum* if and only if there is no $y \in \{0,1\}^n$ that dominates x w.r.t. f. The set of all Pareto optima is called *Pareto set*, and the set of all function values of the Pareto optima is called *Pareto front*. The goal for an MOEA is to find the full Pareto front as much as possible, that is, to compute a not too large set S of solutions such that $f(S)$ equals or approximates well the Pareto front. As common in the mathematical runtime analysis of MOEAs, we call the *runtime* of an algorithm the number of function evaluations until its population P covers the full Pareto front, that is, $f(P)$ contains the Pareto front. We refer to (Neumann and Witt 2010; Auger and Doerr 2011; Jansen 2013; Zhou, Yu, and Qian 2019; Doerr and Neumann 2020) for general introductions to the mathematical runtime analysis of evolutionary algorithms, and to (Brockhoff 2011) for a discussion of runtime analyses of MOEAs.

Mathematical Notations. For $a, b \in \mathbb{N}$ with $a \leq b$, we denote the set of integers in the interval $[a, b]$ by $[a..b] = \{a, a+1, \dots, b\}$. For a point $x \in \{0,1\}^n$, let $x_{[a..b]} := (x_a, x_{a+1}, \dots, x_b)$ and $|x|_1$ be the number of ones in x.

The m-Objective JUMP Function

As mentioned earlier, the NSGA-II was proven to have enormous difficulties in optimizing many-objective problems (Zheng and Doerr 2023b). In that paper, the m-objective counterpart mONEMINMAX of the bi-objective ONEMINMAX benchmark was proposed and analyzed. Since the first runtime work of the SMS-EMOA so far (Bian et al. 2023) analyzed the bi-objective jump problem OJZJ, whose special case with gap size $k = 1$ is (essentially) the ONEMINMAX problem, we shall propose and work with an m-objective version of OJZJ as well. Again, its special case $k = 1$ will be (essentially) equal to the mONEMINMAX problem. With this, our results are comparable both the ones in (Zheng and Doerr 2023b) and (Bian et al. 2023).

mOJZJ

We first recall the definition of the mONEMINMAX problem. For the ease of presentation, we only consider even numbers m of objectives here. In the mONEMINMAX problem, the bit string (of length n) is divided into $m/2$ blocks of equal length $2n/m$. On each of these, a bi-objective ONEMINMAX problem is defined. We note that this general block construction goes back to the seminal paper of (Laumanns, Thiele, and Zitzler 2004).

Definition 1 (Zheng and Doerr (2023b)). *Let m be the number of objectives and be even, and the problem size n be a multiple of $m/2$. Let $n' = 2n/m \in \mathbb{N}$. For any $x = (x_1, \dots, x_n)$, the m-objective function mONEMINMAX $f : \{0,1\}^n \to \mathbb{R}^m$ is defined by*

$$f_i(x) = \begin{cases} \text{ONEMAX}(\bar{x}_{[\frac{i-1}{2}n'+1..\frac{i+1}{2}n']}), & \text{if } i \text{ is odd} \\ \text{ONEMAX}(x_{[\frac{i-2}{2}n'+1..\frac{i}{2}n']}), & \text{else,} \end{cases}$$

where $\bar{x} = (1-x_1, \dots, 1-x_n)$ *and the function* $\text{OneMax} : \{0,1\}^{n'} \to \mathbb{R}$ *is defined by*

$$\text{OneMax}(y) = \sum_{i=1}^{n'} y_i$$

for any $y \in \{0,1\}^{n'}$.

In order to include mOneMinMax as a special case of mOJZJ, we define the m-objective OJZJ in a similar manner, that is, we divide the n bit positions into $m/2$ blocks and define a OJZJ problem in each block.

Definition 2. *Let* m *be the number of objectives and be even, and the problem size* n *be a multiple of* $m/2$. *Let* $n' = 2n/m \in \mathbb{N}$ *and* $k \in [1..n']$. *For any* $x = (x_1, \dots, x_n)$, *the* m*-objective function* $m\text{OJZJ}_k$ $f : \{0,1\}^n \to \mathbb{R}^m$ *is defined by*

$$f_i(x) = \begin{cases} \text{Jump}_{n',k}(x_{[\frac{i-1}{2}n'+1..\frac{i+1}{2}n']}), & \text{if } i \text{ is odd} \\ \text{Jump}_{n',k}(\bar{x}_{[\frac{i-2}{2}n'+1..\frac{i}{2}n']}), & \text{else,} \end{cases}$$

where $\bar{x} = (1-x_1, \dots, 1-x_n)$ *and the function* $\text{Jump}_{n',k} : \{0,1\}^{n'} \to \mathbb{R}$ *is defined by*

$$\text{Jump}_{n',k}(y) = \begin{cases} k + |y|_1, & \text{if } |y|_1 \le n-k \text{ or } y = 1^{n'} \\ n - |y|_1, & \text{else} \end{cases}$$

for any $y \in \{0,1\}^{n'}$.

We note that the function $\text{Jump}_{n',k}$ used in the definition above is the famous Jump benchmark. It was first defined by Droste, Jansen, and Wegener (2002) and has quickly become the most employed multimodal benchmark in the theory of randomized search heuristics, leading to many fundamental results on how these algorithms cope with local optima (Jansen and Wegener 2002; Dang et al. 2018; Doerr 2021; Benbaki, Benomar, and Doerr 2021; Hevia Fajardo and Sudholt 2022; Rajabi and Witt 2022; Witt 2023; Doerr et al. 2024).

Obviously, mOneMinMax and $\text{mOJZJ}_{n,1} - 1$ are equivalent problems, and mOJZJ is the previously defined OJZJ problem when $m = 2$.

Characteristics

We now give more details on this mOJZJ function. Let $B_i := [(i-1)n'+1..in']$, $i \in [1..m/2]$, be the i-th block in the partition of n bit positions. From Definition 2, we know that the bit values in the block B_i only influence the objectives f_{2i-1} and f_{2i}. Figure 1 plots the objective values of f_{2i-1} and f_{2i} relative to the number of ones in this block. Obviously, mOJZJ is multimodal with respect to the definition of multimodality of multiobjective problems in (Zheng and Doerr 2023d).

It is not difficult to see that the Pareto set is $S^* = \{x \in \{0,1\}^n \mid |x_{B_i}|_1 \in [k..n-k] \cup \{0, n\}, i \in [1..m/2]\}$. Hence, the Pareto front is

$$F^* := \{(a_1, n'+2k-a_1, \dots, a_{m/2}, n'+2k-a_{m/2}) \mid a_1, \dots, a_{m/2} \in [2k..n'] \cup \{k, n'+k\}\}$$

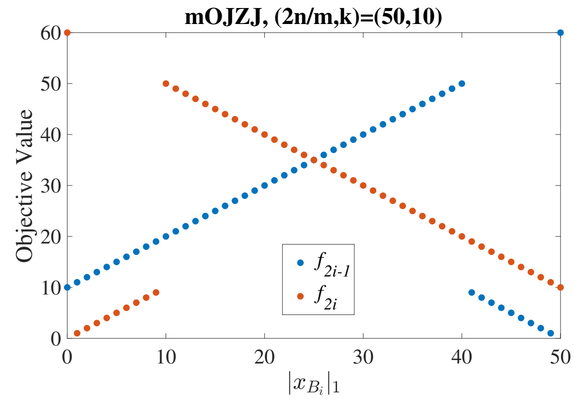

Figure 1: The objective values of f_{2i-1} and f_{2i} in mOJZJ w.r.t. $|x_{B_i}|_1$, the number of ones in the block B_i.

and the Pareto front size is $M := |F^*| = (n'-2k+3)^{m/2}$.

We recall at this point that we defined the runtime of an MOEA as the time to cover the full Pareto front F^*, that is, the first time to reach a population P with $F^* \subseteq f(P)$.

Note that for $k > 1$, not all points are Pareto optimal and not all sets of mutually non-dominated points are the subsets of the Pareto front. However, we have the following result showing that the maximum number of mutually non-dominated points is at most the size of the Pareto front. Due to the limited space, all mathematical proofs could only be sketched or had to be omitted completely. They can be found in the preprint (Zheng and Doerr 2023c).

Lemma 3. *Let* $k \le n'/2$ *and* P *be a set of solutions such that any two are not weakly dominated (w.r.t.* mOJZJ*). Then* $|P| \le M$.

For ease of reading of the proof sketches in the following sections, we call a Pareto optimum x an *inner Pareto optimum* if for all blocks $B_i = [(i-1)n'+1..in']$, $i \in [1..m/2]$, we have $x_{B_i} \notin \{0^{n'}, 1^{n'}\}$. We call $u = (u_1, n'+2k-u_1, \dots, u_{m/2}, n'+2k-u_{m/2})$ and $v = (v_1, n'+2k-v_1, \dots, v_{m/2}, n'+2k-v_{m/2})$ *neighbors* if there exists $i \in [1..m/2]$ such that $u_i - v_i = 1$ and $u_j = v_j$ for all $j \ne i$.

The SMS-EMOA Can Optimize mOJZJ

As discussed before, the original NSGA-II with non-dominated sorting and crowding distance needs at least exponential runtime to cover the full Pareto front of mOneMinMax, $m \ge 3$, (Zheng and Doerr 2023b), which is a special case of mOJZJ. In this section, we analyze the runtime of the SMS-EMOA on this problem and show that it does not encounter such problems.

Algorithm Description

The SMS-EMOA is a steady-state variant (that is, the offspring size is smaller than the parent size) of the NSGA-II. Like the NSGA-II, the SMS-EMOA works with a population of fixed size μ. However, in each iteration, only one offspring x' is generated from the parent population P_t.

Algorithm 1: SMS-EMOA

Initialize P_0 by generating μ solutions uniformly at random from $\{0,1\}^n$
for $t = 0, 1, 2, \ldots,$ **do**
  Select a solution x uniformly at random from P_t
  Generate x' by flipping each bit of x independently with probability $1/n$
  Use fast-non-dominated-sort() (Deb et al. 2002) to partition $R_t = P_t \cup \{x'\}$ into $F_1, \ldots, F_{i^*}$
  Calculate $\Delta_r(z, F_{i^*})$ for all $z \in F_{i^*}$ and find $D = \arg\min_{z \in F_{i^*}} \Delta_r(z, F_{i^*})$
  Uniformly at random pick $z' \in D$ and set $P_{t+1} = R_t \setminus \{z'\}$
end for

From the $\mu + 1$ individuals in the combined parent and offspring population $R_t = P_t \cup \{x'\}$, a single individual is removed. To this aim, the SMS-EMOA like the NSGA-II uses non-dominated sorting, that is, it partitions R_t into fronts $F_1, \ldots, F_{i^*}$, where F_i contains all non-dominated individuals in $R_t \setminus (\bigcup_{j=1}^{i-1} F_j)$. Different from the original NSGA-II that now uses the crowding distance as the secondary criterion for the removal of individuals, the SMS-EMOA removes the individual with the smallest hypervolume contribution in the critical front (which here always is the last front F_{i^*}).

The hypervolume of a set S of individuals w.r.t. a reference point r in the objective space is defined as

$$\mathrm{HV}_r(S) = \mathcal{L}\left(\bigcup_{u \in S} \{h \in \mathbb{R}^m \mid r \le h \le f(u)\}\right),$$

where $\mathcal{L}$ is the Lebesgue measure. The hypervolume contribution of an individual $x \in F_{i^*}$ is calculated via

$$\Delta_r(x, F_{i^*}) := \mathrm{HV}_r(F_{i^*}) - \mathrm{HV}_r(F_{i^*} \setminus \{x\}).$$

Algorithm 1 gives the pseudocode of the SMS-EMOA.

Runtime of SMS-EMOA

We now analyze the runtime of the SMS-EMOA, that is, the time until its population covers the full Pareto front of mOJZJ. We start by proving the following result ensuring that certain individuals survive in the population or are replaced by at least as good individuals. We formulate this result in manner more general than what we need since we expect this general version to be useful in other analyses of the SMS-EMOA.

The key to the proof is that (i) if there is a point with a rank larger than one, then all solutions with rank one (that is, in F_1) will survive, and (ii) if F_1 contains two or more solutions with same function value, then only such a solution can be removed from F_1. If the population size is large enough, by the pigeon-hole principle, one of these two cases comes true.

Lemma 4. *Consider any m-objective optimization problem. Let $M \in \mathbb{N}$ be such that any set S of pairwise non-dominated solutions having different objective values satisfies $|S| \le M$. Consider solving this problem via the SMS-EMOA with population size $\mu \ge M$ and using a reference point r such that $\mathrm{HV}_r(\{x\}) > 0$ for any individual $x \in \{0,1\}^n$.*

Then the following is true. If at some time t the combined parent and offspring population R_t of the SMS-EMOA contains some solution x (and thus in particular if $x \in P_t$), then at any later time $s > t$ the parent population P_s contains a solution y such that $y \succeq x$.

In particular, if R_t contains a Pareto optimum x, then all future generations contain a solution y with $f(y) = f(x)$.

Recall that an inner Pareto optimum is a Pareto optimum with $x_{B_i} \notin \{0^{n'}, 1^{n'}\}$ for all blocks $B_i = [(i-1)n' + 1..in'], i \in [1..m/2]$, as defined before. While it is very likely that at least one initial solution is an inner Pareto optimum, this does not happen with probability one, and hence in the following lemma we estimate the time until the population contains an inner Pareto optimum. Since this time usually is much smaller than the time to generate all remaining Pareto optima, we do not care that this is estimate could easily be improved.

The key to the proof is to note that changing a block x_{B_i} to a bit-string with between k and $n' - k$ ones is relatively easy. By Lemma 4, an individual with this f_i value will remain in the population. Hence a total of at most m such block changes (applied to the right individual) suffice to obtain an inner Pareto optimum.

Lemma 5. *Let $k \le n'/2$. Consider using SMS-EMOA with $\mu \ge M$ to optimize the mOJZJ problem. Then after at most $e\mu(mk)^k(1 + \ln m)$ iterations in expectation, the population (and also the populations afterwards) contains at least one inner Pareto optimum.*

In the next lemma, we consider the stage of covering all inner Pareto front points once at least one such point is in the population. The key to the proof is that as long as we have not yet discovered the full inner Pareto front, there always exists a missing Pareto front point that is a neighbor of a point that is already covered by the population. Hence choosing the right parent and flipping the right single bit suffices to cover the desired point on the Pareto front.

Lemma 6. *Let $k \le n'/2$. Consider using SMS-EMOA with $\mu \ge M$ to optimize the mOJZJ problem. Assume that the current population contains at least one inner Pareto optimum. Then after at most $en\mu M$ iterations in expectation, all inner Pareto front points are covered.*

The last stage is to cover the remaining Pareto front points. The following lemma shows the runtime. The key of the proof is that we can divide the Pareto front into several levels and any individual in the i-th level can be generated from a point in the $(i-1)$-th level by flipping the right k bits.

Lemma 7. *Let $k \le n'/2$. Consider using SMS-EMOA with $\mu \ge M$ to optimize the mOJZJ problem. Assume that the current population covers all inner Pareto front points. Then after at most $eM\mu n^k$ iterations in expectation, the full Pareto front is covered.*

Summing up the runtime of all stages from Lemmas 5 to 7, we have the following theorem for the runtime of the full coverage of the Pareto front.

Theorem 8. *Let $k \leq n'/2$. Consider using SMS-EMOA with $\mu \geq M$ to optimize the mOJZJ problem. Then after at most $e\mu(mk)^k(1+\ln m) + e\mu Mn + e\mu Mn^k$ iterations ($\mu + e\mu(mk)^k(1+\ln m) + e\mu Mn + e\mu Mn^k$ function evaluations) in expectation, the full Pareto front is covered.*

Since $k \leq n'/2 = n/m$ and $M \geq 2^{m/2}$, we easily see that both runtime expressions are $O(\mu Mn^k)$, even when allowing k, m and μ to depend on n.

Runtime of the GSEMO on mOJZJ

Since our arguments above can easily be extended to analyze the runtime of the GSEMO on the mOJZJ, we quickly do so for reasons of completeness. The GSEMO is a multi-objective counterpart of the $(1+1)$ EA. The initial population consists of a single randomly generated solution. In each iteration, a solution is uniformly at random picked from the population to generate an offspring via standard bit-wise mutation. If this offspring is not dominated by any solution in the population, it is added to the population and all solutions weakly dominated by it are removed. It is not difficult to see that all solutions in the population of the GSEMO are mutually non-dominated. Hence, the population size of the GSEMO is at most M by Lemma 3.

When analyzing the runtime of the GSEMO on mOJZJ, the main difference is that SMS-EMOA requires a statement like Lemma 4 to ensure that previous progress is not lost via unlucky selection decisions. For the GSEMO, this property follows immediately from the selection mechanism, which keeps all non-dominated solutions. Together with the upper bound M of the population size, we obtain the following theorem.

Theorem 9. *Let $k \leq n'/2$. Consider using the GSEMO to optimize the mOJZJ problem. Then after an expected number of at most $eM(mk)^k(1+\ln m) + eM^2n + eM^2n^k$ iterations (or $1 + eM(mk)^k(1+\ln m) + eM^2n + eM^2n^k$ fitness evaluations), the full Pareto front is covered.*

Reduced Impact of Stochastic Population Update

Bian, Zhou, Li, and Qian (2023) proposed a stochastic population update mechanism for the SMS-EMOA and proved that, somewhat unexpectedly given the state of the art, it can lead to significant performance gains. More precisely, it was proven that the classic SMS-EMOA with reasonable population size $\mu = \Theta(n)$ solves the bi-objective OJZJ problem in an expected number of $O(n^{k+2}) \cap \Omega(n^{k+1})$ function evaluations, whereas for the SMS-EMOA with stochastic population update, $O(n^{k+2}\min\{1, 2^{-k/4}n\})$ suffice. Hence, for $k = \omega(\log n)$, a super-polynomial speed-up was shown.

In this section, we extend the analysis of (Bian et al. 2023) to m objectives. Unfortunately, we will observe that the larger population sizes necessary here reduce the impact of the stochastic population update. As in (Bian et al. 2023), we have no proven tight lower bounds for the SMS-EMOA with stochastic population update, but our upper-bound proofs suggest that the reduced effect of the stochastic population update on the runtime guarantee is real, that is, the impact on the true runtime is diminishing with the larger population sizes necessary in the many-objective setting.

Stochastic Population Update

The rough idea of the stochastic selection proposed by Bian et al. (2023) is that a random half of the individuals survive into the next generation regardless of their quality. The individual to be discarded is chosen from the remaining individuals according to non-dominated sorting and hypervolume contribution as in the classic SMS-EMOA. This approach resembles the random mixing of two acceptance operators in the Move Acceptance Hyper-Heuristic studied recently (Lehre and Özcan 2013; Lissovoi, Oliveto, and Warwicker 2023; Doerr et al. 2023).

More precisely, after generating the offspring x', the SMS-EMOA with stochastic selection chooses from $P_t \cup \{x'\}$ a set R' of $\lfloor(\mu+1)/2\rfloor$ solutions randomly with replacement. The individual z' to be removed from $P_t \cup \{x'\}$ is then determined via non-dominated sorting and hypervolume contribution applied to R' only. We note that with this mechanism, even the worst solution enters the next generation with probability at least $1/2$.

Runtime

To analyze the runtime of the SMS-EMOA with stochastic population update, we first derive and formulate separately two insights on the survival of solutions. The following Lemmas 10 is an elementary consequence of the algorithm definition, whereas Lemma 11 builds on Lemma 4.

Lemma 10. *Consider using the SMS-EMOA with stochastic population update to optimize the mOJZJ problem. For any iteration t and $x \in R_t$, we have that there is a y with $f(y) = f(x)$ in P_{t+1} with probability at least $1/2$.*

Lemma 11. *Consider using the SMS-EMOA with stochastic population update and with $\mu \geq 2(M+1)$ to optimize the mOJZJ problem. If at some time t the combined parent and offspring population R_t of the SMS-EMOA contains some solution x, then at any later time $s > t$ the parent population P_s contains a solution y such that $y \succeq x$.*

Since the proofs of Lemmas 5 and 6 for the classic SMS-EMOA mostly relied on elementary properties of standard bit-wise mutation and on the survival guarantee of Lemma 4, we can now use the survival guarantee of Lemma 11 to obtain analogous results for the SMS-EMOA with stochastic selection (at the price of requiring essentially twice the population size). This yields the following estimates for the time to obtain at least one inner Pareto optimum and the time to cover all inner Pareto front points starting from a population with at least one inner Pareto optimum.

Lemma 12. *Let $k \leq n'/2$. Consider using the SMS-EMOA with stochastic population update and with $\mu \geq 2(M+1)$ to optimize the mOJZJ problem. Then*

- *after at most $e\mu(mk)^k(1+\ln m)$ iterations in expectation, the population (and also the populations afterwards) contains at least one inner Pareto optimum;*

- *after another at most $en\mu M$ iterations in expectation, all inner Pareto front points are covered.*

Now we consider the runtime for the full coverage of the Pareto front after all inner Pareto front points are covered. Similar to the proof of Lemma 7 for the original SMS-EMOA, we partition the Pareto front points into $m/2+1$ levels based on the number of extreme blocks they contain. The difference here is that we consider the effect of the gap points, similar to Bian et al. (2023).

Lemma 13. *Let $k \le n'/2$. Consider using the SMS-EMOA with stochastic population update and with $\mu \ge 2(M+1)$ to optimize the mOJZJ problem. Assume that the current population covers all inner Pareto front points. Then after at most $O(\min\{\frac{\mu k^{1/2}}{2^{k/2}}, 1\}M\mu n^k)$ iterations in expectation, the full Pareto front is covered.*

Combining Lemmas 12 and 13, we have the runtime of the SMS-EMOA with stochastic population update in the following theorem.

Theorem 14. *Let $k \le n'/2$. Consider using SMS-EMOA with $\mu \ge 2(M+1)$ and stochastic population update to optimize the mOJZJ problem. Then after at most $e\mu(mk)^k(1+\ln m) + en\mu M + O(\min\{\frac{\mu k^{1/2}}{2^{k/2}}, 1\}M\mu n^k)$ iterations in expectation, the full Pareto front is covered.*

Comparing the runtime guarantees or Theorem 8 (classic SMS-EMOA) and Theorem 14 (SMS-EMOA with stochastic selection), we see that stochastic selection can at most lead to a speed-up by a factor of order $2^{k/2}/\mu k^{1/2}$. Now μ is at least $\Omega(M) = \Omega((2n/m-2k+3)^{m/2})$. Consequently, the advantage of stochastic selection is rapidly decreasing with growing numbers of objectives and vanishes, e.g., when $m \ge k$.

Heavy-Tailed Mutation Helps

In the previous section, we saw that the advantages of a stochastic population update do not well generalize from bi-objective optimization to many-objective optimization. We now regard another design choice that so far was only analyzed in bi-objective optimization, namely a heavy-tailed mutation operator. We shall prove that the $k^{\Omega(k)}$ factor speed-up observed in bi-objective optimization extends to many objectives.

Heavy-Tailed Mutation

Different from the standard bit-wise mutation operator, which flips each bit independently with probability $1/n$, the heavy-tailed mutation operator proposed in (Doerr et al. 2017) flips each bit independently with probability α/n, where α follows a power-law distribution with parameter β. The number α is sampled anew in each application of the heavy-tailed mutation operator. The underlying power-law is defined as follows.

Definition 15. *Let $n \in \mathbb{N}$ and $\beta > 1$. We say α follows a power-law distribution with (negative) exponent β if for all $i \in [1..n/2]$, we have $\Pr[\alpha = i] = \left(\sum_{j=1}^{n/2} k^{-\beta}\right)^{-1} i^{-\beta}$.*

The power-law feature extends to the heavy-tailed mutation operator in the sense that it generates an offspring with Hamming distance j from the parent with probability $\Omega(j^{-\beta})$. This facilitates larger jumps in the search space, and thus escaping from a local optimum.

The heavy-tailed mutation operator has resulted in asymptotic performance gains by a factor of $k^{\Omega(k)}$ for the $(1+1)$ EA optimizing single-objective (classic and generalized) JUMP functions with gap size k (Doerr et al. 2017; Bambury, Bultel, and Doerr 2021). For the bi-objective OJZJ benchmark, again a speed-up of $k^{\Omega(k)}$ was proven when optimized via the GSEMO (Zheng and Doerr 2023d) and the NSGA-II (Doerr and Qu 2023a). Several other positive theoretical results exist for this heavy-tailed mutation, or more generally, other heavy-tailed parameter choices (Friedrich, Quinzan, and Wagner 2018; Wu, Qian, and Tang 2018; Quinzan et al. 2021; Corus, Oliveto, and Yazdani 2021; Dang et al. 2022; Antipov, Buzdalov, and Doerr 2022; Doerr, Ghannane, and Ibn Brahim 2022; Doerr and Rajabi 2023).

Prior to this work, no theoretical analysis of the heavy-tailed mutation operator in many-objective optimization existed.

Runtime

We now analyze the runtime of the SMS-EMOA using the heavy-tailed mutation operator instead of bit-wise mutation, the standard choice for this algorithm.

Since the heavy-tailed mutation does not change the survival selection of the original SMS-EMOA, we immediately have the following result on the survival of the individuals.

Corollary 16. *The assertion of Lemma 4 extends to the SMS-EMOA with heavy-tailed mutation.*

With proof ideas similar to those in Lemma 5 to 7, but noticing the different probabilities of generating solutions with a specific Hamming distance, we obtain the following runtime guarantee.

Theorem 17. *Let $k \le n'/2$ and $\beta > 1$. Consider using SMS-EMOA with $\mu \ge M$ and with heavy-tailed mutation to optimize the mOJZJ problem. Then after at most $O(M\mu k^{\beta-0.5-k}n^k)$ iterations or function evaluations in expectation, the full Pareto front is covered.*

Compared to the runtime guarantee of Theorem 8 for the original SMS-EMOA, the guarantee of Theorem 17 above for the SMS-EMOA with heavy-tailed mutation is by a factor of asymptotically $k^{k+0.5-\beta}$ stronger.

The SMS-EMOA Also Performs Well for Bi-objective Optimization

The above sections discussed the runtime of the SMS-EMOA for many objectives, and showed that it performs well for many objectives, which is different from the original NSGA-II. Since the only theory paper (Bian et al. 2023) on the SMS-EMOA merely considers its performance on the bi-objective OJZJ problem, to broaden our understanding of this algorithm we now analyze its runtime for the two most prominent bi-objective benchmarks OneMinMax and LOTZ.

OneMinMax and LOTZ

The OneMinMax benchmark introduced by Giel and Lehre (2010) (and also the CountingOnesCountingZeroes benchmark proposed by Laumanns et al. (2002)) are bi-objective counterparts of the famous single-objective OneMax benchmark. Likewise, the LOTZ benchmark defined by Laumanns, Thiele, and Zitzler (2004) (also the weighted version WLPTNO of Qian, Yu, and Zhou (2013)) are bi-objective counterparts of the classic single-objective LeadingOnes benchmark.

For OneMinMax, the two objectives count the number of ones and zeros in a given bit-string, respectively. For LOTZ, the first objective is the number of contiguous ones starting from the first bit position, and the second objective is the number of contiguous zeros starting from the last bit position. Both benchmarks with their simple and clear structure facilitate the fundamental theoretical understanding of MOEAs. We note that there are other benchmarks used in the mathematical analysis of MOEAs (see, e.g. Horoba and Neumann 2008; Brockhoff et al. 2009; Qian, Tang, and Zhou 2016; Li et al. 2016; Dang et al. 2023b), but clearly OneMinMax and LOTZ are the most prominent ones.

Here are the formal definitions of OneMinMax and LOTZ.

Definition 18 (Giel and Lehre (2010))**.** *Let $n \in \mathbb{N}$ be the problem size. The* OneMinMax *function* : $\{0,1\}^n \to \mathbb{R}$ *is defined by*

$$\textsc{OneMinMax}(x) = \left(\sum_{i=1}^{n}(1-x_i), \sum_{i=1}^{n} x_i\right)$$

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

Definition 19 (Laumanns, Thiele, and Zitzler (2004))**.** *Let $n \in \mathbb{N}$ be the problem size. The* LOTZ *function* : $\{0,1\}^n \to \mathbb{R}$ *is defined by*

$$\mathrm{LOTZ}(x) = \left(\sum_{i=1}^{n}\prod_{j=1}^{i} x_j, \sum_{i=1}^{n}\prod_{j=i}^{n}(1-x_j)\right)$$

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

Runtime

We now analyze the runtime of the SMS-EMOA on the OneMinMax and LOTZ benchmarks. Note that the Pareto front, and more generally the size of any set of pairwise non-dominated solutions, for OneMinMax and LOTZ have sizes at most $n+1$. Hence we can use Lemma 4 with $M = n+1$.

Theorem 20 below gives an upper bound for the runtime of SMS-EMOA on OneMinMax. This bound (in terms of fitness evaluations) is of the same asymptotic order as the runtime guarantee for the NSGA-II (Zheng, Liu, and Doerr 2022). Knowing from Lemma 4 that we cannot lose Pareto points, the proof of our results consists of adding the waiting times for finding a new Pareto point from an already existing neighboring one.

Theorem 20. *Consider using the SMS-EMOA with $\mu \geq n+1$ to optimize the* OneMinMax *problem with problem size n. Then after at most $2e\mu n(\ln n + 1)$ iterations (or $\mu + 2e\mu n(\ln n + 1)$ fitness evaluations) in expectation, the full Pareto front is covered.*

Our runtime guarantee for LOTZ, see Theorem 21 below, again coincides with the guarantee for the NSGA-II (Zheng, Liu, and Doerr 2022). In the proof, the process is divided into two stages. The first stage is to cover a Pareto front point for the first time and the second stage is to cover the full Pareto front starting from one covered Pareto front point. For the first stage, we pessimistically consider the time to reach 1^n. For the second stage, we consider the time to generate $1^i0^{n-i}, i = n-1, \dots, 0$ one after the other. In both stages, we heavily rely on Lemma 4 asserting that important progress is not lost, which allows to add waiting times of successive progress events.

Theorem 21. *Consider using the SMS-EMOA with $\mu \geq n+1$ to optimize the* LOTZ *problem with problem size n. Then after at most $2e\mu n^2$ iterations (or fitness evaluations) in expectation, the full Pareto front is covered.*

Note that the runtimes of the GSEMO on OneMinMax and LOTZ are $O(n^2 \ln n)$ and $O(n^3)$, respectively (Giel and Lehre 2010; Laumanns, Thiele, and Zitzler 2004). Hence, they agree with our bounds for the SMS-EMOA when using the most interesting population size $\mu = \Theta(n)$.

Conclusion

Motivated by the observed difficulty of the NSGA-II for many objectives, this paper resorted to the SMS-EMOA, a variant of the steady-state NSGA-II, and proved that, different from the NSGA-II, it efficiently solves the mOJZJ problem. Noting that the SMS-EMOA also employs non-dominated sorting, but replaces the crowding distance with the hypervolume, this result together with the one of Wietheger and Doerr (2023) supports the conclusion that non-dominated sorting is a good building block for MOEAs, but the crowding distance has deficiencies for more than two objectives.

We also showed that the stochastic population update proposed in (Bian et al. 2023) for the bi-objective SMS-EMOA becomes less effective for many objectives. All these results in a very rigorous manner support the general knowledge that multi-objective optimization becomes increasingly harder with growing numbers of objectives.

On the positive side, we showed that the advantages of heavy-tailed mutation, previously observed in single- and bi-objective optimization, remain when increasing the number of objectives.

Given that our first result shows advantages of the SMS-EMOA and there is only one previous work performing rigorous runtime analyses for this algorithm, we extended our knowledge in this direction by proving competitive runtime guarantees for this algorithm on the two most prominent bi-objective benchmarks.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 62306086), Science, Technology and Innovation Commission of Shenzhen Municipality (Grant No. GXWD20220818191018001), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2019A1515110177). This research benefited from the support of the FMJH Program Gaspard Monge for optimization and operations research and their interactions with data science.

References

Antipov, D.; Buzdalov, M.; and Doerr, B. 2022. Fast mutation in crossover-based algorithms. *Algorithmica*, 84: 1724–1761.

Auger, A.; and Doerr, B., eds. 2011. *Theory of Randomized Search Heuristics*. World Scientific Publishing.

Bambury, H.; Bultel, A.; and Doerr, B. 2021. Generalized jump functions. In *Genetic and Evolutionary Computation Conference, GECCO 2021*, 1124–1132. ACM.

Benbaki, R.; Benomar, Z.; and Doerr, B. 2021. A rigorous runtime analysis of the 2-MMAS$_{\mathrm{ib}}$ on jump functions: ant colony optimizers can cope well with local optima. In *Genetic and Evolutionary Computation Conference, GECCO 2021*, 4–13. ACM.

Beume, N.; Naujoks, B.; and Emmerich, M. 2007. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181: 1653–1669.

Bian, C.; and Qian, C. 2022. 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*, 428–441. Springer.

Bian, C.; Zhou, Y.; Li, M.; and Qian, C. 2023. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, 5513–5521. ijcai.org.

Brockhoff, D. 2011. Theoretical aspects of evolutionary multiobjective optimization. In Auger, A.; and Doerr, B., eds., *Theory of Randomized Search Heuristics*, 101–140. World Scientific Publishing.

Brockhoff, D.; Friedrich, T.; Hebbinghaus, N.; Klein, C.; Neumann, F.; and Zitzler, E. 2009. On the effects of adding objectives to plateau functions. *IEEE Transactions on Evolutionary Computation*, 13: 591–603.

Cerf, S.; Doerr, B.; Hebras, B.; Kahane, J.; and Wietheger, S. 2023. 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*, 5522–5530. ijcai.org.

Corus, D.; Oliveto, P. S.; and Yazdani, D. 2021. Fast immune system-inspired hypermutation operators for combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 25: 956–970.

Dang, D.; Eremeev, A. V.; Lehre, P. K.; and Qin, X. 2022. Fast non-elitist evolutionary algorithms with power-law ranking selection. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, 1372–1380. ACM.

Dang, D.; Friedrich, T.; Kötzing, T.; Krejca, M. S.; Lehre, P. K.; Oliveto, P. S.; Sudholt, D.; and Sutton, A. M. 2018. Escaping local optima using crossover with emergent diversity. *IEEE Transactions on Evolutionary Computation*, 22: 484–497.

Dang, D.-C.; Opris, A.; Salehi, B.; and Sudholt, D. 2023a. Analysing the robustness of NSGA-II under noise. In *Genetic and Evolutionary Computation Conference, GECCO 2023*, 642–651. ACM.

Dang, D.-C.; Opris, A.; Salehi, B.; and Sudholt, D. 2023b. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Conference on Artificial Intelligence, AAAI 2023*, 12390–12398. AAAI Press.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6: 182–197.

Doerr, B. 2021. The runtime of the compact genetic algorithm on Jump functions. *Algorithmica*, 83: 3059–3107.

Doerr, B.; Dremaux, A.; Lutzeyer, J. F.; and Stumpf, A. 2023. How the move acceptance hyper-heuristic copes with local optima: drastic differences between jumps and cliffs. In *Genetic and Evolutionary Computation Conference, GECCO 2023*, 990–999. ACM.

Doerr, B.; Echarghaoui, A.; Jamal, M.; and Krejca, M. S. 2024. Lasting diversity and superior runtime guarantees for the $(\mu + 1)$ genetic algorithm. In *Conference on Artificial Intelligence, AAAI 2024*. AAAI Press.

Doerr, B.; Ghannane, Y.; and Ibn Brahim, M. 2022. Towards a stronger theory for permutation-based evolutionary algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, 1390–1398. ACM.

Doerr, B.; Le, H. P.; Makhmara, R.; and Nguyen, T. D. 2017. Fast genetic algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2017*, 777–784. ACM.

Doerr, B.; and Neumann, F., eds. 2020. *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*. Springer. Also available at http://www.lix.polytechnique.fr/Labo/Benjamin.Doerr/doerr_neumann_book.html.

Doerr, B.; and Qu, Z. 2023a. A first runtime analysis of the NSGA-II on a multimodal problem. *IEEE Transactions on Evolutionary Computation*, 27: 1288–1297.

Doerr, B.; and Qu, Z. 2023b. From understanding the population dynamics of the NSGA-II to the first proven lower bounds. In *Conference on Artificial Intelligence, AAAI 2023*, 12408–12416. AAAI Press.

Doerr, B.; and Qu, Z. 2023c. Runtime analysis for the NSGA-II: Provable speed-ups from crossover. In *Conference on Artificial Intelligence, AAAI 2023*, 12399–12407. AAAI Press.

Doerr, B.; and Rajabi, A. 2023. Stagnation detection meets fast mutation. *Theoretical Computer Science*, 946: 113670.

Droste, S.; Jansen, T.; and Wegener, I. 2002. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276: 51–81.

Friedrich, T.; Quinzan, F.; and Wagner, M. 2018. Escaping large deceptive basins of attraction with heavy-tailed mutation operators. In *Genetic and Evolutionary Computation Conference, GECCO 2018*, 293–300. ACM.

Giel, O.; and Lehre, P. K. 2010. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18: 335–356.

Hevia Fajardo, M. A.; and Sudholt, D. 2022. Theoretical and empirical analysis of parameter control mechanisms in the $(1 + (\lambda, \lambda))$ genetic algorithm. *ACM Transactions on Evolutionary Learning and Optimization*, 2: 13:1–13:39.

Horoba, C.; and Neumann, F. 2008. Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization. In *Genetic and Evolutionary Computation Conference, GECCO 2008*, 641–648. ACM.

Jansen, T. 2013. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Springer.

Jansen, T.; and Wegener, I. 2002. The analysis of evolutionary algorithms – a proof that crossover really can help. *Algorithmica*, 34: 47–66.

Laumanns, M.; Thiele, L.; and Zitzler, E. 2004. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8: 170–182.

Laumanns, M.; Thiele, L.; Zitzler, E.; Welzl, E.; and Deb, K. 2002. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving from Nature, PPSN 2002*, 44–53. Springer.

Lehre, P. K.; and Özcan, E. 2013. A runtime analysis of simple hyper-heuristics: to mix or not to mix operators. In *Foundations of Genetic Algorithms, FOGA 2013*, 97–104. ACM.

Li, Y.-L.; Zhou, Y.-R.; Zhan, Z.-H.; and Zhang, J. 2016. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20: 563–576.

Lissovoi, A.; Oliveto, P. S.; and Warwicker, J. A. 2023. When move acceptance selection hyper-heuristics outperform Metropolis and elitist evolutionary algorithms and when not. *Artificial Intelligence*, 314: 103804.

Neumann, F.; and Witt, C. 2010. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer.

Qian, C.; Tang, K.; and Zhou, Z.-H. 2016. Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization. In *Parallel Problem Solving from Nature, PPSN 2016*, 835–846. Springer.

Qian, C.; Yu, Y.; and Zhou, Z. 2013. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence*, 204: 99–119.

Quinzan, F.; Göbel, A.; Wagner, M.; and Friedrich, T. 2021. Evolutionary algorithms and submodular functions: benefits of heavy-tailed mutations. *Natural Computing*, 20: 561–575.

Rajabi, A.; and Witt, C. 2022. Self-adjusting evolutionary algorithms for multimodal optimization. *Algorithmica*, 84: 1694–1723.

Wietheger, S.; and Doerr, B. 2023. A mathematical runtime analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, 5657–5665. ijcai.org.

Witt, C. 2023. How majority-vote crossover and estimation-of-distribution algorithms cope with fitness valleys. *Theoretical Computer Science*, 940: 18–42.

Wu, M.; Qian, C.; and Tang, K. 2018. Dynamic mutation based Pareto optimization for subset selection. In *Intelligent Computing Methodologies, ICIC 2018, Part III*, 25–35. Springer.

Zheng, W.; and Doerr, B. 2022. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, 611–619. ACM.

Zheng, W.; and Doerr, B. 2023a. Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II). *Artificial Intelligence*, 325: 104016.

Zheng, W.; and Doerr, B. 2023b. Runtime analysis for the NSGA-II: proving, quantifying, and explaining the inefficiency for many objectives. *IEEE Transactions on Evolutionary Computation*. In press, https://doi.org/10.1109/TEVC.2023.3320278.

Zheng, W.; and Doerr, B. 2023c. Runtime analysis of the SMS-EMOA for many-objective optimization. *CoRR*, abs/2312.10290.

Zheng, W.; and Doerr, B. 2023d. Theoretical analyses of multiobjective evolutionary algorithms on multimodal objectives. *Evolutionary Computation*, 31: 337–373.

Zheng, W.; Li, M.; Deng, R.; and Doerr, B. 2024. How to use the Metropolis algorithm for multi-objective optimization? In *Conference on Artificial Intelligence, AAAI 2024*. AAAI Press.

Zheng, W.; Liu, Y.; and Doerr, B. 2022. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Conference on Artificial Intelligence, AAAI 2022*, 10408–10416. AAAI Press.

Zhou, Z.-H.; Yu, Y.; and Qian, C. 2019. *Evolutionary Learning: Advances in Theories and Algorithms*. Springer.