Why Popular MOEAs are Popular: Proven Advantages in Approximating the Pareto Front

Mingfeng Li[1]*, **Qiang Zhang**[1]*, **Weijie Zheng**[1,2]†, **Benjamin Doerr**[3]
[1]School of Computer Science and Technology,
State Key Laboratory of Smart Farm Technologies and Systems,
International Research Institute for Artificial Intelligence,
Harbin Institute of Technology, Shenzhen, China
[2]Pencheng Laboratory, Shenzhen, China
[3]Laboratoire d'Informatique (LIX), CNRS, École Polytechnique,
Institut Polytechnique de Paris, Palaiseau, France
limingfeng@stu.hit.edu.cn, zhengweijie@hit.edu.cn, doerr@lix.polytechnique.fr

Abstract

Recent breakthroughs in the analysis of multi-objective evolutionary algorithms (MOEAs) are mathematical runtime analyses of those algorithms which are intensively used in practice. So far, most of these results show the same performance as previously known for simpler algorithms like the GSEMO. The few results indicating advantages of the popular MOEAs share the same shortages: They consider the performance for the problem of computing the full Pareto front, (of some algorithms enriched with newly invented mechanisms,) and this on newly designed benchmarks. In this work, we overcome these shortcomings by analyzing how existing popular MOEAs approximate the Pareto front of the established LARGEFRONT benchmark. We prove that several popular MOEAs, including NSGA-II (with current crowding distance), NSGA-III, SMS-EMOA, and SPEA2, only need an expected time of $O(n^2 \log n)$ fitness evaluations to compute an additive ε-approximation of the Pareto front of the LARGEFRONT benchmark. This contrasts with the already proven exponential runtime (with high probability) of the GSEMO on the same task. This result is the first mathematical runtime analysis showing and explaining the superiority of popular MOEAs over simple ones like the GSEMO for the central task of computing good approximations to the Pareto front.

1 Introduction

Theoretical runtime analysis of the evolutionary algorithms is always hard to crack. Early theoretical results for the multi-objective evolutionary algorithms (MOEAs) [LTZ+02, LTZ04] focused on the simpler ones like the (G)SEMO, which use only dominance criterion for survival selection. Recent breakthroughs like the runtime analysis on the most widely used MOEAs, NSGA-II [DPAM02], successfully conducted in [ZLD22, ZD23], have triggered a new era for the theory of MOEAs. Other algorithms that are intensively used in practice, like the NSGA-III [DJ14], SMS-EMOA [BNE07], and SPEA2 [ZLT01], are theoretically analyzed thereafter [WD23, BZLQ23, RBLQ24]. Theory on these popular MOEAs has become a hot topic [BQ22, DQ23, DOSS23b, BZLQ23, DDHW23, WD23, ZD24b, ZLDD24, DZL+24, ZD24a, ODNS24, RBLQ24, DIK25, DZD25, LZD25, DZD25].

Interestingly, despite the rapid progress in the analysis of practical MOEAs, only a few results have demonstrated theoretical advantages of popular MOEAs over simpler algorithms. Dang et

*Equal Contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

al. [DOSS23a] introduced the Bernoulli noise model, and showed that the GSEMO fails badly on every noisy fitness function while NSGA-II can cover the whole Pareto front of noisy LOTZ in polynomial fitness evaluations. Dang et al. [DOS24] proved that three popular MOEAs, i.e. NSGA-II, NSGA-III and SMS-EMOA, enhanced with a mild diversity mechanism (avoiding genotype duplication), require $O(n \log n)$ expected fitness evaluations to cover the whole Pareto front of their designed ONETRAPZEROTRAP, which only has two extremal points as the whole Pareto front. In contrast, simpler algorithm GSEMO requires at least n^n number of fitness evaluations in expectation. Meanwhile, another very recent work [DOS25] constructed an artificial problem with a small Pareto set where almost all pairs of search points are incomparable, also with only two points in the whole Pareto front, and proved that any black-box MOEA using only dominance-based selection and bit-value-invariant variation operators takes exponential time with high probability, while three popular MOEAs, i.e. NSGA-II, NSGA-III, and SMS-EMOA efficiently cover the Pareto front in expected quadratic time.

However, we see that the above results [DOSS23a, DOS24, DOS25] indicating the advantages of the popular MOEAs share the same shortages. They consider the performance for the problem of computing the full Pareto front, (of some algorithms enriched with newly invented mechanisms), and this on newly designed benchmarks. In practice, one cannot know the Pareto front beforehand. The newly invented mechanisms or newly designed benchmarks place the question on the generality of tailored results. Till now, it is still not convincingly proved in theory why popular MOEAs are popular in practice.

Our contributions: This work takes such an attempt to overcome these shortages and fill this gap by analyzing how several popular MOEAs (NSGA-II, NSGA-III, SMS-EMOA, and SPEA2) approximate the Pareto front of the LARGEFRONT$'_\varepsilon$ benchmark (denoted by LF$'_\varepsilon$) proposed in [HN09]. Note that we do not consider MOEA/D here, since it is structurally very different from the domination-based algorithms analyzed in this work and poses additional challenges due to its decomposition mechanism. We prove that, for LF$'_\varepsilon$ with problem size n, these four popular MOEAs achieve an additive ε-approximation of LF$'_\varepsilon$ in expected $O(n^2 \log n)$ number of fitness evaluations (See Theorems 9, 12, 14 and 16). In contrast, existing result [HN09] showed that the GSEMO fails in expected polynomial time (See Theorem 5). We also provide a general theorem of expected $O(\mu n \log n)$ number of fitness evaluations for any MOEA with Property $\mathcal{A}$ to achieve an additive ε-approximation of LF$'_\varepsilon$ (See Theorem 7) where μ denotes the maximum population size. Compared with the GSEMO, which only applies the dominance criterion for survival selection, these popular MOEAs additionally use a criterion to increase the diversity of the survived individuals in the next population. This will result in a better approximation when the number of Pareto front points is large (a property that naturally and widely holds for the Pareto front curve containing a continuous segment in continuous optimization). This provides the first mathematical runtime analysis showing the superiority of popular MOEAs over simpler ones like the GSEMO for the central task of computing good approximations to the Pareto front.

The rest of the paper is organized as follows. Section 2 introduces the approximation measurement and existing LARGEFRONT$_\varepsilon$ benchmark. Section 3 presents a general approximation theorem, and Section 4 applies it to establish the runtime of NSGA-II, NSGA-III, SMS-EMOA, and SPEA2 for a good approximation. Section 5 concludes our paper.

2 Preliminaries

2.1 Additive ε-Approximation

Here are some basic definitions for the maximization of a bi-objective problem $f = (f_1, f_2) : \Omega \to \mathbb{R}^2$ defined on Ω. For $x, y \in \Omega$, we say x *weakly dominates* y, written as $x \succeq y$, if $f_1(x) \geq f_1(y)$ and $f_2(x) \geq f_2(y)$, and x *dominates* y, written as $x \succ y$, if at least one inequality is strict. A solution is a *Pareto optimum* if no other solution dominates it. The *Pareto set* consists of all Pareto optima and the set of corresponding objective values is called the *Pareto front*.

When the Pareto front is unknown beforehand, or the number of Pareto front points is exponential or infinite (like a segment in continuous space), covering the whole Pareto front is infeasible. A good approximation of the Pareto front becomes a natural goal. There are multiple approximation measures, such as ε-dominance [LTDZ02], generational distances [VVL98, BT03, CCRS04], hypervolume indicator [ZT98] or maximal empty interval size [ZD25]. Here we adhere to the original LARGEFRONT$'_\varepsilon$ work [HN09], and use additive ε-approximation (See Definition 1) to evaluate how a set of points approximates the Pareto front. It is built on the additive ε-dominance, first defined in [LTDZ02], that relaxes the usual dominance relation by allowing an additive slack ε in each objective.

Definition 1 ([LTDZ02])**.** *A set of objective vectors T is an additive ε-approximation of $f : \{0,1\}^n \to \mathbb{R}^m$ if and only if for each objective vector $v \in f(\{0,1\}^n)$, there exists at least one objective vector $u \in T$ that additively ε-dominates v, where an objective vector u is called additively ε-dominates v (written as $u \succeq_\varepsilon v$) if and only if $u_i + \varepsilon \geq v_i$ for all $i \in \{1, \ldots, m\}$.*

2.2 LargeFront Benchmark

LargeFront$_\varepsilon$ is a benchmark proposed by [HN08] and [HN09], and contains two variants, LF$_\varepsilon$ [HN08] and LF$'_\varepsilon$ [HN09]. Different from existing benchmarks for theoretical analysis, like COCZ [LTZ04], LOTZ [LTZ04], OJZJ [DZ21], which contain the polynomial number of Pareto front points, both variants have exponential Pareto front points (See Lemma 3). We see its similarity to the continuous optimization as a segment of a continuous curve in the Pareto front contains infinite points. It compensates for the situation where the theory of evolutionary algorithms is majorly built on the discrete space. Since LF$'_\varepsilon$ shows more similarity to the arguably most popular OneMax benchmark, this paper will only discuss it, and we believe our findings will inspire the analyses of MOEAs on the other variant LF$_\varepsilon$. Following is the definition of LF$'_\varepsilon$.

Definition 2 ([HN09])**.** *Let $n \in \mathbb{N}$ be even and $\varepsilon > 0$. The function $LF'_\varepsilon(x) = (LF'_{\varepsilon,1}(x), LF'_{\varepsilon,2}(x)) : \{0,1\}^n \to \mathbb{R}^2$ is defined by*

$$\mathrm{LF}'_{\varepsilon,1}(x) := \begin{cases} \left(2|x'|_1 + 2^{-n/2} BV(x'')\right) \cdot \varepsilon & \min\{|x'|_1, |x'|_0\} \geq \sqrt{n} \\ 2|x'|_1 \cdot \varepsilon & \textit{otherwise} \end{cases}$$
$$\mathrm{LF}'_{\varepsilon,2}(x) := \begin{cases} \left(2|x'|_0 + 2^{-n/2} BV(\overline{x''})\right) \cdot \varepsilon & \min\{|x'|_1, |x'|_0\} \geq \sqrt{n} \\ 2|x'|_0 \cdot \varepsilon & \textit{otherwise} \end{cases},$$

where x' and x'' are the prefix of length $n/2$ and suffix of length $n/2$ of x, $|\cdot|_1$ and $|\cdot|_0$ calculate the number of ones and the number of zeros in this bitstring respectively, and $\mathrm{BV}(y) : \{0,1\}^{n'} \to \mathbb{R}$ is defined by $\mathrm{BV}(y) = \sum_{i=1}^{n'} 2^{n'-i} \cdot y_i$, computing the decimal value of the n'-bit binary number y.

To have an intuitive feeling on this function, Figure 1 plots the objective space of LF'_ε for $\varepsilon = 1$ and $n = 36$.

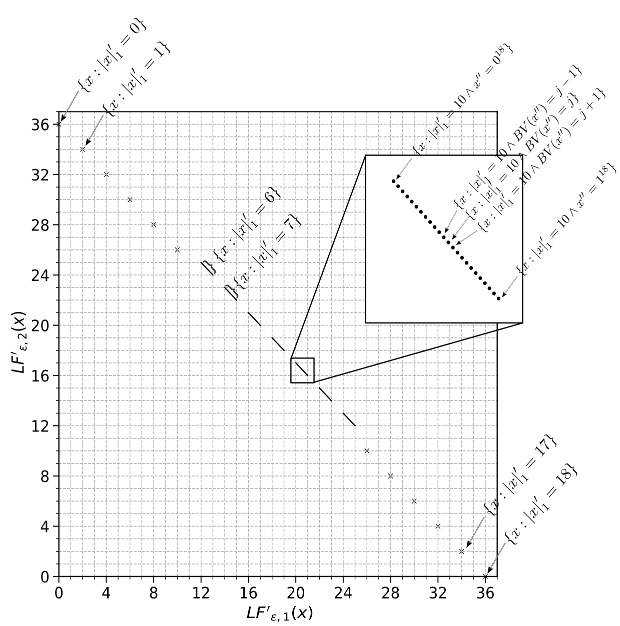

Figure 1: Objective space of LF'_ε for $\varepsilon = 1$ and $n = 36$.

As stated in [HN09], all bitstrings are Pareto optimal. Since there exist solutions x and y such that $|x'|_1 = |y'|_1$, but $x'' \neq y''$ and such solutions can be mutually non-dominated, we could easily see that the size of the Pareto front grows exponentially with the problem size n, also shown in Figure 1. The following lemma collects the results of the Pareto set and the Pareto front w.r.t. LF'_ε from [HN09].

Lemma 3 ([HN09]). *The Pareto set of LF'_ε is $S^* = \{0,1\}^n$, that is, every bitstring of length n is Pareto optimal. The Pareto front is $F^* = \{((2k + 2^{-n/2}\alpha)\varepsilon, (n - 2k + 2^{-n/2}(2^{n/2} - 1 - \alpha))\varepsilon) \mid k \in [0..n/2], \alpha \in [0..2^{n/2} - 1]\}$, where $\alpha = 0$ for $k < \sqrt{n}$ or $k > n/2 - \sqrt{n}$, and for $\sqrt{n} \leq k \leq n/2 - \sqrt{n}$, α ranges over all integers in $[0..2^{n/2} - 1]$.*

The following lemma from [HN09] gives the necessary and sufficient condition to reach an additive ε-approximation w.r.t. LF'_ε.

Lemma 4 ([HN09]). *A set is an additive ε-approximation of LF'_ε if and only if there exists a solution $x \in \{0,1\}^n$ with $|x'|_1 = k$ for each $k \in \{0, \dots, n/2\}$.*

As common in the evolutionary computation theory community [NW10, AD11, Jan13, ZYQ19], the *runtime* usually means the number of fitness evaluations to reach a specific goal. Horoba and Neumann [HN09] proved that the GSEMO fails to achieve an additive ε-approximation of LF'_ε in polynomial runtime (See the following theorem).

Theorem 5 ([HN09]). *The runtime until the GSEMO has achieved an additive ε-approximation of LF'_ε is $2^{\Omega(n^{1/4})}$ with probability $1 - 2^{-\Omega(n^{1/4})}$.*

As stated in the above, in this work, we aim to analyze the runtime guarantees of popular MOEAs required to achieve an additive ε-approximation of LF'_ε. Besides, we use the notation of $[a..b] := \{a, a+1, \dots, b\}$ for $a \leq b$ and $a, b \in \mathbb{Z}$.

3 General Approximation Theorem

Before stepping into specific runtime results of popular MOEAs, this section will introduce a general theorem of the runtime guarantee (to reach an additive ε-approximation w.r.t. LF'_ε) for a general MOEA framework with specific property. It will be then used to prove the runtime of popular MOEAs, in next section, and we believe that it will be useful for future research on LF'_ε. Algorithm 1 states the procedure of the general MOEA framework. The population (the set of solutions) is initialized uniformly at random. In each generation, the algorithm chooses the mating population, generates λ offspring individuals, and then uses a survival selection to determine the next population. We note that this framework with $\lambda = 1$, $p_c = 0$, random parent selection, bit-wise mutation, and dominance-only survival selection gives the GSEMO. The setting of $\lambda = |P_t|$, the survival selection of non-dominated sorting and crowding distance gives the NSGA-II. The NSGA-III corresponds to $\lambda = |P_t|$, survival selection by non-dominated sorting and reference point mechanism. The setting of $\lambda = |P_t|$, the survival selection of non-dominated sorting and hypervolume contribution indicator gives the SMS-EMOA. The SPEA2 employs strength-and-density estimation in its survival selection step.

Algorithm 1: A General MOEA Framework

Initialize P_0 uniformly at random;
for $t = 0, 1, 2, \dots$ **do**
    Choose λ individuals from P_t to form the mating population P'_t;
    Generate offspring population Q_t with λ individuals from P'_t via applying crossover and mutation with crossover rate p_c;
    Generate the next population P_{t+1} from $R_t = P_t \bigcup Q_t$ via a specific survival selection;

Here we extract the following property for the survival selection that will be used for our general theorem.

Definition 6 (Property $\mathcal{A}$). *An MOEA or a survival selection procedure satisfies Property $\mathcal{A}$ on LF'_ε for any time t, if $R_t = P_t \bigcup Q_t$ contains x with $|x'|_1 = k$, then P_{t+1} contains y with $|y'|_1 = k$, where x' (y') is the first half sub-bitstring of x (y).*

Property $\mathcal{A}$ ensures that once the value of k 1-bits in the first half of bitstring is discovered, it will never be lost. Together with Lemma 4, we bound the expected number of fitness evaluations required to achieve an additive ε-approximation of $\mathrm{LF}'_{\varepsilon}$ by $O(\mu n \log n)$ in the following theorem. Due to the space limit, we omit all our proofs and put them into the appendix.

Theorem 7. *Let the crossover rate $p_c \in [0,1)$, μ be the maximum size of parent population P with $\mu > n/2$, and λ the size of offspring population Q with $\lambda = O(\mu)$. Consider using Algorithm 1 with random selection, one-bit mutation or bit-wise mutation to generate P', and survival selection with Property $\mathcal{A}$, to optimize $\mathrm{LF}'_{\varepsilon}$. Then the expected number of fitness evaluations for achieving an additive ε-approximation w.r.t. $\mathrm{LF}'_{\varepsilon}$ is $O(\mu n \log n)$.*

4 Approximation Guarantees for Popular MOEAs

Based on the general approximation theorem (Theorem 7) in the previous section, this section will prove $O(n^2 \log n)$ expected runtime for obtaining an additive ε-approximation w.r.t. $\mathrm{LF}'_{\varepsilon}$, for four widely used MOEAs, NSGA-II, NSGA-III, SMS-EMOA, and SPEA2, all by majorly proving that these popular MOEAs satisfy Property $\mathcal{A}$.

4.1 NSGA-II Using the Current Crowding Distance

The Non-dominated Sorting Genetic Algorithm II (NSGA-II [DPAM02]), is the most widely used MOEA in practice. As stated in the Section 1, Zheng et al. [ZLD22, ZD23] conducted the first mathematical runtime analysis of the NSGA-II, inspiring a series of follow-up studies. Among them, only Zheng and Doerr [ZD22, ZD25] analyzed how the NSGA-II approximates the Pareto front. In these works, they proved the possible difficulties of the original NSGA-II, but proved that a simple modification, like using the current crowding distance in the survival selection [KD06], or a steady-state mode [DNLA09], will result in a near ideal approximation on the Pareto front for ONEMINMAX. Since the proofs are quite similar for these two variants, this work will only discuss the NSGA-II variant with the current crowding distance, and conjecture the similar result of the steady-state variant.

The NSGA-II (See Algorithm 2) fits into the general MOEA framework (Algorithm 1), with fixed $|P_t|$ size of N, offspring population size $\lambda = N$, and special survival selection. The survival selection uses the dominance as the first criterion, preferring the non-dominated ones, and uses the non-dominated sorting to divide the combined population R_t into several fronts $F_1, F_2, \dots$. For the critical front F_{i^*} with $\sum_{i=1}^{i^*-1} |F_i| < N \le \sum_{i=1}^{i^*} |F_i|$, the crowding distance is calculated (See Algorithm 3). The original NSGA-II directly removes $|\bigcup_{i=1}^{i^*} F_i| - N$ individuals with smallest crowding distance in F_{i^*} and selects the remaining ones in F_{i^*}. This strategy only uses the initial crowding distance, and ignores the changes of crowding distance of remaining individuals after each removal. Hence, Kukkonen and Deb [KD06] proposed the survival selection with the current crowding distance and Zheng and Doerr [ZD25] proved its approximation advancing against the original one. Since each removal only affects the crowding distance of four individuals, the update of the crowding distance can be effectively implemented [ZD25].

The following lemma shows that the NSGA-II with current crowding distance satisfies Property $\mathcal{A}$ on $\mathrm{LF}'_{\varepsilon}$ when the population is large enough.

Lemma 8. *Let $N \ge \frac{2n}{3} + 3$. Consider using the NSGA-II with the survival selection based on the current crowding distance to optimize $\mathrm{LF}'_{\varepsilon}$ with problem size n. Assume that at some iteration t, the combined parent and offspring population $R_t = P_t \bigcup Q_t$ contains an individual x with $|x'|_1 = k$, then the next parent population P_{t+1} also contains an individual y with $|y'|_1 = k$.*

With Lemma 8, we then easily apply Theorem 7 to obtain $O(n^2 \log n)$ (when setting $N = \Theta(n)$) expected runtime to reach an additive ε-approximation w.r.t. $\mathrm{LF}'_{\varepsilon}$.

Theorem 9. *Let $N \ge \frac{2n}{3} + 3$ and $p_c \in [0,1)$. Consider using the NSGA-II with the survival selection based on the current crowding distance and employing uniform selection and one-bit mutation or bit-wise mutation to optimize $\mathrm{LF}'_{\varepsilon}$ with problem size n. Then after an expected $O(Nn \log n)$ fitness evaluations, the population reaches an additive ε-approximation w.r.t. $\mathrm{LF}'_{\varepsilon}$.*

Algorithm 2: NSGA-II using current crowding distance [KD06, ZD25]

```
Generate $P_0$ by selecting $N$ solutions uniformly and randomly from $\{0,1\}^n$ with replacement;
for $t = 0, 1, 2, \ldots$ do
    Generate the offspring population $Q_t$ with size $N$;
    Use fast-non-dominated-sort() in [DPAM02] to divide $R_t$ into fronts $F_1, F_2, \ldots$;
    Find $i^* \geq 1$ such that $|\bigcup_{i=1}^{i^*-1} F_i| < N$ and $|\bigcup_{i=1}^{i^*} F_i| \geq N$;
    Use Algorithm 3 to separately calculate the crowding distance of each individual in
      $F_1, \ldots, F_{i^*}$;
    while $|\bigcup_{i=1}^{i^*} F_i| \neq N$ do
        Let $x$ be the individual with the smallest crowding distance in $F_{i^*}$, chosen at random in
          case of a tie;
        Find four neighbors of $x$, two in the sorted list with respect to $f_1$ and two for $f_2$. Update
          the crowding distance of these four neighbors;
        $F_{i^*} = F_{i^*} \setminus \{x\}$;
    $P_{t+1} = \bigcup_{i=1}^{i^*} F_i$
```

Algorithm 3: Computation of the crowding distance $\text{cDis}(S)$ [DPAM02]

```
Input: $S = \{S_1, S_2, \ldots, S_{|S|}\}$
Output: $\text{cDis}(S) = (\text{cDis}(S_1), \text{cDis}(S_2), \ldots, \text{cDis}(S_{|S|}))$, where $\text{cDis}(S_i)$ is the crowding
        distance for $S_i$
$\text{cDis}(S) = (0, \ldots, 0)$;
for each objective $f_i$ do
    Sort $S$ in order of descending $f_i$ value: $S_{i.1}, \ldots, S_{i.|S|}$;
    $\text{cDis}(S_{i.1}) = +\infty$, $\text{cDis}(S_{i.|S|}) = +\infty$;
    for $j = 2, \ldots, |S| - 1$ do
        $\text{cDis}(S_{i.j}) = \text{cDis}(S_{i.j}) + \frac{f_i(S_{i.j-1}) - f_i(S_{i.j+1})}{f_i(S_{i.1}) - f_i(S_{i.|S|})}$;
```

4.2 NSGA-III

The NSGA-II was reported experimentally to encounter difficulties for problems with more objectives (and is recently proven that at least an exponential runtime is needed to cover the full Pareto front for mONEMINMAX, with three and more objectives [ZD24a]). Deb and Jain [DJ14] proposed a new variant called the Non-dominated Sorting Genetic Algorithm III, NSGA-III, to overcome this difficulty. It also uses two criteria for the survival selection, but replaces the second criterion of the crowding distance in NSGA-II by the reference point mechanism. Other components are the same as the NSGA-II, see Algorithm 5.

We now give a brief introduction on the reference point mechanism. After dividing the combined population R_t into serval fronts, all fronts F_i with $i < i^*$ are selected and denoted as Z_t. Following the first theory work of the NSGA-III [WD23], we use the improved and more detailed normalization in [BDR19], by one of the two original authors among others [DJ14]. That is, all individuals in Z_t are normalized by $f_j^n(x) = \frac{f_j(x) - \hat{z}_j^*}{\hat{z}_j^{\text{nad}} - \hat{z}_j^*}$, where $\hat{z}_j^*$ and $\hat{z}_j^{\text{nad}}$ are the ideal point estimate and the Nadir point estimate of objective j. Each normalized individual is then associated with a reference point with the smallest distance. Finally it repeatedly selects the reference point with the fewest already-chosen solutions (breaking ties randomly), then adds the unselected solution closest to that reference point (again breaking ties randomly) until $N - \sum_{i=1}^{i^*-1} |F_i|$ number of solutions are selected. See Algorithm 6 for more details.

The runtime analysis of the NSGA-III starts since 2023, see [WD23, ODNS24, WD24], and all focus on the performance to cover the full Pareto front. Very recently, Deng et al. [DZD25] established the first approximation guarantee of the NSGA-III and proved that the number of reference points is more important than the population size which is suggested important in [ZD22, ZD25]. Till now,

Algorithm 4: Normalization [BDR19]

Input: $F_1, \ldots, F_{i^*}$: non-dominated fronts; $f = (f_1, \ldots, f_m)$: objective function;
$z_j^w \in \mathbb{R}^m$: observed max in each objective; $z_j^* \in \mathbb{R}^m$: observed min in each objective;
$E \subseteq \mathbb{R}^m$: extreme points of previous iteration, initially $\{\infty\}$;

for $j = 1, 2, \ldots, m$ **do**
  $\hat{z}_j^* = \min\{z_j^*, \min_{z \in R_t} f_j(z)\}$;
  $z_j^w = \max\{z_j^w, \max_{z \in R_t} f_j(z)\}$;
  Determine an extreme point $e^{(j)}$ in the j-th objective from $R \bigcup E$ using an achievement scalarization function;
  $E = E \bigcup \{e^{(j)}\}$;
$valid$ = False;
if $e^{(1)}, \ldots, e^{(m)}$ are linearly independent **then**
  $valid$ = True;
  Let H be the hyperplane spanned by $e^{(1)}, \ldots, e^{(m)}$;
  for $j = 1, 2, \ldots, M$ **do**
    Determine the intercept I_j of H with the j-th objective axis;
    if $I_j \geq \epsilon_{\text{nad}}$ and $I_j \leq z_j^w$ **then**
      $\hat{z}_j^{\text{nad}} = I_j$;
    else
      $valid$ = False;
      break;
if $valid$ = False **then**
  for $j = 1, \ldots, M$ **do**
    $\hat{z}_j^{\text{nad}} = \max_{x \in F_1} f_j(x)$;
for $j = 1, 2, \ldots, m$ **do**
  if $\hat{z}_j^{\text{nad}} < \hat{z}_j^* + \epsilon_{\text{nad}}$ **then**
    $\hat{z}_j^{\text{nad}} = \max_{x \in F_1 \bigcup \cdots \bigcup F_{i^*}} f_j(x)$;
Define $f_j^n(x) = \frac{f_j(x) - \hat{z}_j^{\min}}{\hat{z}_j^{\text{nad}} - \hat{z}_j^{\min}}$ for each $x \in \{0,1\}^n$ and $j \in 1, \ldots, m$;

there is no other approximation theory for the NSGA-III. Before we prove that the NSGA-III satisfies Property $\mathcal{A}$, we first show the following lemma that the extremal objective values in the combined population R_t will pass on the next population P_{t+1}. Note that Deng et al. [DZD25] proved the optimal setting of $N = N_r$ for approximating OneMinMax, and note that Deb and Jain [DJ14] suggests $N \approx N_r$ for the general setting. Here we only consider the setting of $N = N_r$. It is not difficult to see from the proofs that our results also hold for $N \geq N_r$.

Algorithm 5: NSGA-III [DJ14]

Generate P_0 by selecting N solutions uniformly and randomly from $\{0,1\}^n$ with replacement;
for $t = 0, 1, 2, \ldots$ **do**
  Generate the offspring population Q_t with size N;
  Use fast-non-dominated-sort() [DPAM02] to divide $R_t = P_t \bigcup Q_t$ into fronts $F_1, F_2, \ldots$;
  Find $i^* \geq 1$ such that $|\bigcup_{i=1}^{i^*-1} F_i| < N$ and $|\bigcup_{i=1}^{i^*} F_i| \geq N$;
  $Z_t = \bigcup_{i=1}^{i^*-1} F_i$;
  Use Algorithm 6 to select $\tilde{F_{i^*}} \subseteq F_{i^*}$ such that $|Z_t \bigcup \tilde{F_{i^*}}| = N$;
  $P_{t+1} = Z_t \bigcup \tilde{F_{i^*}}$;

Lemma 10. *Let $N = N_r \geq 2n + 3$ and a given positive threshold $\epsilon_{nad} \geq n\varepsilon$. Consider using the NSGA-III to optimize LF'_ε with problem size n. Define $z_j^{\min} := \min\{f_j(x) \mid x \in R_t\}$ and*

Algorithm 6: Selection based on a set U of reference points when maximizing f [DJ14]

Input: Z_t: the multi-set of already selected individuals;
$F_t^{i^*}$: the multi-set of individuals to choose from;
f_n: Normalize(f, $Z = Z_t \bigcup F_t^{i^*}$);

Associate each individual $x \in Z_t \bigcup F_t^{i^*}$ to the reference point $rp(x)$ based on the smallest distance to the reference rays;
For each reference point $r \in U$, initialize $\rho_r := |\{x \in Z_t \mid rp(x) = r\}|$;
Initialize $\tilde{F}_t^{i^*} = \emptyset$ and $U' = U$;
while True **do**
  Let $r_{\min} \in U'$ such that $\rho_{r_{\min}}$ is minimal(breaking ties randomly);
  Let $x_{r_{\min}} \in F_t^{i^*} \setminus \tilde{F}_t^{i^*}$ which is associated with $r_{\min}$ and minimizes the distance between $f_n(x_{r_{\min}})$ and $r_{\min}$ (breaking ties randomly);
  if $x_{r_{\min}}$ exists **then**
    $\tilde{F}_t^{i^*} = \tilde{F}_t^{i^*} \bigcup \{x_{r_{\min}}\}$;
    $\rho_{r_{\min}} = \rho_{r_{\min}} + 1$;
    if $|Z_t| + |\tilde{F}_t^{i^*}| = N$ **then**
      return $\tilde{F}_t^{i^*}$
  else
    $U' = U' \setminus \{r_{\min}\}$

$z_j^{\max} := \max\{f_j(x) \mid x \in R_t\}$, $j = 1, 2$. *Then the next parent population P_{t+1} will preserve two individuals x, y such that $f_1(x) = z_1^{\min}$ and $f_1(y) = z_1^{\max}$.*

With Lemma 10, we easily see that once $(0, n\varepsilon)$ and $(n\varepsilon, 0)$ are covered by R_t, they will be covered by the next population. The following lemma even shows that after $(0, n\varepsilon)$ and $(n\varepsilon, 0)$ are covered by R_t, Property $\mathcal{A}$ will be satisfied.

Lemma 11. *Let $N = N_r \geq 2n + 3$ and $\epsilon_{nad} \geq n\varepsilon$. Consider using the NSGA-III to optimize LF'_ε with problem size n. Assume that at some iteration t, the two extreme points $(0, n\varepsilon)$ and $(n\varepsilon, 0)$ are covered by the combined parent and offspring population $R_t = P_t \bigcup Q_t$. If R_t contains an individual x with $|x'|_1 = k$, then the next parent population P_{t+1} also contains an individual $\tilde{x}$ with $|\tilde{x}'|_1 = k$, and covers $(0, n\varepsilon)$ and $(n\varepsilon, 0)$ as well.*

With Lemma 10, it is not difficult to obtain the runtime to cover $(0, n\varepsilon)$ and $(n\varepsilon, 0)$. Then from Lemma 11 that Property $\mathcal{A}$ is satisfied, we use the general approximation theorem (Theorem 7) to obtain $O(n^2 \log n)$ (when setting $N = \Theta(n)$) expected runtime to reach an additive ε-approximation w.r.t. LF'_ε.

Theorem 12. *Let $N = N_r \geq 2n + 3$, $\epsilon_{nad} \geq n\varepsilon$ and $p_c \in [0, 1)$. Consider using the NSGA-III with uniform selection and one-bit mutation or bit-wise mutation to optimize LF'_ε with problem size n. Then after an expected number of $O(Nn \log n)$ fitness evaluations, the population achieves an additive ε-approximation of LF'_ε.*

4.3 SMS-EMOA

The SMS-EMOA [BNE07] can be seen as a steady-state variant of the NSGA-II in which crowding distance is replaced by the hypervolume contribution indicator. In each generation, it generates one offspring and then only removes one individual from R_t. The hypervolume indicator is the most widely used measure for approximation quality in evolutionary multi-objective optimizations [SIHP20]. Given a reference point r, the hypervolume of a population S is calculated as

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

where $\mathcal{L}$ is the Lebesgue measure. The hypervolume contribution of an individual $x \in F_{i^*}$ is defined as $\Delta_r(x, F_{i^*}) := \mathrm{HV}_r(F_{i^*}) - \mathrm{HV}_r(F_{i^*} \setminus \{x\})$ for $x \in F_{i^*}$. Algorithm 7 details the full procedure.

Algorithm 7: SMS-EMOA [BNE07]

Generate P_0 by selecting N solutions uniformly and randomly from $\{0,1\}^n$ with replacement;
for $t = 0, 1, 2, \dots,$ **do**
    Generate one offspring y;
    Use fast-non-dominated-sort() [DPAM02] to divide $R_t = P_t \cup \{y\}$ into $F_1, \dots, F_{i^*}$;
    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 $q \in D$ and $P_{t+1} = R_t \setminus \{q\}$;

It fits into our general MOEA framework (Algorithm 1) with fixed population size of N, offspring population size $\lambda = 1$, and the survival selection based on hypervolume contribution.

Although Bian et al. [BZLQ23] and Zheng and Doerr [ZD24a] have analyzed the runtime of the SMS-EMOA on bi- and many-objective benchmarks, its theoretical approximation performance remains unstudied. Brockhoff et al. [BFN08] proved that the $(\mu+1)$-$SIBEA$ algorithm, the simplified version of the SMS-EMOA without fast non-dominated sorting, achieves a multiplicative ε-approximation of another LargeFront$_\varepsilon$ variant LF_ε in expected $O(\mu n \log n)$ number of fitness evaluations. No approximation theory for the SMS-EMOA on LF'_ε is given. As in the previous sections (also similar to the proof of $(\mu+1)$-$SIBEA$ on LF_ε [BFN08]), we first show that the SMS-EMOA has Property $\mathcal{A}$, w.r.t. LF'_ε.

Lemma 13. *Let $N \geq \frac{n}{2} + 3$ and $r = (r_1, r_2)$ with $r_1 \leq -\varepsilon$, $r_2 \leq -\varepsilon$. Consider using the SMS-EMOA to optimize LF'_ε with problem size n. Assume that at some iteration t the combined parent and offspring population R_t contains an individual x with $|x'|_1 = k$, then the next parent population P_{t+1} contains an individual y that $|y'|_1 = k$.*

Combining Lemma 13 with our general theorem (Theorem 7), we obtain the expected runtime of $O(Nn \log n)$, which is $O(n^2 \log n)$ when $N = \Theta(n)$, required to reach an additive ε-approximation w.r.t. LF'_ε.

Theorem 14. *Let $N \geq n/2 + 3$, $r = (r_1, r_2)$ with $r_1 \leq -\varepsilon$, $r_2 \leq -\varepsilon$ and $p_c \in [0, 1)$. Consider using the SMS-EMOA to optimize LF'_ε using uniform selection and one-bit mutation or bit-wise mutation with problem size n. Then after an expected $O(Nn \log n)$ number of fitness evaluations, the population achieves an additive ε-approximation w.r.t. LF'_ε.*

4.4 SPEA2

The SPEA2 algorithm [ZLT01] is one of the most popular MOEA. In survival selection at generation t, it creates a new parent population P_{t+1} by selecting all non-dominated solutions from R_t. If $|P_{t+1}|$ is smaller than the population size N, it is then supplemented with the best dominated individuals determined by the strength and density estimates. For an individual $u \in \{0,1\}^n$, the strength and density estimate is defined as $F(u) = R(u) + D(u)$, where $R(u) = \sum_{v \in R_t, v \succ u} S(v)$ of which $S(v)$ denotes the number of solutions it dominates, and $D(u) = \frac{1}{\sigma_u^k + 2}$ of which σ_u^k denotes the Euclidean distance (in objective space) of the individual u to its k-th nearest neighbor in R_t with $k = \sqrt{N + \lambda}$. If the number of non-dominated individuals exceeds the population size N, a truncation operator is used to iteratively remove solutions from P_{t+1} until $|P_{t+1}| = N$. At each removal, individual u is chosen for removal with $u \leq_d v$ for all $v \in P_{t+1}$ where

$$\begin{aligned} u \leq_d v :\Leftrightarrow & \forall 0 < k < |P_{t+1}| : \sigma_u^k = \sigma_v^k \vee \\ & \exists 0 < k < |P_{t+1}| : [(\forall 0 < l < k : \sigma_u^l = \sigma_v^l) \wedge \sigma_u^k < \sigma_v^k]. \end{aligned}$$

In other words, at each removal, it removes the individual with the smallest nearest-neighbor distance and ties are broken by comparing their second-nearest distances and so forth. Once a solution is removed, its distances to other solutions are no longer considered. See Algorithm 8 for more details. The SPEA2 fits into the general MOEA framework (Algorithm 1) with uniform parent selection, and the survival selection based on strength-and-density estimation.

The first runtime analysis of the SPEA2 was proposed very recently [RBLQ24], where they proved the runtime bounds for the SPEA2 on three commonly used multi-objective problems, i.e., mOneMinMax, mLOTZ, and mOJZJ. Prior work by Horoba and Neumann [HN09] studied the

Algorithm 8: SPEA2[[ZLT01]]

$Q_0 \leftarrow \lambda$ solutions uniformly and randomly selected from $\{0,1\}^n$ with replacement and $P_0 \leftarrow \emptyset$;
for $t = 0, 1, 2, \dots$ **do**
  $P_{t+1} \leftarrow$ non-dominated solutions in $R_t = P_t \cup Q_t$;
  if $|P_{t+1}| > N$ **then**
    Reduce P_{t+1} by means of the truncation operator;
  else if $|P_{t+1}| < N$ **then**
    Fill P_{t+1} with dominated individuals in R_t;
  for $i = 0, 1, 2, \dots, \lambda$ **do**
    Generate one offspring x';
    $Q_{t+1} \leftarrow Q_{t+1} \cup \{x'\}$;

approximation performance of RADEMO, a simplified version of SPEA2, for solving LF'_ε. Till now, no approximation guarantee for the SPEA2 on LF'_ε is given. Same as the previous sections, we first prove that the SPEA2 also maintains Property $\mathcal{A}$ required for our general approximation theorem.

Lemma 15. *Let $N \geq n/2 + 2$. Consider using the SPEA2 to optimize LF'_ε with problem size n. If at some iteration, the combined population R_t contains an individual x with $|x'|_1 = k$, then the next population P_{t+1} will also include an individual y with $|y'|_1 = k$.*

With Lemma 15, we derive an expected runtime of $O(n^2 \log n)$ to reach an additive ε-approximation w.r.t. LF'_ε, by setting $N = \Theta(n)$ in our general approximation theorem.

Theorem 16. *Let $N \geq n/2 + 2$ and $p_c \in [0, 1)$. Consider using the SPEA2 with uniform selection and one-bit mutation or bit-wise mutation to optimize LF'_ε with problem size n. Then after an expected $O(Nn \log n)$ number of fitness evaluations, the population achieves an additive ε-approximation w.r.t. LF'_ε.*

5 Conclusion and Discussion

The question of why popular MOEAs are popular in practice was not convincingly answered in theory, as the few results indicating their advantage only considered the performance to cover the full Pareto front on newly designed benchmarks. This work tackled this question by considering the approximation ability of several popular MOEAs on the established $\textsc{LargeFront}'_\varepsilon$ benchmark. In contrast to $2^{\Omega(n^{1/4})}$ number of fitness evaluations (with high probability) for the GSEMO to reach an additive ε-approximation w.r.t. LF'_ε, we gave a general theorem of polynomial runtime for any MOEA with Property $\mathcal{A}$, and proved $O(n^2 \log n)$ expected runtime for four widely used MOEAs, i.e., NSGA-II, NSGA-III, SMS-EMOA, and SPEA2. The reason is the second criterion of these popular MOEAs maintains a good diversity in the survival selection, compared with the GSEMO that relies only on the dominance criterion. This is the first mathematical runtime analysis showing and explaining the superiority of popular MOEAs over simpler ones like the GSEMO for the central task of computing good approximations to the Pareto front.

Although our results might also indicate the advantages in approximation for other benchmark with large number of Pareto front points (a property that naturally holds for Pareto front curve containing a continuous segment in continuous optimization), a more thorough and rigorous analysis on a more general benchmark classes will make our theoretical findings more appreciated, and we are optimistic and shall try to address in our future work.

Acknowledgments and Disclosure of Funding

This work was supported by National Natural Science Foundation of China (Grant No. 62306086, 62350710797), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2025A1515011936), Xinjiang Tianshan Innovative Research Team (2025D14009), and Science, Technology and Innovation Commission of Shenzhen Municipality (Grant No. GXWD20220818191018001). This research benefited from the support of the FMJH Program PGMO.

References

[AD11] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics*. World Scientific Publishing, 2011.

[BDR19] Julian Blank, Kalyanmoy Deb, and Proteek Chandan Roy. Investigating the normalization procedure of nsga-iii. In *International conference on evolutionary multi-criterion optimization*, pages 229–240. Springer, 2019.

[BFN08] Dimo Brockhoff, Tobias Friedrich, and Frank Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature, PPSN 2008*, pages 651–660. Springer, 2008.

[BNE07] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181:1653–1669, 2007.

[BQ22] Chao Bian and Chao Qian. Better running time of the non-dominated sorting genetic algorithm II (NSGA-II) by using stochastic tournament selection. In *Parallel Problem Solving From Nature, PPSN 2022*, pages 428–441. Springer, 2022.

[BT03] Peter A.N. Bosman and Dirk Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7:174–188, 2003.

[BZLQ23] Chao Bian, Yawen Zhou, Miqing Li, and Chao Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 5513–5521. ijcai.org, 2023.

[CCRS04] Carlos A Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004. Proceedings 3*, pages 688–697. Springer, 2004.

[DDHW23] Matthieu Dinot, Benjamin Doerr, Ulysse Hennebelle, and Sebastian Will. Runtime analyses of multi-objective evolutionary algorithms in the presence of noise. In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 5549–5557. ijcai.org, 2023.

[DIK25] Benjamin Doerr, Tudor Ivan, and Martin S. Krejca. Speeding up the NSGA-II with a simple tie-breaking rule. In *Conference on Artificial Intelligence, AAAI 2025*, pages 26964–26972. AAAI Press, 2025.

[DJ14] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18:577–601, 2014.

[DNLA09] Juan J Durillo, Antonio J Nebro, Francisco Luna, and Enrique Alba. On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In *Evolutionary Multi-Criterion Optimization: 5th International Conference, EMO 2009, Nantes, France, April 7-10, 2009. Proceedings 5*, pages 183–197. Springer, 2009.

[DOS24] Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Illustrating the efficiency of popular evolutionary multi-objective algorithms using runtime analysis. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 484–492, 2024.

[DOS25] Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Why dominance is not enough: Lessons from practical evolutionary multi-objective algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2025*, pages 1604–1612. ACM, 2025.

[DOSS23a] Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. Analysing the robustness of NSGA-II under noise. In *Genetic and Evolutionary Computation Conference, GECCO 2023*, pages 642–651. ACM, 2023.

[DOSS23b] Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Conference on Artificial Intelligence, AAAI 2023*, pages 12390–12398. AAAI Press, 2023.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[DQ23] Benjamin Doerr and Zhongdi Qu. A first runtime analysis of the NSGA-II on a multimodal problem. *IEEE Transactions on Evolutionary Computation*, 27:1288–1297, 2023.

[DZ21] Benjamin Doerr and Weijie Zheng. Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In *Conference on Artificial Intelligence, AAAI 2021*, pages 12293–12301. AAAI Press, 2021.

[DZD25] Renzhong Deng, Weijie Zheng, and Benjamin Doerr. The first theoretical approximation guarantees for the non-dominated sorting genetic algorithm III (NSGA-III). In *International Joint Conference on Artificial Intelligence, IJCAI 2025*, pages 8867–8875. ijcai.org, 2025.

[DZL+24] Renzhong Deng, Weijie Zheng, Mingfeng Li, Jie Liu, and Benjamin Doerr. Runtime analysis for state-of-the-art multi-objective evolutionary algorithms on the subset selection problem. In Michael Affenzeller, Stephan M. Winkler, Anna V. Kononova, Heike Trautmann, Tea Tusar, Penousal Machado, and Thomas Bäck, editors, *Parallel Problem Solving from Nature, PPSN 2024, Part III*, pages 264–279. Springer, 2024.

[HN08] Christian Horoba and Frank Neumann. Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization. In *Genetic and Evolutionary Computation Conference, GECCO 2008*, pages 641–648. ACM, 2008.

[HN09] Christian Horoba and Frank Neumann. Additive approximations of pareto-optimal sets by evolutionary multi-objective algorithms. In *Foundations of Genetic Algorithms, FOGA 2009*, pages 79–86. ACM, 2009.

[Jan13] Thomas Jansen. *Analyzing Evolutionary Algorithms – The Computer Science Perspective*. Springer, 2013.

[KD06] Saku Kukkonen and Kalyanmoy Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *Conference on Evolutionary Computation, CEC 2006*, pages 1179–1186. IEEE, 2006.

[LTDZ02] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10:263–282, 2002.

[LTZ+02] Marco Laumanns, Lothar Thiele, Eckart Zitzler, Emo Welzl, and Kalyanmoy Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving from Nature, PPSN 2002*, pages 44–53. Springer, 2002.

[LTZ04] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8:170–182, 2004.

[LZD25] Mingfeng Li, Weijie Zheng, and Benjamin Doerr. Scalable speed-ups for the SMS-EMOA from a simple aging strategy. In *International Joint Conference on Artificial Intelligence, IJCAI 2025*, pages 8885–8893. ijcai.org, 2025.

[NW10] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer, 2010.

[ODNS24] Andre Opris, Duc Cuong Dang, Frank Neumann, and Dirk Sudholt. Runtime analyses of NSGA-III on many-objective problems. In *Genetic and Evolutionary Computation Conference, GECCO 2024*, pages 1596–1604. ACM, 2024.

[RBLQ24] Shengjie Ren, Chao Bian, Miqing Li, and Chao Qian. A first running time analysis of the Strength Pareto Evolutionary Algorithm 2 (SPEA2). In *Parallel Problem Solving from Nature, PPSN 2024, Part III*, pages 295–312. Springer, 2024.

[SIHP20] Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20, 2020.

[VVL98] David A Van Veldhuizen and Gary B Lamont. Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, pages 221–228. Stanford University Bookstore, 1998.

[WD23] Simon Wietheger and Benjamin Doerr. A mathematical runtime analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). In *International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 5657–5665. ijcai.org, 2023.

[WD24] Simon Wietheger and Benjamin Doerr. Near-tight runtime guarantees for many-objective evolutionary algorithms. In *Parallel Problem Solving from Nature, PPSN 2024, Part IV*, pages 153–168. Springer, 2024.

[ZD22] Weijie Zheng and Benjamin Doerr. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, pages 611–619. ACM, 2022.

[ZD23] Weijie Zheng and Benjamin Doerr. Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II). *Artificial Intelligence*, 325:104016, 2023.

[ZD24a] Weijie Zheng and Benjamin Doerr. Runtime analysis for the NSGA-II: proving, quantifying, and explaining the inefficiency for many objectives. *IEEE Transactions on Evolutionary Computation*, 28:1442–1454, 2024.

[ZD24b] Weijie Zheng and Benjamin Doerr. Runtime analysis of the SMS-EMOA for many-objective optimization. In *Conference on Artificial Intelligence, AAAI 2024*, pages 20874–20882. AAAI Press, 2024.

[ZD25] Weijie Zheng and Benjamin Doerr. Approximation guarantees for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). *IEEE Transactions on Evolutionary Computation*, 29:891–905, 2025.

[ZLD22] Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Conference on Artificial Intelligence, AAAI 2022*, pages 10408–10416. AAAI Press, 2022.

[ZLDD24] Weijie Zheng, Mingfeng Li, Renzhong Deng, and Benjamin Doerr. How to use the Metropolis algorithm for multi-objective optimization? In *Conference on Artificial Intelligence, AAAI 2024*, pages 20883–20891. AAAI Press, 2024.

[ZLT01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK report*, 103, 2001.

[ZT98] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *Parallel Problem Solving from Nature, PPSN 1998*, pages 292–301. Springer, 1998.

[ZYQ19] Zhi-Hua Zhou, Yang Yu, and Chao Qian. *Evolutionary Learning: Advances in Theories and Algorithms*. Springer, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main contribution is that we establish the first mathematical runtime analysis showing the superiority of popular MOEAs over the simpler ones like the GSEMO for the central task of computing good approximations to the Pareto front. Our technical claim is that all four widely used MOEAs (NSGA-II, NSGA-III, SMS-EMOA, SPEA2) achieve an additive ε-approximation of the LF'_ε benchmark in expected $O(n^2 \log n)$ number of fitness evaluations, contrasting the existing result of the exponential lower bound for the GSEMO. This claim accurately reflects our contribution based on the background introduction and literature review in abstract and introduction.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 5. Although our results might also indicate the advantages in approximation for other benchmark with large number of Pareto front points (a property that naturally holds for Pareto front curve containing a continuous segment in continuous optimization), a more thorough and rigorous analysis on a more general benchmark classes will make our theoretical findings more appreciated, and we are optimistic and shall try to address our future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All the theorems, lemmas, and proofs in this paper are numbered, cross-referenced. All required assumptions are stated precisely, e.g., Theorem 7 specifies the Property $\mathcal{A}$ and the bounds on μ, λ, p_c. Due to the page limit, complete, formal proofs are available in the supplementary material. Moreover, all intermediate results and classical lemmas invoked, e.g., Lemma 4, are cited with precise references.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper is a theory paper and does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: The paper does not include experiments requiring code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have read and strictly follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work is purely theoretical, deriving approximation guarantees for widely used MOEAs on $\mathrm{LF}'_{\varepsilon}$. While these insights may eventually deepen our understanding of algorithm behavior and have potential societal consequences, none of which we feel must be explicitly emphasized here. The main contribution of this paper is to advance the theoretical understanding of popular MOEAs.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.