One for Exploration and Another for Exploitation: A Dual-Population MOEA Framework with Provable Benefits

Chenglin Jiang[1] , **Shengjie Ren**[1] , **Zimin Liang**[2] , **Miqing Li**[2] and **Chao Qian**[1]

[1]State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210023, China
School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
[2]School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K.
qianc@nju.edu.cn

Abstract

Evolutionary Algorithms (EAs) are currently the most popular tool for solving multi-objective optimization problems. Balancing exploration and exploitation is fundamental to the performance of Multi-Objective EAs (MOEAs). Achieving this requires maintaining a set of high-quality solutions for effective exploitation while simultaneously enabling exploration of diverse regions of the search space. In MOEAs, these two roles are typically fulfilled by a single evolutionary population. However, high-quality solutions that are promising for exploitation may lack the potential to guide the search toward different regions of the search space. Conversely, solutions with high exploratory potential that can lead the search toward more promising regions may themselves be of low quality. To address this issue, this paper proposes a dual-population MOEA with Exploration and Exploitation Decoupled (MOEA/EED): the exploration population uses a simple aging mechanism, while the exploitation population preserves the currently optimal solutions. We theoretically prove the benefits of MOEA/EED, i.e., it achieves the upper bounds, $O(n^{k+1} \cdot \min\{1, (e\ln(3e)/k)^k\})$ and $O((5e\ln(3e))^{n/5})$, on the expected running time for solving two commonly studied bi-objective problems OneJumpZeroJump and bi-objective RealRoyalRoad, which are significantly better (even exponentially smaller) than that of the popular SMS-EMOA, NSGA-II, and their non-elitist variants. Lastly, we empirically show its effectiveness on a wide range of practical optimization problems.

1 Introduction

Multi-objective Optimization Problems (MOPs) refer to optimization scenarios involving multiple, often conflicting objectives that need to be optimized simultaneously. Unlike single-objective optimization, an MOP usually has no single optimal solution, but instead a set of Pareto optimal solutions representing different trade-offs among the objectives [Deb, 2001; Zhou *et al.*, 2019]. The corresponding objective vectors of these Pareto optimal solutions constitute the Pareto front. The goal of a multi-objective optimizer is to find the Pareto front or approximate it as accurately as possible. Evolutionary Algorithms (EAs), inspired by natural evolution and operating with populations, are particularly well-suited to MOPs and have been widely applied in real-world domains [Deb, 2001; Zhou *et al.*, 2019]. Over the years, numerous Multi-Objective EAs (MOEAs) have been developed, including the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb *et al.*, 2002], MOEA based on Decomposition (MOEA/D) [Zhang and Li, 2007], and $\mathcal{S}$-Metric Selection Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) [Beume *et al.*, 2007].

As in many stochastic optimization processes, balancing exploration and exploitation is fundamental to the performance of MOEAs. To support effective exploration, the population in MOEAs needs to maintain a diverse set of solutions capable of guiding the search towards different regions of the search space. At the same time, effective exploitation requires the population to retain high-quality solutions found so far, enabling intensive local search for further improvement.

However, these two roles are not always fully aligned. Solutions with high exploratory potential may be of low quality - for example, they lie near the middle of multiple attraction basins. In contrast, high-quality solutions may lack the potential to guide the search toward different regions of the search space - for example, they are located in a local optimal region. As such, using a single population in MOEAs may not always help. This is echoed by recent studies showing that introducing randomness, non-elitism or aging mechanisms into population update of MOEAs (thereby allowing inferior solutions to survive longer) can benefit the search, both theoretically [Bian *et al.*, 2025; Zheng and Doerr, 2024b; Opris, 2025a; Li *et al.*, 2025] and empirically [Tanabe and Ishibuchi, 2019; Sanchez *et al.*, 2023; Liang *et al.*, 2023]. And, adding an external archive to store the non-dominated solutions (despite not being used for exploitation [Li *et al.*, 2024b]) can further enhance exploration by freeing the evolutionary population from the role of preserving solution quality [Ishibuchi *et al.*, 2020; Ren *et al.*, 2025].

In this work, we propose a simple yet effective algorithmic framework that alleviates the conflict between exploration and exploitation by decoupling these two aspects, i.e., a dual-population MOEA with Exploration and Exploitation Decoupled (MOEA/EED). Specifically, we use a probability p to

		OneJumpZeroJump	**Bi-objective RealRoyalRoad**
SMS-EMOA	Original algorithm	$O(\mu n^k)$ [Bian *et al.*, 2025] $[\mu \geq n-2k+3]$	$\Omega(n^{n/5-1})$ [Bian *et al.*, 2025] $[\mu = poly(n)]$
	Stochastic method	$O(\mu n^k \cdot \min\{1, (e\ln(2e\mu)/k)^{k-1}\})$ $[\mu \geq 2(n-2k+4)]$ [Ren *et al.*, 2025]	$O(\mu^2 n^{n/5+1/2}/2^{n/10})$ [Bian *et al.*, 2025] $[\mu \geq 2(2n/5+2)]$
	Aging method	$O(k\tau(e^2 n/(k(1-e^{-\tau/\mu})))^k)$ $[\mu \geq n-2k+4+\tau]$ [Li *et al.*, 2025]	–
NSGA-II	Original algorithm	$O(\mu n^k)$ [Doerr and Qu, 2023] $[\mu \geq 4(n-2k+3)]$	$\Omega(n^{n/5-1})$ [Bian *et al.*, 2025] $[\mu = poly(n)]$
	Stochastic method	$O(\mu n^k \cdot \min\{1, (e\ln(4e)/k)^{k-1}\})$ $[\mu \geq 8(n-2k+3)]$ [Ren *et al.*, 2025]	$O(\mu\sqrt{n}(20e^2)^{n/5})$ [Bian *et al.*, 2025] $[\mu \geq 8(2n/5+1)]$
MOEA/EED		$O(n^{k+1} \cdot \min\{1, (e\ln(3e)/k)^k\})$ [Thm 1]	$O((5e\ln(3e))^{n/5})$ [Thm 2]

Table 1: The expected running time (i.e., number of fitness evaluations) of MOEA/EED and different versions of SMS-EMOA and NSGA-II for solving OneJumpZeroJump and bi-objective RealRoyalRoad, where n denotes the problem size, k denotes the parameter of OneJumpZeroJump ($2 \leq k < n/2$), μ denotes the population size, and τ denotes the maximum lifespan for the aging method. The results of MOEA/EED are obtained under the condition of $p = 1/2$, where p denotes the probability of selecting a parent solution from the exploitation population. The expected running time of the aging-based NSGA-II on the RealRoyalRoad problem has not yet been analyzed so far.

decide to exploit (select a parent solution from the exploitation population P_{exploit}) or explore (select a parent solution from the exploration population P_{explore}). The exploitation population P_{exploit} stores only non-dominated solutions with unique objective vectors; the exploration population P_{explore} uses an aging mechanism, meaning that solutions are preserved for a fixed duration regardless of their quality. Both populations are updated after a new solution is generated.

We analyze the expected running time of the proposed MOEA/EED on two commonly studied bi-objective problems OneJumpZeroJump [Doerr and Zheng, 2021] and bi-objective RealRoyalRoad [Dang *et al.*, 2024], and obtain a significantly better upper bound compared to previously analyzed single-population algorithms. The results are shown in Table 1. The original SMS-EMOA [Beume *et al.*, 2007] and NSGA-II [Deb *et al.*, 2002] serve as the baselines for comparison, as they do not incorporate any non-elitist mechanisms. Stochastic population update (stochastic method) is an approach that introduces randomness into the population update process of MOEAs (randomly selects a fraction of the current population and the offspring solution(s) to directly survive into the next generation), which has been theoretically shown to enhance the search capability [Bian *et al.*, 2025; Zheng and Doerr, 2024b; Ren *et al.*, 2025]. Very recently, Li *et al.* [2025] analytically showed that SMS-EMOA equipped with an aging method (where solutions increment their age by 1 in every generation, and are excluded from population updates until their age reaches a threshold τ) achieves greater acceleration. The main comparison results are as follows.

- On the OneJumpZeroJump problem, the expected running time of SMS-EMOA and NSGA-II is both $O(\mu n^k)$ [Bian *et al.*, 2025; Doerr and Qu, 2023]. SMS-EMOA combined with the stochastic method achieves a speedup of $\Theta(\max\{1, (k/(e\ln(2e\mu)))^{k-1}\})$ [Ren *et al.*, 2025]; when using an aging method (with $\tau = n-2k+4$ and $\mu = 2\tau$), it further reduces the upper bound by a factor of $\Theta(((1-e^{-1/2})\ln(2e\mu)/e)^{k-1})$ [Li *et al.*, 2025]. On the other hand, NSGA-II that employs the $(\mu+\mu)$ update mode and the stochastic method achieves an upper bound comparable to that of the aging-based SMS-EMOA. By decoupling exploration and exploitation with two distinct populations, our proposed MOEA/EED achieves the best expected running time upper bound, faster than the stochastic NSGA-II by a factor of $\Theta(k \cdot (\log_{3e}(4e))^{k-1})$.
- On the bi-objective RealRoyalRoad problem, the expected running time of SMS-EMOA and NSGA-II is both $\Omega(n^{n/5-1})$ [Bian *et al.*, 2025]. SMS-EMOA combined with the stochastic method achieves a speedup of $\Omega(2^{n/10}/(\mu^2 n^{3/2}))$, while NSGA-II with the stochastic method further reduces the upper bound by a factor of $\Theta(\mu(n/(20\sqrt{2}e^2))^{n/5})$. Our proposed MOEA/EED can further reduce the upper bound of the stochastic NSGA-II by a factor of $\Theta(\mu\sqrt{n}(4e/\ln(3e))^{n/5})$.

The above proved (even exponential) speedup of MOEA/EED over the stochastic and aging-based methods stems from the fact that the latter operate in the search process of a single population. Consequently, the only way to increase the intensity of exploration in inferior, yet potentially promising regions is by increasing the proportion of non-elite solutions in the population. Although some non-elite solutions are helpful, most of them offer little exploratory value, and an excessive number of such solutions waste search resources. In contrast, MOEA/EED decouples exploration and exploitation by maintaining two distinct populations and uses a selection probability to directly determine the intensity of exploration in inferior regions. This allows it to allocate a reasonably sized non-elite population for exploration, thereby reducing the chance of selecting useless individuals and leading to a more efficient search. The advantage of MOEA/EED is also validated on well-known practical problems at the end of this paper.

The main contribution of this work is proposing the framework of MOEAs with exploration and exploitation decoupled, and showing that even a simple instantiation can work well both theoretically and empirically. Note that the use of dual-population or dual-archive mechanism (where the archive is used to generate solutions) is not uncommon in empirical studies of MOEAs (see [Li *et al.*, 2024b] for a survey). However, existing methods do not explicitly separate exploration and exploitation by assigning each role to a dedicated population. Instead, they employ multiple populations for other purposes: for example, considering a different selection criterion each (e.g., Pareto dominance, indicator or decomposition criteria [Li *et al.*, 2018]), or introducing an additional population to address problem-specific characteristics such as constraints [Xie *et al.*, 2025] and multi-modality [Liu *et al.*, 2018], alongside the main population.

Finally, we provide a brief overview of running time analyses of MOEAs, which have received growing interests over the last two decades. Early theoretical works [Laumanns *et al.*, 2004; Neumann, 2007; Giel and Lehre, 2010; Doerr *et al.*, 2013; Qian *et al.*, 2013; Qian *et al.*, 2016; Bian *et al.*, 2018] primarily focused on simple MOEAs such as (G)SEMO, showing results on their expected running time. Recently, attention has shifted toward practical, widely used MOEAs. Huang *et al.* [2021] analyzed MOEA/D and evaluated the impact of different decomposition approaches. Zheng *et al.* [2022] conducted the first runtime analysis of NSGA-II. Bian *et al.* [2025] studied SMS-EMOA and showed that stochastic population update can yield significant acceleration. Wietheger and Doerr [2023] showed that NSGA-III outperforms NSGA-II on a tri-objective problem. Ren *et al.* [2024a] and Opris [2025b] provided a first runtime analysis of SPEA2 and PAES-25, respectively. Additional theoretical studies on established MOEAs include [Zheng and Doerr, 2024a; Bian *et al.*, 2024; Dang *et al.*, 2024; Doerr *et al.*, 2025; Benjamin *et al.*, 2025; Opris, 2025b].

2 Preliminaries

In this section, we first introduce the basic concepts of multi-objective optimization, followed by a description of the bi-objective problems OneJumpZeroJump and bi-objective RealRoyalRoad, studied in this paper.

Multi-objective optimization aims to simultaneously optimize two or more typically conflicting objective functions. Without a canonical complete order in the solution space $\mathcal{X}$, solutions are compared using the *domination* relationship in Definition 1. A solution is *Pareto optimal* if no other solution in $\mathcal{X}$ dominates it. The set of objective vectors corresponding to all Pareto optimal solutions constitutes the *Pareto front*. The goal of multi-objective optimization is to find the Pareto front or its good approximation.

Definition 1. *Let* $\boldsymbol{f} = (f_1, f_2, \ldots, f_m) : \mathcal{X} \to \mathbb{R}^m$ *be the objective vector. For two solutions* $\boldsymbol{x}$ *and* $\boldsymbol{y} \in \mathcal{X}$*:*

- $\boldsymbol{x}$ weakly dominates $\boldsymbol{y}$ *(denoted as* $\boldsymbol{x} \succeq \boldsymbol{y}$*) if for any* $1 \le i \le m, f_i(\boldsymbol{x}) \ge f_i(\boldsymbol{y})$*;*
- $\boldsymbol{x}$ dominates $\boldsymbol{y}$ *(denoted as* $\boldsymbol{x} \succ \boldsymbol{y}$*) if* $\boldsymbol{x} \succeq \boldsymbol{y}$ *and* $f_i(\boldsymbol{x}) > f_i(\boldsymbol{y})$ *for some* i*;*
- $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *are* incomparable *if neither* $\boldsymbol{x} \succeq \boldsymbol{y}$ *nor* $\boldsymbol{y} \succeq \boldsymbol{x}$*.*

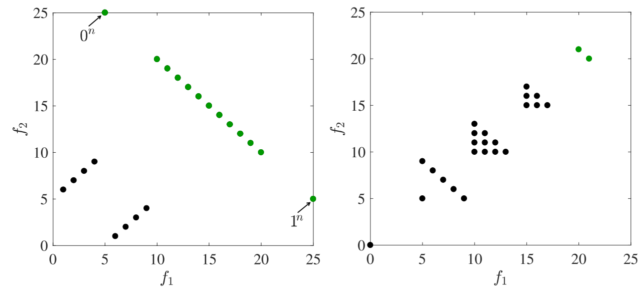

Figure 1: Illustration of the objective vectors of solutions in the (f_1, f_2) objective space for the OneJumpZeroJump problem with $n = 20$ and $k = 5$ (left subfigure), and the bi-objective RealRoyalRoad problem with $n = 5$ (right subfigure).

Next, we introduce the benchmark problems OneJumpZeroJump and bi-objective RealRoyalRoad studied in this paper. The OneJumpZeroJump problem as presented in Definition 2 is constructed based on the Jump problem [Doerr and Neumann, 2020], and has been widely used in MOEAs' theoretical analyses [Doerr and Zheng, 2021; Doerr and Qu, 2023; Lu *et al.*, 2024; Ren *et al.*, 2024b]. Its first objective is the same as the Jump problem, which aims to maximize the number of 1-bits of a solution, except for a valley around 1^n (the solution with all 1-bits) where the number of 1-bits should be minimized. The second objective is isomorphic to the first one, with the roles of 1-bits and 0-bits reversed.

Definition 2 (OneJumpZeroJump [Doerr and Zheng, 2021])**.** *The OneJumpZeroJump problem is to find* n *bits binary strings which maximize*

$$f_1(\boldsymbol{x}) = \begin{cases} k + |\boldsymbol{x}|_1, & \text{if } |\boldsymbol{x}|_1 \le n - k \text{ or } \boldsymbol{x} = 1^n, \\ n - |\boldsymbol{x}|_1, & \text{else}, \end{cases}$$

$$f_2(\boldsymbol{x}) = \begin{cases} k + |\boldsymbol{x}|_0, & \text{if } |\boldsymbol{x}|_0 \le n - k \text{ or } \boldsymbol{x} = 0^n, \\ n - |\boldsymbol{x}|_0, & \text{else}, \end{cases}$$

where $k \in \mathbb{Z} \wedge 2 \le k < n/2$*, and* $|\boldsymbol{x}|_1$ *and* $|\boldsymbol{x}|_0$ *denote the number of 1-bits and 0-bits in* $\boldsymbol{x} \in \{0,1\}^n$*, respectively.*

The left subfigure of Figure 1 illustrates the objective vectors of OneJumpZeroJump in the (f_1, f_2) objective space, where the green points represent Pareto-optimal solutions and the black points denote dominated solutions. A solution is Pareto optimal if and only if it contains $(a - k)$ 1-bits, for $a \in [2k..n] \cup \{k, n+k\}$; the corresponding objective vector is $(a, n + 2k - a)$. Consequently, the Pareto front is given by$\{(a, n + 2k - a) \mid a \in [2k..n] \cup \{k, n + k\}\}$, whose size is $n - 2k + 3$. We denote the inner part of the Pareto front by $F_I^* = \{(a, n + 2k - a) \mid a \in [2k..n]\}$. As observed, the Pareto front consists of F_I^* and two isolated extreme points corresponding to 0^n and 1^n. The OneJumpZeroJump problem serves as a representative benchmark where some adjacent Pareto optimal solutions in the objective space locate far away in the decision space.

The bi-objective RealRoyalRoad problem, formally defined in Definition 3, is originally proposed in [Dang *et al.*, 2024] to evaluate the ability of MOEAs to escape from local optima. It consists of two objectives that combine the number of 1-bits with structural information captured by the number of trailing zeros (TZ) and leading zeros (LZ), where $\mathrm{TZ}(\boldsymbol{x}) = \sum_{i=1}^{n} \prod_{j=i}^{n}(1 - x_j)$ and $\mathrm{LZ}(\boldsymbol{x}) = \sum_{i=1}^{n} \prod_{j=1}^{i}(1 - x_j)$. The solution space is partitioned into two special regions G and H. The set G contains all solutions with at most $3n/5$ 1-bits, while the set H consists of all solutions with exactly $4n/5$ 1-bits and satisfying $\mathrm{LZ}(\boldsymbol{x}) + \mathrm{TZ}(\boldsymbol{x}) = n/5$. For solutions in G or H, $f_1(\boldsymbol{x})$ is defined as the sum of $n \cdot |\boldsymbol{x}|_1$ and the number of trailing zeros $\mathrm{TZ}(\boldsymbol{x})$, whereas all remaining solutions receive objective value 0. $f_2(\boldsymbol{x})$ is defined analogously, replacing TZ with LZ.

Definition 3 (Bi-objective RealRoyalRoad [Dang *et al.*, 2024])**.** *The bi-objective RealRoyalRoad problem is to find n bits binary strings which maximize*

$$f_1(\boldsymbol{x}) = \begin{cases} n \cdot |\boldsymbol{x}|_1 + \mathrm{TZ}(\boldsymbol{x}), & \textit{if } \boldsymbol{x} \in G \cup H, \\ 0, & \textit{else}, \end{cases}$$

$$f_2(\boldsymbol{x}) = \begin{cases} n \cdot |\boldsymbol{x}|_1 + \mathrm{LZ}(\boldsymbol{x}), & \textit{if } \boldsymbol{x} \in G \cup H, \\ 0, & \textit{else}, \end{cases}$$

where $n/5 \in \mathbb{Z}^+$, $G = \{\boldsymbol{x} \mid |\boldsymbol{x}|_1 \le 3n/5\}$ and $H = \{\boldsymbol{x} \mid |\boldsymbol{x}|_1 = 4n/5 \wedge \mathrm{LZ}(\boldsymbol{x}) + \mathrm{TZ}(\boldsymbol{x}) = n/5\}$. Here, $|\boldsymbol{x}|_1$, $\mathrm{LZ}(\boldsymbol{x})$, and $\mathrm{TZ}(\boldsymbol{x})$ denote the number of 1-bits, leading zeros, and trailing zeros in $\boldsymbol{x} \in \{0,1\}^n$, respectively.

The right subfigure of Figure 1 gives an example illustration of bi-objective RealRoyalRoad in the (f_1, f_2) objective space, where green points denote Pareto-optimal solutions and black points denote dominated ones. The set of Pareto optimal solutions is exactly H, and the corresponding Pareto front is $\{(4n^2/5 + a, 4n^2/5 + n/5 - a) \mid a \in [0..n/5]\}$, which has size $n/5 + 1$. We further denote by $G' = \{\boldsymbol{x} \mid |\boldsymbol{x}|_1 = 3n/5 \wedge \mathrm{LZ}(\boldsymbol{x}) + \mathrm{TZ}(\boldsymbol{x}) = 2n/5\}$ the set of non-dominated solutions located in the local region G. Thus, the bi-objective RealRoyalRoad problem simulates scenarios in which the search landscape contains structural barriers, posing significant challenges for MOEAs to transition from local to global optimality (i.e., from G' to H).

Algorithm 1 MOEA/EED

Input: objectives $f_1, \ldots, f_m$, maximum lifespan τ for the exploration population P_{explore}, number of initial solutions μ and selection prob. p for the exploitation population P_{exploit}
Output: solutions from P_{exploit}

1: $P_{\text{explore}} \leftarrow \emptyset$;
2: $P_{\text{exploit}} \leftarrow$ initialize μ solutions by uniform and random sampling from $\{0,1\}^n$, and retain the non-dominated solutions with distinct objective vectors;
3: **while** criterion is not met **do**
4: sample u from uniform distribution over $[0,1]$;
5: **if** $u < p$ or $P_{\text{explore}} = \emptyset$ **then**
6: select a solution $\boldsymbol{x}$ from P_{exploit} randomly
7: **else**
8: select a solution $\boldsymbol{x}$ from P_{explore} randomly
9: **end if**
10: apply bit-wise mutation on $\boldsymbol{x}$ to generate $\boldsymbol{x}'$;
11: add $(\boldsymbol{x}', 0)$ to P_{explore};
12: **if** $\nexists \boldsymbol{z} \in P_{\text{exploit}}$ such that $\boldsymbol{z} \succeq \boldsymbol{x}'$ **then**
13: $P_{\text{exploit}} \leftarrow (P_{\text{exploit}} \setminus \{\boldsymbol{z} \in P_{\text{exploit}} \mid \boldsymbol{x}' \succ \boldsymbol{z}\}) \cup \{\boldsymbol{x}'\}$
14: **end if**
15: **for** each $(\boldsymbol{x}, age) \in P_{\text{explore}}$ **do**
16: $age \leftarrow age + 1$
17: **end for**
18: remove all $(\boldsymbol{x}, age)$ from P_{explore} with $age > \tau$
19: **end while**
20: return P_{exploit}

3 Proposed MOEA with Exploration and Exploitation Decoupled

The proposed algorithm MOEA/EED as shown in Algorithm 1 considers two populations for exploitation and exploration (denoted as P_{exploit} and P_{explore}), respectively. In each generation, a parent is selected for variation either from P_{exploit} to exploit with probability p, or from P_{explore} to explore with probability $1 - p$. The exploitation population retains only high-quality solutions (i.e., nondominated solutions found) for individual refinement, while the exploration population disregards solution quality and uses the age of solutions during the evolutionary process to promote the search in "under-developed" regions. Note that aging mechanisms are widely used in evolutionary algorithms and artificial immune systems [Oliveto and Sudholt, 2014; Stefano *et al.*, 2016; Corus *et al.*, 2020], and recently analyzed for their advantages in multi-objective optimization [Li *et al.*, 2025]. In such mechanisms, each solution is assigned an age that increases by one in each generation, and solutions whose age exceeds a maximum lifespan τ are typically removed. The algorithmic procedure is as follows.

At the beginning, MOEA/EED initializes an empty population P_{explore} and a population P_{exploit} by uniformly sampling μ solutions from $\{0,1\}^n$, and retaining the non-dominated solutions with distinct objective vectors (lines 1–2). In each generation, MOEA/EED uniformly at random selects a parent solution $\boldsymbol{x}$ from P_{exploit} with probability p (or from P_{exploit} if $P_{\text{explore}} = \emptyset$), and otherwise selects a parent solution from P_{explore} (lines 4–9). Then, bit-wise mutation flips each bit of $\boldsymbol{x}$ with probability $1/n$ to produce an offspring $\boldsymbol{x}'$ (line 10), which is added to P_{explore} directly (line 11). Afterwards, $\boldsymbol{x}'$ will be tested if it can enter P_{exploit}. If there is no solution in P_{exploit} that weakly dominates $\boldsymbol{x}'$, it will be placed in P_{exploit}, and meanwhile those solutions dominated by the new solution will be deleted from P_{exploit} as the classic GSEMO/SEMO [Giel, 2003] does (lines 12–14). Afterwards, the ages of all individuals in P_{explore} are increased by 1 (lines 15–17). Finally, all solutions with the age more than τ are removed (line 18).

4 Theoretical Analysis of MOEA/EED

In this section, we theoretically analyze the expected running time of MOEA/EED on the OneJumpZeroJump and bi-objective RealRoyalRoad problems. We show that the expected running time is $O((n^{k+1}/p) \cdot \min\{1, (e(\ln C)/k)^k\})$ and $O((5e\ln C)^{n/5}/p)$, respectively, where $C = e(2-p)/(1-p)$. These bounds are much better than those of the original NSGA-II [Doerr and Qu, 2023] and SMS-EMOA [Bian *et al.*, 2025] and their non-elite methods [Li *et al.*, 2025; Ren *et al.*, 2025]. Note that the running time of EAs is often measured by the number of fitness evaluations.

4.1 Runtime Complexity on OneJumpZeroJump

We first show that the expected running time for MOEA/EED solving OneJumpZeroJump is $O((n^{k+1}/p) \cdot \min\{C, (e(\ln C)/k)^k\})$, where $C = e(2-p)/(1-p)$, which is better than that, i.e., $O(\mu n^k)$, of the original SMS-EMOA and NSGA-II [Bian *et al.*, 2025; Doerr and Qu, 2023], where $\mu = \Theta(n)$. Intuitively, the exploration population P_{explore} considers only the age of solutions but not their quality. Thus, the evolutionary search has a chance to go along inferior regions which are close to Pareto-optimal regions, making the search easier. Actually, non-elitist strategies have been shown to be effective in multi-objective optimization in prior works: Li *et al.* [2025] analyzed SMS-EMOA with aging mechanisms, while Bian *et al.* [2025] studied stochastic population updates on SMS-EMOA and NSGA-II. Compared with these strategies, our MOEA/EED framework also achieves a better upper bound due to its dual-population mechanism.

Theorem 1. *For MOEA/EED solving OneJumpZeroJump with $n - 2k = \Omega(n)$, if using the maximum lifespan $\tau = O(n(\ln C)/k)$, and the number of initial solutions $\mu = \Omega(\log n)$, then the expected running time for finding the whole Pareto front is $O((n^{k+1}/p) \cdot \min\{1, (e(\ln C)/k)^k\})$, where $C = e(2-p)/(1-p)$.*

Proof Sketch of Theorem 1. The optimization is split into two phases: finding the inner Pareto front F_I^*, and then locating the extreme solutions 1^n and 0^n.

For the first phase, we consider the case that there is at least one initial solution $\boldsymbol{x}$ such that $|\boldsymbol{x}|_1 \in [k..n-k]$; otherwise, the probability of all initial solutions having at most $(k-1)$ or at least $(n-k+1)$ 1-bits is $(\exp(-\Omega(n)))^\mu$ by Chernoff bound. Starting from $\boldsymbol{x}$, in each generation the algorithm selects a parent (whose neighboring solution is Pareto optimal but not yet in P_{exploit}) from P_{exploit}, flips one of its 1-bits (or 0-bits), and thereby generates a new Pareto-optimal solution. By repeating this process until F_I^* is covered, the expected running time of the first phase is $O(n^2(\ln n)/p)$.

Next, we consider the second phase. By using the aging-based population P_{explore}, any newly generated solution (including dominated ones) can survive for τ generations. This allows the algorithm to gradually reach 1^n through the dominated solutions $\boldsymbol{x} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 \in [n-k+1..n-1]\}$. A symmetric process applies for reaching 0^n. Similar to the proof technic in [Doerr *et al.*, 2023] and [Ren *et al.*, 2025], we assume a "jump" to be an event where a solution $\boldsymbol{x}$ with $|\boldsymbol{x}|_1 \in [n-k..n-1]$ is selected, and a new dominated solution closer to 1^n is generated and added to P_{explore}. Thus, 1^n can be reached more easily through multiple jumps across the gap of dominated solution set (i.e., $\{\boldsymbol{x} \mid |\boldsymbol{x}|_1 \in [n-k+1..n-1]\}$). We refer to the dominated solutions along the multiple jumps as "stepping stones" and assume that 1^n can be reached through M stepping stones. Then, we consider $M+1$ jumps, which start from a solution $\boldsymbol{x}$ with $n-k$ 1-bits, and continue to generate the next stepping-stone solution until finding 1^n. We analyze the probability that the first jump succeeds within n generations and that each of the remaining jumps succeed within τ generations. If any intermediate jump fails, the process restarts from a solution with $n-k$ 1-bits. By enumerating all possible jump sizes (the sum of which is required to be k), we prove that the $M+1$ jumps can happen with probability at least $p(M+1)^k/(e(1+p)n^kC^M)$, where $C = e(2-p)/(1-p)$. Thus, 1^n can be found after $O(n^kC^M/(p(M+1)^k))$ trials in expectation. As each trial requires up to $n + M\tau$ generations, the expected number of generations for finding 1^n is

$$(n + M\tau) \cdot O(n^k \cdot C^M/(p(M+1)^k)).$$

Finally, we minimize this upper bound by taking $M = \lceil (k-1)/\ln C - 1\rceil$ when $k > e\ln C$, and $M = 0$ when $k \le e\ln C$. This leads to that the expected running time of the second phase is at most $O((n^{k+1}/p) \cdot \min\{1, (e(\ln C)/k)^k\})$, where $C = e(2-p)/(1-p)$.

Combining the two phases, the total expected running time is $O((n^{k+1}/p) \cdot \min\{1, (e(\ln C)/k)^k\})$, where $O(n^2(\ln n)/p)$ required by the first phase is dominated. □

On OneJumpZeroJump, the expected running time of SMS-EMOA and NSGA-II is both $O(\mu n^k)$ [Bian *et al.*, 2025; Doerr and Qu, 2023]. SMS-EMOA combined with the stochastic method achieves a speedup of $\Theta(\max\{1, (k/(e(\ln 2e\mu)))^{k-1}\})$ [Ren *et al.*, 2025]; when using an aging method (with $\tau = n - 2k + 4$ and $\mu = 2\tau$), it further reduces the upper bound by a factor of $\Theta(((1 - \exp(-1/2))\ln(2e\mu)/e)^{k-1})$ [Li *et al.*, 2025]. The advantage stems from that under the stochastic method, a solution that is poor in objective values but beneficial for exploration can survive for about 2 generations on average, whereas the aging method guarantees its preservation for τ generations, further increasing exploration chances in promising regions. On the other hand, the NSGA-II that employs $(\mu+\mu)$ update mode and the stochastic method has a high probability of exploring promising regions in each generation. As a result, it achieves an upper bound comparable to that of the aging-based SMS-EMOA. Our proposed MOEA/EED further achieves a better upper bound than NSGA-II using the stochastic method by a factor of $\Theta(k \cdot (\log_{3e}(4e))^{k-1})$ when p is commonly set to $1/2$. The speedup stems from the fact that both the stochastic method and the aging-based approach are applied to a single population during the search. Consequently, the probability of selecting a solution with exploratory value in inferior regions is limited by the proportion of non-elite individuals in the population, which is at most $O(1/n)$ per generation. In contrast, MOEA/EED structurally decouples exploration and exploitation. It uses a selection probability p to control the intensity of exploration in inferior regions, allowing it to allocate a reasonably sized non-elite population for exploration. This structural separation inherently favors newly generated ("young") individuals by preserving them in the non-elite population for exploration. Consequently, it implies a probability of $(1-p)/\tau$ to choose a solution with exploratory value in inferior regions per generation, leading to a speedup. It is also worth noting that the classic GSEMO equipped with a heavy-tailed mutation operator obtains a competitive runtime of $O(n^{k+1}k^{-\Theta(k)})$ on OneJumpZeroJump [Doerr and Zheng, 2021]. Integrating similar operators with the dual-population mechanism may yield even greater acceleration, which we leave as an interesting future work.

4.2 Runtime Complexity on Bi-objective RealRoyalRoad

We next theoretically show that the expected running time for MOEA/EED solving the bi-objective RealRoyalRoad problem is $O((5e\ln C)^{n/5}/p)$, where $C = e(2-p)/(1-p)$, which is significantly better than that, i.e., $\Omega(n^{n/5-1})$, of the original SMS-EMOA and NSGA-II [Bian *et al.*, 2025]. Thanks to the dual-population that decouples exploration and exploitation, MOEA/EED even achieves a significantly greater speedup compared to both SMS-EMOA and NSGA-II with the stochastic method.

Theorem 2. *For MOEA/EED solving the bi-objective RealRoyalRoad problem, if using the maximum lifespan $\tau = O(\ln C)$, and the number of initial solutions $\mu = \Omega(\log n)$, then the expected running time for finding the whole Pareto front is $O((5e\ln C)^{n/5}/p)$, where $C = e(2-p)/(1-p)$.*

Proof Sketch of Theorem 2. We divide the optimization process into five phases, where the first phase aims at finding a solution with $3n/5$ 1-bits, the second phase aims at finding a solution in $G' = \{\boldsymbol{x} \mid |\boldsymbol{x}|_1 = 3n/5 \wedge \mathrm{LZ}(\boldsymbol{x}) + \mathrm{TZ}(\boldsymbol{x}) = 2n/5\}$, the third phase aims at finding the whole G', the fourth phase aims at finding a solution in H (i.e., a Pareto optimal solution), and the fifth phase aims at finding all solutions in H (i.e., all Pareto optimal solutions).

For the first phase, we consider the case that there is at least one initial solution $\boldsymbol{x}$ such that $|\boldsymbol{x}|_1 \leq 3n/5$; otherwise, the probability of all initial solutions having more than $3n/5$ 1-bits is $(\exp(-\Omega(n)))^{\mu}$ by Chernoff bound. Note that for any two solutions $\boldsymbol{x}, \boldsymbol{y} \in G$ with $|\boldsymbol{x}|_1 < |\boldsymbol{y}|_1 \leq 3n/5$, the solution $\boldsymbol{x}$ is dominated by $\boldsymbol{y}$. Thus, the number of 1-bits of solutions in P_{exploit} is non-decreasing (and increases until reaching $3n/5$). In each generation, a new non-dominated solution with more 1-bits can be generated by selecting any parent from P_{exploit} and flipping one of its 0-bits without disturbing the others. By repeating this process until a solution with $3n/5$ 1-bits is found, the expected running time of the first phase is $O(n(\log n)/p)$.

Next, we consider the second phase. By selecting a solution $\boldsymbol{x}$ in P_{exploit} where all the solutions have $3n/5$ 1-bits and flipping the last 1-bit as well as one 0-bit before the last 1-bit, we keep $\mathrm{LZ}(\boldsymbol{x}^*)$ and $|\boldsymbol{x}^*|_1$ of the offspring $\boldsymbol{x}^*$ unchanged, and increase $\mathrm{TZ}(\boldsymbol{x}^*)$ by one. Then by repeating the process at most $2n/5$ times, we can have a solution in G'. Thus, the expected running time of this phase is at most $O(n^4/p)$.

Now we consider the third phase. Suppose G' is not fully covered; then there exists a solution $\boldsymbol{y} \in G' \setminus P_{\text{exploit}}$ that can be generated from some $\boldsymbol{x} \in G' \cap P_{\text{exploit}}$ by either flipping the first 1-bit of $\boldsymbol{x}$ together with the first 0-bit in its trailing 0-bit block, or by flipping the last 1-bit of $\boldsymbol{x}$ together with the last 0-bit in its leading 0-bit block. Intuitively, this operation shifts the position of the 1-bit block without changing its length, hence producing a non-dominated offspring that belongs to G'. By repeating this process at most $2n/5$ times, the entire G' can be found in expected running time of $O(n^4/p)$.

Next, we consider the fourth phase. By using the aging-based population P_{explore}, any newly generated solution (including dominated ones) can survive for τ generations. This allows the algorithm to gradually reach H through the dominated solutions $\boldsymbol{x} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 \in [3n/5+1..4n/5-1]\}$. Similar to the analysis in Theorem 1, we assume a "jump" to be an event where a dominated solution is selected, and a new dominated solution closer to H is generated and added to P_{explore}. Let $G'' := \{0^i 1^{3n/5} 0^{2n/5-i} \mid n/10 \leq i \leq 3n/10\}$ denote a subset of G'. For any $\boldsymbol{x} \in G''$, flipping the consecutive j 0-bits before the block of 1-bits and the consecutive $n/5 - j$ 0-bits after the block of 1-bits generates a solution in H, where the number of valid choices for j is at least $n/10$. That is, there are at least $n/10$ different ways to generate a solution in H from one solution in G''. To increase the success probability of reaching H, we consider $M + 1$ consecutive jumps, which start from a solution in G'', and continue to generate the next stepping-stone solution until finding a solution in H. If any intermediate jump fails, the process restarts from a solution in G''. By enumerating all possible jump sizes (the sum of which is required to be $n/5$), we prove that the $M + 1$ jumps can happen with probability at least $p(M + 1)^{n/5}/(2en^{n/5}C^M)$, where $C = e(2-p)/(1-p)$. Since there are at least $n/10$ different ways to reach H, and each trial needs at most $1 + M\tau$ generations, the expected number of generations for reaching H is at most $(1+M\tau)\cdot(20en^{n/5-1}C^M/p(M+1)^{n/5})$. Finally, we minimize this upper bound by taking $M = \lceil n/(5\ln C) - 1 \rceil$. This leads to that the expected running time of the fourth phase is at most $O((5e\ln C)^{n/5}/p)$, where $C = e(2-p)/(1-p)$.

Finally, we consider the last phase. The proof is almost the same as that of the third phase, except that we only need to find at most $n/5$ remaining solutions in H. Thus, we can directly derive that the total expected number of generations of this phase is at most $O(n^4/p)$. Combining the five phases, the total expected running time is $O((5e\ln C)^{n/5}/p)$. $\square$

On the bi-objective RealRoyalRoad problem, the expected running time of SMS-EMOA and NSGA-II is both $\Omega(n^{n/5-1})$ [Bian *et al.*, 2025]. SMS-EMOA combined with the stochastic method achieves a speedup of $\Omega(2^{n/10}/(\mu^2 n^{3/2}))$ [Bian *et al.*, 2025], while NSGA-II with the stochastic method further achieves an additional speedup factor of $\Theta(\mu(n/20\sqrt{2}e^2)^{n/5})$. Yet, single-population approaches remain structurally bottlenecked. Our proposed MOEA/EED shows yet another improvement, with a speedup factor of $\Theta(\mu\sqrt{n}(4e/\ln(3e))^{n/5})$ when p is commonly set to $1/2$. As on OneJumpZeroJump, the improvement stems from decoupling exploitation and exploration through a dual-population mechanism. This mechanism avoids restricting the number of non-elite solutions, resulting in a higher probability, i.e., $(1-p)/\tau$, of selecting solutions with exploration value in inferior regions and thus achieving a speedup.

Problem	Budget	NSGA-II	NSGA-II-Stochastic	SMS-EMOA	SMS-EMOA-Aging	MOEA/EED
KP-100	10^4	2.289e+03 (5.2e+03)†	3.036e+03 (4.9e+03)†	2.549e+04 (2.1e+04)†	2.140e+04 (2.2e+04)†	**1.826e+05 (2.5e+04)**
	10^5	2.355e+05 (1.4e+04)†	2.518e+05 (1.6e+04)†	2.563e+05 (1.3e+04)†	2.532e+05 (1.5e+04)†	**2.917e+05 (5.3e+03)**
KP-200	10^4	0.000e+00 (0.0e+00)†	0.000e+00 (0.0e+00)†	0.000e+00 (0.0e+00)†	0.000e+00 (0.0e+00)†	**1.315e+05 (4.9e+04)**
	10^5	3.155e+05 (4.2e+04)†	3.227e+05 (4.7e+04)†	3.457e+05 (3.8e+04)†	3.546e+05 (4.4e+04)†	**5.097e+05 (2.5e+04)**
TSP-100	10^4	3.479e+01 (9.7e+00)†	3.037e+01 (7.7e+00)†	7.966e+01 (1.0e+01)†	8.145e+01 (1.2e+01)†	**4.188e+02 (2.9e+01)**
	10^5	6.630e+02 (1.3e+01)†	6.306e+02 (1.7e+01)†	6.929e+02 (1.2e+01)†	6.789e+02 (1.6e+01)†	**8.093e+02 (2.0e+01)**
TSP-200	10^4	0.000e+00 (0.0e+00)†	0.000e+00 (0.0e+00)†	0.000e+00 (0.0e+00)†	0.000e+00 (0.0e+00)†	**3.270e+01 (3.6e+01)**
	10^5	2.859e+02 (2.5e+01)†	2.493e+02 (2.8e+01)†	4.038e+02 (3.0e+01)†	3.923e+02 (3.5e+01)†	**7.510e+02 (5.6e+01)**
QAP-100	10^4	7.801e+13 (1.4e+13)†	6.809e+13 (1.4e+13)†	1.382e+14 (1.9e+13)†	1.309e+14 (2.1e+13)†	**5.093e+14 (6.0e+13)**
	10^5	9.662e+14 (4.9e+13)†	9.334e+14 (5.4e+13)†	1.008e+15 (3.6e+13)†	9.869e+14 (5.1e+13)†	**1.137e+15 (6.4e+13)**
QAP-200	10^4	4.271e+11 (1.9e+12)†	0.000e+00 (0.0e+00)†	4.457e+13 (4.0e+13)†	4.315e+13 (3.2e+13)†	**1.315e+15 (7.7e+14)**
	10^5	3.409e+15 (1.7e+15)†	3.113e+15 (1.5e+15)†	3.912e+15 (1.9e+15)†	3.824e+15 (1.8e+15)†	**4.927e+15 (2.2e+15)**
NK-100	10^4	2.006e−02 (1.4e−03)†	1.957e−02 (2.0e−03)†	2.259e−02 (1.5e−03)†	2.251e−02 (2.0e−03)†	**2.934e−02 (2.1e−03)**
	10^5	3.627e−02 (2.4e−03)†	3.779e−02 (2.8e−03)	3.433e−02 (2.2e−03)†	3.476e−02 (1.9e−03)†	**3.894e−02 (2.9e−03)**
NK-200	10^4	5.653e−03 (6.3e−04)†	5.409e−03 (7.6e−04)†	7.594e−03 (9.3e−04)†	7.413e−03 (7.7e−04)†	**1.411e−02 (1.5e−03)**
	10^5	2.228e−02 (1.4e−03)†	2.265e−02 (1.2e−03)†	2.160e−02 (1.2e−03)†	2.148e−02 (1.5e−03)†	**2.490e−02 (1.4e−03)**

"†" indicates that the algorithm is significantly different from MOEA/EED, based on the Wilcoxon rank-sum test with 95% confidence.

Table 2: The HV (mean and standard deviation) of the NSGA-II, NSGA-II-Stochastic, SMS-EMOA, SMS-EMOA-Aging and MOEA/EED on KP, NK, TSP and QAP. The number next to each problem denotes the number of variables. The best mean is highlighted in bold.

5 Experiments

In this section, we conduct experiments on four well-known practical problems: the multi-objective 0/1 knapsack (KP) [Teghem, 1994], travelling salesman problem (TSP) [Ribeiro *et al.*, 2002], quadratic assignment problem (QAP) [Knowles and Corne, 2003], and NK-landscapes (NK) [Aguirre and Tanaka, 2004], following the settings in [Li *et al.*, 2024a; Liang and Li, 2026]. These problems cover binary representation (KP, NK), order-based (TSP) and position-based (QAP) permutation representations, constraint (KP), highly rugged landscape (NK when $K = 10$) and relatively smooth landscapes (KP, TSP). Each problem is considered at two sizes: 100 and 200 decision variables.

We consider five algorithms: NSGA-II, the stochastic variant of NSGA-II (NSGA-II-Stochastic), SMS-EMOA, the aging variant of SMS-EMOA (SMS-EMOA-Aging) and the proposed MOEA/EED, as described in the above sections. NSGA-II, SMS-EMOA and their variants follow similar settings: 1) a population size of 200; 2) the parent selection uses uniform random selection; 3) the variation operator uses only a problem-specific mutation. Following the practice in [Li *et al.*, 2024a], for pseudo-Boolean problems (KP, NK), we use the bit-wise mutation [Eiben and Smith, 2015] with a rate of $1/n$, where n is the number of variables. For permutation problems (TSP, QAP), we use the 2-opt and 2-swap mutations [Eiben and Smith, 2015], respectively, and the mutation rate is set to 1.0. For the proposed MOEA/EED, the same problem-specific mutation is used; the maximum lifespan τ is set to 10; the number μ of initial solutions is set to 1; the probability p of selecting the exploitation population is set to 0.75. The two non-elitist variants of NSGA-II and SMS-EMOA adopt this ratio, as well. For the NSGA-II-Stochastic, $1/4$ of the parent$\cup$offspring (100 out of 400 solutions) are randomly selected to enter the next generation. For the SMS-EMOA-Aging, the maximum lifespan is set to $1/4$ of the population size (i.e., 50), so $1/4$ of the population are preserved at each generation regardless of their quality. For all algorithms, we consider two common computational budgets: 10^4 and 10^5 evaluations; the number of independent runs is 30.

We use the hypervolume (HV) [Zitzler and Thiele, 1999] to compare the performance of the algorithms. According to [Li *et al.*, 2022], the reference point of HV is set to $\mathrm{nadir} - 0.1 \times (\mathrm{ideal} - \mathrm{nadir})$ for maximization problems (KP, NK), and to $\mathrm{nadir} + 0.1 \times (\mathrm{nadir} - \mathrm{ideal})$ for minimization problems (TSP, QAP), where the ideal/nadir point is computed from the non-dominated set of all algorithms at 10^4 evaluations.

Table 2 gives the HV results of the five algorithms on the four practical problems. The proposed MOEA/EED achieves the best HV across all problem instances and budgets, with statistically significant differences over the others in almost all cases. The difference is substantial (one to two orders of magnitude in HV) in some instances under the small budget. This indicates that MOEA/EED gains advantages quickly, particularly for larger problem instances (e.g., QAP-200). Indeed, for some large instances (KP-200 and TSP-200), the solutions obtained by the other four competitors after 10^4 evaluations fail to match the nadir point of the non-dominated set obtained by MOEA/EED, resulting in a HV value of zero.

6 Conclusion

In this paper, we propose a dual-population MOEA with Exploration and Exploitation Decoupled (MOEA/EED): the exploration population uses a simple aging mechanism, and the exploitation population stores only non-dominated solutions. Exploration and exploitation are regulated by a probability p instead of the proportion of non-elite populations. We theoretically show that MOEA/EED achieves significantly better expected running time bounds on the OneJumpZeroJump and bi-objective RealRoyalRoad problems than SMS-EMOA, NSGA-II, and their non-elitist variants. Finally, the theoretical benefits are empirically validated on a range of practical optimization problems.

Acknowledgments

This work was supported by the National Science and Technology Major Project (2022ZD0116600), the NSFC (62276124), the Fundamental Research Funds for the Central Universities (14380020), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM118), and the "111 Center" (B26023). Chao Qian is the corresponding author.

References

[Aguirre and Tanaka, 2004] Hernán Aguirre and Kiyoshi Tanaka. Effects of elitism and population climbing on multiobjective MNK-landscapes. In *CEC*, Portland, OR, 2004.

[Benjamin *et al.*, 2025] Doerr Benjamin, Tudor Ivan, and Martin S. Krejca. Speeding up the NSGA-II with a simple tie-breaking rule. In *AAAI*, Philadelphia, Pennsylvania, 2025.

[Beume *et al.*, 2007] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *EJOR*, 181:1653–1669, 2007.

[Bian *et al.*, 2018] Chao Bian, Chao Qian, and Ke Tang. A general approach to running time analysis of multi-objective evolutionary algorithms. In *IJCAI*, Stockholm, Sweden, 2018.

[Bian *et al.*, 2024] Chao Bian, Shengjie Ren, Miqing Li, and Chao Qian. An archive can bring provable speed-ups in multi-objective evolutionary algorithms. In *IJCAI*, Jeju Island, South Korea, 2024.

[Bian *et al.*, 2025] Chao Bian, Yawen Zhou, Miqing Li, and Chao Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. *AIJ*, 341:104308, 2025.

[Corus *et al.*, 2020] Dogan Corus, Pietro Simone Oliveto, and Donya Yazdani. When hypermutations and ageing enable artificial immune systems to outperform evolutionary algorithms. *TCS*, 832:166–185, 2020.

[Dang *et al.*, 2024] Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. *AIJ*, 330:104098, 2024.

[Deb *et al.*, 2002] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE TEVC*, 6(2):182–197, 2002.

[Deb, 2001] Kalyanmoy Deb. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[Doerr and Neumann, 2020] Benjamin Doerr and Frank Neumann. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*. Springer, 2020.

[Doerr and Qu, 2023] Benjamin Doerr and Zhongdi Qu. A first runtime analysis of the NSGA-II on a multimodal problem. *IEEE TEVC*, 27(5):1288–1297, 2023.

[Doerr and Zheng, 2021] Benjamin Doerr and Weijie Zheng. Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In *AAAI*, Virtual, 2021.

[Doerr *et al.*, 2013] Benjamin Doerr, Bojana Kodric, and Marco Voigt. Lower bounds for the runtime of a global multi-objective evolutionary algorithm. In *CEC*, Cancun, Mexico, 2013.

[Doerr *et al.*, 2023] Benjamin Doerr, Arthur Dremaux, Johannes Lutzeyer, and Aurelien Stumpf. How the move acceptance hyper-heuristic copes with local optima: Drastic differences between jumps and cliffs. In *GECCO*, Lisbon, Portugal, 2023.

[Doerr *et al.*, 2025] Benjamin Doerr, Martin S. Krejca, and Gunter Rudolph. Runtime analysis for multi-objective evolutionary algorithms in unbounded integer spaces. In *AAAI*, Philadelphia, Pennsylvania, 2025.

[Eiben and Smith, 2015] Agoston Eiben and Jim Smith. *Introduction to Evolutionary Computing*. Springer, 2015.

[Giel and Lehre, 2010] Oliver Giel and Per Kristian Lehre. On the effect of populations in evolutionary multi-objective optimisation. *ECJ*, 18(3):335–356, 2010.

[Giel, 2003] Oliver Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *CEC*, Canberra, Australia, 2003.

[Huang *et al.*, 2021] Zhengxin Huang, Yuren Zhou, Chuan Luo, and Qingwei Lin. A runtime analysis of typical decomposition approaches in MOEA/D framework for many-objective optimization problems. In *IJCAI*, Virtual, 2021.

[Ishibuchi *et al.*, 2020] Hisao Ishibuchi, Lie Meng Pang, and Ke Shang. A new framework of evolutionary multi-objective algorithms with an unbounded external archive. In *ECAI*, Santiago, Spain, 2020.

[Knowles and Corne, 2003] Joshua Knowles and David Corne. Instance generators and test suites for the multi-objective quadratic assignment problem. In *EMO*, Faro, Portugal, 2003.

[Laumanns *et al.*, 2004] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE TEVC*, 8(2):170–182, 2004.

[Li *et al.*, 2018] Ke Li, Renzhi Chen, Guangtao Fu, and Xin Yao. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE TEVC*, 23(2):303–315, 2018.

[Li *et al.*, 2022] Miqing Li, Tao Chen, and Xin Yao. How to evaluate solutions in pareto-based search-based software engineering: A critical review and methodological guidance. *IEEE TSE*, 48(5):1771–1799, 2022.

[Li *et al.*, 2024a] Miqing Li, Xiaofeng Han, Xiaochen Chu, and Zimin Liang. Empirical comparison between MOEAs and local search on multi-objective combinatorial optimisation problems. In *GECCO*, Melbourne, Australia, 2024.

[Li *et al.*, 2024b] Miqing Li, Manuel López-Ibáñez, and Xin Yao. Multi-objective archiving. *IEEE TEVC*, 28(3):696–717, 2024.

[Li *et al.*, 2025] Mingfeng Li, Weijie Zheng, and Benjamin Doerr. Scalable speed-ups for the SMS-EMOA from a simple aging strategy. In *IJCAI*, Montreal, Canada, 2025.

[Liang and Li, 2026] Zimin Liang and Miqing Li. Random is faster than systematic in multi-objective local search. In *AAAI*, Singapore, 2026.

[Liang *et al.*, 2023] Zimin Liang, Miqing Li, and Per Kristian Lehre. Non-elitist evolutionary multi-objective optimisation: Proof-of-principle results. In *GECCO*, Lisbon, Portugal, 2023.

[Liu *et al.*, 2018] Yiping Liu, Gary G. Yen, and Dunwei Gong. A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. *IEEE TEVC*, 23(4):660–674, 2018.

[Lu *et al.*, 2024] Tianhao Lu, Chao Bian, and Chao Qian. Towards running time analysis of interactive multi-objective evolutionary algorithms. In *AAAI*, Vancouver, Canada, 2024.

[Neumann, 2007] Frank Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *EJOR*, 181(3):1620–1629, 2007.

[Oliveto and Sudholt, 2014] Pietro S. Oliveto and Dirk Sudholt. On the runtime analysis of stochastic ageing mechanisms. In *GECCO*, Vancouver, Canada, 2014.

[Opris, 2025a] Andre Opris. A first runtime analysis of NSGA-III on a many-objective multimodal problem: Provable exponential speedup via stochastic population update. In *IJCAI*, Montreal, Canada, 2025.

[Opris, 2025b] Andre Opris. A first runtime analysis of the PAES-25: An enhanced variant of the pareto archived evolution strategy. In *FOGA*, Leiden, The Netherlands, 2025.

[Qian *et al.*, 2013] Chao Qian, Yang Yu, and Zhi-Hua Zhou. An analysis on recombination in multi-objective evolutionary optimization. *AIJ*, 204:99–119, 2013.

[Qian *et al.*, 2016] Chao Qian, Ke Tang, and Zhi-Hua Zhou. Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization. In *PPSN*, Edinburgh, Scotland, 2016.

[Ren *et al.*, 2024a] Shengjie Ren, Chao Bian, Miqing Li, and Chao Qian. A first running time analysis of the strength Pareto evolutionary algorithm 2 (SPEA2). In *PPSN*, Hagenberg, Austria, 2024.

[Ren *et al.*, 2024b] Shengjie Ren, Zhijia Qiu, Chao Bian, Miqing Li, and Chao Qian. Maintaining diversity provably helps in evolutionary multimodal optimization. In *IJCAI*, Jeju Island, South Korea, 2024.

[Ren *et al.*, 2025] Shengjie Ren, Zimin Liang, Miqing Li, and Chao Qian. A theoretical perspective on why stochastic population update needs an archive in evolutionary multi-objective optimization. In *IJCAI*, Montreal, Canada, 2025.

[Ribeiro *et al.*, 2002] Celso C. Ribeiro, Pierre Hansen, Pedro Castro Borges, and Michael Pilegaard Hansen. A study of global convexity for a multiple objective travelling salesman problem. *Essays and Surveys in Metaheuristics*, pages 129–150, 2002.

[Sanchez *et al.*, 2023] Victor Manuel Sanchez Sanchez, Carlos Gershenson Garcia, and Carlos Ignacio Hernandez Castellanos. On the effect of temporal heterogeneity on selection pressure of evolutionary algorithms. In *GECCO*, Lisbon, Portugal, 2023.

[Stefano *et al.*, 2016] Antonino Di Stefano, Alessandro Vitale, Vincenzo Cutello, and Mario Pavone. How long should offspring lifespan be in order to obtain a proper exploration? In *SSCI*, Athens, Greece, 2016.

[Tanabe and Ishibuchi, 2019] Ryoji Tanabe and Hisao Ishibuchi. Non-elitist evolutionary multi-objective optimizers revisited. In *GECCO*, Prague, Czech Republic, 2019.

[Teghem, 1994] Jacques Teghem. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, January 1994.

[Wietheger and Doerr, 2023] Simon Wietheger and Benjamin Doerr. A mathematical runtime analysis of the non-dominated sorting genetic algorithm III (NSGA-III). In *IJCAI*, Macao, SAR, China, 2023.

[Xie *et al.*, 2025] Shumin Xie, Kangshun Li, Wenxiang Wang, Hui Wang, Chaoda Peng, and Hassan Jalil. A tractive population-assisted dual-population and two-phase evolutionary algorithm for constrained multiobjective optimization. *IEEE TEVC*, 29(1):31–45, 2025.

[Zhang and Li, 2007] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE TEVC*, 11(6):712–731, 2007.

[Zheng and Doerr, 2024a] Weijie Zheng and Benjamin Doerr. Runtime analysis for the NSGA-II: Proving, quantifying, and explaining the inefficiency for many objectives. *IEEE TEVC*, 28(5):1442–1454, 2024.

[Zheng and Doerr, 2024b] Weijie Zheng and Benjamin Doerr. Runtime analysis of the SMS-EMOA for many-objective optimization. In *AAAI*, Vancouver, Canada, 2024.

[Zheng *et al.*, 2022] Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the non-dominated sorting genetic algorithm II (NSGA-II). In *AAAI*, Virtual, 2022.

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

[Zitzler and Thiele, 1999] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE TEVC*, 3(4):257–271, 1999.