Not Just for Archiving: Provable Benefits of Reusing the Archive in Evolutionary Multi-objective Optimization

Shengjie Ren[1,2], **Zimin Liang**[3], **Miqing Li**[3], **Chao Qian**[1,2]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
[3]School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K.
rensj@lamda.nju.edu.cn, zxl525@student.bham.ac.uk, m.li.8@bham.ac.uk, qianc@nju.edu.cn

Abstract

Evolutionary Algorithms (EAs) have become the most popular tool for solving widely-existed multi-objective optimization problems. In Multi-Objective EAs (MOEAs), there is increasing interest in using an archive to store non-dominated solutions generated during the search. This approach can 1) mitigate the effects of population oscillation, a common issue in many MOEAs, and 2) allow for the use of smaller, more practical population sizes. In this paper, we analytically show that the archive can even further help MOEAs through reusing its solutions during the process of new solution generation. We first prove that using a small population size alongside an archive (without incorporating archived solutions in the generation process) may fail on certain problems, as the population may remove previously discovered but promising solutions. We then prove that reusing archive solutions can overcome this limitation, resulting in at least a polynomial speedup on the expected running time. Our analysis focuses on the well-established SMS-EMOA algorithm applied to the commonly studied OneJumpZeroJump problem as well as one of its variants. We also show that reusing archive solutions can be better than using a large population size directly. Finally, we validate our theoretical findings through experiments on well-known practical optimization problems.

Code — https://github.com/Zim-L/AAAI26ArchiveReuse
Extended version — https://arxiv.org/abs/2508.16993

Introduction

Multi-objective optimization deals with problems in which multiple, often conflicting, objectives must be optimized simultaneously. Unlike single-objective optimization, there is generally no single optimal solution for a Multi-objective Optimization Problem (MOP). Instead, the goal is to find a set of solutions representing different optimal trade-offs among the objectives, called Pareto optimal solutions. The corresponding objective vectors of these solutions constitute the Pareto front. Thus, the goal of multi-objective optimization is to find the Pareto front or its good approximation. Evolutionary Algorithms (EAs), a class of randomized heuristic optimization algorithms inspired by natural evolution, have proven to be well-suited for solving MOPs due to their population-based nature. They have been widely applied across various real-world domains (Deb 2001; Zhou, Yu, and Qian 2019). Over the years, numerous well-established Multi-Objective EAs (MOEAs) have been developed, including the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al. 2002), the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) (Zhang and Li 2007), and the $\mathcal{S}$-Metric Selection Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) (Beume, Naujoks, and Emmerich 2007).

In the area of MOEAs, there has recently been growing interest in using an archive to store non-dominated solutions generated during the search process (Li, López-Ibáñez, and Yao 2024). The main reason for this practice is as follows. In MOPs, there can be infinite many Pareto optimal solutions (Figueira et al. 2017), and it may not be practical to use a very large population aiming for accommodating all of them. However, using a small population may easily lead to the oscillation phenomenon, even when an elite preservation mechanism is employed (Knowles and Corne 2003b). This is because, during the evolutionary process, the number of generated non-dominated solutions usually exceeds the population size. As a result, population maintenance becomes necessary to remove excess non-dominated solutions (e.g., based on criteria of crowding distance). Yet, the algorithm may later accept new solutions that are non-dominated w.r.t. the current population but are actually dominated by the previously discarded solutions. Consequently, MOEAs may end up with a final population containing a large number of sub-optimal (i.e., globally dominated) solutions. This issue affects all practical MOEAs (Li, López-Ibáñez, and Yao 2024). For example, Li and Yao (2019) reports that on some problems like DTLZ7 (Deb et al. 2005), nearly half of the final population produced by NSGA-II and MOEA/D are not globally optimal (i.e., being dominated by previously discarded solutions). This phenomenon leads to the popularity of using an archive in MOEAs that store non-dominated solutions found during the search process, e.g., (Fieldsend, Everson, and Singh 2003; Krause et al. 2016; Brockhoff and Tušar 2019; Ishibuchi, Pang, and Shang 2020). Recently, theoretical studies (Bian et al. 2024) also confirm this practice, showing that using an archive can provide a speedup, particularly when working with stochastic population updates in MOEAs (Ren et al. 2025).

		OneJumpZeroJump	**OneJumpZeroJump$_{SS}$**
SMS-EMOA	Archive without reuse	$n^{\Omega(n)}$ (Theorem 1) [$\mu = 2, k \leq n/8$]	$\Omega(n^k)$ (Theorem 3) [$\mu \leq n - 2k + 1, k \geq 13, a \geq k/4$]
	Archive with reuse	$O(\mu n^k)$ (Theorem 2) [$\mu \geq 2, k \geq 3$]	$O(\max\{\mu n^a/k, n^{k-a+1}, n^2 \log n\})$ (Theorem 4) [$\mu \geq 2$]
	Original algorithm with a large population	$O(\mu n^k)$ (Bian et al. 2025) [$\mu \geq n - 2k + 3$]	$O(\mu \cdot \max\{n^a/k, n^{k-a}\})$ (Theorem 5) [$\mu \geq n - 2k + 5$]

Table 1: The expected number of generations of SMS-EMOA for solving OneJumpZeroJump and OneJumpZeroJump$_{SS}$ when 1) considering an archive without the reuse, 2) considering an archive with the reuse, and 3) employing a large population size, where n denotes the problem size, k and a denote the parameters of OneJumpZeroJump ($2 \leq k < n/2$) and OneJumpZeroJump$_{SS}$ ($3 \leq k < n/2, 2 \leq a < k$), and μ denotes the population size.

In this paper, we theoretically show that the archive can further help MOEAs by reusing its solutions during the process of new solution generation. One drawback of using a small population alongside an archive is that some promising non-dominated solutions discovered by an MOEA may be removed from the population due to its limited size, preventing the algorithm from further exploiting them. A reuse of the archive where all non-dominated solutions are preserved can "revisit" these solutions, resulting in a speedup. Specifically, in this study we analyze the expected running time of the well-established SMS-EMOA for solving OneJumpZeroJump and its variant OneJumpZeroJump$_{SS}$, and prove that a simple way of reusing the archive (i.e., with probability 1/2, randomly choosing solutions in the archive to generate offspring) helps. We compare three cases of the algorithm: 1) using an archive solely to store non-dominated solutions (as is common in many MOEAs), 2) reusing the archive, and 3) employing a large population size (i.e., the original version of the algorithm). The results are given in Table 1. Our findings can be summarized as follows.

- Reusing the archive can help find a more diverse and well-covered Pareto front. By comparing the results of Theorem 1 and Theorem 2 in Table 1, we show that the expected running time of SMS-EMOA for solving OneJumpZeroJump when only using an archive to store solutions is $n^{\Omega(n)}$, whereas it is reduced to $O(\mu n^k)$ when the archive is reused to generate new solutions. The reason is that a small population may miss some regions of the Pareto front during the early stages of the search, and re-finding those solutions becomes difficult afterward. In contrast, reusing the archived solutions enables the algorithm to re-explore the previously missed regions easily.
- Reusing the archive can benefit exploration. By comparing the results of Theorem 3 and Theorem 4 in Table 1, we show that the expected running time of SMS-EMOA for solving OneJumpZeroJump$_{SS}$ using an archive that only stores solutions is $\Omega(n^k)$, whereas it is reduced to $O(\max\{\mu n^a/k, n^{k-a+1}, n^2 \log n\})$ (where $\mu \geq 2$ and $2 \leq a < k$) when the archive is reused. The reason is that, for most practical MOEAs like SMS-EMOA, when the number of non-dominated solutions exceeds the population size, diversity maintenance mechanisms in the objective space are typically employed to prune the population. As a result, some non-dominated solutions that are sparse in the decision space (often beneficial for exploration), but are crowded in the objective space, are likely to be discarded. Since such solutions are preserved in the archive, allowing reusing the archive helps the algorithm explore sparse regions in the decision space.
- Reusing the archive with a small population can be better than directly using a large population size. By comparing the results in the last two rows of Table 1, we show that for SMS-EMOA solving OneJumpZeroJump and OneJumpZeroJump$_{SS}$, reusing the archive can reduce the upper bound on the expected running time compared to using a large population size. The reason is that reusing the archive allows for a constant population size $\mu \geq 2$, which brings a speedup of $\Theta(n)$. On OneJumpZeroJump$_{SS}$, this improvement becomes evident only when $a > k/2$. Actually, Bian et al. (2024) showed that compared to using a large population size, using an archive can provide a polynomial-speedup for SMS-EMOA solving OneMinMax and LeadingOnesTrailingZeroes. In this paper, we also show that reusing the archive does not diminish this advantage.

We also conduct experiments on the artificial problems studied in this paper and four well-known practical problems: the multi-objective 0/1 knapsack (Teghem 1994), TSP (Ribeiro et al. 2002), QAP (Knowles and Corne 2003a), and NK-landscapes (Aguirre and Tanaka 2004). The results validate our theoretical findings.

It is worth pointing out that there have been many important studies on the running time of MOEAs over the past two decades. Early theoretical works (Laumanns, Thiele, and Zitzler 2004a,b; Neumann 2007; Giel and Lehre 2010; Neumann and Theile 2010; Doerr, Kodric, and Voigt 2013; Qian, Yu, and Zhou 2013; Qian, Tang, and Zhou 2016; Bian, Qian, and Tang 2018) mainly focus on simple MOEAs like (G)SEMO. Recently, researchers have begun to examine practical MOEAs. Huang *et al.* (2021) investigated MOEA/D. Zheng *et al.* (2022) conducted the first theoretical analysis of NSGA-II. Bian *et al.* (2025) analyzed the running time of SMS-EMOA and showed that stochastic population update can bring acceleration. Wietheger and Doerr (2023) showed that NSGA-III (Deb and Jain 2014) outperforms

NSGA-II on 3OneMinMax. Ren *et al.* (2024a) analyzed SPEA2. Some other works on well-established MOEAs include (Bian and Qian 2022; Zheng and Doerr 2024a; Cerf et al. 2023; Dang et al. 2023a,b; Doerr and Qu 2023b,c; Dang, Opris, and Sudholt 2024; Opris, Dang, and Sudholt 2024; Ren et al. 2024b; Wietheger and Doerr 2024; Doerr, Krejca, and Rudolph 2025; Doerr, Ivan, and Krejca 2025; Opris 2025). There are also a series of theoretical studies showing that MOEAs can achieve good approximation guarantees for single-objective submodular optimization with constraints (Qian, Yu, and Zhou 2015; Friedrich and Neumann 2015; Qian et al. 2017, 2019)

Preliminaries

In this section, we first give basic concepts in multi-objective optimization, followed by a description of the SMS-EMOA algorithm and the proposed archive reusing mechanism. Lastly, we describe the OneJumpZeroJump problem and its variant OneJumpZeroJump$_{SS}$, studied in this paper.

Multi-objective Optimization

Multi-objective optimization aims to optimize two or more objective functions simultaneously, as shown in Definition 1. The objectives are typically conflicting, meaning that there is no canonical complete order in the solution space $\mathcal{X}$. To compare solutions, we use the *domination* relationship in Definition 2. 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 forms the *Pareto front*. The goal of multi-objective optimization is to find the Pareto front or its good approximation.

Definition 1 (Multi-objective Optimization)**.** *Given a feasible solution space $\mathcal{X}$ and objective functions $f_1, f_2, \ldots, f_m$, multi-objective optimization can be formulated as*

$$\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x}) = \max_{\boldsymbol{x}\in\mathcal{X}} \big(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})\big).$$

Definition 2. *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 all $1 \leq i \leq m, f_i(\boldsymbol{x}) \geq f_i(\boldsymbol{y})$;*
- $\boldsymbol{x}$ 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 $\boldsymbol{y} \succeq \boldsymbol{x}$.*

SMS-EMOA and Archiving Mechianism

SMS-EMOA (Beume, Naujoks, and Emmerich 2007) presented in Algorithm 1 is a popular MOEA, which employs non-dominated sorting and hypervolume indicator to update the population. It starts with an initial population of μ solutions (line 1). In each generation, it randomly selects a solution $\boldsymbol{x}$ from the current population (line 3) for reproduction. Afterwards, bit-wise mutation flips each bit of $\boldsymbol{x}$ with probability $1/n$ to produce an offspring $\boldsymbol{x}'$ (line 4). Then, the union of the current population and the offspring solution is partitioned into non-dominated sets $R_1, \ldots, R_v$ (line 5), where R_1 contains all non-dominated solutions in $P \cup \{\boldsymbol{x}'\}$, and R_i ($i \geq 2$) contains all non-dominated solutions in $P \cup \{\boldsymbol{x}'\} \setminus \cup_{j=1}^{i-1} R_j$. A solution $\boldsymbol{z} \in R_v$ is then removed (lines 6–7) by minimizing $\Delta_{\boldsymbol{r}}(\boldsymbol{x}, R_v) := HV_{\boldsymbol{r}}(R_v) - HV_{\boldsymbol{r}}(R_v \setminus \{\boldsymbol{x}\})$, where $HV_{\boldsymbol{r}}(X)$ denotes the hypervolume of the solution set X with respect to a reference point $\boldsymbol{r} \in \mathbb{R}^m$ ($\forall i, r_i \leq \min_{\boldsymbol{x}\in\mathcal{X}} f_i(\boldsymbol{x})$), i.e., the volume between the reference point and the objective vectors of the solution set. A larger hypervolume indicates better approximation of the Pareto front in terms of convergence and diversity. For bi-objective problems, as defined in the original SMS-EMOA (Beume, Naujoks, and Emmerich 2007), the algorithm omits the reference point and directly preserves the two boundary points, i.e., the two solutions with the maximum f_1 and f_2 values, respectively, allowing the hypervolume contribution to be calculated accordingly.

Algorithm 1: SMS-EMOA

Input: objective functions $f_1, f_2 \ldots, f_m$, population size μ
Output: μ solutions from $\{0,1\}^n$

1: $P \leftarrow \mu$ solutions uniformly and randomly selected from $\{0,1\}^n$ with replacement;
2: **while** criterion is not met **do**
3: select a solution $\boldsymbol{x}$ from P uniformly at random;
4: apply bit-wise mutation on $\boldsymbol{x}$ to generate $\boldsymbol{x}'$;
5: partition $P \cup \{\boldsymbol{x}'\}$ into non-dominated sets $R_1, ..., R_v$;
6: let $\boldsymbol{z} = \arg\min_{\boldsymbol{x}\in R_v} \Delta_{\boldsymbol{r}}(\boldsymbol{x}, R_v)$;
7: $P \leftarrow (P \cup \{\boldsymbol{x}'\}) \setminus \{\boldsymbol{z}\}$
8: **end while**
9: **return** P

An archive in MOEAs is used to store non-dominated solutions found during the search, preventing them from being discarded due to limited population capacity. Once a new solution is generated, the solution will be tested if it can enter the archive. If there is no solution in the archive that weakly dominates the new solution, then the solution will be placed in the archive, and meanwhile those solutions dominated by the new solution will be deleted from the archive. Algorithmic steps incurred by adding an archive in SMS-EMOA are given as follows. In Algorithm 1, an empty set A is initialized in line 1, the set A instead of P is returned in the last line, and the following lines are added after line 4:

if $\nexists \boldsymbol{z} \in A$ such that $\boldsymbol{z} \succeq \boldsymbol{x}'$ **then**
  $A \leftarrow (A \setminus \{\boldsymbol{z} \in A \mid \boldsymbol{x}' \succ \boldsymbol{z}\}) \cup \{\boldsymbol{x}'\}$
end if

Reusing the Archive

Reusing the archive means allowing the solutions stored in it to re-participate in the evolutionary process. It can be materialized in different ways. Here we consider a straightforward one. During the parent selection, SMS-EMOA uniformly randomly selects a parent solution from population P with probability $1/2$, and otherwise selects a parent solution from archive A. Specifically, the following lines replace line 3 in Algorithm 1:

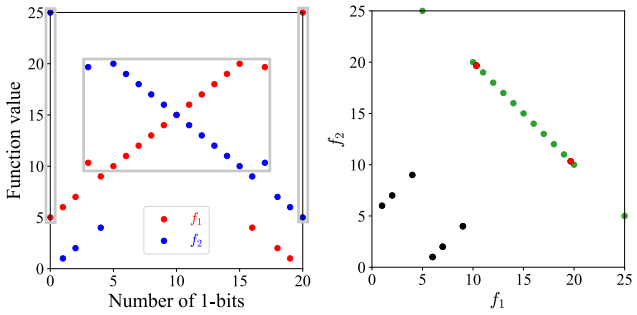

Figure 1: Illustration of the OneJumpZeroJump$_{SS}$ problem when $n = 20$, $k = 5$ and $a = 2$. The left subfigure: the function values f_1 and f_2 vs. the number of 1-bits of a solution; the right subfigure: f_2 vs. f_1.

sample u from uniform distribution over $[0, 1]$;
if $u < 1/2$ **then**
  select a solution $\boldsymbol{x}$ from P uniformly at random;
else
  select a solution $\boldsymbol{x}$ from A uniformly at random;
end if

In this paper, we set the archive reusing rate to $1/2$, which can be adjusted in practice.

Benchmark Problems

Next, we introduce benchmark problems OneJumpZeroJump and its variant OneJumpZeroJump$_{SS}$, studied in this paper. The OneJumpZeroJump problem as presented in Definition 3 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 2023a; Lu, Bian, and Qian 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, with the roles of 1-bits and 0-bits reversed. The Pareto front of the OneJumpZeroJump problem is $\{(c, n + 2k - c) \mid c \in [2k..n] \cup \{k, n + k\}\}$, whose size is $n - 2k + 3$, and the Pareto optimal solution corresponding to $(c, n + 2k - c)$ is any solution with $(c - k)$ 1-bits. We use $S_I^* = \{\boldsymbol{x} \mid |\boldsymbol{x}|_1 \in [k..n - k]\}$ and $F_I^* = \{(c, n + 2k - c) \mid c \in [2k..n]\}$ to denote the internal part of Pareto set and Pareto front, respectively.

Definition 3 (Doerr and Zheng (2021))**.** *The OneJumpZeroJump problem is to find n bits binary strings which maximize*

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

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

where $k \in \mathbb{Z} \wedge 2 \leq k < n/2$, 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.

In this paper, we propose a new variant of OneJumpZeroJump, denoted as OneJumpZeroJump$_{SS}$ (OneJumpZeroJump with stepping stone), as presented in Definition 4. Compared to OneJumpZeroJump, the difference lies in that, among solutions whose number of 1-bits falls in the intervals $[1, \ldots, k-1]$ and $[n-k+1, \ldots, n-1]$, the solutions with $k-a$ and $n-(k-a)$ 1-bits are non-dominated. They lie between the internal Pareto front and the extremal points (i.e., 0^n and 1^n) in the solution space, but their corresponding objective vectors are in a crowded region of the objective space. The right subfigure of Figure 1 shows an example of the objective vectors and the Pareto front. The red points indicate the two newly introduced Pareto front points, which we denote as $G = \{(n - 1/n, 2k + 1/n), (2k + 1/n, n - 1/n)\}$. We also use $S_I^* = \{\boldsymbol{x} \mid |\boldsymbol{x}|_1 \in [k..n - k]\}$ and $F_I^* = \{(c, n + 2k - c) \mid c \in [2k..n]\}$ to denote the internal part of Pareto set and Pareto front. The left subfigure of Figure 1 shows the values of f_1 and f_2 with respect to the number of 1-bits of a solution. The boxed regions represent the Pareto front. We use OneJumpZeroJump$_{SS}$ to characterize a class of problems that some non-dominated solutions are sparse in the solution space, making them helpful for exploration, but crowded in the objective space, reflecting scenarios where the solution space and objective space exhibit inconsistency.

Definition 4. *The OneJumpZeroJump$_{SS}$ problem is to find n bits binary strings which maximize*

$$f_1(\boldsymbol{x}) = \begin{cases} 2k + 1/n, & \text{if } |\boldsymbol{x}|_1 = k - a, \\ n - 1/n, & \text{else if } |\boldsymbol{x}|_1 = n - (k - a), \\ k + |\boldsymbol{x}|_1, & \text{else if } |\boldsymbol{x}|_1 \leq n - k \text{ or } \boldsymbol{x} = 1^n, \\ n - |\boldsymbol{x}|_1, & \text{else}, \end{cases}$$

$$f_2(\boldsymbol{x}) = f_1(\bar{\boldsymbol{x}}),$$

where $a, k \in \mathbb{Z}, 3 \leq k < n/2, 2 \leq a < k$, and $\bar{\boldsymbol{x}} = (1 - x_1, \cdots, 1 - x_n)$.

Provable Benefits of Reusing the Archive

In this section, we prove that reusing the archive is beneficial for SMS-EMOA solving OneJumpZeroJump and OneJumpZeroJump$_{SS}$, compared to using an archive without reuse. Note that SMS-EMOA generates and evaluates only one offspring solution in each generation, thus its running time is equivalent to the number of generations. Due to space limitation, we only give proof sketches, and the full proofs are given in the extended version.

Archive Reuse Enhances Pareto Front Coverage

First, we theoretically show that reusing the archive could enhance Pareto front coverage. In Theorem 1, we prove that when using a small population size $\mu = 2$ and an archive without reuse, the expected number of generations for SMS-EMOA solving OneJumpZeroJump is $n^{\Omega(n)}$. We use this example to illustrate a scenario where, during the early stages of the search, a small population may miss some regions easy to reach, and re-finding these regions later can become difficult as the population becomes more uniformly distributed and stabilizes. The landscape of OneJumpZeroJump (consisting of a plain and two valleys) is simpler than

those of real-world problems, and our analysis relies on a population size $\mu = 2$. For more complex, real-world problems, similar issues may arise with larger populations.

Theorem 1. *For SMS-EMOA solving OneJumpZeroJump with $k \leq n/8$, if using a population size $\mu = 2$ and an archive without reuse, then the expected running time for finding the Pareto front is $n^{\Omega(n)}$.*

Proof Sketch of Theorem 1. We first show that A_1, which denotes the event that the two initial solutions belong to S_I^* (i.e., solutions with number of 1-bits between k and $n-k$) and are located on opposite sides of the central region of the Pareto front (i.e., solutions with $\lfloor n/2 \rfloor$ 1-bits), occurs with probability at least $1-e^{-\Omega(n)}-1/\Omega(\sqrt{n})$. Then conditioned on the occurrence of event A_1, we prove that A_2, which denotes the event that the central point on the Pareto front (i.e., solutions with $\lfloor n/2 \rfloor$ 1-bits) can only be generated from 0^n and 1^n, occurs with probability at least $e^{-8e(e-1)-1/\Omega(n)}$. Finally, conditioned on the occurrence of events A_1 and A_2, we show that the expected number of generations for finding the whole Pareto front is at least $n^{\Omega(n)}$.

For event A_1, by the Chernoff bound and the binomial distribution, we estimate the probability that the two initial solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy $k \leq |\boldsymbol{x}|_1 < \lfloor n/2 \rfloor$ and $\lfloor n/2 \rfloor < |\boldsymbol{y}|_1 \leq n-k$, respectively. For event A_2, since SMS-EMOA always preserves the two boundary solutions with the maximum f_1 and f_2 values, we analyze the probability of increasing f_2 without finding a solution with $\lfloor n/2 \rfloor$ 1-bits, until 0^n is found. A symmetric argument applies for increasing f_1 until 1^n is obtained. Finally, conditioned on the occurrence of both A_1 and A_2, the central point on the Pareto front (i.e., solutions with $\lfloor n/2 \rfloor$ 1-bits) can only be generated from 0^n or 1^n by flipping at least $\lfloor n/2 \rfloor$ bits, which occurs with probability $1/n^{\Omega(n)}$. Thus, the expected running time is at least $\Pr[A_1] \cdot \Pr[A_2] \cdot n^{\Omega(n)} = n^{\Omega(n)}$. □

Through analysis of Theorem 1, we find that using a small population may miss the central region of the Pareto front during the early stages of search, and re-finding these solutions becomes increasingly difficult as the search progresses. In Theorem 2, we prove that when reusing the archive, the expected running time for SMS-EMOA solving OneJumpZeroJump is $O(\mu n^k)$.

Theorem 2. *For SMS-EMOA solving OneJumpZeroJump with $k \geq 3$ and $n - 2k = \Omega(n)$, if using a population size $\mu \geq 2$ and reusing the archive, then the expected running time for finding the Pareto front is $O(\mu n^k)$.*

Proof Sketch of Theorem 2. We divide the evolutionary process into three phases, where the first phase aims at finding one point on internal Pareto front F_I^*, the second phase aims at finding 1^n and 0^n, and the third phase aims at finding the whole internal Pareto front F_I^*.

For the first phase, we consider the case that all the initial solutions have at most $k-1$ or at least $n-k+1$ 1-bits. By the Chernoff bound, this occurs with probability $\exp(-\Omega(n))^\mu$, leading to an expected running time of $O(1)$ to find one point on F_I^*. For the second phase, we prove that the expected running time is $O(\mu n^k)$ by analyzing the process of increasing f_1 value through selecting the parent solution from the population P and generating an offspring with more 1-bits in each step, until 1^n is found. A symmetric bound holds for finding 0^n. For the third phase, we prove that the expected running time is $O(n^2 \log n)$ by analyzing the process of finding the missing points on F_I^* through selecting the parent solution from archive A. Combining the three phases, the total expected running time is $O(\mu n^k)$, where $k \geq 3$. □

Comparing with the result in Theorem 1, the running time for SMS-EMOA solving OneJumpZeroJump reduces from $n^{\Omega(n)}$ to $O(\mu n^k)$ if reusing the archive. Theorem 2 reveals that when using a small population for exploration, reusing the archive allows the algorithm to revisit previously missed regions, namely finding a better-covered Pareto front.

Archive Reuse Helps Exploration

In the last subsection, we have theoretically shown that reusing the archive helps re-find regions missed during early search stages, thus enhances Pareto coverage. Next, we show that reusing the archive can also benefit exploration, thereby finding regions relatively hard to reach. In Theorem 3, we prove that when using a small population size $\mu \leq n-2k+1$ and an archive without reuse, the expected running time for SMS-EMOA solving OneJumpZeroJump$_{SS}$ is $\Omega(n^k)$. For practical MOEAs, e.g., SMS-EMOA, diversity in the objective space always plays an important role in the population update process (i.e., among non-dominated solutions, those in crowded regions are more likely to be removed). This example shows a scenario where some non-dominated solutions are sparse in the solution space and helpful for exploration, but are crowded in the objective space. With a small population and no archive reuse, these useful solutions may not be preserved in the population, limiting the exploration.

Theorem 3. *For SMS-EMOA solving OneJumpZeroJump$_{SS}$ with $k \geq 13$, $a \geq k/4$ and $n - 2k = \Omega(n)$, if using a population size $\mu \leq n - 2k + 1$ and an archive without reuse, then the expected running time for finding the Pareto front is $\Omega(n^k)$.*

Proof Sketch of Theorem 3. We first prove that B_1, which denotes the event that the two boundary solutions of F_I^* (i.e., the solutions with k and $n-k$ 1-bits) are found before a solution $\boldsymbol{x} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = k-a \vee |\boldsymbol{z}|_1 = n-k+a\} \cup \{0^n, 1^n\}$, occurs with probability at least $(1-e^{-\Omega(n)}) \cdot e^{-1/\Omega(n^{1/4})}$. Then conditioned on the occurrence of event B_1, we prove that B_2, which denotes the event that no duplicate objective vectors are maintained in the population before finding a solution $\boldsymbol{x} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = k-a \vee |\boldsymbol{z}|_1 = n-k+a\} \cup \{0^n, 1^n\}$, occurs with probability at least $e^{-1/\Omega(n^{1/4})}$. Finally, conditioned on the occurrence of events B_1 and B_2, we show that the expected number of generations for solving OneJumpZeroJump$_{SS}$ is $\Omega(n^k)$.

For event B_1, we consider the case where all initial solutions have 1-bits between k and $n-k$, and analyze the probability of increasing f_2 value without finding a solution $\boldsymbol{x} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = k-a \vee |\boldsymbol{z}|_1 = n-k+a\} \cup \{0^n, 1^n\}$, until a solution with k 1-bits is found. A symmetric argument

applies for finding a solution with $n-k$ 1-bits. For event B_2, we analyze the probability of generating new points on F_I^* continuously without finding a solution $\boldsymbol{x} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = k-a \vee |\boldsymbol{z}|_1 = n-k+a\} \cup \{0^n, 1^n\}$, until no duplicate objective vectors are maintained in the population. Finally, conditioned on the occurrence of events B_1 and B_2, the newly generated solutions $\boldsymbol{y} \in \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = k-a \vee |\boldsymbol{z}|_1 = n-k+a\}$ cannot be preserved in the population due to its small $\Delta_{\boldsymbol{r}}$ value, until 0^n or 1^n is found. Thus, 0^n (or 1^n) can only be generated from solutions in S_I^* by flipping at least k 1-bits (or 0-bits) with probability at most $1/n^k$. Thus, the expected running time is at least $\Pr[B_1] \cdot \Pr[B_2] \cdot n^k = \Omega(n^k)$. □

Through the analysis of Theorem 3, we observe that using a small population size makes it difficult to retain solutions with objective vectors $(2k+1/n, n-1/n)$ or $(n-1/n, 2k+1/n)$, due to their small $\Delta_{\boldsymbol{r}}$ values. This limits the exploration ability to find 0^n and 1^n. In Theorem 4, we prove that if reusing the archive, the expected number of generations for SMS-EMOA solving OneJumpZeroJump$_{SS}$ is $O(\max\{\mu n^a/k, n^{k-a+1}, n^2 \log n\})$.

Theorem 4. *For SMS-EMOA solving OneJumpZeroJump$_{SS}$ with $n-2k = \Omega(n)$, if using a population size $\mu \geq 2$ and reusing the archive, then the expected running time for finding the Pareto front is $O(\max\{\mu n^a/k, n^{k-a+1}, n^2 \log n\})$.*

Proof Sketch of Theorem 4. We divide the evolutionary process into four phases. The first phase aims at finding one point on F_I^*, the second phase aims at finding one solution $\boldsymbol{x} \in \{1^n\} \cup \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = n-k+a\}$ and one solution $\boldsymbol{y} \in \{0^n\} \cup \{\boldsymbol{z} \mid |\boldsymbol{z}|_1 = k-a\}$, the third phase aims at finding 0^n, 1^n and all points on G, and the fourth phase aims at finding the whole internal Pareto front F_I^*.

For the first phase, the analysis is the same as that in Theorem 2, implying an expected running time of $O(1)$. For the second phase, we prove that the expected running time is $O(\mu n^a/k)$ by analyzing the process of finding a boundary solution $\boldsymbol{x}$ with k 1-bits and then generating a solution $\boldsymbol{y}$ with $k-a$ 1-bits by selecting $\boldsymbol{x}$ from population P and flipping a 1-bits. A symmetric bound holds for finding a solution with $n-k+a$ 1-bits. For the third phase, we prove that the expected running time is $O(n^{k-a+1})$ by analyzing the process of finding 0^n through selecting a solution with $k-a$ 1-bits from the archive A and flipping $k-a$ 1-bits. If 0^n is found in the second phase, a similar analysis applies to finding a solution with $k-a$ 1-bits from 0^n. For the fourth phase, the analysis is the same as that in Theorem 2, implying an expected running time of $O(n^2 \log n)$. Combining the four phases, the total expected running time is $O(\max\{\mu n^a/k, n^{k-a+1}, n^2 \log n\})$. □

Comparing with the result in Theorem 3, when reusing the archive, the expected running time for SMS-EMOA solving OneJumpZeroJump$_{SS}$ reduces from $\Omega(n^k)$ to $O(\max\{\mu n^a/k, n^{k-a+1}, n^2 \log n\})$. Theorem 4 reveals that in scenarios where the distribution of solutions in decision space and objective space is inconsistent, using a small population may lead to the loss of some solutions helpful for exploration, while reusing the archive enables revisiting these helpful solutions, thereby enhancing exploration.

Reusing Archive vs. Using A Large Population

In the last section, we theoretically show that reusing the archive is beneficial compared to using an archive without reuse. Now, we show that reusing the archive with a small population can be better than directly using a large population. Specifically, we further show that when reusing the archive, the upper bound on the expected running time for SMS-EMOA solving OneJumpZeroJump$_{SS}$, OneJumpZeroJump, OneMinMax, and LeadingOnesTrailingZeroes is lower than that of using a large population size. We prove in Theorem 5 that the expected running time for SMS-EMOA solving OneJumpZeroJump$_{SS}$ is $O(\mu \cdot \max\{n^a/k, n^{k-a}\})$ if using a population size $\mu \geq n-2k+5$. The proof idea is to divide the optimization procedure into three phases, where the first phase aims at finding the whole internal Pareto front F_I^*, the second phase aims at finding the two points on G, and the third phase aims at finding 1^n and 0^n. In Theorem 6, Bian et al. (2025) showed that the expected running time for SMS-EMOA solving OneJumpZeroJump is $O(\mu n^k)$ if using a population size $\mu \geq n-2k+3$.

Theorem 5. *For SMS-EMOA solving OneJumpZeroJump$_{SS}$ with $n-2k = \Omega(n)$, if using a population size $\mu \geq n-2k+5$, then the expected running time for finding the Pareto front is $O(\mu \cdot \max\{n^a/k, n^{k-a}\})$.*

Theorem 6 (Bian et al. (2025))**.** *For SMS-EMOA solving OneJumpZeroJump with $n-2k = \Omega(n)$, if using a population size $\mu \geq n-2k+3$, then the expected running time for finding the Pareto front is $O(\mu n^k)$.*

Comparing the results in Theorems 4 and 5, as well as those in 2 and 6, the upper bound on the expected running time for SMS-EMOA solving OneJumpZeroJump$_{SS}$ and OneJumpZeroJump can be reduced by a factor of $\Theta(n)$. The reason is that, with archive reuse, the population size can be constant, rather than the size of the Pareto front (i.e., $n-2k+5$ or $n-2k+3$). On OneJumpZeroJump$_{SS}$, this improvement becomes evident only when $a > k/2$.

Bian et al. (2024) showed that for SMS-EMOA solving OneMinMax and LeadingOnesTrailingZeroes, if using a small population and an archive without reuse, the expected running time can be reduced by a factor of $\Theta(n)$, compared to using a large population size $\mu \geq n+1$ (i.e., the size of Pareto front). Then, we show that archive reuse does not diminish this advantage. Specifically, we consider the case where all Pareto-optimal solutions are generated from parents selected from the population, with each parent selected from the population with probability $1/2$. The rest of the analysis follows that in (Bian et al. 2024), yielding an asymptotically same expected running time. Definitions of OneMinMax and LeadingOnesTrailingZeroes, and full proofs, are provided in the extended version.

Theorem 7 (Zheng et al. (2024b))**.** *Consider using the SMS-EMOA with $\mu \geq n+1$ to optimize the OneMinMax problem. Then after at most $2e\mu n(\ln n + 1)$ iterations in expectation, the Pareto front is covered.*

Theorem 8. *For SMS-EMOA solving OneMinMax, if using a population size $\mu \geq 2$, one-point crossover and reusing the archive, then the expected running time for finding the Pareto front is $O(\mu n \log n)$.*

size n	SMS-EMOA$_L$	SMS-EMOA$_A$	SMS-EMOA$_{AR}$
15	3032.58	59244.63	2991.15
20	9984.90	141679.07	6075.74
25	20410.68	296777.68	9891.29
30	34534.04	517542.01	15485.38

Table 2: Average number of generations over 1,000 independent runs for SMS-EMOA$_L$ (a large population size), SMS-EMOA$_A$ (archive only to store non-dominated solutions), and SMS-EMOA$_{AR}$ (archive with reuse) solving OneJumpZeroJump$_{SS}$ with $k = 3$ and $a = 2$.

Theorem 9 (Zheng et al. (2024b))**.** *Consider using the SMS-EMOA with $\mu \geq n + 1$ to optimize the LeadingOnesTrailingZeroes problem. Then after at most $2e\mu n^2$ iterations in expectation, the Pareto front is covered.*

Theorem 10. *For SMS-EMOA solving LeadingOnesTrailingZeroes, if using a population size $\mu \geq 2$, one-point crossover and reusing the archive, then the expected running time for finding the Pareto front is $O(\mu n^2 + \mu^2 n \log n)$.*

The expected running time for SMS-EMOA (without using an archive) solving OneMinMax and LeadingOnesTrailingZeroes has been shown to be $O(\mu n \log n)$ and $O(\mu n^2)$ in Theorems 7 and 9, respectively, where the population size $\mu \geq n + 1$ (Zheng and Doerr 2024b). Thus, our results in Theorems 8 and 10 show that if a constant population size is used for SMS-EMOA reusing the archive, the expected running time can be reduced by a factor of $\Theta(n)$.

Experiments

In this section, we conduct experiments on two artificial problems OneJumpZeroJump and OneJumpZeroJump$_{SS}$ in this paper, and 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 2003a) and NK-landscapes (NK) (Aguirre and Tanaka 2004), following the settings in (Li et al. 2024). Table 2 presents the results of SMS-EMOA solving OneJumpZeroJump$_{SS}$ with problem size $n \in \{15, 20, 25, 30\}$, $k = 3$, and $a = 2$. We present the average number of generations over 1000 runs for three variants: 1) SMS-EMOA with a large population size $\mu = n - 2k + 5$, 2) SMS-EMOA with an archive storing all non-dominated solutions ($\mu = 5$), and 3) SMS-EMOA with archive reuse ($\mu = 5$). Results show that reusing the archive significantly reduces the running time compared to using an archive only to store non-dominated solutions, and using a large population size. Similar results can be seen on OneJumpZeroJump in the extended version.

As for the four practical problems, each problem is instantiated at three sizes: 100, 200, and 500 variables. We conduct experiments using three SMS-EMOA variants: 1) With large population size ($\mu = 1000$), 2) Archive without reuse ($\mu = 200$), and 3) Archive with reuse ($\mu = 200$). These variants follow common parameter settings: 1) the parent selection uses uniform random selection; 2) the crossover and mutation operators are problem-specific. For pseudo-Boolean problems (KP, NK), we use the uniform crossover (Eiben and Smith 2015) with a rate of 1.0 and the bit-wise mutation (Eiben and Smith 2015) with a rate of $1/n$, where n is the number of variables. For TSP, we use the order-based crossover and the 2-opt mutation (Eiben and Smith 2015). For QAP, we use the cycle-based crossover and the 2-swap mutation (Eiben and Smith 2015). For both TSP and QAP, the crossover rate is 1.0 and the mutation rate is 0.05, following the practice in (Li et al. 2024). The budget for the algorithms is 10,000,000 fitness evaluations and the number of independent runs is set to 30. Table 3 presents the result of a widely used indicator, Hypervolume (HV) (Zitzler and Thiele 1999) of the three SMS-EMOA variants on the four problems. To compute the HV reference point, we sample 100,000 random solutions from the decision space and use the worst objective values among the non-dominated ones. Table 3 shows that reusing the archive can lead to a better mean HV across all problem instances.

"†" indicates that SMS-EMOA$_A$ or SMS-EMOA$_L$ is significantly worse than SMS-EMOA$_{AR}$ according to the Wilcoxon rank-sum test ($\alpha = 0.05$).

Problem	SMS-EMOA$_L$	SMS-EMOA$_A$	SMS-EMOA$_{AR}$
KP$_{100}$	2.19e6 (3.2e4)†	2.24e6 (9.3e3)†	**2.25e6** (7.7e3)
KP$_{200}$	6.44e6 (8.4e4)†	6.86e6 (2.9e4)†	**6.88e6** (3.7e4)
KP$_{500}$	4.20e7 (4.9e5)†	4.52e7 (3.1e5)†	**4.55e7** (3.3e5)
NK$_{100}$	6.15e-2 (3.7e-3)†	7.06e-2 (3.0e-3)†	**7.23e-2** (3.6e-3)
NK$_{200}$	6.04e-2 (3.3e-3)†	7.24e-2 (2.8e-3)	**7.36e-2** (2.6e-3)
NK$_{500}$	4.45e-2 (2.3e-3)†	5.91e-2 (1.7e-3)	**5.92e-2** (1.7e-3)
QAP$_{100}$	2.80e15 (9.8e13)†	3.03e15 (1.4e14)†	**3.16e15** (1.4e14)
QAP$_{200}$	2.28e16 (6.3e14)†	2.66e16 (8.3e14)†	**2.76e16** (8.7e14)
QAP$_{500}$	2.29e17 (7.3e15)†	3.16e17 (8.2e15)†	**3.30e17** (6.4e15)
TSP$_{100}$	2.18e3 (4.0e1)†	2.67e3 (1.7e1)	**2.68e3** (2.0e1)
TSP$_{200}$	6.15e3 (1.8e2)†	9.03e3 (9.2e1)	**9.05e3** (7.2e1)
TSP$_{500}$	2.35e4 (1.1e3)†	4.87e4 (8.4e2)†	**4.92e4** (6.7e2)

Table 3: Hypervolume (Zitzler and Thiele 1999) results (mean and standard deviation) of SMS-EMOA$_L$ (a large population size), SMS-EMOA$_A$ (archive without reuse), and SMS-EMOA$_{AR}$ (archive with reuse) on practical problems. The number in each problem name denotes the number of variables. The best mean per row is highlighted in bold.

Conclusion

This paper analytically shows that apart from storing non-dominated solutions, the archive can be reused to benefit the search of MOEAs. We prove that for SMS-EMOA solving OneJumpZeroJump and one of its variants, reusing the archive is significantly superior in terms of expected running time compared to merely using an archive to store non-dominated solutions. The analysis shows that reusing the archive brings benefits in two ways: 1) it enables revisiting regions missed in the early stages of the search, thus helping to find a better-covered Pareto front; and 2) it alleviates the issue of small populations losing potentially promising non-dominated solutions due to inconsistencies of solution distribution between the decision and objective spaces, thereby enhancing exploration ability. We also prove that reusing archive solutions can be better than using a large population size directly. These theoretical findings are empirically validated on both artificial problems and practical problems.

Acknowledgements

This work was supported by the National Science and Technology Major Project (2022ZD0116600), the National Science Foundation of China (62276124), and the Fundamental Research Funds for the Central Universities (14380020). Chao Qian is the corresponding author.

References

Aguirre, H. E.; and Tanaka, K. 2004. Effects of elitism and population climbing on multiobjective MNK-landscapes. In *CEC*, volume 1, 449–456. Portland, OR.

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

Bian, C.; and Qian, C. 2022. Better running time of the non-dominated sorting genetic algorithm II (NSGA-II) by using stochastic tournament selection. In *PPSN*, 428–441. Dortmund, Germany.

Bian, C.; Qian, C.; and Tang, K. 2018. A general approach to running time analysis of multi-objective evolutionary algorithms. In *IJCAI*, 1405–1411. Stockholm, Sweden.

Bian, C.; Ren, S.; Li, M.; and Qian, C. 2024. An archive can bring provable speed-ups in multi-objective evolutionary algorithms. In *IJCAI*, 6905–6913. Jeju Island, South Korea.

Bian, C.; Zhou, Y.; Li, M.; and Qian, C. 2025. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. *Artificial Intelligence*, 341: 104308.

Brockhoff, D.; and Tušar, T. 2019. Benchmarking algorithms from the platypus framework on the biobjective bbob-biobj testbed. In *GECCO*, 1905–1911. Prague, Czech Republic.

Cerf, S.; Doerr, B.; Hebras, B.; Kahane, Y.; and Wietheger, S. 2023. The first proven performance guarantees for the non-dominated sorting genetic algorithm II (NSGA-II) on a combinatorial optimization problem. In *IJCAI*, 5522–5530. Macao, SAR, China.

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

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

Dang, D.-C.; Opris, A.; and Sudholt, D. 2024. Level-based theorems for runtime analysis of multi-objective evolutionary algorithms. In *PPSN*, 246–263. Hagenberg, Austria.

Deb, K. 2001. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley.

Deb, K.; and Jain, H. 2014. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4): 577–601.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197.

Deb, K.; Thiele, L.; Laumanns, M.; and Zitzler, E. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization: theoretical advances and applications*, 105–145. Springer.

Doerr, B.; Ivan, T.; and Krejca, M. S. 2025. Speeding up the NSGA-II with a simple tie-breaking rule. In *AAAI*, 25, 26964–26972. Philadelphia, Pennsylvania.

Doerr, B.; Kodric, B.; and Voigt, M. 2013. Lower bounds for the runtime of a global multi-objective evolutionary algorithm. In *CEC*, 432–439. Cancun, Mexico.

Doerr, B.; Krejca, M. S.; and Rudolph, G. 2025. Runtime analysis for multi-objective evolutionary algorithms in unbounded integer spaces. In *AAAI*, 25, 26955–26963. Philadelphia, Pennsylvania.

Doerr, B.; and Neumann, F. 2020. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*. Springer.

Doerr, B.; and Qu, Z. 2023a. A first runtime analysis of the NSGA-II on a multimodal problem. *IEEE Transactions on Evolutionary Computation*, 27(5): 1288–1297.

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

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

Doerr, B.; and Zheng, W. 2021. Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In *AAAI*, 12293–12301. Virtual.

Eiben, A. E.; and Smith, J. E. 2015. *Introduction to Evolutionary Computing*. Heidelberg, Germany: Springer Berlin Heidelberg.

Fieldsend, J. E.; Everson, R. M.; and Singh, S. 2003. Using unconstrained elite archives for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 7(3): 305–323.

Figueira, J.; Fonseca, C. M.; Halffmann, P.; Klamroth, K.; Paquete, L.; Ruzika, S.; Schulze, B.; Stiglmayr, M.; and Willems, D. 2017. Easy to say they are hard, but hard to see they are easy—towards a categorization of tractable multi-objective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 24(1-2): 82–98.

Friedrich, T.; and Neumann, F. 2015. Maximizing submodular functions under matroid constraints by evolutionary algorithms. *Evolutionary Computation*, 23(4): 543–558.

Giel, O.; and Lehre, P. K. 2010. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18(3): 335–356.

Huang, Z.; Zhou, Y.; Luo, C.; and Lin, Q. 2021. A runtime analysis of typical decomposition approaches in MOEA/D framework for many-objective optimization problems. In *IJCAI*, 1682–1688. Virtual.

Ishibuchi, H.; Pang, L. M.; and Shang, K. 2020. A new framework of evolutionary multi-objective algorithms with an unbounded external archive. In *ECAI 2020*, 283–290. Santiago, Spain.

Knowles, J.; and Corne, D. 2003a. Instance generators and test suites for the multiobjective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization*, 295–310.

Knowles, J.; and Corne, D. 2003b. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 7(2): 100–116.

Krause, O.; Glasmachers, T.; Hansen, N.; and Igel, C. 2016. Unbounded population MO-CMA-ES for the bi-objective BBOB test suite. In *GECCO*, 1177–1184. Denver, CO.

Laumanns, M.; Thiele, L.; and Zitzler, E. 2004a. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing*, 3: 37–51.

Laumanns, M.; Thiele, L.; and Zitzler, E. 2004b. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8(2): 170–182.

Li, M.; Han, X.; Chu, X.; and Liang, Z. 2024. Empirical comparison between MOEAs and local search on multi-objective combinatorial optimisation problems. In *GECCO*, 547–556. ACM.

Li, M.; López-Ibáñez, M.; and Yao, X. 2024. Multi-objective archiving. *IEEE Transactions on Evolutionary Computation*, 28(3): 696–717.

Li, M.; and Yao, X. 2019. An empirical investigation of the optimality and monotonicity properties of multiobjective archiving methods. In *International conference on evolutionary multi-criterion optimization*, 15–26.

Lu, T.; Bian, C.; and Qian, C. 2024. Towards running time analysis of interactive multi-objective evolutionary algorithms. In *AAAI*, 20777–20785. Vancouver, Canada.

Neumann, F. 2007. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181(3): 1620–1629.

Neumann, F.; and Theile, M. 2010. How crossover speeds up evolutionary algorithms for the multi-criteria all-pairs-shortest-path problem. In *PPSN*, 667–676. Krakow, Poland.

Opris, A. 2025. A first runtime analysis of the PAES-25: An enhanced variant of the Pareto archived evolution strategy. *CORR abs/2507.03666*.

Opris, A.; Dang, D.-C.; and Sudholt, D. 2024. Runtime analyses of NSGA-III on many-objective problems. In *GECCO*, 1596–1604. Melbourne, Australia.

Qian, C.; Shi, J.-C.; Yu, Y.; Tang, K.; and Zhou, Z.-H. 2017. Subset selection under noise. In *NIPS*, 3562–3572. Long Beach, CA.

Qian, C.; Tang, K.; and Zhou, Z.-H. 2016. Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization. In *PPSN*, 835–846. Edinburgh, Scotland.

Qian, C.; Yu, Y.; Tang, K.; Yao, X.; and Zhou, Z.-H. 2019. Maximizing submodular or monotone approximately submodular functions by multi-objective evolutionary algorithms. *Artificial Intelligence*, 275: 279–294.

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

Qian, C.; Yu, Y.; and Zhou, Z.-H. 2015. Subset selection by Pareto optimization. In *NIPS*, 1765–1773. Montreal,Canada.

Ren, S.; Bian, C.; Li, M.; and Qian, C. 2024a. A first running time analysis of the strength Pareto evolutionary algorithm 2 (SPEA2). In *PPSN*, 295–312. Hagenberg, Austria.

Ren, S.; Liang, Z.; Li, M.; and Qian, C. 2025. A theoretical perspective on why stochastic population update needs an archive in evolutionary multi-objective optimization. In *IJCAI*. Montreal, Canada.

Ren, S.; Qiu, Z.; Bian, C.; Li, M.; and Qian, C. 2024b. Maintaining diversity provably helps in evolutionary multimodal optimization. In *IJCAI*, 7012–7020. Jeju Island, South Korea.

Ribeiro, C. C.; Hansen, P.; Borges, P. C.; and Hansen, M. P. 2002. A study of global convexity for a multiple objective travelling salesman problem. *Essays and Surveys in Metaheuristics*, 129–150.

Teghem, J. 1994. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*.

Wietheger, S.; and Doerr, B. 2023. A mathematical runtime analysis of the non-dominated sorting genetic algorithm III (NSGA-III). In *IJCAI*, 5657–5665. Macao, SAR, China.

Wietheger, S.; and Doerr, B. 2024. Near-tight runtime guarantees for many-objective evolutionary algorithms. In *PPSN*, 153–168. Hagenberg, Austria.

Zhang, Q.; and Li, H. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6): 712–731.

Zheng, W.; and Doerr, B. 2024a. Runtime analysis for the NSGA-II: Proving, quantifying, and explaining the inefficiency for many objectives. *IEEE Transactions on Evolutionary Computation*, 28(5): 1442–1454.

Zheng, W.; and Doerr, B. 2024b. Runtime analysis of the SMS-EMOA for many-objective optimization. In *AAAI*, 20874–20882. Vancouver, Canada.

Zheng, W.; Liu, Y.; and Doerr, B. 2022. A first mathematical runtime analysis of the non-dominated sorting genetic algorithm II (NSGA-II). In *AAAI*, 10408–10416. Virtual.

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

Zitzler, E.; and Thiele, L. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4): 257–271.