Theoretical Analysis of Multi-Objective Evolutionary Algorithms on Integer Spaces with Local Optima

Yuetong Sun[1,2], **Zeqiong Lv**[1], **Shengjie Ren**[1], **Zimin Liang**[3], **Miqing Li**[3] 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]Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K.
qianc@nju.edu.cn

Abstract

Multi-objective evolutionary algorithms (MOEAs) are popular tools for multi-objective optimization (MOO), and have been successfully applied to many real-world MOO problems. However, the theoretical study has lagged behind their practical success and remains largely confined to synthetic pseudo-Boolean functions. To close this gap, this paper–drawing inspiration from a class of popular continuous problems with real-world relevance–introduces a multi-objective benchmark defined on an integer space, featuring an analyzable landscape and the presence of local optima. We conduct a running time analysis on the proposed benchmark and derive several theoretical results. Specifically, we prove that a widely-studied MOEA, GSEMO, using unit-step mutation can be trapped in local optimal regions and fail to identify the Pareto front. Fortunately, we find that this difficulty can be overcome either by incorporating an ageing mechanism or using heavy-tailed mutations that allow multi-valued changes along each dimension of an individual. In addition, we demonstrate the extendability of the proposed benchmark to more complex landscapes with numerous local optima, resembling well-established problems in the field (e.g., those from the ZDT and DTLZ suites). We hope this work is a step forward for the theoretical study of MOEAs on problems that are closely related to those commonly investigated in empirical research.

1 Introduction

Multi-objective optimization (MOO) problems arise in many real-world applications, where multiple conflicting objectives must be optimized simultaneously. The goal in MOO is to identify a set of Pareto-optimal solutions that represent trade-offs among conflicting objectives. Multi-objective evolutionary algorithms (MOEAs) have become one of the most widely used approaches for solving MOO problems [Deb, 2001; Zhou *et al.*, 2019]. By maintaining a population of candidate solutions, MOEAs can approximate the Pareto front in a single run. Well-established MOEAs include the Global Simple Evolutionary Multi-objective Optimizer (GSEMO) [Giel, 2003], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb *et al.*, 2002], $\mathcal{S}$-Metric Selection Evolutionary Multi-objective Optimization Algorithm (SMS-EMOA) [Beume *et al.*, 2007], MOEA based on Decomposition (MOEA/D) [Zhang and Li, 2007], and SPEA2 [Zitzler *et al.*, 2001]. These MOEAs have been successfully applied to many real-world domains, such as neural architecture search [Lu *et al.*, 2019] and production scheduling [Ojstersek *et al.*, 2020]. Despite their widespread use, theoretical study of MOEAs has lagged behind practical applications.

In recent years, theoretical analysis of MOEAs, particularly running time analysis, has attracted considerable attention, aiming to reveal, for example, which evolutionary mechanisms are provably effective and under which conditions they ensure efficient optimization [Auger and Doerr, 2011; Neumann and Witt, 2010; Zhou *et al.*, 2019; Doerr and Neumann, 2020]. Early studies mainly focused on simple MOEAs (e.g., GSEMO), while recent work shifted to practical MOEAs, including MOEA/D [Li *et al.*, 2015; Corus *et al.*, 2025], NSGA-II [Zheng *et al.*, 2022], SPEA2 [Ren *et al.*, 2024], SMS-EMOA [Bian *et al.*, 2025], and NSGA-III [Wietheger and Doerr, 2024a]. Beyond the algorithm framework, theoretical studies have investigated the influence of specific components of MOEAs, such as selection mechanisms [Doerr *et al.*, 2025a; Ren *et al.*, 2026], archiving [Bian *et al.*, 2024; Ren *et al.*, 2025; Ren *et al.*, 2026], population update mechanisms [Bian *et al.*, 2025], mutation operators [Doerr *et al.*, 2025b], and crossover operators [Doerr and Qu, 2023; Dang *et al.*, 2024a], thereby providing insights for efficient algorithm design. At the problem-setting level, theoreticians [Wietheger and Doerr, 2024b; Opris, 2025] have analyzed many-objective optimization problems (with more than two objectives), aiming to explain the increasing difficulty of MOEAs as the number of objectives grows.

However, these studies in MOEAs are primarily focused on synthetic pseudo-Boolean functions, such as ONEMINMAX [Giel and Lehre, 2010], LEADINGONESTRAILINGZEROES [Laumanns *et al.*, 2004b], COUNTONESCOUNTZEROS [Laumanns *et al.*, 2004b], and ONEJUMPZEROJUMP [Doerr and Zheng, 2021] benchmarks, all of which are obtained by extending classical single-objective functions [Liang and Li, 2025]. For instance, ONEMINMAX simultaneously maximizes the number of 1-bits (extended from

ONEMAX [Droste *et al.*, 2002]) and the number of 0-bits of a binary bit string; LEADINGONESTRAILINGZEROES simultaneously maximizes the number of leading 1-bits (extended from LEADINGONES [Rudolph, 1997]) and the number of trailing 0-bits; and ONEJUMPZEROJUMP generalizes the JUMP [Doerr and Neumann, 2020] function to a bi-objective setting. There also exist other benchmarks, such as ONETRAPZEROTRAP [Dang *et al.*, 2024b], and multi-objective REALROYALROAD [Dang *et al.*, 2023]. Beyond these synthetic benchmarks, only a few studies consider MOEAs on multi-objective combinatorial optimization problems, such as multi-objective knapsack [Laumanns *et al.*, 2004a], minimum spanning tree [Cerf *et al.*, 2023], and shortest path [Horoba, 2009].

Practical MOO problems, however, are not limited to pseudo-Boolean domains, which can lead to MOEAs exhibiting behaviors that differ substantially from those observed on synthetic problems [Liang and Li, 2026]. Recently, there has been progress in extending the running time analysis of MOEAs to integer spaces. To our knowledge, the only work in this vein is from [Doerr *et al.*, 2025b] which analyzes a benchmark with an unbounded integer space (presented by [Rudolph, 2023]). This problem is defined over n-dimensional integer vectors and aims to minimize the distance of a solution to two target points. It induces a simple, unimodal fitness landscape in which the Pareto optimal solutions lie on the line connecting the two target points, and any move towards the Pareto set results in an improvement in fitness. Doerr et al. [2025b] analyzed the performance of (G)SEMO with different mutation operators, and showed that unit-step mutation can be slow when initial solutions are far from the Pareto front, whereas exponential-tail and power-law mutations yield consistently better performance across different problem parameters and starting points.

In this paper, we present a class of integer-valued test functions that can incorporate diverse problem features, serving as a benchmark for the theoretical analysis of MOEAs on integer spaces. The main contributions are as follows:

- We introduce a class of multi-objective integer optimization functions, termed the L1DM-2SIP problem, which has an analyzable structure as well as practical relevance. This benchmark is inspired by the multi-point distance minimization problem (MP-DMP) [Köppen and Yoshida, 2007; Ishibuchi *et al.*, 2010; Fieldsend *et al.*, 2021] in the continuous domain, which has been used to visually investigate the behavior of MOEAs. We instantiate a simple function of L1DM-2SIP that features local-optimal regions (Definition 5) and thus complements the current simple unimodal benchmark [Rudolph, 2023; Doerr *et al.*, 2025b].

  Compared to pseudo-Boolean benchmarks, the proposed benchmark exhibits a high degree of consistency between the solution and objective spaces. This consistency, arising from the L1 distance linking solution positions with dominance relations in objective space, makes the search process more intuitive and traceable in the solution space, especially when progressing from the local Pareto optimal set to the global Pareto optimal set.

  Note that the presented L1DM-2SIP problem is of practical relevance and can model real-world problems such as site location selection in urban planning, where the goal is to identify possible sites that simultaneously minimize the Manhattan distances to multiple facilities.
- For the considered L1DM-2SIP function, we prove in Theorem 1 that the widely-studied GSEMO with unit-step mutation [Doerr *et al.*, 2025b] can be trapped in local-optimal regions and fails to identify the Pareto front. However, this difficulty can be overcome by introducing an ageing mechanism [Li *et al.*, 2025] (as shown in Theorem 2) or employing heavy-tailed mutation operators, including exponential-tail and power-law mutations [Doerr *et al.*, 2025b] (as shown in Theorem 3), which enables GSEMO escape local-optimal regions and successfully identify the Pareto front within a bounded expected running time.
- To show the extendability of the presented benchmark, we also provide an additional instantiation with a more complex landscape featuring numerous local optima (Definition 6). This instantiation more closely resembles the landscape characteristics of multi-modal problems commonly used in empirical research, such as those in the ZDT [Zitzler *et al.*, 2000] and DTLZ [Deb *et al.*, 2005] suites.

Finally, we also conduct empirical studies to complement the theoretical results. We hope this work is a step forward for the theoretical study of MOEAs on problems that are closely related to those commonly investigated in empirical research.

2 Proposed Benchmark L1DM-2SIP

In this section, we first introduce the basic concepts of multi-objective optimization, and then describe the bi-objective optimization problem in integer space studied by Doerr *et al.* [2025b]. Finally, we present a new integer-valued benchmark problem L1DM-2SIP, and provide one of its problem instances. In this paper, we use $[p..q]$ to represent $[p, q] \cap \mathbb{Z}$ for $p \leq q$ and $p, q \in \mathbb{Z}$.

2.1 Multi-objective Optimization

Multi-objective optimization aims to simultaneously optimize two or more objective functions. In this paper, we focus on minimization. In multi-objective optimization, 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 given 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 forms the *Pareto front* (PF). The goal of multi-objective optimization is to find the Pareto front or a good approximation of it. In addition to global optimality, we also consider *local optimality* [Lapucci *et al.*, 2023]. Given a neighborhood structure on $\mathcal{X}$, a solution is *locally Pareto-optimal* if it is not dominated by any of its neighboring solutions. The definition of neighboring solutions studied in this paper is given by Definition 3. That is, two solutions are neighboring if and only if they differ by 1 in exactly one dimension. Note that a locally Pareto-optimal solution is not necessarily Pareto-optimal.

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

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

Definition 2 (Local Pareto Optimality)**.** *Let $\mathcal{N}(\boldsymbol{x}) \subseteq \mathcal{X}$ denote the neighborhood of a solution $\boldsymbol{x} \in \mathcal{X}$. A solution $\boldsymbol{x}$ is called* locally Pareto-optimal *(with respect to $\mathcal{N}$) if there exists no solution $\boldsymbol{y} \in \mathcal{N}(\boldsymbol{x})$ such that $\boldsymbol{y} \prec \boldsymbol{x}$.*

Definition 3 (ℓ_∞-Neighborhood)**.** *For $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{Z}^n$, we have*

$$\mathcal{N}_\infty(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{X} \mid \|\boldsymbol{y} - \boldsymbol{x}\|_\infty = 1\}.$$

2.2 Existing Integer-Space Benchmark

Doerr *et al.* [2025b] conducted the first running time analysis of (G)SEMO with an unbounded integer search space, using the problem benchmark (shown in Definition 4) presented by Rudolph [2023]. Their results showed that unit-step mutation is slow when starting far from the PF, while exponential-tail and power-law mutations perform better.

Definition 4 ([Rudolph, 2023])**.** *Consider the bi-objective optimization problem $\boldsymbol{f}_g : \mathbb{Z}^n \to \mathbb{N}_0^2$ with*

$$\boldsymbol{f}_g(\boldsymbol{x}) = \begin{pmatrix} \|\boldsymbol{x} - \boldsymbol{z}_1\|_1 \\ \|\boldsymbol{x} - \boldsymbol{z}_2\|_1 \end{pmatrix},$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm, $\boldsymbol{z}_1 = (c, 0, ..., 0)^{\mathrm{T}}$, and $\boldsymbol{z}_2 = (-c, 0, ..., 0)^{\mathrm{T}}$.

The Pareto set of $\boldsymbol{f}_g$ is $X_g^* = \{(k, 0, \dots, 0)^{\mathrm{T}} \in \mathbb{Z}^n \mid k \in [-c..c]\}$, and the Pareto front of $\boldsymbol{f}_g$ is $F_g^* = \{(k, 2c - k)^{\mathrm{T}} \in \mathbb{N}^2 \mid k \in [0..2c]\}$, where $|F^*| = 2c + 1$. This problem is simple and exhibits a unimodal global fitness landscape, in which every point that moves towards the Pareto set achieves an improvement in fitness. Next, we will introduce a new benchmark with local optima.

2.3 Proposed Integer-Space Benchmark L1DM-2SIP

We consider a multi-objective optimization problem that simultaneously minimizes the L1-distance regarding a point to two sets of m points in an integer spac, called L1-distance minimization to two sets of integer points (L1DM-2SIP). Each set contains multiple candidate points, and typically each point in one set is paired with a corresponding point in the other set. One instance of the proposed L1DM-2SIP with two pairs of points is given as follows.

Definition 5 (L1DM-2SIP: instance I)**.** *Consider the problem of L1-distance minimization to two sets ($\{\boldsymbol{z}_1, \boldsymbol{u}_1\}$ and $\{\boldsymbol{z}_2, \boldsymbol{u}_2\}$) of integer points: $\mathbb{Z}^n \to \mathbb{N}_0^2$ with*

$$\textit{L1DM-2SIP}(\boldsymbol{x}) = \begin{pmatrix} \min\{\|\boldsymbol{x} - \boldsymbol{z}_1\|_1, \|\boldsymbol{x} - \boldsymbol{u}_1\|_1\} \\ \min\{\|\boldsymbol{x} - \boldsymbol{z}_2\|_1, \|\boldsymbol{x} - \boldsymbol{u}_2\|_1\} \end{pmatrix},$$

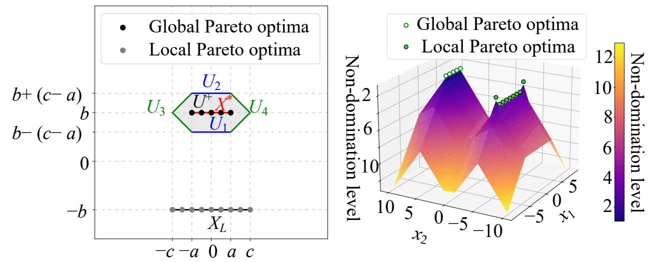

Figure 1: An instance of L1DM-2SIP (Definition 5) in the decision space, and its corresponding fitness landscape, shown by Pareto Landscape [Liang *et al.*, 2024], where $a = 2$, $b = 5$, and $c = 4$.

where $\|\cdot\|_1$ denotes the ℓ_1-norm, $\boldsymbol{z}_1 = (-c, -b, 0, ..., 0)^{\mathrm{T}}$, $\boldsymbol{z}_2 = (c, -b, 0, ..., 0)^{\mathrm{T}}$, $\boldsymbol{u}_1 = (-a, b, 0, ..., 0)^{\mathrm{T}}$, $\boldsymbol{u}_2 = (a, b, 0, ..., 0)^{\mathrm{T}}$, $a < c$, and $a + 2b - c > 1$.

Under the conditions of $a < c$ and $a + 2b - c > 1$, L1DM-2SIP admits both Pareto-optimal and locally Pareto-optimal solutions. The set of locally Pareto-optimal solutions is given by $X_L = \{(x_1, -b, 0, \dots, 0)^{\mathrm{T}} \in \mathbb{Z}^n \mid x_1 \in [-c..c]\}$, with $|X_L| = 2c + 1$. The Pareto set of L1DM-2SIP is $X^* = \{(x_1, b, 0, \dots, 0)^{\mathrm{T}} \in \mathbb{Z}^n \mid x_1 \in [-a..a]\}$, and the Pareto front is $F^* = \{(k, 2a - k)^{\mathrm{T}} \in \mathbb{N}^2 \mid k \in [0..2a]\}$, with $|X^*| = |F^*| = 2a + 1$. Figure 1 shows an example of the above function with two decision variables, along with its fitness landscape under the Pareto dominance relation (i.e., Pareto landscape [Liang *et al.*, 2024]).

In the left sub-figure of Figure 1, $U_1 = \{(x_1, b - c + a, 0, \cdots, 0)^T \mid x_1 \in [-a..a]\}$ and $U_2 = \{(x_1, b + c - a, 0, \cdots, 0)^T \mid x_1 \in [-a..a]\}$ (blue lines) have the same objective values as $X_{L_a} = \{(x_1, -b, 0, \dots, 0)^T \in \mathbb{Z}^n \mid x_1 \in [-a..a]\}$ (i.e., the set of solutions that satisfy $x_1 \in [-a..a]$ in X_L and also a mapping of U_1 onto X_L). The solutions in $U_3 = \{(x_1, x_2, 0, \cdots, 0)^T \mid x_1 \in [-c.. - a], x_2 \in \{b - c - x_1, b + c + x_1\}\}$ (the green line on the left) have the same objective values as $(-c, b, 0, \cdots, 0)^T$, and the solutions in $U_4 = \{(x_1, x_2, 0, \cdots, 0)^{\mathrm{T}} \mid x_1 \in [a..c], x_2 \in \{b - c + x_1, b + c - x_1\}\}$ (the green line on the right) have the same objective values as $(c, b, 0, \cdots, 0)^T$. Thus, the solutions in the boundary $U = U_1 \cup U_2 \cup U_3 \cup U_4$ (the boundary formed by the blue and green lines) and X_L are incomparable or have the same objective values. The internal region enclosed by U is denoted as U^+ (the shaded area), which dominates the local Pareto optimal set X_L. That is, any solution in U^+ is not dominated by the local Pareto optimal solutions in X_L, while for any local Pareto optimal solution in X_L, there exist solutions in U^+ dominating it.

It is worth noting that when $c - a \ge 3$, two new sets of local Pareto optimal solutions X_{L_1} and X_{L_2} appear, where X_{L_1} consists of all $(x_1, x_2, \dots, x_n)^{\mathrm{T}} \in \mathbb{Z}^n$ with $-c \le x_1 \le -a - 3$ and $\lceil \frac{a-c}{2} \rceil + 1 \le x_2 \le \lceil \frac{a-c}{2} \rceil - x_1 - a - 2$, and X_{L_2} consists of all $(x_1, x_2, \dots, x_n)^{\mathrm{T}} \in \mathbb{Z}^n$ with $a + 3 \le x_1 \le c$ and $\lceil \frac{a-c}{2} \rceil + 1 \le x_2 \le \lceil \frac{a-c}{2} \rceil + x_1 - a - 2$. However, X_{L_1} and X_{L_2} do not form a peak and will not affect the analysis of GSEMO, so the local Pareto optimal solutions mentioned later all refer to X_L.

3 Running Time Analysis of GSEMO on L1DM-2SIP

In this section, we analyze the expected running time of GSEMO (an MOEA widely used in theoretical analysis of MOEAs [Neumann and Witt, 2010; Zhou *et al.*, 2019]) with the same mutation operators as those used in [Doerr *et al.*, 2025b] for solving the proposed integer-space benchmark L1DM-2SIP. As presented in Algorithm 1, GSEMO maintains a population of unfixed size to store all solutions that are not dominated by any other solution encountered so far. It starts with an initial solution randomly selected from the solution space. In each iteration, a solution $\boldsymbol{x}$ is randomly selected from the current population, and a new solution $\boldsymbol{y}$ is generated by applying mutation to $\boldsymbol{x}$. If no solution in the current population dominates $\boldsymbol{y}$, all individuals weakly dominated by $\boldsymbol{y}$ will be deleted from the population and $\boldsymbol{y}$ is added to the population.

For the mutation operator, we consider full-dimensional mutation, which independently modifies each dimension x_i of a solution $\boldsymbol{x}$ with probability $1/n$. That is, each dimension y_i of the offspring solution $\boldsymbol{y}$ generated from $\boldsymbol{x}$ is

$$y_i = \begin{cases} x_i + Z_i, & \text{with probability } 1/n, \\ x_i, & \text{otherwise,} \end{cases} \tag{1}$$

where Z_i is an integer-valued random variable whose distribution depends on the chosen mutation operator.

Lemma 1 shows the maximum population size maintained by GSEMO during the optimization of L1DM-2SIP, which will be used in the proofs of Theorems 2 and 3. The proof of Lemma 1 proceeds by first dividing the solution space into different regions based on whether the solution is closer to $\boldsymbol{z}_1$ ($\boldsymbol{z}_2$) or $\boldsymbol{u}_1$ ($\boldsymbol{u}_2$), the integer points used in the definition of L1DM-2SIP. Then, in each region, the size of the non-dominated solution set that can be retained in the population is determined based on the already established form of the objective functions. Since there are dominance relationships between the non-dominated solution sets in different regions, the size of the largest non-dominated solution set among them represents the maximum population size. The detailed proofs are provided in the Appendix due to space limitation.

Lemma 1. *When using GSEMO to optimize L1DM-2SIP, the population size is at most* $2c + 1$.

In the following, we will first illustrate the risk of the GSEMO algorithm with unit-step mutation getting stuck in local Pareto optima when optimizing L1DM-2SIP, and then demonstrate that using an ageing strategy or heavy-tailed mutation operators can help escape from local Pareto optima and find the Pareto front.

3.1 Unit-step Mutation fails

We first consider GSEMO equipped with the basic unit-step mutation operator, and show in Theorem 1 that it runs the risk of getting stuck in local Pareto optima, preventing it from ever finding the Pareto front.

Unit-step mutation. The random variable Z_i in Eq. (1) is 1 with probability $1/2$, and -1 otherwise.

Algorithm 1 Global Simple Evolutionary Multi-objective Optimizer (GSEMO) [Laumanns *et al.*, 2004b]

Select a solution $\boldsymbol{x}_0$ uniformly from $\mathcal{X}$ at random;
$P \leftarrow \{\boldsymbol{x}_0\}$;
while termination condition is not met **do**
  Select a solution $\boldsymbol{x}$ from P uniformly at random ;
  Apply mutation on $\boldsymbol{x}$ to generate $\boldsymbol{y}$;
  if $\nexists \boldsymbol{z} \in P$ satisfies $\boldsymbol{z} \prec \boldsymbol{y}$ **then**
    $P \leftarrow (P \setminus \{\boldsymbol{z} \in P \mid \boldsymbol{y} \preceq \boldsymbol{z}\}) \cup \{\boldsymbol{y}\}$
  end if
end while

Theorem 1. *For GSEMO with unit-step mutation optimizing L1DM-2SIP, the probability that the population* P *cannot find the Pareto front is greater than* 0.

Proof. Suppose that the initial solution $\boldsymbol{x}_0$ randomly sampled from the solution space belongs to the local Pareto optimal set X_L, which happens with some probability larger than 0. Then, the population will converge to the local optimal region X_L, from which the algorithm cannot escape by plus/minus one on one dimension (using unit-step mutation). Thus, the Pareto front can never be found. □

3.2 Ageing is Provably Helpful

Next, we show that for L1DM-2SIP, using an ageing strategy can help GSEMO escape from local Pareto optima. Ageing has several variants. A common version, i.e., static pure ageing, can automatically restart the algorithm after detecting stagnation in local optima [Horoba *et al.*, 2009; Jansen and Zarges, 2011], and has been shown able to allow $(\mu+1)$-EA to escape from local optima [Oliveto and Sudholt, 2014; Corus *et al.*, 2019]. Here, we adopt the ageing strategy proposed by Li *et al.* [2025], which assigns each individual an age, increases it by one in each generation, and lets individuals whose age exceeds τ participate in survival selection. This strategy ensures that any newly generated individual can survive for at least τ generations, allowing inferior solutions to remain in the population long enough to reduce the risk of prematurely discarding potentially useful search directions. Meanwhile, the survival selection can keep non-dominated solutions encountered so far.

Algorithm 2 presents GSEMO with the ageing strategy. The age of the initial solution is set to 1. In each iteration, a solution $\boldsymbol{x}$ is randomly selected from the population and mutated. The new solution $\boldsymbol{y}$ has an age of 0. For individuals in the population who have reached age τ, only non-dominated solutions are retained, and among solutions with identical objective values, only one is randomly selected and kept. Individuals with an age less than τ are retained without further filtering. Finally, the new solution $\boldsymbol{y}$ is added to the population, and the age of all solutions is incremented by one.

In Theorem 2, we prove that the expected running time of GSEMO using the ageing strategy and unit-step mutation for finding the Pareto front of L1DM-2SIP can be upper bounded. Note that the running time is measured by the number of fitness evaluations.

Algorithm 2 GSEMO with ageing

```
1: Select a solution x_0 ∈ 𝒳 randomly, and set its age to 1;
2: P ← {x_0};
3: while termination condition is not met do
4:     Select a solution x from P uniformly at random;
5:     Apply mutation on x to generate y and set y.age = 0;
6:     P_τ ← {z ∈ P | z.age ≥ τ};
7:     P_s = P \ P_τ;
8:     P_τ ← Retain the non-dominated solutions with distinct objective vectors in P_τ;
9:     P ← P_τ ∪ {y} ∪ P_s;
10:    Increase the age of all solutions in P by 1
11: end while
```

Theorem 2. *For GSEMO with ageing and unit-step mutation optimizing L1DM-2SIP, the expected running time for finding the Pareto front is upper bounded by $O((c+D(\boldsymbol{x}_0))(\tau+c)n+(2b-c+a)\tau(\tau+c)n(2en(\tau+2c+1)/\tau+1)^{2b-c+a})$, where $\boldsymbol{x}_0$ is the initial solution, $D(\boldsymbol{x}_0)=\min_{\boldsymbol{v}\in U\cup X_L}||\boldsymbol{x}_0-\boldsymbol{v}||_1$, and $\tau\geq 1$.*

Proof. We divide the optimization process into five phases. The first phase aims to find a boundary solution in $U=U_1\cup U_2\cup U_3\cup U_4$ or a local Pareto optimal solution in X_L. The second phase aims to find a boundary solution in U or a local Pareto optimal solution in X_{L_a}. The third phase is to find a solution in U^+ (i.e., the interior region surrounded by U). The fourth phase aims to find a global Pareto optimal solution in X^*, and the last phase is to find the complete Pareto set X^*.

According to Lemma 1, we know that the maximum size of the non-dominated solution set generated by GSEMO during the search process is $2c+1$. For GSEMO with ageing, since solutions with age less than τ are always retained, and only non-dominated solutions among those whose age reaches τ are kept, the maximum population size during the search is $P_{\max}=\tau+2c+1$. During the analysis of the first two phases, we pessimistically assume that the solutions in U^+ are not found; otherwise, we directly go to the fourth phase.

Let $\mathrm{dis}_Q(\boldsymbol{x})=\min_{\boldsymbol{v}\in Q}||\boldsymbol{x}-\boldsymbol{v}||_1$ denote the distance from a point $\boldsymbol{x}$ in the solution space to a region Q. In the first phase, we consider $D(P)=\min_{\boldsymbol{x}\in P}\mathrm{dis}_{U\cup X_L}(\boldsymbol{x})$, i.e., the minimum distance from the population P to the boundary U and the local Pareto optimal set X_L. Because a solution will be deleted only if a new solution weakly dominating it appears, and the domination relationship between two solutions is consistent with the relationship between their values of $\mathrm{dis}_{U\cup X_L}(\boldsymbol{x})$, $D(P)$ will not increase. To decrease $D(P)$, it is sufficient to select a solution $\boldsymbol{x}'=\arg\min_{\boldsymbol{x}\in P}\mathrm{dis}_{U\cup X_L}(\boldsymbol{x})$ and change the value of one of its first two dimensions in the correct direction. For example, if $\boldsymbol{x}'=(c,-b-2,0,\dots,0)^T$, we only need to add one to its second dimension to obtain a solution that is closer to X_L, and then $D(P)$ will decrease by 1. For Algorithm 2, the probability of selecting $\boldsymbol{x}'$ is at least $1/P_{\max}$, the probability of changing the value in the corresponding dimension is $(1/n)\cdot(1-1/n)^{n-1}\geq 1/(en)$, and the probability of unit-step mutation in the correct direction is $1/2$. Thus, $D(P)$ can decrease by 1 in each iteration with probability at least $1/(2enP_{\max})$. According to additive drift analysis (Lemma 4 in the Appendix) [He and Yao, 2001], the expected running time of the first phase is at most $D(\boldsymbol{x}_0)\cdot 2enP_{\max}$, as the initial distance $D(P)=D(\boldsymbol{x}_0)$.

For the second phase, we assume that only solutions in $X_L\setminus X_{L_a}$ are found after the first phase; otherwise, the goal of the second phase has reached. As the solutions in $X_L\setminus X_{L_a}$ can only be weakly dominated by the solutions in U^+, they will be always kept in the population. By selecting the solution $\boldsymbol{x}'$ in $X_L\setminus X_{L_a}$ which is closest to X_{L_a}, and only changing x_1' in the right direction by unit-step mutation, we can move closer to X_{L_a}. It is sufficient to perform such steps for $c-a$ times. As such a step happens with probability at least $1/(2enP_{\max})$ in each iteration, the expected running time of the second phase is at most $(c-a)\cdot(2enP_{\max})$.

Now we are in the third phase, where some solutions in $U\cup X_{L_a}$ have been found, and the goal is to reach the interior region U^+. Consider starting from a solution $\boldsymbol{u}\in X_{L_a}$ in the current population. First, we select $\boldsymbol{u}$ for mutation and increase only its second dimension value u_2, resulting in the offspring solution $\boldsymbol{u}'$. The probability of this event occurring is at least $1/(2enP_{\max})$. Next, we select $\boldsymbol{u}'$ from the population and continue increasing its second dimension value. Since $\boldsymbol{u}'$ is retained in the population for τ iterations, the probability that this event happens at least once in τ iterations is at least $p=1-(1-1/(2enP_{\max}))^\tau\geq 1-e^{-\frac{\tau}{2enP_{\max}}}$. Since $e^{-x}\leq 1/(x+1)$ for $x\geq -1$, we have $p\geq 1-\frac{1}{1+\tau/(2enP_{\max})}=\frac{\tau}{2enP_{\max}+\tau}$. This process of selecting a specific solution and increasing its second dimension value needs to be repeated $2b-c+a$ times to find a solution in U^+, so the probability of successfully finding a solution in U^+ is at least $\frac{1}{2enP_{\max}}\cdot(\frac{\tau}{2enP_{\max}+\tau})^{2b-c+a}$. Note that such one successful attempt requires $(2b-c+a)\cdot\tau+1$ iterations. If we consider starting from a boundary solution in U in the third phase, we can directly select it for mutation and change only one of its first two dimensions to enter U^+, which happens with probability at least $1/(2enP_{\max})$ in one iteration. Thus, the expected running time of the third phase is $O((2b-c+a)\tau nP_{\max}\cdot(2enP_{\max}/\tau+1)^{2b-c+a})$.

In the fourth phase, we have found one solution in U^+, and aim to find a Pareto optimal solution in X^*. For this purpose, we only need to select the solution nearest to X^* for mutation and change one of its first two dimensions towards the direction of X^*, which happens with probability at least $1/(2enP_{\max})$. After performing such a step for $c-a-1$ times, X^* will be reached. Thus, the expected running time of the fourth phase is at most $(c-a-1)\cdot(2enP_{\max})$.

Finally, we are to find the whole Pareto optimal set X^*. By selecting a proper Pareto optimal solution $\boldsymbol{u}\in X^*\cap P$ as the parent and only changing u_1 in the right direction, we can obtain a new Pareto optimal solution in $X^*\setminus P$. Such a behavior happens with probability at least $1/(2enP_{\max})$. As the size of X^* is $2a+1$, the expected running time of the last phase is at most $2a\cdot(2enP_{\max})$.

Combining the above five phases, the total expected running time is $D(\boldsymbol{x}_0)\cdot 2enP_{\max}+(c-a)\cdot(2enP_{\max})+O((2b-c+a)\tau nP_{\max}\cdot(2enP_{\max}/\tau+1)^{2b-c+a})+(c-a-1)\cdot(2enP_{\max})+2a\cdot(2enP_{\max})=O((c+D(\boldsymbol{x}_0))(\tau+c)n+(2b-c+a)\tau(\tau+c)n(2en(\tau+2c+1)/\tau+1)^{2b-c+a})$, where the equality holds by $P_{\max} = \tau + 2c + 1$. □

The benefit of ageing has been demonstrated in [Li *et al.*, 2025], which can accelerate SMS-EMOA in solving the m-objective pseudo-Boolean JUMP benchmark for all even m. Our results in Theorem 2 confirm its advantage in solving MOO problems in integer space. The ageing strategy plays a key role in the analysis of the third phase of the above proof. That is, it enables inferior solutions to survive in the population for a while, and helps the algorithm go across the inferior region to reach U^+ from the local Pareto optimal set X_L.

3.3 Heavy-Tailed Mutations are Provably Helpful

In this section, we replace the unit-step mutation operator with two heavy-tailed mutation operators [Doerr *et al.*, 2025b], i.e., exponential-tail [Rudolph, 1994] and power-law [Doerr *et al.*, 2017] mutations, to allow multi-valued changes in each dimension of a solution, and analyze the expected running time of GSEMO on the L1DM-2SIP problem.

Exponential-tail mutation. The random variable Z_i in Eq. (1) satisfies $\Pr(Z_i = k) = \alpha(2-\alpha)^{-1}(1-\alpha)^{|k|}$, where $\alpha \in (0, 1)$ is a decay parameter.

Power-law mutation. The random variable Z_i in Eq. (1) satisfies $\Pr(Z_i = k) = |k|^{-\beta}/(2\zeta(\beta))$, where $\beta \in (1, 2)$ is the power-law exponent and $\zeta(\beta) = \sum_{k=1}^{\infty} 1/k^{\beta}$ is the Riemann zeta function.

We prove in Theorem 3 that GSEMO can find the whole Pareto front when using heavy-tailed mutations (either exponential-tail or power-law mutation). Unlike the unit-step mutation that only changes the value of each dimension by one, heavy-tailed mutations allow each dimension to change by a larger value. This property enables the algorithm to jump directly from an arbitrary state to a target state, which is crucial for escaping local Pareto optima. It would also be interesting to see whether GSEMO equipped with both ageing and heavy-tailed mutation [Corus *et al.*, 2021] works, which we leave as a future study.

Theorem 3. *Consider GSEMO optimizing L1DM-2SIP with the initial solution $\boldsymbol{x}_0$. Let $D(\boldsymbol{x}_0) = \min_{\boldsymbol{v} \in U \cup X_L} ||\boldsymbol{x}_0 - \boldsymbol{v}||_1$. The expected running time for finding the whole Pareto front satisfies the following bounds.*

(1) If the exponential-tail mutation with parameter $\alpha \in (0, 1)$ is used, then the expected running time is

$$O\big(cn\big(c+D(\boldsymbol{x}_0)+\tfrac{n}{\alpha}+\tfrac{c-a}{\alpha(1-\alpha)}+\tfrac{1}{\alpha(1-\alpha)^{2b-c+a+1}}\big)\big).$$

(2) If the power-law mutation with exponent $\beta \in (1, 2)$ is used, then the expected running time is

$$O\big(n\zeta(\beta)\big(\tfrac{c(b+D(\boldsymbol{x}_0))}{\beta-1}+\tfrac{a^2}{(1-1.5^{1-\beta})^3}+c\big(b^2+\ln a\big)\big)\big).$$

The proof idea is similar to that of Theorem 2, i.e., dividing the optimization procedure into five phases and analyzing the expected running time of each phase. The main difference lies in that the different mutation operators are employed, and the largest population size is reduced from $2c+1+\tau$ to $2c+1$ without using the ageing strategy. In particular, in Phases 1 and 4, we apply Lemma 2 to bound the expected running time for reaching the target solution set $U \cup X_L$ or X^*. This lemma is developed from [Doerr *et al.*, 2025b, Lemmas 14 and 18], with the main difference that their original analysis considers the point $(0, 0, \ldots, 0)$ as the search target, whereas in our setting the target is a set of solutions (i.e., $U \cup X_L$ or X^*); the largest population size changes from $2a + 1$ to $2c + 1$. Phase 2 uses heavy-tailed mutation instead of unit-step mutation to find a solution in X_{L_a} from a solution in $X_L \setminus X_{L_a}$. In Phase 3, we can use exponential-tail or power-law mutation to directly reach the interior region U^+ from a local Pareto optimal solution $\boldsymbol{x}$ in X_{L_a} within a single mutation step, e.g., by increasing x_2 from $-b$ to the range of $[b-c+a+1..b+c-a-1]$, which happens with probability $\sum_{k=2b-c+a+1}^{2b+c-a-1} \frac{\alpha}{2-\alpha}(1-\alpha)^k$ by exponential-tail mutation or $\sum_{k=2b-c+a+1}^{2b+c-a-1} \frac{k^{-\beta}}{2\zeta(\beta)}$ by power-law mutation. Finally, in Phase 5, Lemma 3 (adapted from [Doerr *et al.*, 2025b, Lemmas 15 and 19] by replacing the largest population size $2a+1$ in their analysis with $2c + 1$ here) is used to bound the expected running time for extending from a single Pareto optimal solution to the entire Pareto front. Due to space limitation, the full proof of Theorem 3 is provided in the Appendix.

Lemma 2 (Adapted from [Doerr *et al.*, 2025b])**.** *Let $Q \in \{U \cup X_L, X^*\}$ be a target set, and let $D(P) := \min_{\boldsymbol{x} \in P, \boldsymbol{g} \in Q} \|\boldsymbol{x} - \boldsymbol{g}\|_1$ denote the minimum ℓ_1-distance between two solutions in the population P and the target set Q. The expected running time of GSEMO for finding a solution in Q starting from P is bounded as follows:*

(1) If using the exponential-tail mutation with parameter $\alpha \in (0, 1)$, it is upper bounded by $((2c + 1)en/K)(n\pi^2/(6\alpha)+4\alpha D(P))$, where K is a constant.

(2) If using the Power-law mutation with exponent $\beta \in (1, 2)$, it is upper bounded by $(2c + 1) \cdot 2en\zeta(\beta)\big(2^{1/(2-\beta)} + 2(2-\beta)(\beta-1)^{-1}(D(P))^{\beta-1}\big)$.

Lemma 3 (Adapted from [Doerr *et al.*, 2025b])**.** *If a population P contains a Pareto optimal solution in X^*, then the expected running time of GSEMO for finding the whole Pareto front F^* is bounded as follows:*

(1) If using the exponential-tail mutation with parameter $\alpha \in (0, 1)$, it is upper bounded by $C(cn \max\{\ln(a + 1)/(a\alpha), a\alpha + \ln(a + 1)\})$, where C is a large constant.

(2) If using the Power-law mutation with exponent $\beta \in (1, 2)$, it is upper bounded by $4\ln(2)en\,\zeta(\beta)(\beta - 1)(1 - 1.5^{1-\beta})^{-1}(2a + 1)^{\beta}(1 - 2^{1-\beta})^{-2} + (2c + 1) \cdot 2en\,\zeta(\beta)(\ln(a + 1) + 1)$.

4 Extension of L1DM-2SIP

In this section, we give a more general form of the benchmark problem L1DM-2SIP proposed in Section 2, which includes complex local optima regions.

Definition 6 (L1DM-2SIP)**.** *Consider the problem of L1-distance minimization to two sets of integer points: $\mathbb{Z}^n \to \mathbb{N}_0^2$ with*

$$\text{L1DM-2SIP}(\boldsymbol{x}) = \begin{pmatrix} \min\{\|\boldsymbol{x} - \boldsymbol{u}_1\|_1, h_1(\boldsymbol{x}), e_1(\boldsymbol{x})\} \\ \min\{\|\boldsymbol{x} - \boldsymbol{u}_2\|_1, h_2(\boldsymbol{x}), e_2(\boldsymbol{x})\} \end{pmatrix}.$$

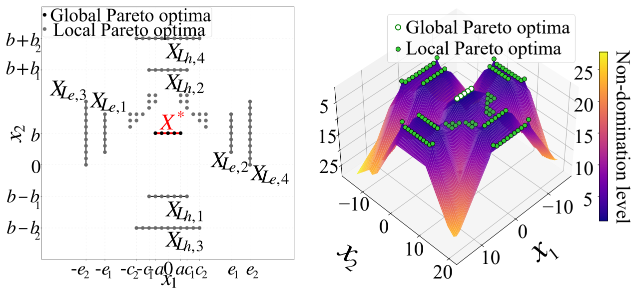

Figure 2: An instance of L1DM-2SIP (Definition 6) in the decision space, and its corresponding fitness landscape, shown by Pareto Landscape [Liang *et al.*, 2024].

where $h_r(\boldsymbol{x}) = \min_{i\in[1..k_z]}(\|\boldsymbol{x} - \boldsymbol{z}_{r,i}\|_1)$ *and* $e_r(\boldsymbol{x}) = \min_{j\in[1..k_v]}(\|\boldsymbol{x} - \boldsymbol{v}_{r,j}\|_1)$. *Setting* $\boldsymbol{u}_1 = (-a, b, 0, ..., 0)^{\mathrm{T}}$, $\boldsymbol{u}_2 = (a, b, 0, ..., 0)^{\mathrm{T}}$, $\boldsymbol{z}_{1,i} = ((-1)^i c_{\gamma(i)}, (-1)^i b_{\gamma(i)} + b, 0, \ldots, 0)^{\mathrm{T}}$, $\boldsymbol{z}_{2,i} = ((-1)^{i+1} c_{\gamma(i)}, (-1)^i b_{\gamma(i)} + b, 0, \ldots, 0)^{\mathrm{T}}$, $\boldsymbol{v}_{1,j} = ((-1)^j e_{\gamma(j)}, (-1)^j g_{\gamma(j)} + b, 0, \ldots, 0)^{\mathrm{T}}$, *and* $\boldsymbol{v}_{2,j} = ((-1)^j e_{\gamma(j)}, (-1)^{j+1} g_{\gamma(j)} + b, 0, \ldots, 0)^{\mathrm{T}}$, *where* $\gamma(i) = 2 \cdot \lceil i/2 \rceil - 1$, c_k, b_k, e_k, *and* g_k *all increase gradually with* $k \geq 1$, *and* $a < c_1 < e_1$.

When $c_1 = c$, $b_1 = 2b$, and L1DM-2SIP depends only on the four points $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{z}_{1,1}$, and $\boldsymbol{z}_{2,1}$, the problem reduces to the instance considered in Sections 2 and 3. For the general form of L1DM-2SIP, under conditions of $b_1 > a + c_1$, $e_1 - a > g_1 + 1$, and for $k \geq 1$, $g_{\gamma(k)} = c_{\gamma(k)}$, $c_k + 2|b_{k+1} - b_k| - c_{k+1} > 1$, and $g_k + 2|e_k - e_{k+1}| - g_{k+1} > 1$, L1DM-2SIP has the Pareto set $X^* = \{(x_1, b, 0, \ldots, 0)^T \in \mathbb{Z}^n \mid x_1 \in [-a..a]\}$, and a series of local Pareto optimal solution sets $X_{L_{h,i}}$ and $X_{L_{e,j}}$. The horizontal $X_{L_{h,i}} = \{(x_1, (-1)^i b_{\gamma(i)} + b, 0, \ldots, 0)^T \in \mathbb{Z}^n \mid x_1 \in [-c_{\gamma(i)}..c_{\gamma(i)}]\}$, and the vertical $X_{L_{e,j}}$ consists of all $((-1)^j e_{\gamma(j)}, x_2, 0, \ldots, 0)^T \in \mathbb{Z}^n$ with $x_2 \in [-g_{\gamma(j)} + b..g_{\gamma(j)} + b]$. According to Definition 2, the solution space may still contain a small number of discrete local Pareto optimal solutions, but these solutions will not form peaks and thus will not cause the algorithm to get stuck. Figure 2 gives an example of such a problem.

This problem can be instantiated to exhibit a more "interesting" structure by gradually increasing the distance between the point pairs from the global optimum towards outer regions, thereby forming a landscape with many local optima of progressively improving quality as they approach the global optimum. Such a landscape resembles those of many well-established functions like ZDT4 [Zitzler *et al.*, 2000], DTLZ1 and DTLZ3 [Deb *et al.*, 2005].

5 Experiments

In this section, we conduct experiments on the proposed integer-space function to complement the theoretical results.

Table 1 presents the results of GSEMO solving L1DM-2SIP with different number of fitness evaluations FEs $\in$ $\{1\times10^3, 5\times10^3, 1\times10^4, 5\times10^4, 1\times10^5, 5\times10^5, 1\times10^6\}$, problem size $n = 2$, $a = 3$, $b = 3$, $c = 4$, and each dimension initialized within the range $[-10, 10]$. We present the average PF coverage and standard deviation over 50 runs for four variants of GSEMO: 1) GSEMO with unit-step mutation, 2) GSEMO with ageing ($\tau = 4$) and unit-step mutation, 3) GSEMO with exponential-tail mutation ($\alpha = 0.5$), and 4) GSEMO with power-law mutation ($\beta = 1.5$). The results show that increasing the number of fitness evaluations (FEs) does not improve the performance of GSEMO with unit-step mutation, indicating that the algorithm may have encountered a bottleneck, specifically, getting stuck in local Pareto optima. Introducing an ageing strategy improved this situation, increasing the coverage of the PF at the same number of function evaluations. To further examine the effect of τ, we added results with $\tau = 2, 6, 8$ in the Appendix, suggesting that τ cannot be set too small in our studied case. However, even with the ageing strategy, the results with unit-step mutation were still inferior to those with heavy-tailed mutation, because the latter allows for mutation strengths greater than 1.

FEs	GSEMO-u	GSEMO-u with ageing	GSEMO-e	GSEMO-p
1×10^3	0.520(0.505)	0.580(0.499)	0.966(0.172)	1.000(0.000)
5×10^3	0.520(0.505)	0.580(0.499)	1.000(0.000)	1.000(0.000)
1×10^4	0.520(0.505)	0.580(0.499)	1.000(0.000)	1.000(0.000)
5×10^4	0.520(0.505)	0.640(0.485)	1.000(0.000)	1.000(0.000)
1×10^5	0.520(0.505)	0.680(0.471)	1.000(0.000)	1.000(0.000)
5×10^5	0.520(0.505)	0.920(0.274)	1.000(0.000)	1.000(0.000)
1×10^6	0.520(0.505)	1.000(0.000)	1.000(0.000)	1.000(0.000)

Table 1: Average PF coverage and standard deviation over 50 runs for GSEMO-u (GSEMO with unit-step mutation), GSEMO-u with ageing (GSEMO with ageing and unit-step mutation), GSEMO-e (GSEMO with exponential-tail mutation), and GSEMO-p (GSEMO with power-law mutation) solving L1DM-2SIP under different FEs.

6 Conclusion

In this paper, we take the first step towards the running time analysis of MOEAs on integer spaces with local optima. We begin by presenting a multi-objective benchmark defined on an integer space, characterized by an analyzable landscape and the presence of local optima. Next, we conduct a running time analysis on this benchmark and derive some theoretical results. Specifically, we prove that the widely-studied GSEMO with unit-step mutation may fail to find the Pareto front, whereas introducing an ageing mechanism or employing heavy-tailed (exponential-tail or power-law) mutation operators can enable GSEMO to escape local optima and identify the Pareto front within a bounded expected running time. Empirical studies also support our theoretical results. As a class of MOO problems with practical relevance, the proposed benchmark narrows the gap between the theoretical analyses and practice. We hope this work will inspire further research along this direction, e.g., considering other well-established MOEAs or the extension of L1DM-2SIP to broaden the theoretical understanding of search algorithms on integer spaces.

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

[Auger and Doerr, 2011] Anne Auger and Benjamin Doerr. *Theory of Randomized Search Heuristics - Foundations and Recent Developments*. World Scientific, 2011.

[Beume *et al.*, 2007] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *EJOR*, 181(3):1653–1669, 2007.

[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.

[Cerf *et al.*, 2023] Sacha Cerf, Benjamin Doerr, Benjamin Hebras, Yakob Kahane, and Simon Wietheger. The first proven performance guarantees for the non-dominated sorting genetic algorithm II (NSGA-II) on a combinatorial optimization problem. In *IJCAI*, Macao, SAR, China, 2023.

[Corus *et al.*, 2019] Dogan Corus, Pietro S. Oliveto, and Donya Yazdani. Artificial immune systems can find arbitrarily good approximations for the NP-hard number partitioning problem. *AIJ*, 274:180–196, 2019.

[Corus *et al.*, 2021] Dogan Corus, Pietro S. Oliveto, and Donya Yazdani. Fast immune system-inspired hypermutation operators for combinatorial optimization. *IEEE TEVC*, 25(5):956–970, 2021.

[Corus *et al.*, 2025] Dogan Corus, Pietro S. Oliveto, and Donya Yazdani. Fast contiguous somatic hypermutations for single-objective optimisation and multi-objective optimisation via decomposition. In *AAAI*, Philadelphia, 2025.

[Dang *et al.*, 2023] Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *AAAI*, Washington, DC, 2023.

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

[Dang *et al.*, 2024b] Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Illustrating the efficiency of popular evolutionary multi-objective algorithms using runtime analysis. In *GECCO*, Melbourne, Australia, 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 *et al.*, 2005] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pages 105–145. Springer, 2005.

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

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

[Doerr and Qu, 2023] Benjamin Doerr and Zhongdi Qu. Runtime analysis for the NSGA-II: Provable speed-ups from crossover. In *AAAI*, Washington, DC, 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.*, 2017] Benjamin Doerr, Huu Phuoc Le, Régis Makhmara, and Ta Duy Nguyen. Fast genetic algorithms. In *GECCO*, Berlin, Germany, 2017.

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

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

[Droste *et al.*, 2002] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *TCS*, 276(1-2):51–81, 2002.

[Fieldsend *et al.*, 2021] Jonathan E Fieldsend, Tinkle Chugh, Richard Allmendinger, and Kaisa Miettinen. A visualizable test problem generator for many-objective optimization. *IEEE TEVC*, 26(1):1–11, 2021.

[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.

[He and Yao, 2001] Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *AIJ*, 127(1):57–85, 2001.

[Horoba *et al.*, 2009] Christian Horoba, Thomas Jansen, and Christine Zarges. Maximal age in randomized search heuristics with aging. In *GECCO*, Montreal, Canada, 2009.

[Horoba, 2009] Christian Horoba. Analysis of a simple evolutionary algorithm for the multiobjective shortest path problem. In *FOGA*, Orlando, FL, 2009.

[Ishibuchi *et al.*, 2010] Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Noritaka Tsukamoto, and Yusuke Nojima. Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In *PPSN*, Krakow, Poland, 2010.

[Jansen and Zarges, 2011] Thomas Jansen and Christine Zarges. On the role of age diversity for effective aging operators. *Evolutionary Intelligence*, 4(2):99–125, 2011.

[Köppen and Yoshida, 2007] Mario Köppen and Kaori Yoshida. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *EMO*, Matsushima, Japan, 2007.

[Lapucci *et al.*, 2023] Matteo Lapucci, Pierluigi Mansueto, and Fabio Schoen. A memetic procedure for global multiobjective optimization. *Mathematical Programming Computation*, 15(2):227–267, 2023.

[Laumanns *et al.*, 2004a] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing*, 3(1):37–51, 2004.

[Laumanns *et al.*, 2004b] 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.*, 2015] Yuan-Long Li, Yu-Ren Zhou, Zhi-Hui Zhan, and Jun Zhang. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms. *IEEE TEVC*, 20(4):563–576, 2015.

[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, 2025] Zimin Liang and Miqing Li. On the problem characteristics of multi-objective pseudo-Boolean functions in runtime analysis. In *FOGA*, Leiden, The Netherlands, 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.*, 2024] Zimin Liang, Zhiji Cui, and Miqing Li. Pareto landscape: Visualising the landscape of multiobjective optimisation problems. In *PPSN*, Hagenberg, Austria, 2024.

[Lu *et al.*, 2019] Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. NSGA-Net: Neural architecture search using multi-objective genetic algorithm. In *GECCO*, Prague, Czech Republic, 2019.

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

[Ojstersek *et al.*, 2020] Robert Ojstersek, Miran Brezocnik, and Borut Buchmeister. Multi-objective optimization of production scheduling with evolutionary computation: A review. *International Journal of Industrial Engineering Computations*, 11(3):359–376, 2020.

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

[Opris, 2025] 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.

[Ren *et al.*, 2024] 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.*, 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.

[Ren *et al.*, 2026] Shengjie Ren, Zimin Liang, Miqing Li, and Chao Qian. Not just for archiving: Provable benefits of reusing the archive in evolutionary multi-objective optimization. In *AAAI*, Singapore, 2026.

[Rudolph, 1994] Günter Rudolph. An evolutionary algorithm for integer programming. In *PPSN*, Jerusalem, Israel, 1994.

[Rudolph, 1997] Günter Rudolph. Convergence properties of evolutionary algorithms. In *Verlag Dr Kovac*, 1997.

[Rudolph, 2023] Günter Rudolph. Runtime analysis of (1+1)-EA on a biobjective test function in unbounded integer search space. In *SSCI*, Mexico City, Mexico, 2023.

[Wietheger and Doerr, 2024a] Simon Wietheger and Benjamin Doerr. A mathematical runtime analysis of the non-dominated sorting genetic algorithm III (NSGA-III). In *GECCO*, Melbourne, Australia, 2024.

[Wietheger and Doerr, 2024b] Simon Wietheger and Benjamin Doerr. Near-tight runtime guarantees for many-objective evolutionary algorithms. In *PPSN*, Hagenberg, Austria, 2024.

[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 *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 *et al.*, 2000] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *ECJ*, 8(2):173–195, 2000.

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