

# On the Problem Characteristics of Multi-objective Pseudo-Boolean Functions in Runtime Analysis

Zimin Liang  
zx1525@student.bham.ac.uk  
University of Birmingham  
Birmingham, United Kingdom

Miqing Li\*  
m.li.8@bham.ac.uk  
University of Birmingham  
Birmingham, United Kingdom

## ABSTRACT

Recently, there has been growing interest within the theoretical community in analytically studying multi-objective evolutionary algorithms. This runtime analysis-focused research can help formally understand algorithm behaviour, explain empirical observations, and provide theoretical insights to support algorithm development and exploration. However, the test problems commonly used in the theoretical analysis are predominantly limited to problems with heavy “artificial” characteristics (e.g., symmetric, homogeneous objectives and linear Pareto fronts), which may not be able to well represent realistic scenarios. In this paper, we first discuss commonly used multi-objective functions in the theory domain and systematically review their features, limitations and implications to practical use. Then, we present several new functions with more realistic features, such as heterogeneous objectives, local optimality and nonlinearity of the Pareto front, through simply mixing and matching classical single-objective functions in the area (e.g., LeadingOnes, Jump and RoyalRoad). We hope these functions can enrich the existing test problem suites, and strengthen the connection between theoretic and practical research.

## CCS CONCEPTS

• **Mathematics of computing** → **Combinatorics**; • **Theory of computation** → **Evolutionary algorithms**; • **Applied computing** → **Multi-criterion optimization and decision-making**.

## KEYWORDS

Multi-objective optimisation, evolutionary computation, runtime analysis, pseudo-Boolean functions.

### ACM Reference Format:

Zimin Liang and Miqing Li. 2025. On the Problem Characteristics of Multi-objective Pseudo-Boolean Functions in Runtime Analysis. In *Foundations of Genetic Algorithms XVIII (FOGA '25)*, August 27–29, 2025, Leiden, Netherlands. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3729878.3746700>

## 1 INTRODUCTION

Many real-world challenges contain multiple conflicting objectives to be optimised simultaneously, known as multi-objective optimisation problems (MOPs). Instead of a single optimal solution, MOPs

yield a set of solutions called *Pareto optimal solutions*, each representing a distinct trade-off between the objectives. Multi-objective evolutionary algorithms (MOEAs) have demonstrated their ability to deal with MOPs in a variety of practical scenarios. Yet, the theoretical understanding of MOEAs, particularly their behaviours and performance guarantees, lags behind their practical success [1, 2].

In this regard, runtime analysis [2] emerges as a very useful tool. First, it can be used to theoretically analyse the expected runtime of MOEAs, including mainstream algorithms like NSGA-II [3–5], SPEA2 [6], SMS-EMOA [7–9], MOEA/D [10–12] and NSGA-III [13], as well as simple heuristics like SEMO [14–16] and G-SEMO [17–19]. Second, runtime analysis can help confirm observations reported from empirical studies, for example, why NSGA-II is less effective for problems with three or more objectives [20]. Third, runtime analysis can provide insight and guidance in algorithm design (e.g., the use of crossover [19], adaptive mutation [21] and the archive [22]), as well as in algorithmic parameter setup (e.g., mutation rate setting [23]). Lastly, it can even be used to challenge conventional practice in the empirical community and to guide the development of different mechanisms, for example introducing randomness in population update of MOEAs [24, 25].

Yet, multi-objective benchmark functions used in runtime analysis are predominantly limited to pseudo-Boolean ones with heavy “artificial” characteristics. They may not be able to well represent realistic scenarios. For example, in OneMinMax [15], a commonly used benchmark, any point in the decision space is a Pareto optimal solution. This means the two objectives are completely conflicting, with no configuration improving both objectives or yielding dominated solutions. This is apparently not very realistic.

Another prominent feature in multi-objective pseudo-Boolean functions is the linear shape of their Pareto fronts, such as in the well-known benchmarks LeadingOnes-TrailingZeroes [26], CountOnes-CountZeroes [14] and OneJump-ZeroJump [27]. This may not be very representative of real-world problems and, more importantly, may favour certain MOEAs (e.g., the decomposition-based algorithm MOEA/D [28]). The performance of some practical MOEAs may heavily depend on the Pareto front shape due to their population maintenance mechanisms [29]. For example, MOEA/D uses a set of uniformly distributed weights in a simplex to maintain the population. As a result, the algorithm may achieve perfect uniformity when the function’s Pareto front is of a linear or simplex-like shape, but may perform very poorly on other shapes [30]. Solely considering such linear Pareto fronts may fail to provide a fair comparison between MOEAs, particularly to those who are robust to Pareto front shapes such as Pareto-based MOEAs [29].

Findings derived from very artificial functions may not hold on practical problems. For example, to our knowledge, all existing

\*Corresponding author

theoretical studies suggest that NSGA-II performs at least as well as SEMO, and sometimes better. However, recent empirical work [31] has shown that SEMO can outperform NSGA-II on many practical problems (e.g., the multi-objective TSP and QAP problems).

In this paper, we conduct a short review of multi-objective pseudo-Boolean functions commonly used in runtime analysis. We discuss the characteristics that make these functions useful for theoretical study, and identify their limitations in capturing realistic features, as well as the implications for the practical community of algorithm development. Moreover, we present several new functions by combining different single-objective functions such as LeadingOnes, Jump, and RoyalRoad. These mix-and-match functions, aside from being composed of heterogeneous objectives on which MOEAs may perform differently [32], can introduce more realistic features such as multi-modality (i.e., Pareto local optimality) and nonlinearity of the Pareto front, while still retaining analytical tractability.

It is worth stating that in this paper we focus on widely used, representative multi-objective pseudo-Boolean functions in runtime analysis, including OneMinMax, LeadingOnes-TrailingZeroes, CountOnes-CountZeroes and OneJump-ZeroJump, as well as “asymmetric” combinations of their single-objective components such as OneMax [33], LeadingOnes [34], Jump [35] and RoyalRoad [36]. There also exist many other benchmarks in this area, such as OneTrap-ZeroTrap [37], LPTNO [38], SPG, ZPLG [39] (extended from PLG [40]), Dec-obj-MOP [41] and multi-objective RealRoyalRoad [19], and single-objective ones such as the Needle [42], Trap [42] and RealRoyalRoad [43]. However, such functions may be either too special (e.g., the global optimum in the Needle and Trap functions too hard to locate), or are regarded too artificial [27].

The rest of the paper is structured as follows. Section 2 gives the preliminaries in multi-objective optimisation and single-objective pseudo-Boolean functions considered in the paper. Section 3 describes the function characteristics of interest. Sections 4 and 5 discuss existing common benchmarks and new mix-and-match functions based on the aforementioned characteristics and their implications, followed by section 6 that gives the limitations of the study. Finally, Section 7 concludes the paper.

Due to space limitations, the new functions’ properties (e.g., what their Pareto (local) optimal set is) are given in the supplementary at [https://github.com/Zim-L/FOGA25ReviewPseudoBoolean/blob/main/FOGA\\_Supplementary\\_Review\\_Pseudo\\_Boolean.pdf](https://github.com/Zim-L/FOGA25ReviewPseudoBoolean/blob/main/FOGA_Supplementary_Review_Pseudo_Boolean.pdf).

## 2 PRELIMINARIES

In this section, we will first introduce basic concepts in multi-objective optimisation. We will then describe several well-known single-objective pseudo-Boolean functions used in runtime analysis and their extension to multi-objective pseudo-Boolean functions.

### 2.1 Multi-Objective Optimisation

Let  $f(x) = (f_1(x), f_2(x))$  be a bi-objective (maximisation) optimisation problem in decision space  $\Omega$ , where  $f_i(x) : \Omega \rightarrow \mathbb{R}$  for all  $x \in \Omega$ . Since we only consider pseudo-Boolean functions, each  $x$  is a bit-string  $\{0, 1\}^n$ .

Considering two solutions  $x, x' \in \Omega$ , we say  $x$  weakly (Pareto) dominates  $x'$ , denoted  $x \succeq x'$ , if  $f_1(x) \geq f_1(x')$  and  $f_2(x) \geq f_2(x')$ ;  $x$  dominates  $x'$  if one of the inequalities is strict, denoted  $x \succ x'$ . A

solution  $x$  is *Pareto optimal* if no other solution in  $\Omega$  dominates it. The set of all such solutions is called the *Pareto set* in the decision space, and their image under  $f$  is the *Pareto front* (objective space).

### 2.2 Single-objective pseudo-Boolean Functions and Their Multi-objective Extensions

In single-objective runtime analysis, a range of pseudo-Boolean functions have been studied. We focus on four representative ones: OneMax, LeadingOnes, Jump, and RoyalRoad.

**2.2.1 OneMax.** OneMax counts the number of 1s in a bit-string.

**Definition 2.1.** (*OneMax* [33]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ . The OneMax function is defined as:

$$\text{OneMax}(x) = \sum_{i=1}^n x_i. \quad (1)$$

Its symmetric counterpart is the OneMin (ZeroMax) function, as the second objective function of the OneMinMax benchmark.

**Definition 2.2.** (*OneMinMax* (OMM) [15]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ . The problem has two objectives:

$$f_1(x) = \sum_{i=1}^n x_i \quad \text{and} \quad f_2(x) = \sum_{i=1}^n (1 - x_i). \quad (2)$$

The goal of the function is to simultaneously maximise both  $f_1(x)$  and  $f_2(x)$ , which is inherently contradictory, as improving one objective results in decreasing the other.

**2.2.2 LeadingOnes.** This problem counts the number of consecutive ones starting from the leftmost side of the bit-string.

**Definition 2.3.** (*LeadingOnes* [34]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ . The LeadingOnes function is defined as:

$$\text{LeadingOnes}(x) = \sum_{i=1}^n \prod_{j=1}^i x_j. \quad (3)$$

The symmetric counterpart of LeadingOnes is the TrailingZeroes function, as the second objective function of the LeadingOnes-TrailingZeroes benchmark in the following.

**Definition 2.4.** (*LeadingOnes-TrailingZeroes* (LOTZ) [26]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ . The problem is defined as:

$$f_1(x) = \sum_{i=1}^n \prod_{j=1}^i x_j \quad \text{and} \quad f_2(x) = \sum_{i=1}^n \prod_{j=i}^n (1 - x_j). \quad (4)$$

The goal here is to maximise both  $f_1(x)$  and  $f_2(x)$ , which encourages solutions with 1s in the beginning and 0s at the end.

**2.2.3 Jump.** The Jump function (also known as the  $k$ -Jump function) is a variation of the OneMax function but involves jumps (across a valley) in the fitness landscape. The objective function penalises solutions that are close to the optimum (the penalty becoming severer as getting closer). Formally, it is defined as follows.

**Definition 2.5.** ( *$k$ -Jump (OneJump)* [35]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$  with a jump parameter  $k \in \{1, 2, \dots, n\}$  indicating

the size of the valley. The  $k$ -Jump function is formalised as:

$$\text{Jump}_k(x) = \begin{cases} k + |x|_1, & \text{if } |x|_1 \leq n - k \text{ or } x = 1^n \\ n - |x|_1, & \text{otherwise.} \end{cases} \quad (5)$$

In the area of multi-objective optimisation, the jump function and its symmetric counterpart – ZeroJump function – composed of the OneJump-ZeroJump benchmark, as in the following.

**Definition 2.6.** (*OneJump-ZeroJump (OJZJ)* [27]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$  and let  $k$  be a fixed jump parameter with  $1 \leq k < \frac{n}{2}$ . The OneJump-ZeroJump problem is defined as:

$$\begin{aligned} f_1(x) &= \begin{cases} k + |x|_1, & \text{if } |x|_1 \leq n - k \text{ or } x = 1^n \\ n - |x|_1, & \text{otherwise} \end{cases} \\ f_2(x) &= \begin{cases} k + |x|_0, & \text{if } |x|_0 \leq n - k \text{ or } x = 0^n \\ n - |x|_0, & \text{otherwise.} \end{cases} \end{aligned} \quad (6)$$

Here,  $f_1(x)$  prefers bit-strings with a larger number of ones until it gets close to  $1^n$ , in which case the objective value drops. Similarly,  $f_2(x)$  prefers strings with a larger number of zeros until it is too large and close to  $0^n$ . This forms two symmetric and (partially) conflicting objectives, which we will explain in detail in Section 4.

**2.2.4 RoyalRoad.** The RoyalRoad function is designed to provide a fitness landscape with explicit building blocks which introduce neutral areas.

**Definition 2.7.** (*RoyalRoad* [36]). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ , partitioned into  $b$  disjoint blocks  $S_1, S_2, \dots, S_b$ , each with the same length  $\ell$  (where  $n = b \cdot \ell$ ,  $b > 1$ ). The RoyalRoad function is defined as:

$$\text{RoyalRoad}(x) = \sum_{j=1}^b \left( \ell \prod_{i \in S_j} x_i \right). \quad (7)$$

Likewise to OneJump, we regard the RoyalRoad function here as OneRoyalRoad, its symmetric counterpart is ZeroRoyalRoad, as the second objective function of the OneRoyalRoadZeroRoyalRoad benchmark in the following.

To our knowledge, the RoyalRoad function has not been considered and studied in multi-objective optimisation<sup>1</sup>, though it can be trivially extended to by optimising the OneRoyalRoad and ZeroRoyalRoad functions.

**Definition 2.8.** (*OneRoyalRoad-ZeroRoyalRoad (ORZR)*). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ , partitioned into  $b$  disjoint blocks  $S_1, S_2, \dots, S_b$ , each with the same length  $\ell$  (where  $n = b \cdot \ell$ ,  $b > 1$ ). The problem is:

$$f_1(x) = \sum_{j=1}^b \left( \ell \prod_{i \in S_j} x_i \right), \quad \text{and} \quad f_2(x) = \sum_{j=1}^b \left( \ell \prod_{i \in S_j} (1 - x_i) \right). \quad (8)$$

Here,  $f_1(x)$  prefers strings that have all bits set to one in each block, while  $f_2(x)$  prefers strings that have all bits set to zero in each block, forming two symmetric and conflicting objectives.

<sup>1</sup>Note that there is a similar but different problem, called RealRoyalRoad [43], has been extended and studied recently in multi-objective optimisation [19, 24, 44].

### 3 CHARACTERISTICS OF MULTI-OBJECTIVE OPTIMISATION PROBLEMS

In this section, we will review representative characteristics of multi-objective optimisation problems. These include characteristics that existing multi-objective pseudo-Boolean functions commonly have, and characteristics that practical multi-objective optimisation problems typically have. In doing so, we try to explain what having (or not having) these characteristics may imply.

#### 3.1 Completely Conflicting Objectives

An interesting characteristic resulting from the symmetric extension of single-objective pseudo-Boolean functions is that some multi-objective functions may have rather conflicting objectives. An extreme example is the OneMinMax function [15] where the two objectives are completely conflicting. That is, an improvement on one objective always leads to a deterioration on the other. This characteristic can be defined as follows.

**Definition 3.1.** (*Completely conflicting objectives*).

$$\begin{aligned} \text{For all } x, x' \in \Omega : \quad & f_1(x') < f_1(x) \implies f_2(x') > f_2(x) \\ & \text{and} \quad f_2(x') < f_2(x) \implies f_1(x') > f_1(x). \end{aligned} \quad (9)$$

When a multi-objective optimisation problem has this characteristic, all solutions in the decision space are Pareto optimal ones. This apparently is not very realistic, though some early continuous benchmark functions do have this characteristic, such as SCH [45]. From the perspective of MOEAs, this type of problem does not test an algorithms' ability to converge, but rather its ability to distribute solutions over the Pareto front. As such, some diversity/novelty-driven search [46, 47] strategies may have the edge in addressing such problems.

#### 3.2 Symmetry

Another characteristic resulting from the straightforward extension of single-objective pseudo-Boolean functions is that the resultant multi-objective problems are symmetric, meaning that the objectives are similar in form or mirror each other. For pseudo-Boolean problems, this means reversing each bit as well as their positions. Formally, it can be defined as follows.

**Definition 3.2.** (*symmetry in pseudo-Boolean functions*). Let  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  and  $g : \{0, 1\}^n \rightarrow \mathbb{R}$  be two functions on binary strings of length  $n$ .  $f$  and  $g$  are **symmetric** if

$$f(x) = g(B(R(x))) \quad \text{and} \quad g(x) = f(B(R(x))) \quad (10)$$

where  $R(x)$  is a function that reverses the order of bit-string  $x$  from  $\{1, 2, \dots, n\}$  to  $\{n, \dots, 2, 1\}$ ;  $B(x)$  is a bitwise complementation function that flips all bits (i.e.,  $x_i = 1 - x_i$  for  $i \in \{1, 2, \dots, n\}$ ).

Crossover may benefit from such a symmetric setup. For example, if the all-zero solution is a non-dominated solution, then the all-one solution must be a non-dominated solution as well, hence being preserved highly likely in the search process of an MOEA. As all the bits of these two solutions are different, crossing them over may generate any bit-string, potentially speeding up the search.

Another implication of having the symmetric setup is that solutions in the objective space are symmetric with respect to the line  $f_1 = f_2$ . Such symmetry is rare in real-world scenarios, where

objectives often differ in scale and structure. For example, in the multi-objective capacitated vehicle routing problem [48], objectives like total distance and vehicle load balance are inherently heterogeneous and asymmetric. As for the effect on algorithms, some types of MOEAs are sensitive to problems with different objective ranges, such as indicator-based and decomposition-based algorithms [29, 49, 50].

### 3.3 Disjoint Pareto Optimal Solutions

Disjointness means that not all the optimal solutions (in decision space) are connected. Connectedness can be defined using the notion of *neighbourhood*, with an intuitive assumption that neighbouring solutions are connected to each other by default. In terms of pseudo-Boolean functions, neighbourhood can be defined by one-bit difference of solutions (i.e. Hamming distance of 1).

**Definition 3.3.** (*Neighbourhood*). The neighbourhood of a solution  $x$  is

$$\mathcal{N}(x) = \{x' \in \Omega \mid \text{Hamming}(x, x') = 1\}. \quad (11)$$

Consider the set of Pareto optimal solutions as a directed graph where a node is a solution and the edge is defined by the neighbourhood relationship, this set is disjoint if the graph is disconnected (i.e., there exists no path between some pairs of nodes). Formally, it can be defined as follows.

**Definition 3.4.** (*Disjoint Pareto optimal solutions*). For a set of Pareto optimal solutions  $\mathcal{PS}$ , denote a graph  $G_{\mathcal{PS}} = (\mathcal{PS}, E)$ , where a node corresponds to a solution in  $\mathcal{PS}$ , and  $E = \{(x, x') \mid x, x' \in \mathcal{PS} \text{ and } x' \in \mathcal{N}(x)\}$ . The problem has disjoint Pareto optimal solutions if  $G_{\mathcal{PS}}$  is disconnected.

In many existing pseudo-Boolean functions for runtime analysis, Pareto optimal solutions are not disjoint, hence forming a connected region, such as in OneMinMax [15] and LeadingOnes-TrailingZeros [26]. In this case, local search can easily reach all Pareto optimal solutions, provided that it hits one of them. However, many practical combinatorial problems have been shown that their Pareto optimal regions are disjoint, such as multi-objective linear programming [51], spanning tree [51] and knapsack [52]. This can also be echoed by the multi-funnel structure in multi-objective combinatorial problems [53]. As for algorithm design, having disjoint Pareto optimal solutions may highlight the importance of maintaining the diversity of solutions in decision space, rather than purely in objective space. Unfortunately, mainstream MOEAs only consider the latter, and performance of such algorithms can be improved substantially through considering the diversity of solutions in decision space [54–56].

### 3.4 Pareto Local Optimality

Local optima are commonly observed in real-world problems, which may cause the search of an optimisation algorithm to be stuck. In single-objective optimisation with fitness function  $f : \Omega \rightarrow \mathbb{R}$ , a solution  $x$  is a local optimum if, for some neighbourhood  $\mathcal{N}(x) \subset \Omega$ , we have  $\nexists x' \in \mathcal{N}(x)$  such that  $f(x') > f(x)$ , meaning that its fitness is not worse than any of its neighbours. There are some single-objective pseudo-Boolean functions having local optimal solutions, such as  $k$ -Jump [35].

Similarly, in the multi-objective case, a solution is a Pareto local optimum if it is not dominated by any of its neighbours [57].

**Definition 3.5.** (*Pareto local optimality*). A solution  $x$  is a Pareto local optimum if

$$\nexists x' \in \mathcal{N}(x) \text{ such that } x' \succ x. \quad (12)$$

Note that global optimal solutions are also local optimal solutions by definition, but for the convenience of discussion, we exclude them when discussing local optimal solutions.

Pareto local optimality is an important characteristic that many practical problems have, and it can significantly increase the level of hardness for MOEAs to deal with. For example, in the DTLZ suite [58], the only difference between DTLZ2 and DTLZ3 is that the latter has a number of local Pareto fronts, which imposes a much bigger challenge for MOEAs to converge [59].

However, many well-established multi-objective pseudo-Boolean functions in runtime analysis do not have Pareto local optimal solutions, though some of their single-objective versions have. For example, OneJump-ZeroJump [27], the extension of  $k$ -Jump, does not have any Pareto local optimal solution. The local optimal solutions in  $k$ -Jump become non-dominated solutions (i.e., global optimal solutions in the multi-objective context) as no solution is better than them on both objectives. We will explain it in the Section 4.3.

### 3.5 Separability of Variables

Separability refers to how variables in the problem interact with one another with respect to the objective function.

**Definition 3.6.** (*Separability* [60]). An objective function  $f(x)$  with  $x \in \Omega$  is **separable** if it can be expressed as:

$$f(x) = g_1(x_1) + g_2(x_2) + \dots + g_n(x_n) \quad (13)$$

where  $g_i$  is a function of the single variable  $x_i$ .

The definition of separability is strict, requiring every variable of the problem is separable. With such a characteristic, an optimiser can optimise the problem variable by variable [60, 61]. However, not many real-world problems meet this condition. Instead, they may be partially separable, i.e., some of the variables can be optimised separately. For example, in many ZDT functions [62], variables can be separated as the first variable  $x_1$  alone controls the first objective and the remaining variables contribute to the second objective.

### 3.6 Ratio of Pareto Optimal Solutions

In real-world problems, Pareto optimal solutions usually only take a small portion of the search space. However, in some multi-objective pseudo-Boolean functions in runtime analysis (e.g., OneMinMax and OneJump-ZeroJump), most of their solutions are Pareto optimal ones. For a pseudo-Boolean function, the ratio of Pareto optimal solutions is  $\frac{|\mathcal{PS}|}{2^n}$ , where  $\mathcal{PS}$  denotes the set of Pareto optimal solutions. We say a function has a low ratio if it converges to 0 as  $n$  grows to infinity. Most practical problems have a low ratio, even though the number of the Pareto optimal solutions can grow exponentially with the problem size [72], such as in bi-objective versions of the shortest path problem [73], minimum spanning tree problem [74] and integer minimum-cost flow problem [75].

**Table 1: Characteristics of practical multi-objective optimisation problems and whether existing and new multi-objective pseudo-Boolean functions in runtime analysis have them.**

Problem	Non-symmetric objectives	Non-completely conflicting objectives	Disjoint optimal solutions	Not fully separable variables	Low ratio of Pareto optimal solutions	Non-linear Pareto front shape	Pareto local optimality	Studied in
OneMinMax (OMM)								[3, 6, 12, 15, 22, 56, 63–68]
LeadingOnes-TrailingZeroes (LOTZ)		✓		✓	✓			[3, 4, 6, 14, 17, 18, 22, 26, 38, 56, 63, 65, 66, 69, 70]
OneJump-ZeroJump (OJZJ)		✓	✓	✓	✓ <sup>1</sup>			[1, 5–7, 27, 71]
CountOnes-CountZeroes (COCZ)	✓	✓			✓			[10, 14, 18, 38, 41]
OneRoyalRoad-ZeroRoyalRoad (ORZR)		✓	✓	✓	✓		✓ <sup>2</sup>	
OneMax-TrailingZeroes (OMTZ)	✓	✓		✓	✓			
OneMax-ZeroJump (OMZJ)	✓	✓	✓	✓	✓			
OneMax-ZeroRoyalRoad (OMZR)	✓	✓	✓	✓	✓			
LeadingOnes-ZeroJump (LOZJ)	✓	✓	✓	✓	✓		✓	
LeadingOnes-ZeroRoyalRoad (LOZR)	✓	✓	✓	✓	✓		✓	
OneJump-ZeroRoyalRoad (OJZR)	✓	✓	✓	✓	✓	✓ <sup>3</sup>	✓	

<sup>1</sup> The ratio of Pareto optimal solutions is not low when the jump parameter  $k \leq \frac{n}{2} - \sqrt{\frac{n \ln 4}{2}}$  (Section 4 in the supplementary material [https://github.com/Zim-L/FOGA25ReviewPseudoBoolean/blob/main/FOGA\\_Supplementary\\_Review\\_Pseudo\\_Boolean.pdf](https://github.com/Zim-L/FOGA25ReviewPseudoBoolean/blob/main/FOGA_Supplementary_Review_Pseudo_Boolean.pdf)).

<sup>2</sup> Pareto local optima exist when block length  $\ell > 3$ .

<sup>3</sup> The shape of the Pareto front is generally non-linear, except for the case that  $(n - k) \bmod \ell = 0$ .

### 3.7 Shape of the Pareto Front

Real-world multi-objective optimisation problems may have a very complex geometry of their Pareto front shapes, such as various convex, concave, mixed, and/or disjoint fronts. However, existing multi-objective pseudo-Boolean functions in runtime analysis usually have a simple linear Pareto front. For MOEAs which aim to approximate a good representation of a problem’s Pareto front with a fixed population size, the shape of the problem matters. Some types of MOEAs favour a linear Pareto front, such as hypervolume-based algorithms [76] and decomposition-based algorithms [30]. In contrast, Pareto-based algorithms are robust to Pareto front shapes [29]. This may be one of the reasons that NSGA-II may generally perform better than SMS-EMOA [77] and MOEA/D [28] in classic combinatorial problems [31] and real-world problems [78].

## 4 REVIEW OF MULTI-OBJECTIVE BOOLEAN FUNCTIONS

In this and the next sections, we will review multi-objective pseudo-Boolean functions based on the characteristics described in Section 3. This section covers the well-known benchmarks such as OneMinMax [15], LeadingOnes-TrailingZeroes [26], OneJump-ZeroJump [27] and CountOnes-CountZeroes [14]; the next section introduces new mix-and-match benchmarks of the single-objective functions. Table 1 summarises how these functions reflect the key characteristics of practical multi-objective optimisation problems.

### 4.1 OneMinMax (OMM)

OMM [15] is to simultaneously maximise the number of ones and zeros in a bit-string, defined in Definition 2.2. It has been commonly used in runtime analysis (see Table 1). OMM is arguably one of the least realistic multi-objective pseudo-Boolean functions. As can be seen in Table 1, OMM has none of the characteristics that practical multi-objective problems may have.

Figure 1 illustrates the 8-bit OMM function from three different perspectives. Specifically, Figure 1(a) plots the solutions on two individual objectives (OneMax and OneMin) with respect to the number of ones. Figure 1(b) plots the objective space, where the number associated with a solution means how many solutions in

the decision space map to that solution. Figure 1(c) plots the non-dominated levels [79] of solutions with respect to the number of ones, which helps indicate how easy solutions may move into a better place with respect to the Pareto non-dominated levels (similar to *Pareto Landscape* [80]).

As can be seen from Figure 1(a), the two objectives OneMax and OneMin are completely conflicting, making all solutions Pareto optimal (Figure 1(b) and (c)). In addition, the objectives in OMM are fully separable with respect to their variables, allowing MOEAs to optimise the problem one variable (bit) at a time.

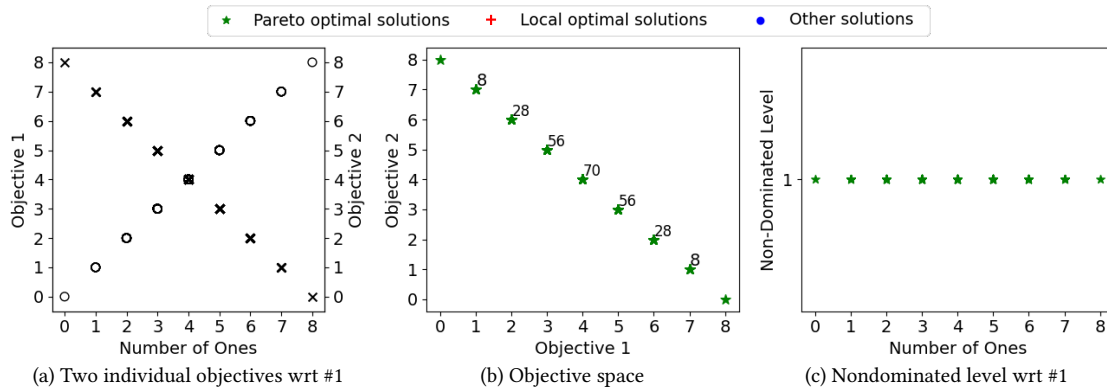
### 4.2 LeadingOnes-TrailingZeroes (LOTZ)

LOTZ [26] is to simultaneously maximise the number of consecutive ones from left to right and the number of consecutive zeroes from right to left. It is one of the earliest and most studied multi-objective pseudo-Boolean functions. As can be seen in Table 1, LOTZ has some unrealistic characteristics such as symmetric objectives, a linear Pareto front, no disjoint optimal solutions nor local optima, (see Figure 2). The last two may make the problem generally easy to solve (i.e., finding all the optimal solutions).

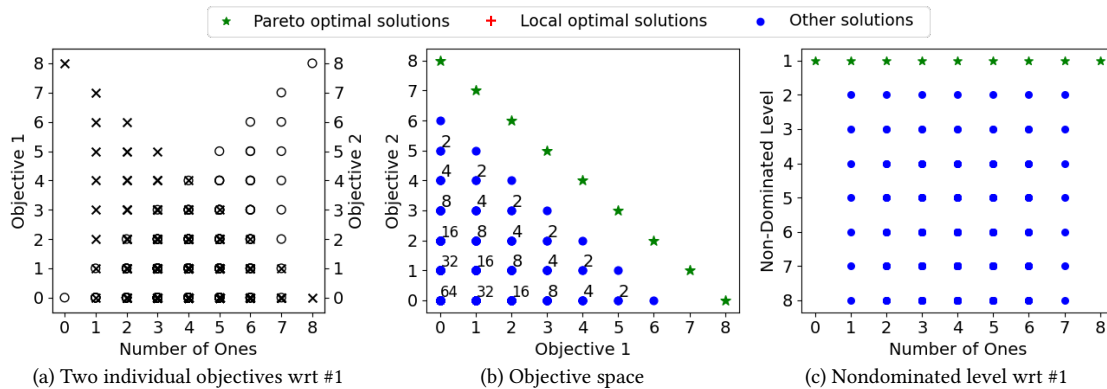
Unlike OneMinMax, the LOTZ function exhibits characteristics such as non-completely conflicting objectives and a low ratio of Pareto optimal solutions. As can be seen from Figure 2(a), the presence of dominated solutions indicates that the objectives are not completely conflicting. The number associated with a point in Figure 2(b) stands for the number of solutions with identical objective values (a point without a number is a unique solution). Clearly, most of the solutions are dominated ones and there are only  $n + 1$  solutions in the Pareto front. In addition, unlike OMM, the objectives in LOTZ are not fully separable, implying that MOEAs can no longer optimise them independently on a bit-wise basis.

### 4.3 OneJump-ZeroJump (OJZJ)

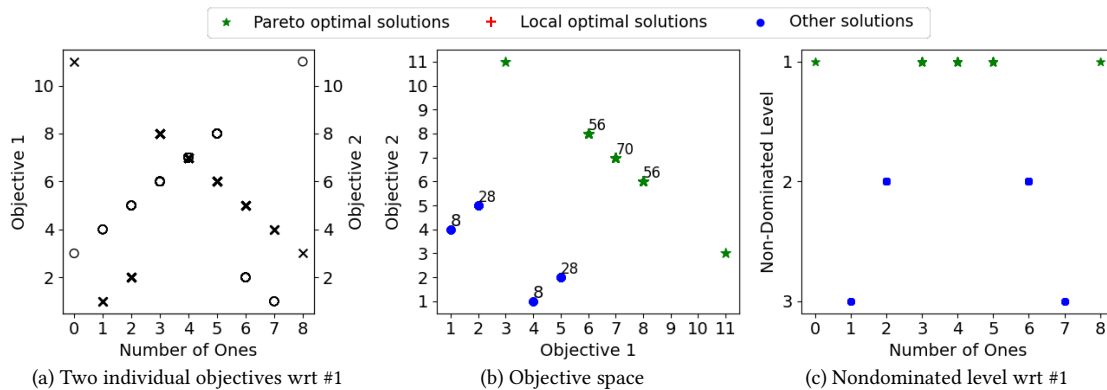
OJZJ [27] is to simultaneously maximise the number of ones and zeros while introducing a deceptive fitness gap that requires crossing a valley in each objective. The size of the valley is controlled by a parameter called the *jump parameter*  $k$ . Its single-objective version, *OneJump* function (Eq. 2.5), is known for the presence of a local



**Figure 1: OneMinMax (OMM) ( $n = 8$ )** (a) Two individual objectives (OneMax and OneMin) of the OMM problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting [79]. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions, and other solutions respectively. In this problem, all solutions are Pareto optimal.



**Figure 2:** LeadingOnes-TrailingZeros (LOTZ) ( $n = 8$ ) (a) Two individual objectives (LeadingOnes and TrailingZeros) of the LOTZ problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions, and other solutions respectively.

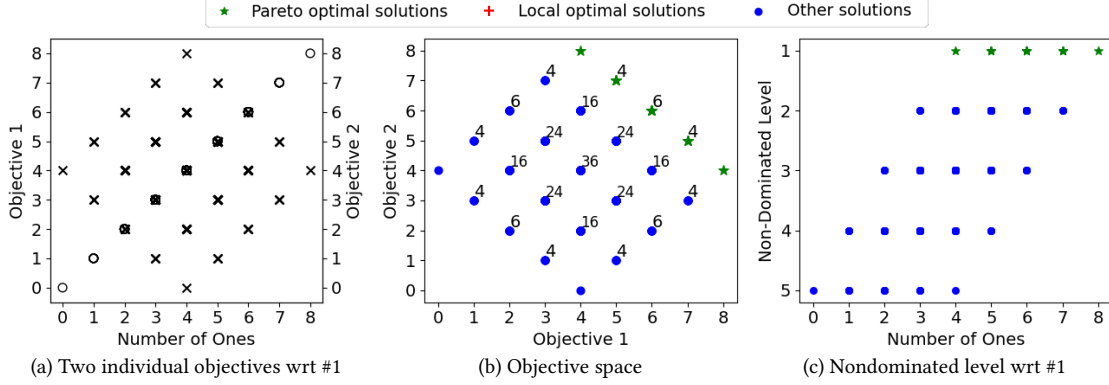


**Figure 3:** OneJump-ZeroJump (OJZJ) ( $n = 8, k = 2$  where  $k$  is the jump parameter) (a) Two individual objectives (OneJump and ZeroJump) of the OJZJ problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions, and other solutions respectively.

optimum. Interestingly, in the multi-objective version, the local optimum becomes a Pareto optimal solution. Figure 3(a) plots the solutions on two individual objectives (OneJump and ZeroJump) with respect to the number of ones for the 8-bit OJZJ problem with

$k = 2$ . As seen, on the objective  $f_1$ , the solution with five ones is a local optimum for that objective; however, it becomes non-dominated if considering the objective  $f_2$ . The global optimal solution on each objective (i.e.,  $(1^n)$  and  $(0^n)$ ) becomes the boundary solution in this two-objective problem.





**Figure 4:** CountOnes-CountZeroes (COCZ) ( $n = 8$ ) (a) Two individual objectives (OneMax and CountOnes) of the COCZ problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions, and other solutions respectively.

A difference of OJZJ from the previous problems is that it has disjoint optimal solutions. As can be seen in Figure 3(c), there is a gap between the boundary solutions and the remaining Pareto optimal solutions. This brings difficulty in finding the whole Pareto front, especially with a big  $k$ . The ratio of Pareto optimal solutions in the problem depends on the size of the valley (i.e.,  $k$ ) as only solutions in the valley are dominated ones. When  $k$  is small, relative to the number of total bits ( $n$ ), the ratio is high. For example, if  $k \leq \frac{n}{2} - \sqrt{\frac{n \ln 4}{2}}$ , the ratio is not less than 0.5 with a sufficiently large  $n$ . When  $k$  is large (e.g.,  $k = \frac{n}{2} - 1$ ), the ratio converges toward 0 with a sufficiently large  $n$ , see Section 4 in the supplementary.

#### 4.4 CountOnes-CountZeroes (COCZ)

COCZ [14] is a problem known for having asymmetric objectives. The problem consists of two parts: a cooperative part (the first half of the decision variables) and a conflicting part (the second half of the decision variables). In the cooperative part, both objectives aim to maximise the number of ones (like two OneMax functions). In the conflicting part, one objective maximises the number of ones and another objective maximises the number of zeroes (like OneMinMax). Formally, COCZ can be formulated below.

**Definition 4.1.** (*CountOne-CountZeroes (COCZ)* [14]) Let  $x \in \{0, 1\}^n$  be a bit-string of even bits. The COCZ function is defined as

$$f_1(x) = \sum_{i=1}^n x_i \quad \text{and} \quad f_2(x) = \sum_{i=1}^{n/2} x_i + \sum_{i=n/2+1}^n (1 - x_i). \quad (14)$$

The first objective is the OneMax function, and the second objective is called the single-objective *CountOnes* function [60] which treats half of the decision variables as OneMax and another half as OneMin. This feature makes the problem non-symmetric and non-completely conflicting with respect to the two objectives.

Figure 4 illustrates the 8-bit COCZ function. As can be seen from Figure 4(b), the solutions in the objective space form a diamond shape. This occurrence can be attributed to the fact that the cooperative part of the bit-string determines the distance between a solution and the Pareto front (from the upper right to the lower left), the conflicting part alone controls the positions of solutions on the Pareto front. This bipartite feature partly resembles several

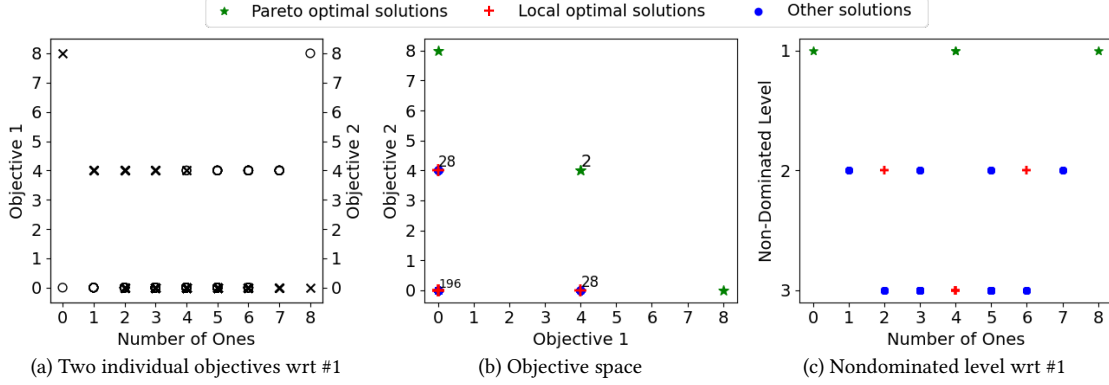
continuous benchmark suites widely used in the empirical community, such as ZDT [62] and WFG [58]. In addition, the ratio of the Pareto optimal solutions is low ( $1/2^{n/2}$ ) as only the conflicting part affects the Pareto front.

#### 4.5 OneRoyalRoad-ZeroRoyalRoad (ORZR)

Unlike the previous problems, to our knowledge ORZR has not yet been considered in runtime analysis. It divides the bit-string into equal-sized blocks, each requiring all ones or zeroes to contribute to the objective (see Definition 2.8). This design leads to disjoint optimal solutions as they are at least  $b$  bits from each other (where  $b$  denotes the number of bits in a block), and a low ratio of Pareto optimal solutions (i.e.,  $\frac{2^{n/b}}{2^n}$ ) as optimal solutions must consist of only blocks of all ones (or all zeroes). For the formal form of Pareto set and Pareto local optimal solutions, see Section 3 in the supplementary.

The single-objective RoyalRoad function [36] is known for the presence of the neutral area, where there exists at least a solution that all of its neighbours are of the same fitness. Such a neutral area is considered a challenging feature [81, 82]. The multi-objective version inherits this feature; there exist solutions such that all of their neighbours are non-dominated to itself. This introduces a feature that is prevalent in practical problems but does not exist in the other existing multi-objective pseudo-Boolean functions – Pareto local optimality. Figure 5 illustrates the 8-bit ORZR with two blocks. As can be seen from Figure 5(c), there are three local optimal solutions whose neighbours are not better than them. It is worth noting that in ORZR the local optimal solutions and their neighbours are not “incomparable” but have the same objective values (i.e., mapped into the same objective vectors, see Figure 5(b)). This may pose the challenge for mainstream MOEAs that maintain diversity only in the objective space, as they cannot identify potential solutions that help in finding promising areas [56].

In addition, ORZR exhibits a property that is common in real-world applications but rare in the pseudo-Boolean benchmarks. While optimal solutions require each block to consist entirely of ones or zeroes, some blocks in random solutions have roughly the same number of ones and zeroes. This creates a position at the “centre” of a plateau where improving the block requires multiple



**Figure 5: OneRoyalRoad-ZeroRoyalRoad (ORZR) ( $n = 8$ ,  $b = 2$  where  $b$  is the number of blocks)** (a) Two individual objectives (OneRoyalRoad and ZeroRoyalRoad) of the ORZR problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions and other solutions respectively. In this problem, local optimal solutions (red crosses) and other solutions (blue dots) overlap in (b).

simultaneous bit flips. This makes ORZR challenging for algorithms like SEMO that relies on local variation.

## 5 MIX-AND-MATCH MULTI-OBJECTIVE BOOLEAN FUNCTIONS

In the previous section, the bi-objective functions are formed by extending the symmetric version of single-objective ones. In this section, we introduce the asymmetric mixes of two different objectives: the first objective remains its original form, while the second is its symmetric counterpart. For example, mixing LeadingOnes and OneJump yields *LeadingOnes-ZeroJump*. Due to space limitations, the mixes involving OneMax are provided separately in the supplementary, as they often trivialise the problem. Formal properties (e.g., the Pareto solution set and local optimal set) of all the new benchmarks are given in Section 3 in the supplementary.

### 5.1 LeadingOnes-ZeroJump (LOZJ)

The problem LOZJ combines the LeadingOnes function with the ZeroJump function, and it simultaneously maximises the number of consecutive ones (from left to the right), and the number of zeroes in the bit-string with a valley (determined by the parameter  $k$ ). Formally, LOZJ can be formulated as follows.

**Definition 5.1.** (*LeadingOnes-ZeroJump*). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$  and  $k$  be the jump parameter with  $1 < k < \frac{n}{2}$ . The problem is defined as:

$$f_1(x) = \sum_{i=1}^n \prod_{j=1}^i x_j, \quad f_2(x) = \begin{cases} k + |x|_0, & \text{if } |x|_0 \leq n - k \text{ or } x = 0^n \\ n - |x|_0, & \text{otherwise.} \end{cases} \quad (15)$$

As LOZJ combines LeadingOnes-TrailingZeroes with OneJump-ZeroJump, it inherits their characteristics, such as having disjoint optimal solutions and a low ratio of optimal solutions (Table 1). However, a distinct characteristic of LOZJ is its Pareto local optimality. Figure 6(c) shows three local optima near the valley (in red), corresponding to ZeroJump's local optima. For these local optimal solutions, flipping any bit reduces the ZeroJump value; hence these solutions are non-dominated by their neighbours. Take the local optimal solution 10010010 in Figure 6(c) as an example,

where the solution is located in non-dominated level 3 with objective values (1, 7). Flipping any of its bits from one to zero will potentially lead to one of the three solutions, 00010010 (level 9), 10000010 and 10010000 (level 8). Likewise, flipping any bit from zero to one will potentially lead to one of the five solutions, 11010010 (level 3), 10110010, 10011010, 10010110 and 10010011 (level 4).

It is worth mentioning is that the valley in LOZJ poses a bigger challenge for MOEAs to reach the all-zero optimal solution than in OneJump-ZeroJump (OJZJ). To do so, (i.e., the top-left solution in Figure 6(b)), MOEAs generally need to find the nearest Pareto optimal solution first, i.e., the one with objective values (3, 7) in Figure 6(b). However, in LOZJ, this solution corresponds to only one bit-string (11100000), whereas in OJZJ, it corresponds to any bit-string with exactly three 1 bits (e.g., 56 bit-strings in Figure 3(b)). This many-to-one mapping in OJZJ enables crossover to speed up the jump [83], but this is not possible in LOZJ. Such hard-to-reach boundary solutions also appear in real-world problems [84, 85].

### 5.2 LeadingOnes-ZeroRoyalRoad (LOZR)

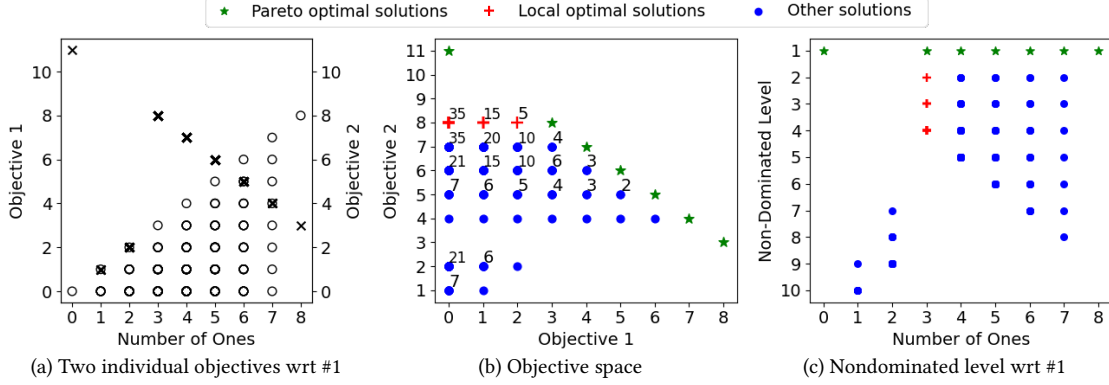
LOZR combines the LeadingOnes function with the ZeroRoyalRoad function, simultaneously maximising the number of consecutive ones (from left to right) in a bit-string, and the number of blocks with all zero bits. Formally, LOZR can be formulated as follows.

**Definition 5.2.** (*LeadingOnes-ZeroRoyalRoad*). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ , partitioned into  $b$  disjoint blocks  $S_1, S_2, \dots, S_b$ , with each of length  $\ell$  (where  $n = b \cdot \ell$ ,  $b > 1$ ).

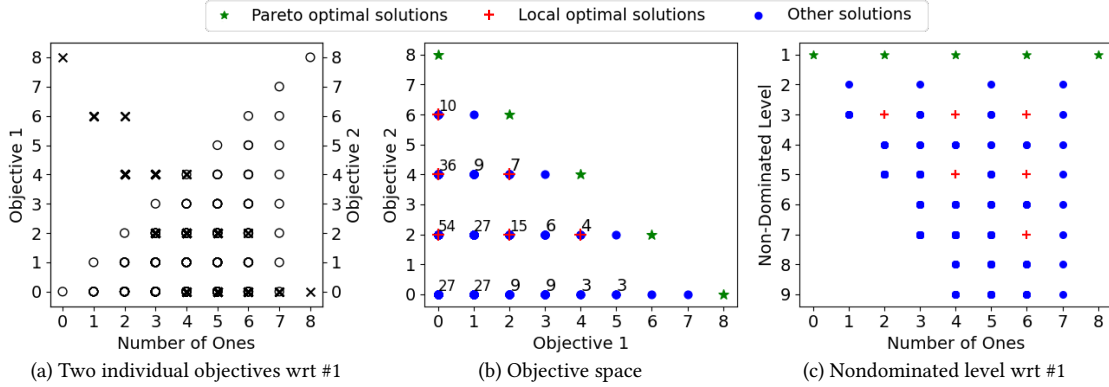
$$f_1(x) = \sum_{i=1}^n \prod_{j=1}^i x_j, \quad \text{and} \quad f_2(x) = \sum_{j=1}^b \left( \ell \prod_{i \in S_j} (1 - x_i) \right). \quad (16)$$

LOZR shares many characteristics with OneRoyalRoad-ZeroRoyalRoad (ORZR), for example, having disjoint optimal solutions, a low ratio of optimal solutions, and Pareto local optimality (Table 1). However, an interesting difference is that in ORZR, Pareto local optimal solutions only exist when the block length is greater than three ( $\ell > 3$ ), whereas in LOZR they exist with any legitimate block length (i.e.,  $\ell \geq 2$ ). The reason for this is as follows.





**Figure 6:** LeadingOnes-ZeroJump (LOZJ) ( $n = 8, k = 3$  where  $k$  is the jump parameter, controlling the size of the valley). (a) Two individual objectives (LeadingOnes and ZeroJump) of the LOZJ problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions and other solutions, respectively.



**Figure 7:** LeadingOnes-ZeroRoyalRoad (LOZR) ( $n = 8, b = 4$  where  $b$  denotes the number of blocks). (a) Two individual objectives (LeadingOnes and ZeroRoyalRoad) of the LOZR problem with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The level of solutions with respect to the number of ones based on the Pareto non-dominated sorting. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions, and other solutions respectively. In this problem, the local optimal solutions (red cross) overlap with other solutions (blue dots) in the (b).

In ORZR, a Pareto local optimal solution needs to contain at least one block that is neither  $1^\ell$  nor  $0^\ell$ , preventing it from being globally optimal. For such a block, flipping a single bit must not turn it into  $1^\ell$  or  $0^\ell$ , as that would improve at least one objective (while the other stays unchanged). This condition can only be met when the block length  $\ell > 3$ , i.e., having at least two 1s and at least two 0s. In contrast, in LOZR a requirement for being a Pareto local optimal solution is that a single-bit flip cannot turn a block into  $0^\ell$  (but can be  $1^\ell$  as it only considers the ZeroRoyalRoad function). As a result, LOZR does not require a block length greater than three because a block of length two (i.e., 11) is sufficient to prevent itself from being turned into an all-zero block via a single-bit flip.

### 5.3 OneJump-ZeroRoyalRoad (OJZR)

OJZR combines the OneJump function with the ZeroRoyalRoad function, simultaneously maximising the number of ones with a valley (determined by the parameter  $k$ ), and the number of blocks with all zero bits. It can be formulated as follows.

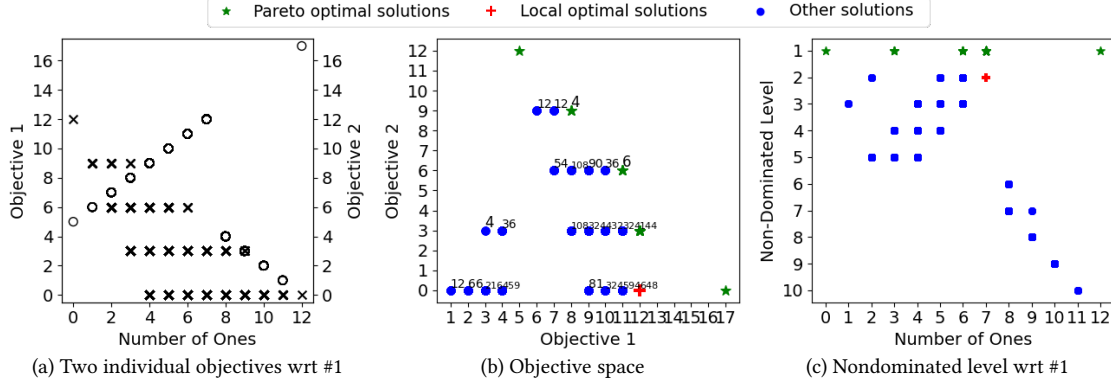
**Definition 5.3.** (*OneJump-ZeroRoyalRoad*). Let  $x \in \{0, 1\}^n$  be a bit-string of length  $n$ , partitioned into  $b$  disjoint blocks  $S_1, S_2, \dots, S_b$ , each of length  $\ell$  (where  $n = b \cdot \ell, b > 1$ ). Let  $k \in \mathbb{Z}^+$  be the jump

parameter with  $1 < k < \lfloor \frac{n}{2} \rfloor$ . The problem is defined as:

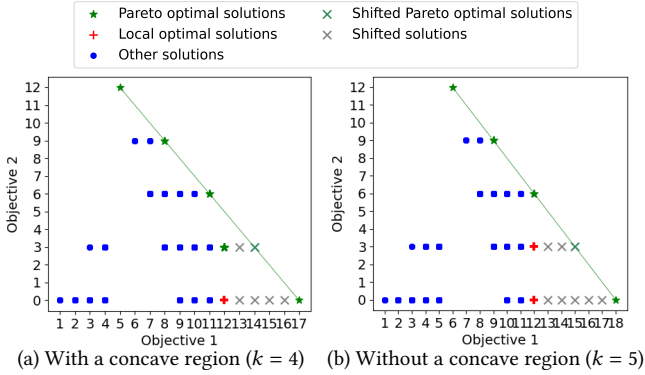
$$f_1(x) = \begin{cases} k + |x|_1, & \text{if } |x|_1 \leq n - k \\ & \text{or } x = 1^n \\ n - |x|_1, & \text{otherwise} \end{cases} \quad f_2(x) = \sum_{j=1}^b (\ell \prod_{i \in S_j} (1 - x_i)). \quad (17)$$

OJZR exhibits all the characteristics listed in Table 1, such as the Pareto local optimality and a low ratio of Pareto optimal solutions. Notably, it is the only problem that has a non-linear Pareto front (having a concave region). Taking the 12-bit OJZR problem ( $k = 5, b = 4$ ) in Figure 8 as an example, the problem has a concave region near the solution with objective values (11, 3) (Figure 8(b)). The reason for this occurrence is as follows.

The occurrence of the concave region can be attributed to the fact that the OneJump objective shifts the positions of some solutions in the objective space and “creates” a new Pareto optimal solution. Figure 9(a) illustrates and explains this in the objective space of a 12-bit OJZR problem, where the grey and green crosses (columns 12, 13, 14, 15 with respect to the first objective) represent the solutions that are shifted to the left and leave a gap in the objective space (as seen in Figure 8(b)). Since the Pareto optimal solution



**Figure 8:** OneJump-ZeroRoyalRoad (OJZR) ( $n = 12, k = 5, b = 4$  where  $k$  is the jump parameter and  $b$  is the number of blocks). (a) Two individual objectives of OJZR with respect to the number of ones. (b) Objective space, where the number associated with a solution means how many solutions in the decision space map to that solution. (c) The non-dominated level of solutions with respect to the number of ones. In (b) and (c), green, red and blue points indicate Pareto optimal solutions, local optimal solutions, and other solutions respectively. In this problem, the local optimal solution (red cross) overlaps with other solutions (blue dots) in (b). Note that the Pareto front in (b) is not linear — the Pareto optimal solution (11, 3) forms a concave region.



**Figure 9:** The objective space of two OneJump-ZeroRoyalRoad (OJZR) instances ( $n = 12, b = 4, k = 5$  or  $k = 6$ ): one having a concave Pareto front and the other having a linear Pareto front, where  $b$  is the number of blocks and  $k$  is the jump parameter. (a) The OJZR instance with  $k = 5$  (b) The OJZR instance with  $k = 6$ . In both cases, by OneJump, several solutions (the crosses) are shifted to the left. When  $k = 5$ , the solution (11, 3) becomes a Pareto optimal solution, creating a concave region in the Pareto front. In contrast, when  $k = 6$  the solution (11, 3) is still a dominated solution since it is dominated by (11, 6).

(13, 3) is shifted to (3, 3), the solution (11, 3) becomes a new Pareto optimal solution. This leads to a concave region below the original straight-line Pareto front (the green line in Figure 9(a)).

However, such a shift does not always create a new Pareto optimal solution, depending on the interplay between the parameters  $n, k, \ell$  (where  $\ell$  is the block length). When  $(n - k) \bmod \ell = 0$ , the concave region no longer appears. As can be seen in Figure 9(b), the solution next to the valley (11, 3) cannot become a new Pareto optimal solution, as it is dominated by another Pareto optimal solution (11, 6). As such, the shape of the Pareto front remains a straight line, though there is a gap between the bottom-right Pareto optimal solution and the others optimal ones.

## 6 DISCUSSION AND LIMITATIONS

While this work focuses on pseudo-Boolean problems, other optimisation types exist (e.g., permutation and integer problems). Theoreticians also investigate many other multi-objective problems, such as minimum spanning tree [86–88], vertex cover problem [89], and integer-based optimisation [90, 91].

In this study, we also limit our scope to bi-objective problems. However, many practical problems have three or more objectives. Adding more objectives may rapidly increase the complexity of optimisation problems [92], and MOEAs may behave very differently [93–96]. Studies in the theory area mainly consider many-objectives problems through extending existing bi-objective problems by partitioning a bit-string into multiple sections, with each section corresponding to two objectives, such as in [11, 13, 16, 20, 97–99]. One feature of such problems is that the number of objectives to be optimised needs to be even. With the mix-and-match approach in this paper, one may naturally consider problems with more than two objectives, for example, an interesting (but maybe challenging) problem can be constructed by considering the three individual functions LeadingOnes, ZeroJump and ZeroRoyalRoad.

A notable feature of all the problems considered in this paper is the common presence of many-to-one mappings between the decision space and objective space, i.e., multiple bit-strings having identical objective values. This is due to the use of counting-based (e.g., OneMax, Jump) and block-based (e.g., RoyalRoad) functions. However, there exist many practical problems that only have one-to-one mappings between the two spaces, in particular many continuous problems like the ZDT [62] and DTLZ [58] suites. The feature of many-to-one mapping may disadvantage MOEAs that only consider the diversity of solutions in objective space, which unfortunately all mainstream MOEAs do. As presented recently, for such problems, considering the diversity of solutions in decision space can substantially speed up the search [56].

Some problems constructed in this paper have local optimal solutions. This is in line with practical scenarios that many multi-objective combinatorial problems have local optimal solutions [57]. In general, there are two types of Pareto local optimal solutions, plateau-type and non-plateau-type. Plateau-type solutions have at least one identical neighbouring solution. In our problems, such plateau local optimal solutions, introduced by the RoyalRoad function, scatter dispersedly in the search space (e.g., in LeadingOnes-ZeroRoyalRoad). This is aligned with some practical optimisation scenarios like time tabling [100]. In contrast, the non-plateau local optimal solutions, primarily caused by the Jump function, concentrate in specific regions of the search space, as seen in the problems

LeadingOnes-ZeroJump and OneJump-ZeroRoyalRoad. However, in some practical problems, the non-plateau local optimal solutions can be more dispersed such as in multi-objective NK-Landscape [101, 102], making MOEAs being trapped in different regions [103].

## 7 CONCLUSION

In this paper, we conducted a short review of pseudo-Boolean problems used in runtime analysis for evolutionary multi-objective optimisation. Based on characteristics that real-world optimisation problems have, we discussed commonly used benchmarks in the area, including their limitations and implications for practical use.

We also presented several new pseudo-Boolean problems by mix-and-matching different single-objective functions, and shown that they have more characteristics commonly seen in real-world applications (see Table 1). This includes plateau-type local optima (OneJump-ZeroRoyalRoad), a non-linear Pareto front (OneJump-ZeroRoyalRoad), and hard-to-reach boundary solutions (LeadingOnes-ZeroJump) which may not be easily obtained by crossover. Of course, these functions may not fully reflect the complexity of practical problems, but we hope they can help strengthen the connection between theoretic and practical research in the area of evolutionary multi-objective optimisation.

## ACKNOWLEDGMENT

The authors would like to acknowledge the discussions with Prof Chao Qian, Dr Chao Bian, Mr Shengjie Ren, Dr Duc-Cuong Dang, and Prof Per Kristian Lehre that motivated this work.

## REFERENCES

- [1] W. Zheng and B. Doerr. Theoretical analyses of multiobjective evolutionary algorithms on multimodal objectives. *Evolutionary Computation*, 31(4):337–373, 2023.
- [2] J. Knowles and W. Zheng. Evolutionary multiobjective optimization (EMO). In *Genetic and Evolutionary Computation Conference Companion*, pages 1432–1459, 2024.
- [3] W. Zheng, Y. Liu, and B. Doerr. A first mathematical runtime analysis of the non-dominated sorting genetic algorithm II (NSGA-II). In *AAAI Conference on Artificial Intelligence*, volume 36, pages 10408–10416, 2022.
- [4] C. Bian and C. Qian. Better running time of the non-dominated sorting genetic algorithm II (NSGA-II) by using stochastic tournament selection. In *Parallel Problem Solving from Nature*, pages 428–441, 2022.
- [5] B. Doerr and Z. Qu. A first runtime analysis of the NSGA-II on a multimodal problem. *Evolutionary Computation*, 27(5):1288–1297, 2023.
- [6] S. Ren, C. Bian, M. Li, and C. Qian. A first running time analysis of the strength pareto evolutionary algorithm 2 (SPEA2). In *Parallel Problem Solving from Nature*, pages 295–312, 2024.
- [7] C. Bian, Y. Zhou, M. Li, and C. Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence*, pages 5513–5521, 2023.
- [8] W. Zheng, M. Li, R. Deng, and B. Doerr. How to use the metropolis algorithm for multi-objective optimization? In *AAAI Conference on Artificial Intelligence*, volume 38, pages 20883–20891, 2024.
- [9] R. Deng, W. Zheng, M. Li, J. Liu, and B. Doerr. Runtime analysis for state-of-the-art multi-objective evolutionary algorithms on the subset selection problem. In *Parallel Problem Solving from Nature*, pages 264–279, 2024.
- [10] Z. Huang, Y. Zhou, Z. Chen, and X. He. Running time analysis of MOEA/D with crossover on discrete optimization problem. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 2296–2303, 2019.
- [11] Z. Huang, Y. Zhou, C. Luo, and Q. Lin. A runtime analysis of typical decomposition approaches in MOEA/D framework for many-objective optimization problems. In *International Joint Conference on Artificial Intelligence*, pages 1682–1688, 2021.
- [12] B. Doerr, M.S. Krejca, and N. Weeks. Proven runtime guarantees for how the MOEA/D: computes the Pareto front from the subproblem solutions. In *Parallel Problem Solving from Nature*, pages 197–212, 2024.
- [13] S. Wiethager and B. Doerr. A mathematical runtime analysis of the non-dominated sorting genetic algorithm III (NSGA-III). In *Genetic and Evolutionary Computation Conference Companion*, pages 63–64, 2024.
- [14] M. Laumanns, L. Thiele, and E. Zitzler. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing*, 3(1):37–51, 2004.
- [15] O. Giel and P.K. Lehre. On the effect of populations in evolutionary multi-objective optimization. In *Genetic and evolutionary computation*, pages 651–658, 2006.
- [16] S. Wiethager and B. Doerr. Near-tight runtime guarantees for many-objective evolutionary algorithms. In *Parallel Problem Solving from Nature*, pages 153–168, 2024.
- [17] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Congress on Evolutionary Computation*, volume 3, pages 1918–1925, 2003.
- [18] C. Bian, C. Qian, and K. Tang. A general approach to running time analysis of multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence*, pages 1405–1411, 2018.
- [19] D.C. Dang, A. Opris, and D. Sudholt. Crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. *Artificial Intelligence*, 330:104098, 2024.
- [20] W. Zheng and B. Doerr. Runtime analysis for the NSGA-II: Proving, quantifying, and explaining the inefficiency for many objectives. *Evolutionary Computation*, pages 1–1, 2024.
- [21] P.K. Lehre and X. Qin. Self-adaptation via multi-objectivisation: a theoretical study. In *Genetic and Evolutionary Computation Conference*, pages 1417–1425, 2022.
- [22] C. Bian, S. Ren, M. Li, and C. Qian. An archive can bring provable speed-ups in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence*, pages 6905–6913, 2024.
- [23] P.K. Lehre. Negative drift in populations. In *Parallel Problem Solving from Nature*, pages 244–253, 2010.
- [24] C. Bian, Y. Zhou, M. Li, and C. Qian. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. *Artificial Intelligence*, page 104308, 2025.
- [25] S. Ren, Z. Liang, M. Li, and C. Qian. Stochastic population update provably needs an archive in evolutionary multi-objective optimization. *arXiv preprint arXiv:2501.16735*, 2025.
- [26] M. Laumanns, L. Thiele, E. Zitzler, E. Mo Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving from Nature*, pages 44–53, 2002.
- [27] B. Doerr and W. Zheng. Theoretical analyses of multi-objective evolutionary algorithms on multi-modal objectives. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 12293–12301, 2021.
- [28] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation*, 11(6):712–731, 2007.
- [29] M. Li, S. Yang, and X. Liu. Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization. *Evolutionary Computation*, 20(5):645–665, 2016.
- [30] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima. Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *Evolutionary Computation*, 21(2):169–190, 2017.
- [31] M. Li, X. Han, X. Chu, and Z. Liang. Empirical comparison between MOEAs and local search on multi-objective combinatorial optimisation problems. In *Genetic and Evolutionary Computation Conference Companion*, pages 547–556, 2024.
- [32] Raphaël Cosson, Roberto Santana, Bilel Derbel, and Arnaud Liefoghe. On bi-objective combinatorial optimization with heterogeneous objectives. *European Journal of Operational Research*, 319(1):89–101, 2024.
- [33] H. Mühlenbein. How genetic algorithms really work: Mutation and hillclimbing. In *Parallel Problem Solving from Nature*, pages 15–26, 1992.
- [34] G. Rudolph. *Convergence Properties of Evolutionary Algorithms*. PhD thesis, Universität Dortmund, 1997.
- [35] T. Jansen and I. Wegener. On the analysis of evolutionary algorithms – A proof that crossover really can help. In *European Symposium on Algorithms*, pages 184–193, 1999.
- [36] M. Mitchell. The royal road for genetic algorithms: Fitness landscapes and GA performance. *Conference on Artificial Life*, 1992.
- [37] Duc-Cuong Dang, Andre Opris, and Dirk Sudholt. Illustrating the Efficiency of Popular Evolutionary Multi-Objective Algorithms Using Runtime Analysis. In *Genetic and Evolutionary Computation Conference*, pages 484–492, 2024.
- [38] C. Qian, Y. Yu, and Z. Zhou. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence*, 204:99–119, 2013.
- [39] C. Qian, K. Tang, and Z. Zhou. Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization. In *Parallel Problem Solving from Nature*, pages 835–846, 2016.
- [40] T. Friedrich, C. Horoba, and F. Neumann. Illustration of fairness in evolutionary multi-objective optimization. *Theoretical Computer Science*, 412(17):1546–1556, 2011.
- [41] Y. Li, Y. Zhou, Z. Zhan, and J. Zhang. A Primary theoretical study on decomposition-based multiobjective evolutionary algorithms. *Evolutionary Computation*, 20(4):563–576, 2016.
- [42] P.S. Oliveto and X. Yao. Runtime analysis of evolutionary algorithms for discrete optimization. In *Theory of Randomized Search Heuristics*, pages 21–52, 2011.
- [43] T. Jansen and I. Wegener. Real royal road functions—where crossover provably is essential. *Discrete Applied Mathematics*, 149(1):111–125, 2005.
- [44] A. Opris. A Many Objective Problem Where Crossover is Provably Indispensable, 2024. arXiv preprint arXiv:2412.18375.

- [45] D. A. Van Veldhuizen. *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations*. PhD thesis, Air Force Institute of Technology, Ohio, 1999.
- [46] D. Antipov, A. Neumann, and F. Neumann. Rigorous runtime analysis of diversity optimization with gSEMO on oneMinMax. In *Foundations of Genetic Algorithms*, pages 3–14, 2023.
- [47] A. Neumann, W. Gao, C. Doerr, F. Neumann, and M. Wagner. Discrepancy-based evolutionary diversity optimization. In *Genetic and Evolutionary Computation Conference*, pages 991–998, 2018.
- [48] N. Jozefowicz, F. Semet, and E. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309, 2008.
- [49] H. Ishibuchi, K. Doi, and Y. Nojima. On the effect of normalization in MOEA/D for multi-objective and many-objective optimization. *Complex & Intelligent Systems*, 3(4):279–294, 2017.
- [50] L. He, Y. Nan, H. Ishibuchi, and D. Srinivasan. Effects of objective space normalization in multi-objective evolutionary algorithms on real-world problems. In *Genetic and Evolutionary Computation Conference Companion*, pages 670–678, 2023.
- [51] M. Ehrgott and Kathrin Klamroth. Connectedness of efficient solutions in multiple criteria combinatorial optimization. *European Journal of Operational Research*, 97(1):159–166, 1997.
- [52] Jochen Gorski, Kathrin Klamroth, and Stefan Ruzika. Connectedness of efficient solutions in multiple objective combinatorial optimization. *Journal of Optimization Theory and Applications*, 150(3):475–497, 2011.
- [53] G. Ochoa, A. Liefoghe, and S. Verel. Funnels in multi-objective fitness landscapes. In *Parallel Problem Solving from Nature*, pages 343–359, 2024.
- [54] O.M. Shir, M. Preuss, B. Naujoks, and M. Emmerich. Enhancing decision space diversity in evolutionary multiobjective algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 95–109, 2009.
- [55] O. Cuate and O. Schütze. Variation rate to maintain diversity in decision space within multi-objective evolutionary algorithms. *Mathematical and Computational Applications*, 24(3):82, 2019.
- [56] S. Ren, Z. Qiu, C. Bian, M. Li, and C. Qian. Maintaining diversity provably helps in evolutionary multimodal optimization. In *International Joint Conference on Artificial Intelligence*, pages 7012–7020, 2024.
- [57] L. Paquete, T. Schiavonotto, and T. Stützle. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research*, 156(1):83–97, 2007.
- [58] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145, 2005.
- [59] M. Li, S. Yang, X. Liu, and R. Shen. A comparative study on evolutionary algorithms for many-objective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 261–275, 2013.
- [60] S. Droste, T. Jansen, and I. Wegener. A rigorous complexity analysis of the  $(1 + 1)$  evolutionary algorithm for separable functions with Boolean inputs. *Evolutionary Computation*, 6(2):185–196, 1998.
- [61] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar. Using well-understood single-objective functions in multiobjective black-box optimization test suites. *Evolutionary Computation*, 30(2):165, 2022.
- [62] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [63] A.Q. Nguyen, A. M. Sutton, and F. Neumann. Population size matters: Rigorous runtime results for maximizing the hypervolume indicator. *Theoretical Computer Science*, 561:24–36, 2015.
- [64] B. Doerr, W. Gao, and F. Neumann. Runtime analysis of evolutionary diversity maximization for OneMinMax. In *Genetic and Evolutionary Computation Conference*, pages 557–564, 2016.
- [65] W. Zheng and B. Doerr. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Genetic and Evolutionary Computation Conference*, pages 611–619, 2022.
- [66] W. Zheng and B. Doerr. Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II). *Artificial Intelligence*, 325:104016, 2023.
- [67] B. Doerr and Z. Qu. From understanding the population dynamics of the NSGA-II to the first proven lower bounds. In *AAAI Conference on Artificial Intelligence*, volume 37, pages 12408–12416, 2023.
- [68] W. Zheng and B. Doerr. Runtime Analysis of the SMS-EMOA For Many-Objective Optimization. In *AAAI Conference on Artificial Intelligence*, 2024.
- [69] D. Brockhoff, T. Friedrich, and F. Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature*, pages 651–660, 2008.
- [70] D.C. Dang, A. Opris, B. Salehi, and D. Sudholt. Analysing the robustness of NSGA-II under noise. *Genetic and Evolutionary Computation Conference Companion*, pages 642–651, 2023.
- [71] C. Doerr and M.S. Krejca. Run time analysis for random local search on generalized majority functions. *Trans. Evol. Comp.*, 27(5):1385–1397, 2023.
- [72] M. Ehrgott. *Multicriteria optimization*. 2006.
- [73] P. Hansen. Bicriterion path problems. In *Multiple Criteria Decision Making Theory and Application*, pages 109–127, 1980.
- [74] H.W. Hamacher and G. Ruhe. On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52(4):209–230, 1994.
- [75] G. Ruhe. Complexity results for multicriterial and parametric network flows using a pathological graph of Zadeh. *Zeitschrift für Operations Research*, 32(1), 1988.
- [76] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: Optimal  $\mu$ -distributions and the choice of the reference point. In *Foundations of Genetic Algorithms*, pages 87–102, 2009.
- [77] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
- [78] H. Ishibuchi, Y. Nan, and L.M. Pang. Performance evaluation of multi-objective evolutionary algorithms using artificial and real-world problems. In *Evolutionary Multi-Criterion Optimization*, pages 333–347, 2023.
- [79] D.E. Goldberg. Genetic algorithms in search, optimization and machine learning, 1989.
- [80] Z. Liang, Z. Cui, and M. Li. Pareto landscape: Visualising the landscape of multi-objective optimisation problems. In *Parallel Problem Solving from Nature*, pages 299–315, 2024.
- [81] E.V. Nimwegen, J.P. Crutchfield, and M. Mitchell. Statistical Dynamics of the Royal Road Genetic Algorithm. *Theoretical Computer Science*, 229(1):41–102, 1999.
- [82] S. Verel, G. Ochoa, and M. Tomassini. Local optima networks of NK landscapes with neutrality. *Evolutionary Computation*, 15(6):783–797, 2011.
- [83] B. Doerr and Z. Qu. Runtime analysis for the NSGA-II: Provable speed-ups from crossover. In *AAAI Conference on Artificial Intelligence*, volume 37, pages 12399–12407, 2023.
- [84] Zhenkun Wang, Yew-Soon Ong, Jianyong Sun, Abhishek Gupta, and Q. Zhang. A generator for multiobjective test problems with difficult-to-approximate pareto front boundaries. *Evolutionary Computation*, 23(4):556–571, 2018.
- [85] X. Chu, X. Han, Maorui Zhang, and M. Li. Improving decomposition-based moeas for combinatorial optimisation by intensifying corner weights. *Swarm and Evolutionary Computation*, 91:101722, 2024.
- [86] F. Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181(3):1620–1629, 2007.
- [87] A.V. Do, A. Neumann, F. Neumann, and A. Sutton. Rigorous runtime analysis of MOEA/D for solving multi-objective minimum weight base problems. *Advances in Neural Information Processing Systems*, 36:36434–36448, 2023.
- [88] F. Neumann and C. Witt. Runtime analysis of single- and multi-objective evolutionary algorithms for chance constrained optimization problems with normally distributed random variables, 2024. arXiv preprint arXiv:2109.05799.
- [89] S. Kratsch and F. Neumann. Fixed-parameter evolutionary algorithms and the vertex cover problem. *Algorithmica*, 65(4):754–771, 2013.
- [90] G. Rudolph. Runtime analysis of  $(1+1)$ -EA on a biobjective test function in unbounded integer search space. In *Symposium Series on Computational Intelligence*, pages 1380–1385, 2023.
- [91] B. Doerr, M.S. Krejca, and G. Rudolph. Runtime analysis for multi-objective evolutionary algorithms in unbounded integer spaces, 2024. arXiv preprint arXiv:2412.11684.
- [92] R. Allmendinger, A. Jaskiewicz, A. Liefoghe, and C. Tammer. What if we increase the number of objectives? theoretical and empirical implications for many-objective combinatorial optimization. *Computers & Operations Research*, 145:105857, 2022.
- [93] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 742–756, 2007.
- [94] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Behavior of evolutionary many-objective optimization. In *Int. Conf. Computer Modeling and Simulation*, 2008.
- [95] M. Li, C. Grosan, S. Yang, X. Liu, and X. Yao. Multi-line distance minimization: A visualized many-objective test problem suite. *Trans. Evol. Comp.*, 22(1):61–78, 2018.
- [96] K. Li, R. Wang, T. Zhang, and H. Ishibuchi. Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access*, 6:26194–26214, 2018.
- [97] W. Zheng, Y. Gao, and B. Doerr. A crowding distance that provably solves the difficulties of the NSGA-II in many-objective optimization, 2024.
- [98] D.C. Dang, A. Opris, and D. Sudholt. Level-based theorems for runtime analysis of multi-objective evolutionary algorithms. In *Parallel Problem Solving from Nature*, pages 246–263, 2024.
- [99] A. Opris, D.C. Dang, F. Neumann, and D. Sudholt. Runtime analyses of NSGA-III on many-objective problems. In *Genetic and Evolutionary Computation Conference*, pages 1596–1604, 2024.
- [100] J. Sakel, J. Fieldsend, and E. Keedwell. Genotype diversity measures for escaping plateau regions in university course timetabling. In *Genetic and Evolutionary Computation Conference Companion*, pages 2090–2098, 2023.
- [101] H. E. Aguirre and K. Tanaka. Insights on properties of multiobjective MNK-landscapes. In *Congress on Evolutionary Computation*, volume 1, pages 196–203, 2004.
- [102] A. Liefoghe, G. Ochoa, S. Verel, and B. Derbel. Pareto local optimal solutions networks with compression, enhanced visualization and expressiveness. In *Genetic and Evolutionary Computation Conference Companion*, pages 713–721, 2023.
- [103] M. Li, X. Han, and X. Chu. MOEAs are stuck in a different area at a time. In *Genetic and Evolutionary Computation Conference*, pages 303–311, 2023.