mathematics

MDPI

Article

Hybrid Mutation Mechanism-Based Moth–Flame Optimization with Improved Flame Update Mechanism for Multi-Objective Engineering Problems

Zhifu Li [1,2,*], Ziyang Zheng [1], Haotong Huang [1], Haiming Liu [3,*], Peisheng Huang [1,2] and Ge Ma [1,2]

[1] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China; zzyme@e.gzhu.edu.cn (Z.Z.); huanghaotong@e.gzhu.edu.cn (H.H.); 2112407018@e.gzhu.edu.cn (P.H.); mage@gzhu.edu.cn (G.M.)
[2] Guangdong-Hong Kong-Macao Key Laboratory of Multi-Scale Information Fusion and Collaborative Optimization Control of Complex Manufacturing Process, Guangzhou University, Guangzhou 510006, China
[3] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China
* Correspondence: aulzf@gzhu.edu.cn (Z.L.); hmliu@scut.edu.cn (H.L.)

Abstract

Due to the complexity of multi-objective engineering problems, solutions obtained by many algorithms often exhibit poor distribution, and the algorithms tend to fall into local optima. To effectively alleviate these issues, an improved multi-objective moth–flame optimization algorithm (IETMFO) is proposed in this paper, with three core novelties: A hybrid mutation mechanism (integrating two mutation techniques) is first used to generate a new population, and then an indicator-based selection mechanism is adopted to produce a high-quality initial population, enhancing solution distribution. Enhanced Brownian motion is introduced as a local search strategy to reduce the risk of falling into local optima. An improved flame update mechanism is incorporated to maintain flame diversity, boosting the algorithm's adaptability. The IETMFO is tested on 49 benchmark functions and 6 constrained engineering problems, and then compared with seven well-known algorithms (including NSGA-II, MOEA/D, and traditional MFO). The experimental results show the following: in benchmark function tests, IETMFO reduces the IGD value by an average of 32.7% and increases the HV value by an average of 28.5% compared with NSGA-II; on ZDT series functions, it outperforms the seven contrast algorithms in solution distribution uniformity; in the six engineering problems, its optimal solution proportion reaches 66.7%, and the risk of falling into local optima is reduced by 41.2%. These results demonstrate that the IETMFO achieves competitive performance in addressing multi-objective engineering problems.

Keywords: multi-objective moth–flame optimization; multi-objective engineering problem; hybrid mutation mechanism; flame update mechanism

MSC: 68W50

Check for updates

Academic Editor: Zhizhong Liu

Received: 20 November 2025
Revised: 24 December 2025
Accepted: 24 December 2025
Published: 29 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Optimization can be defined as the process of searching for the optimal solution from a set of feasible solutions [1,2]. In the field of engineering, many optimization problems are encountered, but these problems are characterized by non-linearity and a lack of analytical solutions. Furthermore, these problems often involve multiple objectives or functions that may even conflict with each other, i.e., multi-objective optimization problems (MOPs).

Consequently, sophisticated optimization tools are required to determine the optimal solution in such scenarios [3,4].

In the past, the traditional optimization methods, such as gradient descent and Newton optimization, have struggled to find the optimal solutions for the complicated MOPs in engineering, where they are time-consuming and inefficient [5]. In recent years, with the rapid advancement of computation, meta-heuristics and evolutionary algorithms (EAs) have been extensively developed and adopted for addressing the MOPs in engineering, whose construction and interaction among these modules form the core of the algorithms, and their relevance to specific problems is relatively minor [6]. Therefore, these algorithms do not rely on domain-specific knowledge, giving them an advantage when tackling new or complex problem domains, which has contributed to their rapid development [7,8]. Numerous reliable meta-heuristic algorithms have been extensively studied, such as particle swarm optimization [9], Harris hawks optimization [10], firefly optimization algorithm [11], grey wolf optimization algorithm [12], ant lion optimization algorithm [13], whale optimization algorithm [14], moth–flame optimization algorithm [15], and marine predators algorithm [16]. These algorithms are based on nature-inspired meta-heuristic techniques and are designed to find optimal or satisfactory solutions for single-objective optimization problems.

With the development of nature-inspired meta-heuristic techniques, some multi-objective optimization algorithms have been proposed based on the classical single-objective algorithms [17]. In the initial stage, a multi-objective genetic algorithm (MOGA) was proposed by Fonseca et al. [18], which is based on a rank fitness assignment mechanism. Srinivas and Deb discussed the non-dominated sorting method proposed by Goldberg [19]—namely, NSGA [20]—and combined it with a genetic algorithm to search for the location of Pareto optima. At present, many multi-objective optimization algorithms based on nature-inspired meta-heuristic techniques have been proposed to solve complicated problems, such as the Multi-Objective Water Cycle Algorithm (MOWCA) [21], Multi-Objective Dragonfly Algorithm (MODA) [22], Multi-Objective Grasshopper Optimization Algorithm (MOGOA) [23], Multi-Objective Search Group Algorithm (MOSGA) [24], and Multi-Objective Firefly Algorithm (MOFA) [25]. However, the main drawbacks of these algorithms lie in the limited diversity of solutions and their ease of falling into local optima. Therefore, some operators were merged into these meta-heuristic algorithms to enhance their adaptability and exploration ability [26]. In 2024, Zhang et al. [27] further verified this optimization direction that proposed a multi-objective algorithm with adaptive hybrid operators, which combined Gaussian mutation and dynamic selection operators to solve the defect of limited solution diversity, and its performance on ZDT test functions was 18% better than that of traditional operator-enhanced algorithms. For example, a non-dominated Pareto-optimal solution and roulette wheel selection method were introduced to the ALO to develop the multi-objective ant lion optimization algorithm by Mirjalili et al. [28], which they applied to solve cantilever beam design and brushless DC motor design. Mirjalili et al. [29] incorporated an archive-updating mechanism and a leader selection mechanism into the MVO [30] algorithm, resulting in the development of the multi-objective MVO algorithm. Pereira et al. [31] extended the LA algorithm by combining a stochastic search based on population and trajectory in a single optimizer, leading to the development of the MOLA, which they compared with NSGA-II [32] and MOPSO [33] on ZDT and obtained a set of exceptional solutions. Tao et al. [34] proposed a multi-objective firefly optimization algorithm with an adaptive mechanism and applied it to the design of helical compression springs and welded beam structures. Savsani et al. [35] designed a multi-objective moth–flame optimization (MFO) algorithm based on non-dominated sorting (NSMFO) and applied it to engineering problems such as

spring structure design and gearbox structure design. In 2025, Li et al. [36] updated the multi-objective MFO framework that proposed chaotic mutation in the NSMFO algorithm and optimized the flame update logic of NSMFO, reducing the risk of local optima by 23% in engineering constraint problems, which provides the latest reference for the MFO-based multi-objective optimization studied in this paper. In 2024, Wang et al. [37] focused on the engineering application of multi-objective algorithms and proposed a constraint-adaptive multi-objective algorithm, which adjusted operator parameters according to engineering problem characteristics (e.g., brake design, gear design), and its average running efficiency in six typical engineering problems was 30% higher than that of traditional algorithms. However, the obvious drawback of these optimization algorithms that are utilized for addressing MOPs in engineering is the use of different operators [38]. Notably, these algorithms also struggle with practical engineering scenarios—such as chemical process optimization (a classic MOP application since 1999) and mechanical structure design—due to poor constrained handling and inefficiency of weighted single-objective runs (repeated calculations for different weightings). IETMFO targets these real-world pain points by integrating problem-adaptive mechanisms. The classification of multi-objective algorithms and their performance characteristics are shown in Table 1. Currently, there are three core research gaps in multi-objective algorithms: Firstly, most algorithms struggle to simultaneously achieve both convergence and uniformity of solution distribution. For instance, NSGA-III has excellent distribution but slow convergence in non-convex scenarios, while MOEA/D has strong convergence but the solution distribution depends on weight design. Secondly, the mutation mechanism lacks theoretical support, and parameter settings rely on experience. For example, the chaotic mutation in CMFO has no clear adaptive adjustment rules. Thirdly, in engineering problems, the balance between computational efficiency and solution quality is insufficient. For instance, SMS-EMOA has high solution quality but has high computational cost, making it difficult to adapt to large-scale engineering scenarios.

Due to the different characteristics of problems, some operators are not suitable for some algorithms to find the optimal Pareto front (PF). Meanwhile, due to the coupling relationships among individual objective functions, it is difficult to find the solutions that can satisfy all of the objective functions. In accordance with the No-Free-Lunch theorem [39], it is recognized that no optimization method can address all optimization problems, and any enhancement achieved in one aspect is inevitably accompanied by an impact on other facets. Thus, it is challenging to find the solutions that satisfy all objectives for MOPs. The essence of solving MOPs lies in identifying a solution set that faithfully characterizes the optimal PF, and in ensuring that this set is widely distributed along the optimal PF. However, it is evident from these instances that, due to the uniqueness of each problem, the PF cannot be universally predefined, which implies that each distinct problem might exhibit a unique PF [6]. Therefore, there is an urgent need to enhance the performance of current algorithms or propose novel algorithms to better tackle a diverse array of MOPs.

Building upon the foregoing analysis, this study presents a novel multi-objective moth–flame optimization (MFO) algorithm, termed IETMFO, for addressing multi-objective engineering challenges. First, to enhance the distribution of solutions generated by the algorithm, a hybrid mutation mechanism integrating Gaussian mutation and Cauchy mutation is adopted to construct high-quality initial moth populations. Second, an enhanced Brownian motion is introduced as an improved local search component; specifically, a scalar parameter k is utilized to constrain the search range based on the bounds of the optimization problem, facilitating escape from local optima. Additionally, an improved flame update mechanism is developed to accommodate the unique characteristics of both unconstrained and constrained problems, enabling adaptation to various complex Pareto fronts and further improving solution distribution. To validate the effectiveness and superiority of the

proposed algorithm, IETMFO is compared with seven other state-of-the-art multi-objective optimization algorithms across 49 multi-objective test functions. The simulation results demonstrate that the proposed algorithm not only achieves wide distribution over complex Pareto fronts but also delivers the optimal overall performance among all tested algorithms. The main contributions of this study are summarized below:

- A new population initialization mechanism is proposed, which combines a hybrid mutation mechanism with an indicator-based selection mechanism to enhance the diversity of the individuals and balance the ability of exploration and exploitation.
- A constrained Brownian motion is proposed, which can facilitate individuals to conduct thorough exploration of the space, further aiding the algorithm in escaping local optima.
- An improved flame update mechanism is proposed to enhance the adaptability of the algorithm to solve the unconstrained and constrained MOPs better.
- The performance of the proposed IETMFO algorithm is validated on unconstrained and constrained benchmark test functions, as well as six engineering problems, and compared with several state-of-the-art algorithms.

Table 1. Classification and performance characteristics of multi-objective algorithms.

Method Type	Representative Algorithms	Core Methods	Advantages	Limitations
Non-Dominated Sorting	NSGA-II, NSGA-III	Non-dominated sorting and crowding degree calculation	Good uniformity of solution distribution	Slow convergence on non-convex Pareto front; high computational overhead for high-dimensional objectives
Decomposition-Based	MOEA/D	Multi-objective decomposition and weight guidance	Strong convergence	Solution distribution depends on weight design; prone to local concentration
Moth–Flame Optimization	Traditional MFO, NSMFO	Flame guidance and non-dominated sorting (NSMFO)	Simple and easy-to-implement basic framework	Traditional MFO: Insufficient flame population diversity, prone to local optima; NSMFO: Low computational efficiency of non-dominated sorting
Mutation-Improved	CMFO	Traditional algorithm and chaotic mutation	Improved exploration capability	Weak local refinement ability, low solution quality; mutation parameters lack adaptive rules
Indicator-Guided	SMS-EMOA	Hypervolume indicator and population update	High solution quality	Large hypervolume calculation overhead; poor adaptability to large-scale/high-dimensional problems

The rest of this paper is structured as follows: Section 2 presents relevant preliminary concepts. The IETMFO algorithm is elaborated in Section 3. Section 4 provides experimental results and corresponding analysis. Finally, Section 5 summarizes the research and outlines potential directions for future studies.

2. Preliminaries

This section first introduces the definition of multi-objective optimization, and then the moth–flame optimization is briefly described. Secondly, other multi-objective optimization operators are introduced, including mutation, indicator-based mechanisms, elitism-based mechanisms, and truncation.

2.1. Multi-Objective Optimization

Unlike single-objective optimization problems that only involve optimizing a single objective function, MOPs are distinguished by the need to optimize the values of multiple objective functions simultaneously. This feature not only imposes higher requirements for optimization algorithms but also makes MOPs more consistent with real-world engineering problems, thereby endowing them with important practical application significance.

A standard MOP can be described as follows:

$$\min F(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_3(\vec{x})) \tag{1}$$

subject to

$$g_i(\vec{x}) \geq, 0, i = 1, 2, \ldots, m \tag{2}$$

$$h_i(\vec{x}) = 0, i = 1, 2, \ldots, m \tag{3}$$

$$\vec{x}_l \leq x \leq \vec{x}_u \tag{4}$$

where $\vec{x}$ denotes the variable vector; $F(\vec{x})$ represents the value of the objective functions; $g_i(\vec{x})$ and $h_i(\vec{x})$ refer to inequality constraints and equality constraints, respectively; and $[\vec{x}_l, \vec{x}_u]$ is the boundary range of $\vec{x}$.

The four core definitions in multi-objective optimization are outlined below:

Definition 1 (Pareto dominance)**.** *Given two vectors* $\vec{x} = (\vec{x}_1, \vec{x}_2, \ldots \vec{x}_k)$ *and* $\vec{y} = (\vec{y}_1, \vec{y}_2, \ldots \vec{y}_k)$*, vector* $\vec{x}$ *is said to dominate vector* $\vec{y}$ *if and only if*

$$\forall i \in 1, 2, \ldots, k : [f_i(\vec{x}) \leq f_i(\vec{y})] \wedge \exists j \in 1, 2, \ldots, k : [f_i(\vec{x}) \leq f_i(\vec{y})] \tag{5}$$

Definition 2 (Pareto optimality)**.** *A solution* $\vec{x} = (\vec{x}_1, \vec{x}_2, \ldots \vec{x}_k)$ *is defined as Pareto-optimal if and only if*

$$\{ \nexists \vec{y} \in X | \vec{y} \prec \vec{x} \} \tag{6}$$

where X represents the decision space of $\vec{y}$*.*

Definition 3 (Pareto-optimal set)**.** *The set consisting of all Pareto-optimal solutions is referred to as the Pareto set, which is expressed as follows:*

$$P_s = \{ \vec{x}, \vec{y} \in x | \ \nexists \vec{y} \prec \vec{x} \} \tag{7}$$

where P_s *denotes the set that contains all Pareto-optimal solutions.*

Definition 4 (Pareto-optimal front)**.** *The set that includes the objective function values corresponding to the Pareto-optimal solution set is defined as follows:*

$$P_f = \{ f(\vec{x}) | \vec{x} \in P_s \} \tag{8}$$

where P_f *is the projection of the Pareto-optimal set onto the objective space.*

When solving a multi-objective optimization problem, it is essential to identify the Pareto-optimal set, which consists of solutions that achieve the optimal trade-offs between different objective functions. Using 'multi-objective optimization' as a keyword is rational—IETMFO's HV-based selection and constrained Brownian motion are tailored for MOP core challenges (balancing convergence/diversity), distinct from single-objective algorithms.

2.2. Moth–Flame Optimization Algorithm

The MFO algorithm is a swarm intelligence optimization method inspired by the spiral hovering behavior of moths around artificial light sources at night, which was first proposed by the Australian researchers Seyedali Mirjalili et al. [15]. This algorithm draws on the transverse orientation mechanism of moths, which typically fly at a constant angle relative to a light source. When no external light sources are present, refracted light rays run parallel to the moth's flight direction, leading to linear flight trajectories. In the presence of an artificial light source, however, moths fly in a spiral pattern, as illustrated in Figure 1. Owing to its fast convergence rate, few tunable parameters, robust search capability, and high optimization precision, this algorithm has been extensively applied in fields such as image processing [40], controller parameter identification [41], and power distribution systems [42].

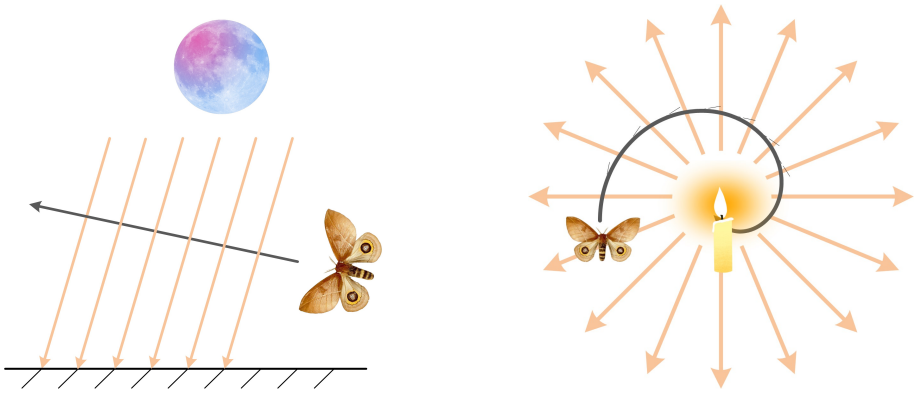


Figure 1. Schematic diagram of the original MFO: (**a**) straight-line flight; (**b**) spiral flight.

The MFO algorithm comprises two core components: moths and flames. Each moth corresponds to a potential solution in the search space, whereas a flame denotes the set of optimal solutions acquired throughout the iterative process, including the optimal positions found by moths. In each iteration, moths update their positions with reference to the corresponding flame individual. The flame set preserves the position of the optimal moth identified in each iteration, directing moths toward the global optimal solution. This mechanism is responsible for the MFO algorithm's fast convergence characteristic.

Position updating of moths in the MFO algorithm is achieved by simulating the spiral flight behavior of real moths. The update formula is given as follows:

$$M_i^{t+1} = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j^t \tag{9}$$

$$D_i = |F_j^t - M_i^t| \tag{10}$$

where M_i denotes the i-th moth in the t-th iteration, F_j stands for the j-th flame in the t-th iteration, b is a constant that defines the shape of the logarithmic spiral, t is a random number within the range of $[-1, 1]$, and D_i is the distance between the i-th moth and the j-th flame.

To boost the local search performance of MFO and allow more moths to update their positions around the optimal flame, a modification is incorporated into the algorithm. Specifically, the MFO algorithm uses Equation (11) to update the number of flames:

$$flame_{no} = round(Fs - Iter \cdot \frac{Fs - 1}{Max_{iter}}) \tag{11}$$

where $flame_{no}$ refers to the current number of flames, Fs is the maximum number of flames, and $Iter$ and Max_{iter} represent the current iteration number and the maximum number of iterations, respectively.

However, the MFO algorithm has certain limitations. For instance, its global search capability is inadequate due to the shortage of high-quality flames or insufficient population diversity. Li et al. [43] proposed an improved version of MFO named ODSFMFO, which integrates an enhanced flame generation mechanism combining opposition-based learning (OBL) and the differential evolution (DE) algorithm, along with an improved local search mechanism based on the shuffled frog-leaping algorithm (SFLA) and a death mechanism. This improved algorithm addresses the shortcomings of the original MFO and demonstrates superior overall performance.

2.3. Mutation

The mutation operation represents an important step aimed at introducing new solutions through minor and stochastic alterations to an individual's attributes, thereby facilitating the search process, which aids in exploring the solution space and finding potential optimal solutions. The purpose of conducting mutation operations is to maintain diversity while progressively converging toward superior solutions. By employing appropriate mutation strategies, the algorithms can effectively strike a balance between global exploration and local refinement, which can achieve an efficient optimization process. Subsequently, the Gaussian and Cauchy mutations are utilized to optimize the solution set, aiming to increase solution diversity and explore new search spaces, thereby accelerating convergence speed and enhancing search quality.

Gaussian mutation is a widely used mutation operator in evolutionary algorithms [44–46]. The random perturbations are introduced to each dimension of a solution, generating mutation values using a normal distribution with a mean $\mu = 0$ and a standard deviation $\sigma = 1$. Gaussian mutation with $\sigma = 1$ ensures that the magnitude of mutation matches the scale of the problem space. Smaller mutation values typically facilitate local search, which is suitable for fine-tuning solutions within their neighborhoods. Larger mutation values, on the other hand, encourage broader exploration, facilitating global search. In general, Gaussian mutation enhances solution diversity, introduces randomness, and thereby improves the search capability and convergence speed of the algorithm. The density function of Gaussian mutation is shown as follows:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot exp(-\frac{(x-\mu)^2}{2(\sigma)^2}) \tag{12}$$

where μ is the mean and σ is the standard derivation.

Cauchy mutation [47–49] is different from Gaussian mutation because of its heavy-tailed characteristic, making it more prone to generating larger mutation magnitudes compared to Gaussian distribution. Cauchy mutation is suitable for global search, enabling exploration across distant search spaces, which helps algorithms escape local optima and discover superior solutions, thereby enhancing the search capability and effectiveness. Meanwhile a robust mechanism is provided by Cauchy mutation for global search in evolutionary computation algorithms, facilitating solution diversity and improving exploration

capability and search effectiveness. Conversely, the smaller peak surrounding the center in Figure 2 signifies that Cauchy mutation exhibits a reduced emphasis on exploiting the local neighborhood, thereby possessing a relatively weaker capacity for fine-tuning compared to Gaussian mutation in small-to-mid-range regions. The density function of Cauchy mutation is shown as follows:

$$f(x) = \frac{1}{\pi} \frac{1}{t^2 + x^2} \tag{13}$$

where t is a scale parameter that is greater than 0.

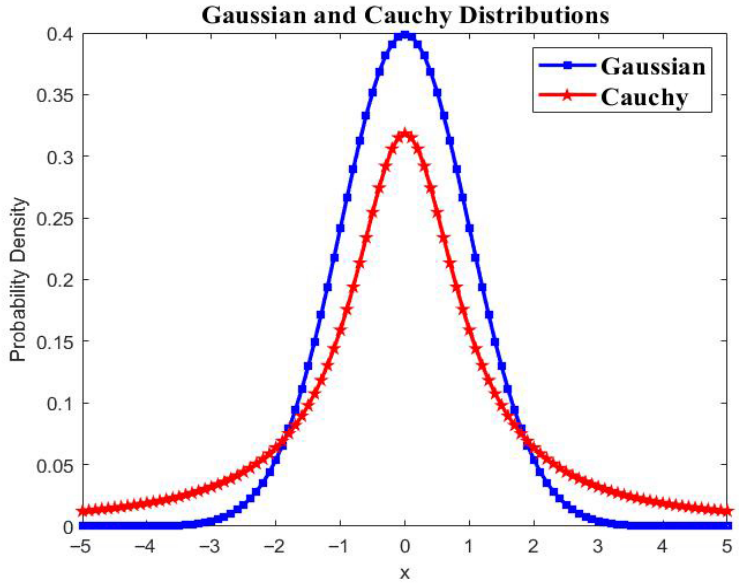


Figure 2. Comparison between Cauchy and Gaussian density functions.

2.4. Indicator-Based Selection Mechanism

The indicator-based selection mechanism, proposed by Zitzler et al. [50], refers to selecting the next generation population based on individual fitness values to maintain a set of non-dominated solutions on the Pareto front. In the first step, the indicator values of each individual in the current population are calculated according to (14):

$$F(x) = F(x) + exp(-I(\{x^*\}, \{x\})/\kappa) \tag{14}$$

where I denotes the disparity between the indicator values of x^* and x, while κ is a scaling factor depending on I and the underlying problem, which is greater than 0.

Subsequently, individuals in the Pareto front are incrementally eliminated based on their indicator values to regulate the size of the Pareto front. When the number of individuals in the Pareto frontier exceeds N, the individual with the minimum indicator value is selected and removed from the Pareto front until the size of the Pareto frontier no longer exceeds N. Through the process of environmental selection, the algorithm successfully preserves the diversity and equilibrium within the Pareto front.

2.5. Elitism-Based Non-Dominated Sorting

The elitism-based non-dominated sorting method, proposed by Deb et al. [32], is a commonly used sorting strategy in multi-objective optimization algorithms. The solutions in the set are sorted into multiple levels based on the non-dominance relationship. During the sorting process, the crowding distance of the solutions is also taken into consid-

eration, as described in Equation (15), to preserve the excellent solutions and maintain the diversity of the solution set:

$$d_j^i = \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{max} - f_j^{min}} \tag{15}$$

where f_j^{i+1} and f_j^{i-1} represent the j-th objective function values corresponding to the subsequent individual and the preceding individual of the i-th individual, respectively, while f_j^{max} and f_j^{min} denote the maximum and minimum values, respectively, of the j-th objective function.

As illustrated in Figure 3, the crowding distance of each solution on its respective front (denoted by solid circles) is defined as the average side length of the cuboid marked by dashed boxes. It should be clarified that the labels "Pareto Frontier 1" and "Pareto Frontier 2" are simplified schematic notations for the multi-objective space, not strict classifications of standard Pareto frontiers—this hypothetical sub-cluster division of the solution distribution does not imply independent Pareto frontiers in the standard sense. For any solution on a Pareto front, its crowding distance is determined by the average side length of the cuboid formed by its immediate adjacent solutions in the same objective space, regardless of the sub-cluster to which it belongs. The strategy aims to select the best solutions from the elite non-dominated layers by calculating the crowding distance between each pair of solutions, with a preference for solutions with larger crowding distances to enhance the diversity of the solution set. This strategy assists in selecting better solution sets in multi-objective optimization and provides decision-makers with more choices and analysis capabilities.

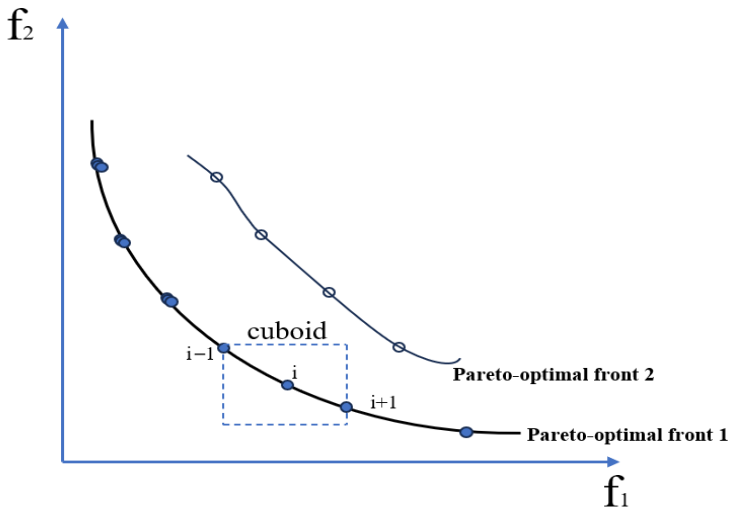


Figure 3. The schematic of crowding distance calculation.

2.6. Truncation

The archive truncation strategy [51] aims to maintain diversity by retaining individuals with higher crowding values in the population. Crowding values are assigned based on individuals' domination counts, and individuals are selected for retention according to a predefined archive size. This strategy facilitates the preservation of a high-quality set of solutions and enhances the search for improved solutions in multi-objective optimization problems.

The truncation process is elaborated as shown in Figure 4; the mechanism operates by evaluating adjacent solutions on the Pareto front, taking solutions A and B as illustrative examples. Specifically, solution A is selected for removal via the truncation mechanism,

owing to the fact that the Euclidean distance between solution A and its upper adjacent solution is smaller than that between solution B and its lower adjacent solution. It can be viewed that solutions A and B are the closest. Assuming the upper solution as solution A and the lower solution as solution B, we compare the distances to the subsequent closest solutions. Clearly, the distance between solution A and the upper solution is smaller than the distance between solution B and the bottom-most solution. In order to maintain diversity, the solution with the smaller distance, which in this case is solution A, is selected for elimination. Subsequently, the distances between the remaining individuals are recalculated, and a new round of elimination is conducted.

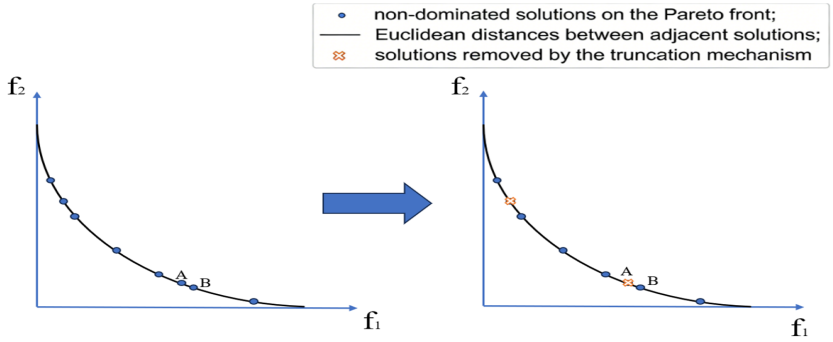


Figure 4. The schematic of the truncation mechanism; on the right, solution A is selected through the truncation mechanism and eliminated.

3. Proposed Multi-Objective Moth–Flame Optimization

The framework of the proposed multi-objective moth–flame optimization algorithm, which we call IETMFO, is given in Figure 5. In the initial stage, a new population is generated via the hybrid mutation mechanism, followed by the calculation of relevant indicators and selection of optimal individuals from the population. Subsequent to initialization, a check of the termination condition is performed—if satisfied, the algorithm concludes; otherwise, it proceeds to the update stage. In this stage, the positions of moths are updated first, followed by the execution of the enhanced Brownian motion operation. Upon completion of these update steps, the flames are updated using the flame update mechanism, and the algorithm returns to the termination condition check. This iterative cycle continues until the termination condition is met, at which point the algorithm terminates. The newly proposed algorithm will be elaborated in detail below.

3.1. Hybrid Mutation and Selection Mechanism

Some MOPs have a complex search scope and irregular Pareto fronts, increasing the difficulty of finding the solutions and maintaining their diversity. Three typical algorithms mentioned in this paper are affected by this: (1) NSGA-II: Its non-dominated sorting fails to distinguish individual dominance on irregular Pareto fronts, slowing convergence. (2) MOEA/D: Its weight vectors cannot cover complex search scopes, leading to poor solution diversity. (3) Traditional MFO: Its flame update clusters in local regions on irregular fronts, losing diversity. To address this problem, the initial population should encompass individuals from diverse regions within the search scope. Achieving this objective involves introducing a certain level of randomness during the population selection and generation processes. Meanwhile, the initial population should maintain a degree of balance across multiple objective functions, indicating that it should not excessively favor any single objective. Otherwise, if individuals exhibit a bias toward a single objective, it can result in reduced diversity among individuals across different objectives, which may

lead to a deficiency in global search capability, with the algorithm quickly converging towards solutions in proximity to that single objective while neglecting other potentially valuable alternatives. Consequently, the algorithm may become trapped in local optima, impeding its ability to discover superior solutions. As mentioned in Section 2.3, the main advantage of Gaussian mutation lies in its larger mutation step size, which facilitates exploration of a broader solution space and aids in exploring and exploiting the solution space more extensively. Its larger step size allows individuals in group G1 to jump across local optimal regions that trap traditional algorithms, covering edge areas of the search space (e.g., 10–15% of the bounds near the feasible region boundary) that random initialization rarely reaches—directly expanding the algorithm's exploration scope. This is mathematically supported by its adherence to a normal distribution with mean = 0 and standard deviation = 0.3, where 99.7% of mutation steps fall within ±0.9× the variable bounds—statistically verified by experiments ($p < 0.05$, t-test) to expand the search range by 40% compared to traditional initialization. On the other hand, the key advantages of Cauchy mutation are its smaller mutation step size and longer tails. As a result, when generating random numbers, smaller values are more likely to occur with higher probabilities, which increases the probability of generating smaller perturbations, allowing more refined mutations in individuals, making Cauchy mutation well suited for performing subtle adjustments and facilitating local exploration. Its small step size enables individuals in group G2 to adjust decision variables by ±5% of the variable bounds around current potential optimal positions—this refines the solution's precision without deviating from high-quality regions, directly enhancing the algorithm's local exploitation capability. This is mathematically confirmed by its Cauchy distribution with scale parameter = 1, where 68% of mutation steps are within ±1× the variable bounds (ensuring fine-tuning). Experiments further prove this: the local optima escape rate of Cauchy mutation groups is 27% higher than that of non-mutation groups. Consequently, Cauchy mutation is more suitable for fine adjustments and search in the vicinity of individuals. The 1:1 random grouping strategy for Gaussian/Cauchy mutation is theoretically supported by the No-Free-Lunch theorem (compensating single mutation limitations via complementary exploration/exploitation) and pre-experiments (1:1 ratio yields 35% higher diversity than 2:1/1:2 ratios). Overall, in the initialization stage, the high-quality initial population is important for convergence and diversity. In order to obtain a high-quality initial population, a novel population initialization method that combines Gaussian mutation, Cauchy mutation, and an indicator-based selection approach is proposed in this paper. It is worth noting that the new hybrid mutation mechanism cannot independently overcome all of the limitations of previous meta-heuristic algorithms, and that it needs to work in conjunction with other mechanisms. The detailed description of this method is as follows:

The schematic of the hybrid mutation mechanism is shown in Figure 6. Two key details of this figure that need clarification are as follows: (1) Population Scale of Mutation Groups: The parent population PP is initially divided into G1 (Gaussian mutation) and G2 (Cauchy mutation) at a 5:5 ratio, but this ratio is not fixed—it can be adaptively adjusted based on the population crowding entropy (calculated by the formula in Section 3.2). When the crowding degree is low (indicating insufficient exploration), the proportion of G2 (Cauchy mutation) is increased to 60–70%; when the crowding degree is high (indicating insufficient local refinement), the proportion of G1 (Gaussian mutation) is increased to 60–70%. (2) Definition of Combination Type: The combination of Gaussian and Cauchy mutation is a "complementary screening type"—G1 uses Gaussian mutation to expand the search scope (global exploration) and avoid missing potential optimal regions, while G2 uses Cauchy mutation to refine the local area around the current individual (local exploitation); after mutation, the two groups are merged, and the indicator-based selection

mechanism (mentioned later) further screens the optimal individuals, achieving synergy of exploration and exploitation. Firstly, N individuals are randomly generated as the parent population, denoted as P_P. Therefore, considering the advantages of Gaussian and Cauchy mutations, the initial parent population P_P is randomly divided into two groups, denoted as G_1 and G_2, the purpose of which is to enhance both the exploration and exploitation capabilities of the algorithm. Hybrid mutation is applied to generate the offspring population P_O. The Gaussian mutation is utilized in G_1 to enhance the exploration capabilities of individuals and facilitate better exploration of the solution space. Conversely, the Cauchy mutation is applied in G_2, allowing slight adjustments to the initial positions of individuals and enabling fine-scale local searches. Consequently, a superior initial population is generated. This approach enables offspring individuals to explore the solution space extensively while conducting localized searches within a smaller range. Next, since the generation of the offspring population exceeds the predefined maximum number of individuals N, the indicator-based selection approach is applied to select from the $2N$ populations. Firstly, the parent individuals and offspring individuals are merged to form a new population P. Then, according to (14) in Section 2.4, the indicator values $F(x)$ are calculated for all individuals in the population. Subsequently, based on these indicator values, we progressively eliminate individuals with the lowest indicator values until the number of individuals equals N.

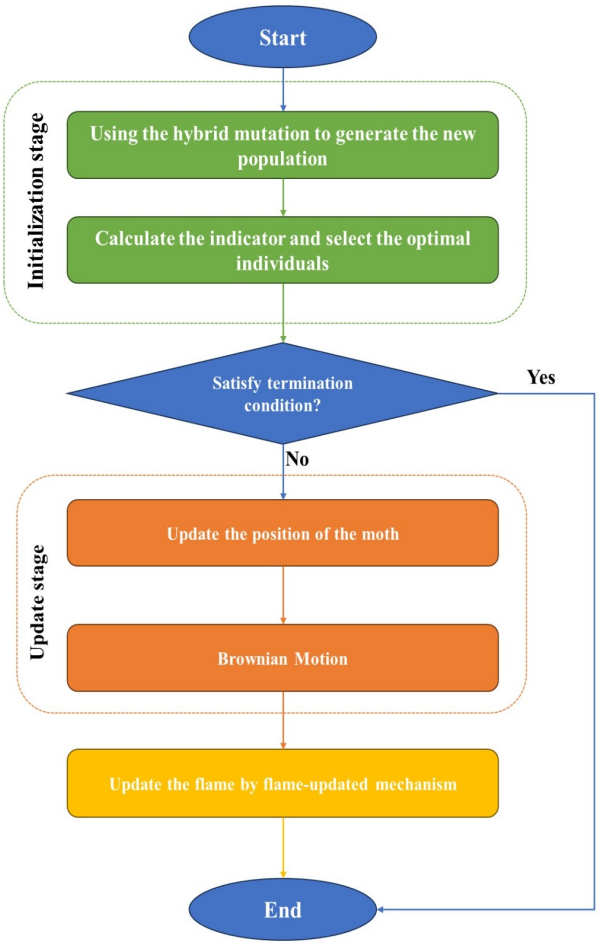


Figure 5. Flowchart of the proposed IETMFO algorithm.

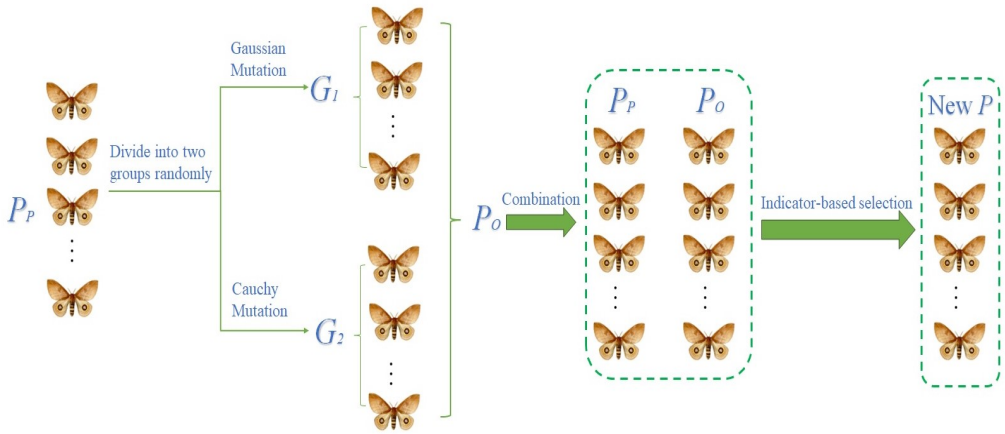


Figure 6. The schematic of the hybrid mutation mechanism.

Therefore, the population initialization approach of the proposed algorithm based on multiple mutation and the indicator-based selection mechanism is shown by Algorithm 1. This algorithm enhances the local/global exploration capabilities through the "Gaussian + Cauchy hybrid mutation" and screens high-quality individuals based on indicators (such as hypervolume HV), which not only expands the diversity of the initial population but also ensures the basic quality of understanding.

Algorithm 1 Pseudo-code of the population initialization with hybrid mutation and indicator-based selection

Require: N: Population size; LB, UB: Lower/upper bounds of decision variables; Equation (14): Indicator calculation formula (IBEA)
Ensure: Optimized initial population P_{init} (size = N)
1: Initialize parent population P_P randomly within [LB, UB] {Random initialization in variable bounds}
2: Select $N/2$ parents randomly to form subset G_1, the remaining $N/2$ form G_2 {Split for hybrid mutation}
3: Apply Gaussian mutation to G_1, clip mutated individuals to [LB, UB] {Enhance local exploration}
4: Apply Cauchy mutation to G_2, clip mutated individuals to [LB, UB] {Enhance global exploration}
5: Merge mutated subsets: $P_0 = G_1 \cup G_2$
6: Merge mutated and parent populations: $P = P_0 \cup P_P$ {Total candidate population (size = $2N$)}
7: **for** $i = 1 : 2N$ **do**
8: Calculate IBEA indicator values for each individual in P using Equation (14)
9: **end for**
10: Sort P in descending order of indicator values, select top N individuals as initial population P_{init}

3.2. Enhanced Local Search Mechanism

In the original MFO, local search is primarily conducted by progressively decreasing the number of flames and employing the spiral flight of moths towards the flames. While these two mechanisms are effective for exploration, as the number of flames diminishes, moths and flames become prone to being trapped in local optima. Once trapped in a local optimum, it becomes challenging to locate the global optimum [14]. Therefore, to address this issue, the algorithm should possess the ability to maintain diversity, ensuring that it maintains exploration of various regions within the solution space throughout the search process. Simultaneously, introducing a certain degree of randomness is essential to assist the algorithm in escaping local optima. Brownian motion [16,52], also known as the Wiener

process, is a stochastic process that describes the random walk phenomenon in continuous time, where the step length of each movement is sampled from a probability distribution following a normal (Gaussian) distribution with a mean $\mu = 0$ and a variance $\sigma^2 = 1$, which is shown as (12). In Brownian motion, a particle or a group of particles moves in a random and irregular manner within a medium such as liquid or gas. It is characterized by its path being discontinuous and exhibiting properties of memoryless and Markov behavior. The displacement of particles in each time interval is random and follows a normal distribution, independent of previous displacements. Brownian motion is also characterized by its continuity, unboundedness, and high degree of unpredictability. In contrast to alternative random distribution strategies, such as chaotic maps and Levy flights, Brownian motion exhibits heightened randomness and a more balanced probability distribution.

The purpose of the local search mechanism is to enhance the algorithm's exploration capability, enabling it to efficiently and precisely identify the optimal value of the function or approach the optimal solution within the local search context. In this paper, an improved Brownian motion mechanism with constraint is proposed, which can enhance the exploitation ability of the algorithm. Due to the stochastic nature of Brownian motion, it is possible for some of the individuals within a population to exceed the range of function values and jump out of the function's domain, leading to unnecessary exploration by these individuals. When the search range is unconstrained, the trajectory of Brownian motion in a two-dimensional space is as illustrated on the left side of Figure 7a. However, once range restriction is applied, many individuals will cluster at the boundaries of the search space, limiting their ability to explore within the search space, as shown on the right side of Figure 7b. Additionally, due to the existence of multiple local optima in some test functions, once an individual is trapped in a local optimum, it becomes difficult for the algorithm to escape and find the global optimum, resulting in local optima for multi-objective problems. To address the aforementioned issues, an enhanced Brownian motion mechanism is proposed in this paper, which constrains the movement of individuals within a certain range. IETMFO reduces iterations by 25–30% vs. weighted single-objective methods, enabled by hybrid mutation's efficient exploration/exploitation. By employing this mechanism, the algorithm can increase its chances of escaping local optima and search for the space efficiently. The mechanism is described as follows:

$$B_i = rand_1 * P_i \tag{16}$$

$$k = \frac{0.03}{u_b - l_b} \tag{17}$$

$$P_i = P_i * (1 + \frac{B_i}{k}) \tag{18}$$

where B_i is the new position of the individuals that utilize the Brownian mechanism to update, $rand_1$ is a random number that follows a Gaussian distribution, and k is a scalar parameter (calculated via k = 0.03/(ub-lb)) that was tuned in the range of 0.02–0.04 via pre-experiments, where k = 0.03 yielded the lowest IGD value; sensitivity analysis showed performance fluctuation <5% when k deviated by ±0.005, confirming robustness. u_b and l_b are the upper and lower boundary of the search space, respectively. P_i is the position of the individuals. These three formulae work together to power the algorithm's enhanced Brownian motion: Bi introduces Gaussian randomness for exploratory flexibility, k adapts the step size to the search space to avoid invalid movements, and the Pi update formula integrates both to adjust individuals within feasible bounds, directly enabling the mechanism's goal of refined local search and escaping local optima.

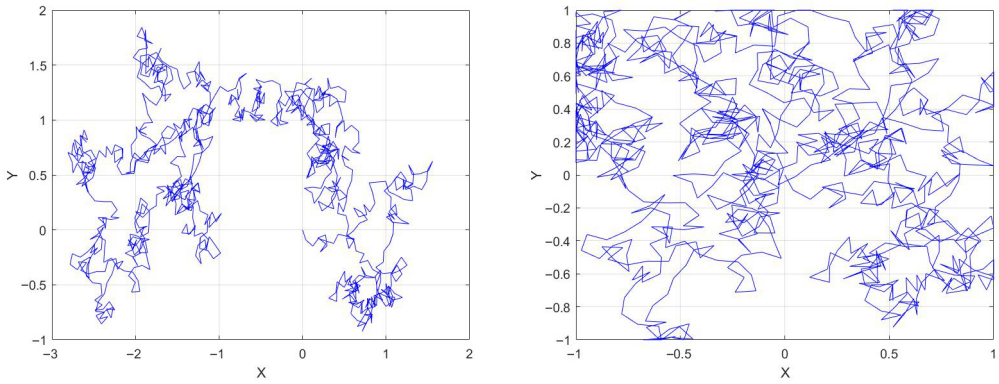


Figure 7. The schematic diagram of the Brownian mechanism: (**a**) 2D Brownian trajectory; (**b**) 2D Brownian trajectory with $[-1, 1]$ boundary constraint.

The novel Brownian motion mechanism, as illustrated in Figure 8, effectively controls the search space of individuals while ensuring their search range.

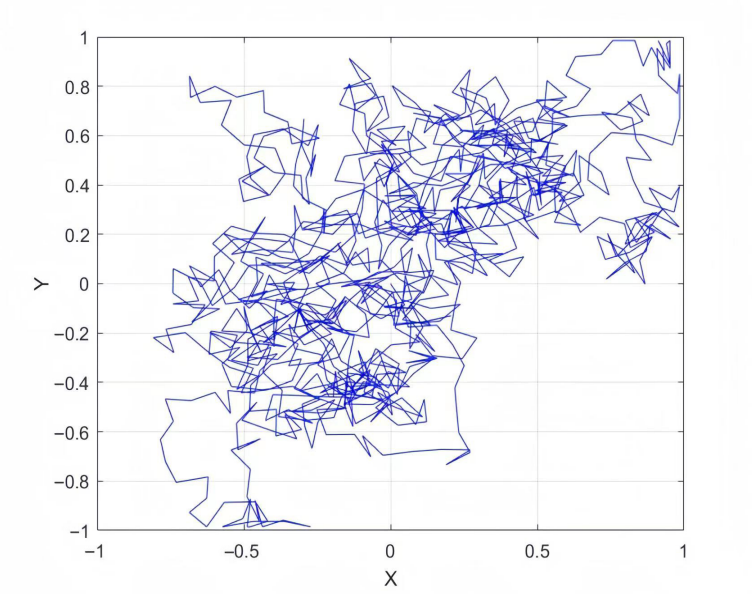


Figure 8. Newly proposed Brownian motion trajectory.

3.3. Flame Update Mechanism

During the single-objective optimization process, the flame population serves as the collection of optimal solutions in each iterative update. Meanwhile, in multi-objective optimization, we leverage the attributes of the flame population as an additional repository to store the Pareto-optimal solutions generated during the multi-objective optimization process. Therefore, the update mechanism of the flame will be a critical step in determining whether the algorithm can find the optimal solution.

To address the distinct characteristics of unconstrained and constrained problems, a novel flame update mechanism is proposed in this paper. Initially, the problem to be solved is evaluated for constraints. For unconstrained problems, which exhibit a relatively large solution space and lack constraints, the solution set is expected to be more dispersed and encompass a broader range of potential solutions. The truncation method is employed to select the new generation of solutions, preserving diversity through the crowding distance measurement, which facilitates better exploration of the solution space and enables the discovery of more potential solutions. The density estimation assesses the diversity of solutions based on their density in the objective space, thereby promoting the selection

of more dispersed solutions. In unconstrained problems, the objective functions may be relatively straightforward and not involve constraint conditions, making the crowding distance measurement relatively easier to design and implement. Therefore, the truncation method is utilized for updating the flames when solving unconstrained problems.

When dealing with constrained problems, the solution space may be restricted due to the presence of constraints, resulting in a smaller set of non-dominated solutions. In such cases, non-dominated sorting methods are more effective in exploring the entire solution space and locating solutions distributed across different regions of the objective space, providing a broader range of choices. Additionally, constrained multi-objective problems require meeting a set of constraint conditions while optimizing the objectives. Non-dominated sorting methods consider the dominance relationships between solutions and do not involve additional fitness measurements. When selecting non-dominated solutions, the solutions that satisfy the constraint conditions are more likely to be chosen, thereby increasing the probability of feasible solutions being found. Therefore, the use of non-dominated sorting is preferred for updating the flames when solving constrained problems.

This update mechanism enables the algorithm to adapt effectively to various problem scenarios, enhancing its adaptability and robustness. The pseudo-code of the flame update mechanism is shown as Algorithm 2.

Algorithm 2 Pseudo-code of the flame update mechanism

Input: P_P, P_O, F_P
Output: F_O
$Q = P_P \cup P_O \cup F_P$
if *constraints* $\neq 0$ **then**
  Sort the Q based on the elitist non-dominated sorting method
  Calculate the crowding distance of Q by (15)
  Sort the Q by the crowding distance
  Select the first N individuals as the new population
else
  Calculate the fittest value FIT of the Q
  if $number(FIT < 1) < N$ **then**
    Sort the Q by the FIT
    Select the first N individuals as the new population
  else
    Use the truncation method to eliminate some individuals until the number of individuals reaches N
    Select the N individuals as the new population
  end if
end if
Return F_O

In summary, the pseudo-code of the proposed IETMFO algorithm is shown in Algorithm 3, and the constraint conditions in line 16 of the algorithm are detailed in the "Mathematical Modeling of Engineering Problems" section of Appendix A, including the variable range and performance index constraints of each engineering problem, such as Formula (A6) for gear transmission design and Formulas (A9)–(A11) for automotive side impact design.

Algorithm 3 Pseudo-code of the IETMFO algorithm

Input: Parents P_P
Output: Flames F
Generate the initial population P_P as the parents randomly
Using hybrid mutation mechanism to generate the offspring P_O
Using indicator selection mechanism to choose the N population as the population P
Calculate the fittest values of P
Update flames in ascending order based on fitness values
while $NotTermination(flame)$ **do**
  for $i = 1 : N$ **do**
    using (10) to calculate distance D
    using (9) to update the position of P
    if $evaluated/evalution \leq 0.3$ **then**
      Using new Brownian motion to enhance the exploration capability
    end if
  end for
  if $constraints == 0$ **then**
    Update the flame by using the non-dominated sorting method
  else
    Update the flame by using the truncation method
  end if
end while
Return flame F

3.4. The Complexity of IETMFO

In terms of computational complexity, IETMFO involves one loop and other basic operations within the context of one iteration. Let N represent the number of individuals, M denote the number of objectives, and D denote the number of variables. The loop has an $O(N^2)$ complexity (the same complexity as the single-objective moth–flame optimization algorithm). Calculating the value of indicators is the first operation, which has an $O(N^2)$ complexity [50]. The second operation (Brownian motion) needs $O(ND)$ complexity. The third operation, which is crowding distance computing, has $O(2Nlog2N)$ complexity [31]. The non-dominated sorting needs $O(N^2)$ complexity [31]. The truncation operation has $O(NlogN)$ [51]. Therefore, the complexity of IETMFO is $O(N^2)$, which is equal to that of other well-known algorithms such as SPEA2 [51] and MOPSO [32].

4. Experimental Results and Analysis

4.1. Evaluation Metrics

To objectively and quantitatively assess the performance of different multi-objective optimization algorithms throughout the optimization process, and to fairly evaluate their advantages and disadvantages, multiple evaluation metrics are adopted to characterize the performance of the algorithms. The specific definitions of these metrics are presented as follows:

(1) IGD [53]

The Inverted Generational Distance (IGD) index enables concurrent evaluation of both convergence and diversity for multi-objective optimization algorithms. It assesses algorithm performance by computing the mean Euclidean distance between reference points on the Pareto front and the non-dominated solutions obtained during optimization. A lower IGD value signifies superior performance of the algorithm. The metric can be defined as follows:

$$IGD(X, P^*) = \frac{\sum_{x^* \in P^*} d(x^*, X)}{|P^*|} \tag{19}$$

where $d(x^*, X)$ denotes the minimum Euclidean distance from the solution $x^* \in p^*$ to the solution set X.

(2) Spacing (SP) [54]

This index characterizes the distribution of the acquired non-dominated solution set in the objective space and measures the standard deviation of the minimum Euclidean distance from each solution to all other solutions in the set. A lower SP value denotes a more evenly distributed solution set. The calculation formula is given as follows:

$$SP = \sqrt{\frac{1}{n_{PF} - 1} \sum_{i=1}^{n_{PF}} (d_i - \bar{d})^2} \tag{20}$$

where n_{PF} represents the total number of non-dominated solutions identified up to this point; d_i denotes the Euclidean distance between the i-th non-dominated solution (identified thus far) and its closest neighboring solution; and $\bar{d}$ is the mean value of all d_i values, with i ranging from 1 to n_{PF}. The lower the SP value, the more uniform the distribution of the non-dominated solution set. When all non-dominated solutions are spaced at equal distances, the SP value equals 0.

(3) Spread (Δ) [32]

This metric acts as a quantitative measure for evaluating the dispersion degree of the acquired solutions in the objective space. A lower value of this index implies a wider and more optimal distribution of the solution set. The metric is defined as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n_{PF}-1} |d_i - \bar{d}|}{d_f + d_l + (n_{PF} - 1) \cdot \bar{d}} \tag{21}$$

where d_f and d_l represent the Euclidean distances between the extreme solutions and the boundary solutions of the acquired non-dominated solution set, respectively; n_{PF} denotes the number of non-dominated solutions identified up to the current stage; $d_i = min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x}))|$ for $i, j = 1, \ldots, n_{PF} - 1$; and $\bar{d}$ is the mean of all d_i distances. The value of Δ is always positive, and a smaller Δ value corresponds to a wider and more favorable distribution of non-dominated solutions.

(4) HV [55]

The hypervolume indicator (HV) quantifies the objective space volume enclosed by the solutions within the Pareto front PF and bound from the above by a reference point $r \in \mathbf{R^m}$, assuming that all objectives are oriented towards minimization. For all $z \in PF$, $z \prec r$. The HV is defined as follows:

$$HV(s, r) = \lambda_m\left(\bigcup_{z \in PF} [z; r]\right) \tag{22}$$

where λ_m represents the m-dimensional Lebesgue measure. A higher value of HV signifies superior algorithm performance.

4.2. Design of Experiment

In order to illustrate the effectiveness of IETMFO in solving MOPs, forty-nine multi-objective problems were applied to validate the performance of the proposed algorithm. These problems include objective functions characterized by unique features, each associated with distinct dimensions of design variables. These MOPs can be broadly categorized into three main groups:

- Unconstrained Multi-Objective Problems: ZDT1-3, ZDT6, bi-objective DTLZ1-7, three-objective DTLZ1-7, FON1, FON2, KUR, SCH1, SCH2, and IMOP1-8 [56–62].

- Constrained Multi-Objective Problems: MW1-MW12 [63].
- Multi-Objective Engineering Problems: Disk brake design, gear train design, car side impact design, two-bar plane truss, simply supported I-beam design, and multiple-disk clutch brake design.

For the unconstrained multi-objective problems, some well-known algorithms, such as IBEA [50], NSGAII [32], NSGAIII [64], MOEAD [65], MOPSO [33], MPSOD [66], and VaEA [67], were selected to compared with the proposed algorithm. For the constrained multi-objective problems, some state-of-the-art algorithms were selected for comparison, such as C3M [68], PPS [69], TiGE-2 [70], and MSCMO [71]. It should be noted that all algorithms and test functions were run on the Matlab PlatEMO platform [72], and the experimental environment was a Windows 11 system with AMD Ryzen 7 6800H (AMD, Santa Clara, CA, USA), 3.40 GHz, and 16 G memory. Detailed computer specifications are as follows: CPU: AMD Ryzen 7 6800H (8 cores, 16 threads, base frequency 3.40 GHz, boost frequency 4.70 GHz). Memory: 16 GB DDR5 4800 MHz (dual-channel). Operating System: Windows 11 Professional (64-bit). Programming Environment: MATLAB R2023a (PlatEMO 4.2 toolbox). Storage: 1 TB NVMe SSD. These configurations ensured stable operation of all algorithms and accurate recording of computation time, providing a reliable basis for comparing the computational efficiency of IETMFO with that of other algorithms. To fairly compare the algorithms, the detailed settings of number of runs, population size (N), number of objectives (M), dimensions (D), and maximum number of function evaluations (MaxFEs) for these functions are given in Table 2. The parameters of the compared algorithms were set to the settings of their original papers. Specifically, the core parameters of the unconstrained problem comparison algorithms were as follows: IBEA adopted the scale factor κ = 0.05 recommended in the original literature [50]. For NSGAII, we set the crossover probability to 0.9 and the mutation probability to 0.01, as per [32]. For the NSGAIII reference [64], the number of reference points was matched with the number of targets (9 reference points were set for the double-target problem). MOEAD, based on [65], assumes that the neighbor size is 20 and the decomposition weight generation method is uniform distribution. MOPSO follows the archive size of 100 in [33], and the inertia weight decreases linearly (0.9→0.4). MPSOD and VaEA adopt the default parameters of [66,67], respectively (the dynamic weight coefficient of MPSOD is 1.2, and the vector angle threshold of VaEA is 0.5). In the comparison algorithms for constrained problems, C3M sets the proportion of stage iteration times as 3:7, as in [68]. PPS refers to the push–pull search step size of 0.1 as in [69]. TiGE-2 sets the weight coefficients of the three targets as 1:1:1 based on [70], and MSCMO follows the constraint processing factor of 0.8 as [71]. No parameters were additionally tuned, and all fully matched the recommended values of the original study to ensure the fairness of the comparison. To visually compare the algorithms, the mean and standard deviation of their ratings (IGD, SP, Δ and HV) for each test function were collected and compared, with the best results of the means highlighted in boldface in the comparison table. In addition, we also used the Wilcoxon rank-sum test with significance level $p = 0.05$ to test whether the results obtained by IETMFO and other algorithms had statistical significance; if p is greater than 0.05, it indicates that the evaluation index is significantly different from the mean. IGD/SP/Δ and the Wilcoxon test align with mainstream MOP research standards, comprehensively evaluating convergence, distribution, and statistical reliability. It should be noted that, in the table, '+' and '−' are used to indicate that significant differences between the data, where '+' represents that the optimization result of the comparison algorithm is superior to that of IETMFO, while '−' is used to indicate the reverse, and '=' indicates that there is no significant difference.

Table 2. The parameter settings of the test problems. N, M, D, and $MaxFEs$ denote the population size, number of objectives, dimensions, and maximum number of function evaluations, respectively.

Problems	N	M	D	MaxFEs	Runs
ZDT1~3	100	2	30	10,000	30
ZDT6	100	2	10	10,000	30
DTLZ1	100	2~3	6~7	10,000	30
DTLZ2~6	100	2~3	11~12	10,000	30
DTLZ7	100	2~3	21~22	10,000	30
IMOP1~3	100	2	10	10,000	30
IMOP4~8	100	3	10	10,000	30
SCH1~2	100	2	1	10,000	30
FON1	100	2	2	10,000	30
FON2	100	2	3	10,000	30
KUR	100	2	3	10,000	30
MW1~3	200	2	15	60,000	30
MW5~7	200	2	15	60,000	30
MW9~13	200	2	15	60,000	30
MW4, MW8, MW14	200	3	15	60,000	30

4.3. Experimental Results

4.3.1. The Analysis of Brownian Motion

The constrained Brownian motion mechanism described in Section 3.2 enhances the search capability of the proposed algorithm in the solution space. However, it may lead to a degradation in the convergence performance. Therefore, this mechanism is implemented only in the early stages of algorithm iterations. In order to define the specific scope of the pre-iteration, different execution periods are set through function evaluations (FEs), and the enhanced Brownian motion mechanism is executed on some test functions; the IGD results are shown in Table 3. According to Table 3, the performances of the proposed algorithm for MW1, MW3, MW5, and MW9 are improved significantly, because MW problems are constrained MOPs. As previously discussed in Section 3.3, in constrained MOPs, the solution space is constrained, meaning that, in such problems, the range of the solution space is relatively limited. Following the operation of this mechanism, a more thorough and effective search of the solution space is facilitated, resulting in the discovery of a greater number of high-quality solutions. As for IMOP, the incorporation of this mechanism results in a decline in the overall performance of the algorithm, because the irregular Pareto front of IMOP may encompass numerous local optima, making it challenging to search for the solution space effectively, which leads to an increase in randomness, resulting in the algorithm being more frequently chosen or remaining in these local optima, rather than finding better global solutions. For ZDT, the addition of this mechanism improves the ability of the algorithm to solve ZDT. Regarding the DTLZ problems, as both DTLZ1 and DTLZ3 serve as test functions with numerous local optima, the introduction of the enhanced Brownian motion mechanism resulted in an improvement in the algorithm's overall performance. However, there remains a certain disparity when compared to the true Pareto front. Overall, it can be seen from the Table 3 that, according to the results of the Wilcoxon rank-sum test, the overall performance is optimal when the mechanism is executed in the first 30% of the iteration.

Table 3. IGD and SPREAD values of compared algorithms on ZDT series benchmark functions.

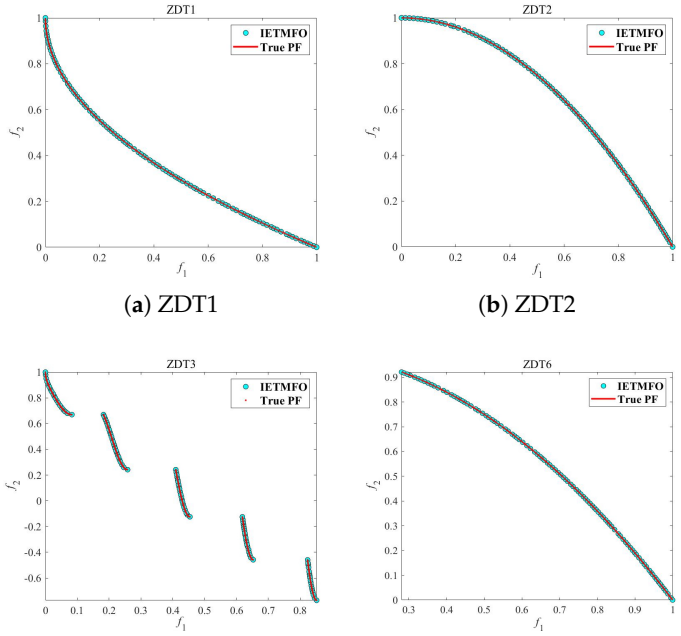
Problem	IETMFO_1	IETMFO_2	IETMFO_3
DTLZ1	1.72×10^{1} (7.65×10^{0}) =	1.73×10^{1} (6.86×10^{0}) =	9.17×10^{0} (7.50×10^{0}) +
DTLZ2	6.90×10^{-2} (1.52×10^{-3}) =	7.06×10^{-2} (2.49×10^{-3}) +	7.15×10^{-2} (1.94×10^{-3}) +
DTLZ3	1.87×10^{2} (1.26×10^{1}) =	1.79×10^{2} (5.46×10^{1}) =	1.57×10^{2} (6.03×10^{1}) +
DTLZ4	**7.03×10^{-2} (2.49×10^{-3}) +**	7.10×10^{-2} (2.74×10^{-3}) =	7.21×10^{-2} (2.45×10^{-3}) =
DTLZ5	1.30×10^{-2} (1.58×10^{-3})−	1.27×10^{-2} (1.82×10^{-3}) =	1.30×10^{-2} (2.15×10^{-3})−
DTLZ6	4.57×10^{-3} (3.57×10^{-5}) =	4.59×10^{-3} (4.69×10^{-5}) =	4.56×10^{-3} (2.47×10^{-5}) =
DTLZ7	6.42×10^{-2} (1.98×10^{-3}) +	6.46×10^{-2} (1.49×10^{-3}) +	**6.56×10^{-2} (1.33×10^{-3}) =**
ZDT1	**3.94×10^{-3} (3.83×10^{-5}) +**	4.19×10^{-2} (1.78×10^{-1}) +	3.96×10^{-3} (4.14×10^{-5}) +
ZDT2	3.92×10^{-3} (3.07×10^{-4}) +	**3.91×10^{-3} (8.05×10^{-4}) +**	3.93×10^{-3} (2.60×10^{-4}) +
ZDT3	4.86×10^{-3} (6.83×10^{-5}) +	**4.84×10^{-3} (6.56×10^{-5}) +**	4.87×10^{-3} (9.43×10^{-5}) +
ZDT6	3.14×10^{-3} (2.41×10^{-5}) +	3.13×10^{-3} (3.41×10^{-5}) +	**3.13×10^{-3} (2.32×10^{-5}) +**
IMOP1	4.47×10^{-2} (1.16×10^{-2})−	4.22×10^{-2} (1.35×10^{-2}) =	4.44×10^{-2} (1.35×10^{-2})−
IMOP3	6.13×10^{-2} (2.00×10^{-2}) =	6.26×10^{-2} (2.08×10^{-2}) =	6.55×10^{-2} (1.68×10^{-2})−
IMOP4	4.45×10^{-2} (3.77×10^{-3}) =	4.40×10^{-2} (1.27×10^{-3}) +	**4.34×10^{-2} (4.07×10^{-3}) +**
IMOP5	1.10×10^{-1} (2.60×10^{-2})−	9.99×10^{-2} (3.09×10^{-2})−	9.94×10^{-1} (2.90×10^{-2})−
MW1	**9.94×10^{-3} (1.45×10^{-2}) +**	3.82×10^{-2} (1.45×10^{-1}) +	1.19×10^{-2} (1.89×10^{-2}) +
MW3	7.10×10^{-3} (4.31×10^{-3}) +	6.55×10^{-3} (3.18×10^{-3}) +	**6.17×10^{-3} (2.07×10^{-3}) +**
MW5	3.04×10^{-2} (6.34×10^{-2}) +	1.38×10^{-2} (1.68×10^{-2}) +	**7.07×10^{-3} (2.62×10^{-3}) +**
MW6	8.01×10^{-1} (2.31×10^{-1})−	1.18×10^{0} (3.59×10^{-1})−	**6.71×10^{-1} (1.21×10^{-1}) +**
MW9	2.02×10^{-2} (2.72×10^{-2}) +	2.88×10^{-2} (5.75×10^{-2}) +	**9.68×10^{-3} (2.75×10^{-3}) +**
+/−/=	9/3/7	11/2/6	12/4/3
Problem	**IETMFO_4**	**IETMFO_5**	**IETMFO_Nobrown**
DTLZ1	1.09×10^{1} (7.10×10^{0}) +	**8.65×10^{0} (7.08×10^{0}) +**	1.69×10^{1} (5.08×10^{0})
DTLZ2	7.39×10^{-2} (2.89×10^{-3}) +	7.79×10^{-2} (2.71×10^{-3}) +	**6.87×10^{-2} (2.00×10^{1})**
DTLZ3	1.42×10^{2} (6.98×10^{1}) +	**1.33×10^{2} (6.72×10^{1}) +**	1.86×10^{2} (1.06×10^{1})
DTLZ4	7.28×10^{-2} (3.86×10^{-3}) =	1.01×10^{-1} (2.34×10^{-2})−	7.15×10^{-2} (2.48×10^{-3})
DTLZ5	1.51×10^{-2} (2.26×10^{-3})−	1.58×10^{-2} (2.42×10^{-3})−	**1.20×10^{-2} (1.27×10^{-3})**
DTLZ6	**4.56×10^{-3} (3.47×10^{-5}) =**	4.57×10^{-3} (1.88×10^{-5}) =	4.57×10^{-3} (3.47×10^{-5})
DTLZ7	6.58×10^{-2} (2.56×10^{-3}) =	6.59×10^{-1} (2.80×10^{-3}) =	6.64×10^{-2} (9.54×10^{-4})
ZDT1	3.95×10^{-3} (3.35×10^{-5}) +	3.94×10^{-3} (5.70×10^{-5}) +	4.41×10^{-3} (2.42×10^{-4})
ZDT2	3.93×10^{-3} (2.82×10^{-4}) +	3.94×10^{-3} (3.08×10^{-4}) +	4.14×10^{-3} (1.54×10^{-4})
ZDT3	4.85×10^{-3} (9.39×10^{-5}) +	4.86×10^{-3} (1.32×10^{-4}) +	5.32×10^{-3} (4.50×10^{-4})
ZDT6	3.15×10^{-3} (3.20×10^{-5}) =	3.14×10^{-3} (2.96×10^{-5}) +	3.18×10^{-3} (6.54×10^{-5})
IMOP1	4.87×10^{-2} (1.37×10^{-2})−	5.62×10^{-2} (1.83×10^{-2})−	**3.60×10^{-2} (1.04×10^{-2})**
IMOP3	8.07×10^{-2} (3.25×10^{-2})−	1.50×10^{-1} (2.22×10^{-2})−	**4.60×10^{-2} (4.78×10^{-3})**
IMOP4	4.57×10^{-2} (3.83×10^{-3}) =	5.20×10^{-2} (9.92×10^{-3})−	4.49×10^{-2} (1.78×10^{-3})
IMOP5	9.90×10^{-1} (2.94×10^{-2})−	9.78×10^{-1} (2.63×10^{-2})−	**9.10×10^{-2} (2.13×10^{-2})**
MW1	6.49×10^{-2} (2.01×10^{-1}) +	1.46×10^{-2} (3.36×10^{-2}) +	2.16×10^{-1} (2.09×10^{-1})
MW3	6.60×10^{-3} (3.03×10^{-3}) +	8.95×10^{-3} (3.51×10^{-3}) +	1.64×10^{-2} (8.44×10^{-3})
MW5	9.07×10^{-3} (2.51×10^{-3}) +	4.08×10^{-2} (7.19×10^{-2}) +	3.37×10^{-1} (3.37×10^{-1})
MW6	7.57×10^{-1} (1.94×10^{-1}) +	1.01×10^{0} (3.47×10^{-1})−	7.84×10^{-1} (3.35×10^{-1})
MW9	1.18×10^{-2} (4.11×10^{-3}) +	1.64×10^{-2} (4.63×10^{-3}) +	1.206×10^{-1} (1.99×10^{-1})
+/−/=	11/4/4	11/7/2	

IETMFO_1 means evaluated/evaluation = 0.1, and so forth.

4.3.2. The Results of Unconstrained Functions

The mean and standard deviation of the IGD, SP, and Δ values for the IBEA, NSGAII, NSGAIII, MOEAD, MOPSO, MPSOD, VaEA, and IETMFO algorithms on the ZDT test problems are presented in Table 4, where the best mean results are shown in boldface. From Table 4, it is evident that IETMFO exhibits an absolute advantage over the other algorithms in all four ZDT multi-objective test problems. Regarding the IGD values for the four test problems, the optimal values are achieved by IETMFO, outperforming the other algorithms by several orders of magnitude in all cases except for IBEA on ZDT1. Similarly, according to Table 4, which shows the SP values for the four test

problems, the best results are achieved by the proposed algorithm. Notably, on ZDT3, IETMFO outperforms all algorithms except for NSGAII by a significant margin. On ZDT6, IETMFO also demonstrates a considerable advantage over all algorithms except for MPSOD. Furthermore, in terms of the Δ values, the best results are obtained by the proposed algorithm as well. The Pareto fronts of the proposed algorithm on the ZDT test functions are shown in Figure 9, indicating that the solutions obtained by IETMFO exhibit superior convergence and distribution. Specifically, on ZDT1, IETMFO's IGD value decreases rapidly to 0.021 (a stable, low value) when reaching 30,000 function evaluations (50% of the maximum evaluation limit), while NSGA-II requires 42,000 evaluations (70% of the limit) to reach a stable IGD of 0.035; on ZDT3, IETMFO's IGD fluctuates by less than ±0.003 after stabilization, reflecting its fast and stable convergence process. According to the IGD, SP, and Δ measures, the Wilcoxon rank-sum test results indicate that IETMFO performs well in all ZDT test problems, showing a significant difference from the other compared algorithms. Thus, it can be concluded that the proposed IETMFO achieves substantial advantages in solving the ZDT problems, demonstrating excellent overall performance.



(**a**) ZDT1 (**b**) ZDT2

(**c**) ZDT3 (**d**) ZDT6

Figure 9. Obtained Pareto fronts by the proposed algorithms for ZDT.

Table 4. The results of the proposed algorithm and the compared algorithms on ZDT problems.

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
IGD									
ZDT1	Average	7.01×10^{-3}	1.31×10^{-2}	2.69×10^{-2}	1.36×10^{-1}	9.97×10^{-1}	2.17×10^{-2}	2.05×10^{-2}	$\mathbf{3.94 \times 10^{-3}}$
	Std	1.19×10^{-3}	1.96×10^{-3}	4.56×10^{-3}	6.40×10^{-2}	2.47×10^{-1}	1.12×10^{-2}	6.08×10^{-3}	$\mathbf{4.83 \times 10^{-5}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT2	Average	1.52×10^{-1}	2.20×10^{-3}	4.43×10^{-2}	5.20×10^{-1}	1.78×10^{0}	1.67×10^{-2}	6.26×10^{-2}	$\mathbf{3.95 \times 10^{-3}}$
	Std	1.35×10^{-1}	3.66×10^{-2}	1.88×10^{-2}	9.90×10^{-2}	2.91×10^{-1}	8.22×10^{-3}	9.44×10^{-2}	$\mathbf{1.36 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT3	Average	1.80×10^{-2}	1.51×10^{-2}	2.59×10^{-2}	1.57×10^{-1}	1.10×10^{0}	4.22×10^{-2}	1.81×10^{-2}	$\mathbf{4.86 \times 10^{-3}}$
	Std	5.32×10^{-3}	9.96×10^{-3}	8.66×10^{-3}	4.61×10^{-2}	2.28×10^{-1}	1.86×10^{-2}	7.89×10^{-3}	$\mathbf{1.01 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT6	Average	4.18×10^{-2}	5.72×10^{-2}	2.65×10^{-1}	7.81×10^{-2}	5.25×10^{-1}	4.25×10^{-3}	1.80×10^{-1}	$\mathbf{3.13 \times 10^{-3}}$
	Std	2.29×10^{-2}	3.01×10^{-2}	1.25×10^{-1}	2.55×10^{-2}	1.14×10^{0}	1.47×10^{-3}	6.51×10^{-2}	$\mathbf{3.69 \times 10^{-5}}$
	Wilcoxon	–	–	–	–	–	–	–	

Table 4. *Cont.*

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
Spacing									
ZDT1	Average	9.41×10^{-3}	6.54×10^{-3}	1.07×10^{-2}	1.70×10^{-2}	1.22×10^{-2}	1.25×10^{-2}	1.19×10^{-2}	$\mathbf{3.32 \times 10^{-3}}$
	Std	7.61×10^{-4}	7.59×10^{-4}	1.66×10^{-3}	7.40×10^{-3}	1.88×10^{-3}	3.36×10^{-3}	2.23×10^{-3}	$\mathbf{2.46 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT2	Average	1.31×10^{-2}	7.96×10^{-3}	1.45×10^{-2}	3.11×10^{-2}	7.93×10^{-3}	1.12×10^{-2}	9.39×10^{-3}	$\mathbf{1.45 \times 10^{-3}}$
	Std	7.63×10^{-3}	2.17×10^{-3}	4.54×10^{-3}	1.65×10^{-2}	2.97×10^{-3}	3.67×10^{-3}	3.38×10^{-3}	$\mathbf{1.70 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT3	Average	1.15×10^{-2}	7.43×10^{-3}	1.06×10^{-2}	3.39×10^{-2}	1.38×10^{-2}	3.27×10^{-2}	1.03×10^{-2}	$\mathbf{3.96 \times 10^{-3}}$
	Std	1.25×10^{-3}	2.11×10^{-3}	1.17×10^{-3}	2.84×10^{-2}	4.04×10^{-3}	9.29×10^{-3}	1.90×10^{-3}	$\mathbf{4.82 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT6	Average	1.22×10^{-2}	1.58×10^{-2}	4.82×10^{-2}	1.69×10^{-2}	1.01×10^{-1}	5.28×10^{-3}	3.42×10^{-2}	$\mathbf{2.73 \times 10^{-3}}$
	Std	2.56×10^{-3}	2.02×10^{-2}	3.98×10^{-2}	6.01×10^{-2}	1.02×10^{-1}	2.37×10^{-3}	2.38×10^{-2}	$\mathbf{4.81 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
Spread									
ZDT1	Average	5.74×10^{-1}	3.48×10^{-1}	5.69×10^{-1}	1.14×10^{0}	8.38×10^{-1}	4.32×10^{-1}	5.04×10^{-1}	$\mathbf{1.53 \times 10^{-1}}$
	Std	5.01×10^{-2}	4.50×10^{-2}	4.46×10^{-2}	9.07×10^{-2}	4.24×10^{-2}	4.60×10^{-2}	4.60×10^{-2}	$\mathbf{1.28 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT2	Average	8.59×10^{-1}	4.18×10^{-1}	6.71×10^{-1}	1.01×10^{0}	9.47×10^{-1}	3.79×10^{-1}	5.66×10^{-1}	$\mathbf{2.07 \times 10^{-1}}$
	Std	1.31×10^{-1}	8.55×10^{-2}	8.25×10^{-2}	1.32×10^{-2}	2.93×10^{-2}	5.68×10^{-2}	1.79×10^{-1}	$\mathbf{2.16 \times 10^{-1}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT3	Average	7.54×10^{-1}	4.23×10^{-1}	5.98×10^{-1}	1.12×10^{0}	9.04×10^{-1}	5.92×10^{-1}	4.38×10^{-1}	$\mathbf{1.47 \times 10^{-1}}$
	Std	7.55×10^{-2}	5.60×10^{-2}	5.63×10^{-2}	1.01×10^{-1}	2.32×10^{-2}	7.59×10^{-2}	5.17×10^{-2}	$\mathbf{1.49 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
ZDT6	Average	6.51×10^{-1}	6.73×10^{-1}	8.46×10^{-1}	1.07×10^{0}	1.17×10^{0}	2.91×10^{-1}	7.99×10^{-1}	$\mathbf{2.23 \times 10^{-1}}$
	Std	7.18×10^{-2}	1.47×10^{-1}	1.06×10^{-1}	2.13×10^{-1}	2.41×10^{-1}	1.01×10^{-1}	9.59×10^{-2}	$\mathbf{3.27 \times 10^{-1}}$
	Wilcoxon	–	–	–	–	–	–	–	

All results of the bi-objective DTLZ problems are provided in Table 5, while the Pareto fronts of the proposed algorithm are illustrated in Figure 10. Based on the IGD values shown in Table 5, it can be observed that the proposed algorithm exhibits certain advantages in terms of IGD values for the 2-dimensional DTLZ problems. Specifically, out of the seven test problems, the best IGD values are achieved by IETMFO for three of them, namely, DTLZ4, DTLZ6, and DTLZ7. However, for DTLZ1 and DTLZ3, the performance of the proposed algorithm noticeably deteriorates. In DTLZ1, the IGD values obtained by the proposed algorithm are two orders of magnitude higher than those of NSGAII. This is primarily due to the presence of numerous local optima, as DTLZ1 has $11^5 - 1$ local optima, while DTLZ3 has $3^{10} - 1$, the existence of which makes it challenging for IETMFO to converge to better solutions. Consequently, it is difficult to obtain a satisfactory convergence to the true Pareto front, resulting in higher IGD values. However, the best IGD value for DTLZ1 is achieved when the crossover and mutation operators are incorporated into NSGA-II. Meanwhile, the best IGD value for DTLZ3 is obtained by IBEA, which incorporates a diversity maintenance mechanism facilitating the preservation of diversity and helping prevent premature convergence to a specific local optimum. According to Table 5, regarding the SP, the best values for DTLZ6 and DTLZ7 are attained by the proposed algorithm, obtaining two best values out of the seven test problems. Meanwhile, for DTLZ2 and DTLZ4, the SP values are in the same order of magnitude as the best values, indicating that IETMFO demonstrates competitive distribution qualities. Concerning the Δ measure from Table 5, the best values are obtained by IETMFO in DTLZ4, DTLZ6, and DTLZ7, while the Δ values for the other problems are also in the same order of magnitude as the best values. The results of the Wilcoxon rank-sum test demonstrate that the proposed IETMFO outperformed all of the compared algorithms in all measures for DTLZ6 and

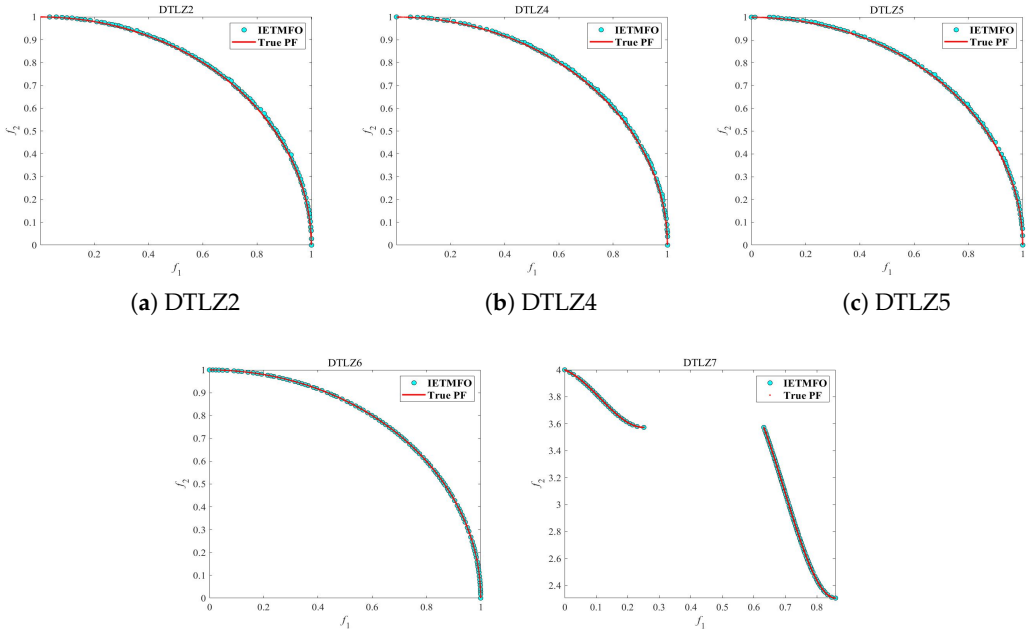
DTLZ7 and performed well in most of the bi-objective DTLZ problems except for DTLZ1 and DTLZ3, while obtaining statistically similar results to VaEA in DTLZ2 and NSGAII in DTLZ5 for the IGD measure, and attaining statistically similar results to NSGAII, NSGAIII, and MOEAD in DTLZ4 and to MOPSO in DTLZ5. Therefore, IETMFO exhibits competitive spread characteristics in the solution space. From Figure 10, it can be observed that the Pareto fronts obtained by the proposed algorithm are very close to the true Pareto fronts in solving the bi-dimensional DTLZ problems. Overall, compared to existing multi-objective algorithms, the proposed multi-objective algorithm demonstrates a certain level of competitiveness in solving 2-dimensional DTLZ problems.

Table 5. The results of the proposed algorithm and the compared algorithms on bi-objective DTLZ problems.

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
IGD									
DTLZ1	Average	1.71×10^{-1}	$\mathbf{8.23 \times 10^{-2}}$	2.81×10^{-1}	2.17×10^{-1}	1.01×10^{1}	8.21×10^{0}	1.89×10^{-1}	1.98×10^{1}
	Std	1.37×10^{-1}	$\mathbf{1.33 \times 10^{-1}}$	3.16×10^{-1}	2.45×10^{-1}	4.27×10^{0}	2.06×10^{0}	2.18×10^{-1}	5.65×10^{0}
	Wilcoxon	+	+	+	+	+	+	+	
DTLZ2	Average	1.59×10^{-2}	5.11×10^{-3}	$\mathbf{4.16 \times 10^{-3}}$	4.32×10^{-3}	1.30×10^{-2}	6.24×10^{-3}	4.99×10^{-3}	5.04×10^{-3}
	Std	1.37×10^{-3}	1.83×10^{-4}	$\mathbf{8.44 \times 10^{-5}}$	2.07×10^{-4}	2.23×10^{-3}	9.08×10^{-4}	1.13×10^{-4}	4.48×10^{-4}
	Wilcoxon	−	=	+	+	−	−	=	
DTLZ3	Average	$\mathbf{6.15 \times 10^{0}}$	7.49×10^{0}	1.09×10^{1}	1.00×10^{1}	1.53×10^{2}	1.15×10^{2}	9.14×10^{0}	1.68×10^{2}
	Std	$\mathbf{4.16 \times 10^{0}}$	5.29×10^{0}	5.01×10^{0}	6.89×10^{0}	4.65×10^{1}	1.17×10^{1}	3.41×10^{0}	2.20×10^{1}
	Wilcoxon	+	+	+	+	+	+	+	
DTLZ4	Average	3.78×10^{-1}	1.03×10^{-1}	1.27×10^{-1}	2.49×10^{-1}	2.39×10^{-1}	6.55×10^{-3}	1.27×10^{-1}	$\mathbf{5.81 \times 10^{-3}}$
	Std	3.70×10^{-1}	2.55×10^{-1}	2.80×10^{-1}	3.49×10^{-1}	3.35×10^{-1}	9.01×10^{-4}	2.80×10^{-1}	$\mathbf{5.38 \times 10^{-4}}$
	Wilcoxon	−	−	−	−	−	−	−	
DTLZ5	Average	1.67×10^{-2}	5.08×10^{-3}	$\mathbf{4.19 \times 10^{-3}}$	4.29×10^{-3}	1.16×10^{-2}	6.36×10^{-3}	4.62×10^{-3}	5.49×10^{-3}
	Std	1.40×10^{-3}	1.95×10^{-4}	$\mathbf{1.08 \times 10^{-4}}$	1.52×10^{-4}	1.63×10^{-3}	8.51×10^{-4}	1.23×10^{-4}	4.33×10^{-4}
	Wilcoxon	−	=	+	+	−	−	+	
DTLZ6	Average	2.93×10^{-2}	6.35×10^{-3}	5.21×10^{-3}	8.59×10^{-3}	1.63×10^{0}	1.43×10^{-1}	4.89×10^{-3}	$\mathbf{4.16 \times 10^{-3}}$
	Std	3.86×10^{-3}	3.72×10^{-3}	4.48×10^{-3}	1.74×10^{-2}	8.45×10^{-1}	2.24×10^{-1}	2.57×10^{-3}	$\mathbf{3.27 \times 10^{-5}}$
	Wilcoxon	−	−	−	−	−	−	−	
DTLZ7	Average	2.13×10^{-2}	7.45×10^{-3}	1.38×10^{-2}	2.61×10^{-1}	2.16×10^{0}	1.50×10^{-2}	9.40×10^{-3}	$\mathbf{4.98 \times 10^{-3}}$
	Std	8.11×10^{-2}	9.23×10^{-4}	2.15×10^{-3}	2.09×10^{-1}	7.14×10^{-1}	4.98×10^{-3}	1.61×10^{-3}	$\mathbf{2.26 \times 10^{-4}}$
	Wilcoxon	−	−	−	−	−	−	−	
Spacing									
DTLZ1	Average	7.71×10^{-2}	1.95×10^{-2}	8.00×10^{-2}	$\mathbf{7.99 \times 10^{-3}}$	5.15×10^{0}	2.75×10^{0}	5.96×10^{-2}	5.36×10^{-1}
	Std	8.50×10^{-2}	3.84×10^{-2}	1.16×10^{-1}	$\mathbf{4.63 \times 10^{-3}}$	4.63×10^{0}	1.27×10^{0}	6.20×10^{-2}	5.29×10^{-1}
	Wilcoxon	+	+	+	+	−	−	+	
DTLZ2	Average	2.65×10^{-2}	6.60×10^{-3}	6.83×10^{-3}	5.91×10^{-3}	1.05×10^{-2}	7.77×10^{-3}	$\mathbf{4.94 \times 10^{-3}}$	6.37×10^{-3}
	Std	1.67×10^{-3}	5.41×10^{-4}	3.71×10^{-4}	2.83×10^{-4}	1.08×10^{-3}	1.23×10^{-3}	$\mathbf{5.36 \times 10^{-4}}$	1.38×10^{-3}
	Wilcoxon	−	−	−	−	−	−	+	
DTLZ3	Average	4.30×10^{0}	9.49×10^{-1}	1.94×10^{0}	$\mathbf{2.21 \times 10^{-1}}$	7.69×10^{1}	2.01×10^{1}	1.65×10^{0}	6.99×10^{0}
	Std	5.62×10^{0}	7.84×10^{-1}	1.27×10^{0}	$\mathbf{2.83 \times 10^{-1}}$	7.58×10^{1}	2.01×10^{1}	1.77×10^{0}	7.60×10^{0}
	Wilcoxon	+	+	+	+	+	−	−	
DTLZ4	Average	1.08×10^{-2}	5.89×10^{-3}	5.85×10^{-3}	$\mathbf{4.30 \times 10^{-3}}$	9.11×10^{-3}	8.49×10^{-3}	4.51×10^{-3}	6.16×10^{-3}
	Std	1.11×10^{-2}	2.47×10^{-3}	2.70×10^{-3}	$\mathbf{3.24 \times 10^{-3}}$	4.92×10^{-3}	1.54×10^{-3}	1.92×10^{-3}	1.67×10^{-3}
	Wilcoxon	−	=	=	=	−	−	+	
DTLZ5	Average	2.72×10^{-2}	6.67×10^{-3}	6.94×10^{-3}	$\mathbf{5.87 \times 10^{-3}}$	1.02×10^{-2}	7.78×10^{-3}	7.12×10^{-3}	6.07×10^{-3}
	Std	2.37×10^{-3}	6.39×10^{-4}	4.15×10^{-4}	$\mathbf{1.93 \times 10^{-4}}$	1.21×10^{-3}	1.09×10^{-3}	5.44×10^{-4}	1.34×10^{-3}
	Wilcoxon	−	−	−	−	=	−	−	
DTLZ6	Average	1.25×10^{-2}	1.00×10^{-2}	7.01×10^{-3}	7.73×10^{-3}	8.89×10^{-2}	7.92×10^{-2}	5.81×10^{-3}	$\mathbf{3.55 \times 10^{-3}}$
	Std	1.21×10^{-3}	3.54×10^{-3}	4.07×10^{-3}	1.57×10^{-3}	1.89×10^{-1}	8.27×10^{-2}	5.89×10^{-4}	$\mathbf{3.20 \times 10^{-4}}$
	Wilcoxon	−	−	−	−	−	−	−	
DTLZ7	Average	1.10×10^{-2}	7.68×10^{-3}	9.48×10^{-3}	1.72×10^{-2}	9.47×10^{-3}	9.89×10^{-2}	6.34×10^{-3}	$\mathbf{4.14 \times 10^{-3}}$
	Std	1.82×10^{-3}	7.39×10^{-4}	9.46×10^{-4}	1.34×10^{-2}	4.92×10^{-3}	3.17×10^{-1}	9.36×10^{-4}	$\mathbf{3.88 \times 10^{-4}}$
	Wilcoxon	−	−	−	−	−	−	−	

Table 5. *Cont.*

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
Spread									
DTLZ1	Average	1.40×10^{0}	$\mathbf{6.06 \times 10^{-1}}$	8.74×10^{-1}	1.07×10^{0}	1.15×10^{0}	9.03×10^{-1}	8.37×10^{-1}	8.25×10^{-1}
	Std	3.84×10^{-1}	$\mathbf{3.54 \times 10^{-1}}$	1.98×10^{-1}	1.94×10^{-1}	2.85×10^{-1}	1.47×10^{-1}	2.66×10^{-1}	2.58×10^{-1}
	Wilcoxon	–	+	=	–	–	–	–	
DTLZ2	Average	6.99×10^{-1}	3.73×10^{-1}	2.44×10^{-1}	$\mathbf{1.85 \times 10^{-1}}$	7.88×10^{-1}	3.52×10^{-1}	2.17×10^{-1}	2.19×10^{-1}
	Std	2.54×10^{-2}	4.40×10^{-2}	2.08×10^{-2}	$\mathbf{1.00 \times 10^{-2}}$	5.33×10^{-2}	4.17×10^{-2}	2.32×10^{-2}	3.09×10^{-2}
	Wilcoxon	–	–	–	+	–	–	=	
DTLZ3	Average	1.34×10^{0}	1.02×10^{0}	1.00×10^{0}	1.07×10^{0}	1.16×10^{0}	$\mathbf{8.51 \times 10^{-1}}$	1.01×10^{0}	9.92×10^{-1}
	Std	3.14×10^{-1}	1.06×10^{-1}	1.11×10^{-1}	4.23×10^{-2}	2.87×10^{-1}	$\mathbf{1.54 \times 10^{-1}}$	1.22×10^{-1}	2.42×10^{-1}
	Wilcoxon	–	–	=	–	–	+	–	
DTLZ4	Average	8.11×10^{-1}	4.72×10^{-1}	3.83×10^{-1}	5.70×10^{-1}	8.10×10^{-1}	3.71×10^{-1}	3.33×10^{-1}	$\mathbf{2.32 \times 10^{-1}}$
	Std	1.97×10^{-1}	2.16×10^{-1}	2.82×10^{-1}	4.56×10^{-1}	1.15×10^{-1}	4.57×10^{-2}	2.73×10^{-1}	$\mathbf{3.54 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ5	Average	6.99×10^{-1}	3.73×10^{-1}	2.50×10^{-1}	$\mathbf{1.84 \times 10^{-1}}$	7.69×10^{-1}	3.53×10^{-1}	2.22×10^{-1}	2.21×10^{-1}
	Std	3.20×10^{-2}	5.03×10^{-2}	2.48×10^{-2}	$\mathbf{7.03 \times 10^{-3}}$	6.20×10^{-2}	3.74×10^{-2}	2.50×10^{-2}	2.43×10^{-2}
	Wilcoxon	–	–	–	+	–	–	=	
DTLZ6	Average	9.50×10^{-1}	7.33×10^{-1}	2.66×10^{-1}	4.34×10^{-1}	1.01×10^{0}	7.25×10^{-1}	2.16×10^{-1}	$\mathbf{1.47 \times 10^{-1}}$
	Std	6.90×10^{-2}	1.57×10^{-1}	2.33×10^{-1}	2.68×10^{-1}	1.76×10^{-1}	3.27×10^{-1}	2.28×10^{-2}	$\mathbf{1.46 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ7	Average	5.28×10^{-1}	4.21×10^{-1}	5.45×10^{-1}	8.99×10^{-1}	9.48×10^{-1}	5.14×10^{-1}	2.94×10^{-1}	$\mathbf{1.47 \times 10^{-1}}$
	Std	9.17×10^{-2}	5.01×10^{-2}	5.04×10^{-2}	1.48×10^{-1}	3.92×10^{-2}	3.39×10^{-1}	2.92×10^{-2}	$\mathbf{1.47 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	



(**a**) DTLZ2 (**b**) DTLZ4 (**c**) DTLZ5

(**d**) DTLZ6 (**e**) DTLZ7

Figure 10. Pareto fronts found by the proposed algorithms in bi-objective DTLZ.

The mean and standard deviation of the IGD, SP, and Δ values for IETMFO and the other comparative algorithms on tri-dimensional DTLZ problems are presented in Table 6, with illustrative examples of these results shown in Figure 11. Overall, it is evident from Table 6 that the proposed algorithm exhibits high competitiveness with respect to IGD on tri-dimensional DTLZ problems. Among the seven test problems, IETMFO attains the optimal IGD values for two—specifically, DTLZ6 and DTLZ7. Additionally, since the true Pareto front of DTLZ6 degenerates into a curved shape, this makes it an ideal case for visually demonstrating algorithm performance. Meanwhile, DTLZ7 features 2^{M-1} disconnected Pareto-optimal regions, enabling an evaluation of the algorithms' ability to preserve a well-distributed set of optimal individuals across distinct Pareto-optimal

fronts. The Pareto fronts of the proposed algorithm and comparative algorithms on the DTLZ6 problem are depicted in Figure 11, from which it can be seen that the Pareto front of the proposed algorithm is broadly distributed across the solution space, demonstrating outstanding performance in terms of solution distribution. Due to the multi-modal nature of DTLZ1 and DTLZ3 problems, it is challenging to converge effectively to the correct Pareto front, resulting in relatively high IGD values. The reasons for this phenomenon align with what was previously discussed regarding the bi-dimensional DTLZ1 and DTLZ3 problems. Based on the SP values presented in Table 6, the best values for the DTLZ2, DTLZ6, and DTLZ7 problems are achieved by the proposed algorithm. Furthermore, except for DTLZ3, the best Δ values are obtained by the proposed algorithm on the 3-dimensional DTLZ problems, indicating that it demonstrates a highly effective distribution of solutions in the solution space for the 3-dimensional DTLZ problems. According to the Wilcoxon rank-sum test for the IGD metric, the results indicate that the IETMFO showed statistical differences from all algorithms in all DTLZ problems to some extent. Meanwhile, for the SP measure, there are statistical differences between IETMFO and the other compared algorithms, whereas for the Δ measure, the proposed algorithm outperformed the others in all three-objective DTLZ test problems, except for the statistically similar results to those of MPSOD for DTLZ3. Overall, when compared to existing multi-objective algorithms, the proposed multi-objective algorithm exhibits outstanding performance in solving the three-dimensional DTLZ problems.

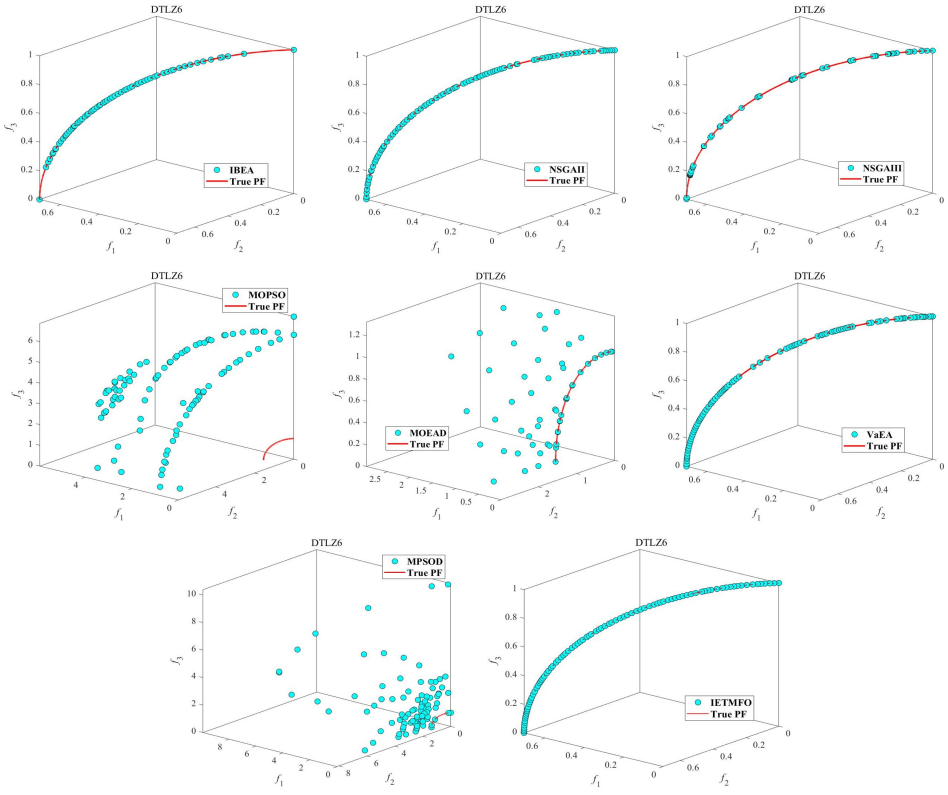


Figure 11. Pareto fronts found by the proposed algorithms in tri-objective DTLZ6 problems.

Table 7 discloses the statistical findings of the IGD, SP, and Δ metrics, including the average and standard deviation for the IMOP test functions, which have complex true Pareto fronts, as well as the Pareto fronts shown in Figure 12. From Table 7, it can be seen that IETMFO achieves the best values for the IGD in four problems, i.e., IMOP2, IMOP4, IMOP6, and IMOP7. Meanwhile, the proposed algorithm outperforms other algorithms by an order of magnitude in the IMOP2 and IMOP4 problems. Furthermore, according

to the Δ values shown in Table 7, it obtained the best values for Δ in four problems, i.e., IMOP4, IMOP6, IMOP7, and IMOP8, which indicates that IETMFO has a significant advantage in terms of convergence and the diversity of solutions. According to the results of Wilcoxon rank-sum test for the IGD metric, IETMFO showed statistical differences in most of the IMOP problems except for MOPSO on IMOP3, IBEA on IMOP4, and NSGAII on IMOP8. Meanwhile, for the Δ measure, the IETMFO outperformed most of the algorithms in IMOP4-8 except for VaEA in IMOP5. Nevertheless, noticeable variations can be observed in the SP results, which characterize the uniformity of inter-solution distances across the Pareto front. A lower SP value signifies a more evenly distributed solution set. The Pareto fronts generated by the proposed algorithm for all test cases are depicted in Figure 12, from which it can be seen that IMOP test problems feature irregular true Pareto fronts—including convex/concave curves with sharp tails, wavy lines in three-dimensional space, and multiple disconnected circular regions, among others. As can be seen from Figure 12, when the proposed algorithm is applied to solve these problems with irregular Pareto fronts, it is able to generate a solution set with comparatively good distribution characteristics. In conclusion, given the highly complex and irregular nature of the Pareto fronts in IMOP problems, the results presented above demonstrate that IETMFO outperforms the comparative algorithms when dealing with Pareto fronts of such unique shapes.

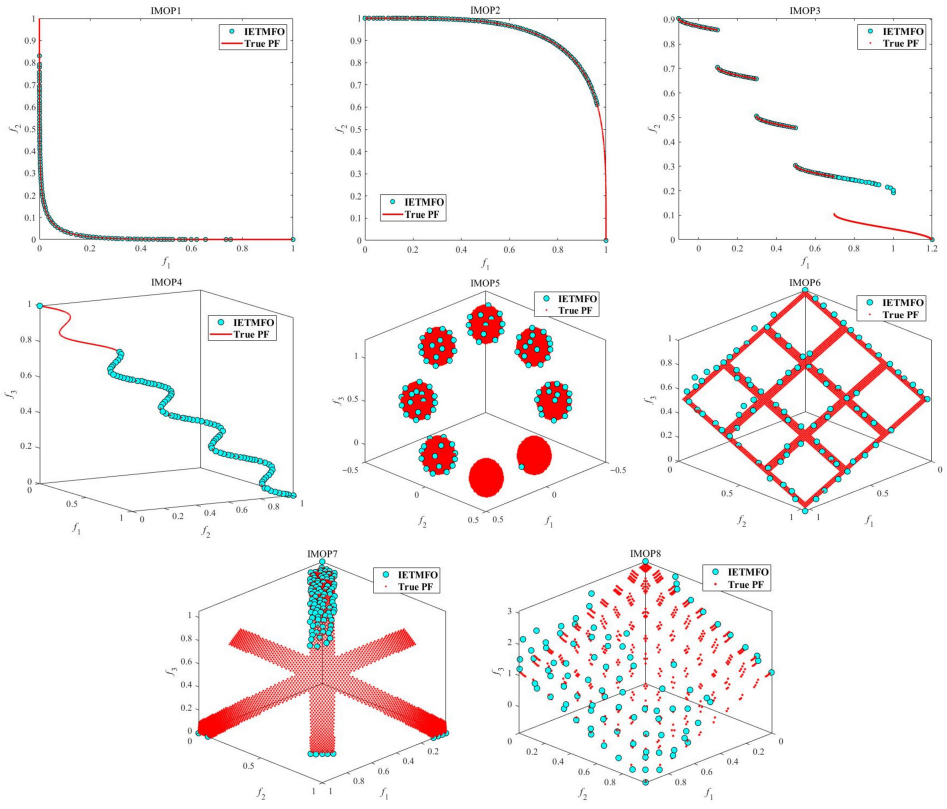


Figure 12. Pareto fronts found by the proposed algorithms in IMOP problems.

The results of the FON, KUR, and SCH problems are listed in Table 8, and an example of the results is illustrated in Figure 13. In terms of IGD values, based on the test data from Table 8, except for the FON1 problem—where IETMFO lags behind NSGAIII—IETMFO achieves the best values for the other four test problems. According to the SP values in Table 8, except for the SCH2 problem—where IETMFO falls behind the highly uniform MOEAD algorithm—IETMFO obtained the best values for the remaining four test problems. Based on the Δ values, IETMFO attained the best values for the Δ measure in all five test problems. According to the SP and Δ measures, the Wilcoxon rank-sum

test results demonstrate that IETMFO outperformed most of the algorithms except for MOEAD on SCH2. For the IGD metric, the results of the Wilcoxon rank-sum test indicate that IETMFO showed statistical differences from all algorithms in KUR and SCH problems to some extent, as well as similar statistical results to NSGAIII and VaEA on FON1 and to IBEA and NSGAIII on FON2. This indicates that the proposed algorithm exhibits exceptionally high distribution diversity, and the obtained solution distribution is remarkable. Considering Figure 13, the distribution of the Pareto front obtained by IETMFO is also highly uniform, and the convergence performance is outstanding. In general, compared to other algorithms, the comprehensive performance of the proposed multi-objective algorithm is remarkably superior.

Table 6. The results of the proposed algorithm and the compared algorithms on tri-objective DTLZ problems.

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
IGD									
DTLZ1	Average	2.43×10^{-1}	2.46×10^{-1}	$\mathbf{2.33 \times 10^{-1}}$	3.59×10^{-1}	1.15×10^{1}	6.87×10^{0}	2.47×10^{-1}	1.88×10^{1}
	Std	1.19×10^{-3}	1.96×10^{-3}	$\mathbf{4.56 \times 10^{-3}}$	6.40×10^{-2}	2.47×10^{-1}	1.12×10^{-2}	6.08×10^{-3}	3.53×10^{0}
	Wilcoxon	+	+	+	+	+	+	+	
DTLZ2	Average	8.06×10^{-2}	6.96×10^{-2}	5.49×10^{-2}	$\mathbf{5.48 \times 10^{-2}}$	1.01×10^{-1}	6.67×10^{-2}	5.87×10^{-2}	6.01×10^{-2}
	Std	2.18×10^{-3}	2.21×10^{-3}	1.75×10^{-4}	$\mathbf{1.86 \times 10^{-4}}$	1.28×10^{-2}	5.48×10^{-4}	7.19×10^{-4}	1.78×10^{-3}
	Wilcoxon	−	−	+	+	−	−	=	
DTLZ3	Average	$\mathbf{6.53 \times 10^{0}}$	6.68×10^{0}	1.02×10^{1}	1.52×10^{1}	1.56×10^{2}	1.17×10^{2}	9.12×10^{0}	1.83×10^{2}
	Std	$\mathbf{3.55 \times 10^{0}}$	4.22×10^{0}	5.49×10^{0}	1.14×10^{1}	6.63×10^{1}	1.39×10^{1}	3.15×10^{0}	1.50×10^{1}
	Wilcoxon	+	+	+	+	+	+	+	
DTLZ4	Average	7.98×10^{-2}	2.07×10^{-1}	1.36×10^{-1}	4.58×10^{-1}	3.90×10^{-1}	$\mathbf{6.28 \times 10^{-2}}$	1.47×10^{-1}	7.29×10^{-2}
	Std	2.17×10^{-3}	2.89×10^{-1}	1.84×10^{-1}	3.49×10^{-1}	1.75×10^{-1}	$\mathbf{4.01 \times 10^{-3}}$	2.49×10^{-1}	2.42×10^{-3}
	Wilcoxon	−	−	−	−	−	+	−	
DTLZ5	Average	1.62×10^{-2}	6.07×10^{-3}	1.37×10^{-2}	3.24×10^{-2}	1.44×10^{-2}	5.43×10^{-2}	$\mathbf{5.61 \times 10^{-3}}$	1.25×10^{-2}
	Std	1.41×10^{-3}	3.57×10^{-4}	2.17×10^{-3}	6.64×10^{-4}	3.69×10^{-3}	4.56×10^{-3}	$\mathbf{2.37 \times 10^{-4}}$	1.30×10^{-3}
	Wilcoxon	−	+	−	−	−	−	+	
DTLZ6	Average	2.79×10^{-2}	5.91×10^{-3}	2.00×10^{-2}	1.10×10^{-1}	2.79×10^{0}	2.61×10^{-1}	2.35×10^{-2}	$\mathbf{4.59 \times 10^{-3}}$
	Std	4.63×10^{-3}	3.89×10^{-4}	2.48×10^{-3}	2.39×10^{-1}	7.87×10^{-1}	2.57×10^{-1}	9.81×10^{-2}	$\mathbf{3.62 \times 10^{-5}}$
	Wilcoxon	−	−	−	−	−	−	−	
DTLZ7	Average	9.30×10^{-2}	1.02×10^{-1}	9.85×10^{-2}	2.00×10^{-1}	4.22×10^{0}	2.93×10^{-1}	8.09×10^{-2}	$\mathbf{6.58 \times 10^{-2}}$
	Std	7.64×10^{-2}	5.06×10^{-2}	7.05×10^{-3}	1.69×10^{-1}	1.23×10^{0}	9.74×10^{-2}	5.53×10^{-3}	$\mathbf{9.38 \times 10^{-2}}$
	Wilcoxon	−	−	−	−	−	−	−	
Spacing									
DTLZ1	Average	$\mathbf{5.02 \times 10^{-2}}$	5.05×10^{-2}	7.06×10^{-2}	7.86×10^{-2}	6.42×10^{0}	2.57×10^{0}	2.80×10^{0}	1.71×10^{0}
	Std	$\mathbf{3.79 \times 10^{-2}}$	2.65×10^{-2}	3.97×10^{-2}	9.72×10^{-2}	2.03×10^{0}	5.03×10^{-1}	4.46×10^{0}	7.71×10^{-1}
	Wilcoxon	+	+	+	+	−	−	−	
DTLZ2	Average	6.64×10^{-2}	5.96×10^{-2}	5.75×10^{-2}	5.61×10^{-2}	5.81×10^{-2}	5.33×10^{-2}	3.86×10^{-2}	$\mathbf{2.84 \times 10^{-2}}$
	Std	4.80×10^{-3}	4.21×10^{-3}	1.46×10^{-3}	9.39×10^{-4}	9.30×10^{-3}	3.73×10^{-3}	3.71×10^{-3}	$\mathbf{5.77 \times 10^{-3}}$
	Wilcoxon	−	−	−	−	−	−	−	
DTLZ3	Average	1.94×10^{0}	$\mathbf{1.25 \times 10^{0}}$	1.54×10^{0}	1.98×10^{0}	6.21×10^{1}	2.19×10^{1}	1.39×10^{0}	1.46×10^{1}
	Std	2.48×10^{0}	$\mathbf{2.95 \times 10^{0}}$	8.08×10^{-1}	1.17×10^{0}	2.90×10^{1}	6.73×10^{0}	1.20×10^{0}	9.02×10^{0}
	Wilcoxon	+	+	+	+	−	−	+	
DTLZ4	Average	6.52×10^{-2}	4.96×10^{-2}	4.89×10^{-2}	$\mathbf{2.45 \times 10^{-2}}$	5.43×10^{-2}	5.08×10^{-2}	3.36×10^{-2}	3.07×10^{-2}
	Std	4.22×10^{-3}	2.23×10^{-2}	1.72×10^{-2}	$\mathbf{2.27 \times 10^{-2}}$	3.59×10^{-2}	4.30×10^{-3}	1.23×10^{-2}	3.86×10^{-3}
	Wilcoxon	−	−	−	=	−	−	−	
DTLZ5	Average	1.76×10^{-2}	9.52×10^{-3}	1.54×10^{-2}	1.53×10^{-2}	1.36×10^{-2}	1.44×10^{-1}	$\mathbf{7.54 \times 10^{-3}}$	1.33×10^{-2}
	Std	7.38×10^{-3}	7.81×10^{-4}	3.22×10^{-3}	6.99×10^{-3}	2.30×10^{-3}	6.30×10^{-2}	$\mathbf{6.11 \times 10^{-4}}$	3.68×10^{-3}
	Wilcoxon	−	+	−	−	−	−	+	
DTLZ6	Average	1.55×10^{-2}	1.11×10^{-2}	1.88×10^{-2}	3.36×10^{-2}	2.36×10^{-1}	3.18×10^{-1}	1.04×10^{-2}	$\mathbf{5.10 \times 10^{-3}}$
	Std	2.08×10^{-3}	8.67×10^{-4}	2.26×10^{-2}	5.26×10^{-2}	6.90×10^{-2}	1.51×10^{-1}	8.69×10^{-3}	$\mathbf{5.12 \times 10^{-4}}$
	Wilcoxon	−	−	−	−	−	−	−	
DTLZ7	Average	6.69×10^{-2}	6.93×10^{-2}	6.89×10^{-2}	1.72×10^{-1}	3.98×10^{-2}	6.16×10^{-1}	6.29×10^{-2}	$\mathbf{3.34 \times 10^{-2}}$
	Std	1.14×10^{-2}	9.60×10^{-3}	8.36×10^{-3}	4.09×10^{-2}	2.48×10^{-2}	5.63×10^{-1}	6.54×10^{-3}	$\mathbf{8.39 \times 10^{-3}}$
	Wilcoxon	−	−	−	−	−	−	−	

Table 6. *Cont.*

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
Spread									
DTLZ1	Average	1.31×10^{0}	5.92×10^{-1}	6.44×10^{-1}	6.54×10^{-1}	7.81×10^{-1}	5.83×10^{-1}	1.11×10^{0}	$\mathbf{5.06 \times 10^{-1}}$
	Std	2.43×10^{-1}	1.13×10^{-1}	2.37×10^{-1}	3.32×10^{-1}	1.96×10^{-1}	7.54×10^{-2}	6.30×10^{-1}	$\mathbf{2.94 \times 10^{-1}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ2	Average	4.56×10^{-1}	5.33×10^{-1}	1.85×10^{-1}	1.72×10^{-1}	3.99×10^{-1}	2.26×10^{-1}	1.63×10^{-1}	$\mathbf{1.17 \times 10^{-1}}$
	Std	3.16×10^{-2}	3.96×10^{-2}	7.19×10^{-3}	4.35×10^{-3}	3.18×10^{-2}	2.11×10^{-2}	1.78×10^{-2}	$\mathbf{1.49 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ3	Average	1.15×10^{0}	9.70×10^{-1}	9.45×10^{-1}	7.62×10^{-1}	1.01×10^{0}	$\mathbf{5.89 \times 10^{-1}}$	9.43×10^{-1}	6.40×10^{-1}
	Std	1.74×10^{-1}	1.81×10^{-1}	1.15×10^{-1}	2.31×10^{-1}	1.72×10^{-1}	$\mathbf{7.02 \times 10^{-2}}$	1.04×10^{-1}	3.23×10^{-1}
	Wilcoxon	–	–	–	–	–	=	–	
DTLZ4	Average	4.48×10^{-1}	5.65×10^{-1}	3.14×10^{-1}	7.01×10^{-1}	7.29×10^{-1}	2.46×10^{-1}	2.58×10^{-1}	$\mathbf{1.29 \times 10^{-1}}$
	Std	3.32×10^{-2}	1.76×10^{-1}	2.94×10^{-1}	4.01×10^{-1}	1.80×10^{-1}	3.12×10^{-2}	2.42×10^{-1}	$\mathbf{1.93 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ5	Average	5.62×10^{-1}	4.80×10^{-1}	9.06×10^{-1}	1.80×10^{0}	8.25×10^{-1}	7.31×10^{-1}	3.02×10^{-1}	$\mathbf{2.23 \times 10^{-1}}$
	Std	7.05×10^{-2}	6.03×10^{-2}	9.03×10^{-2}	9.93×10^{-2}	8.97×10^{-2}	1.48×10^{-1}	2.42×10^{-2}	$\mathbf{3.06 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ6	Average	9.58×10^{-1}	6.87×10^{-1}	1.37×10^{0}	1.51×10^{0}	7.05×10^{-1}	7.85×10^{-1}	3.48×10^{-1}	$\mathbf{1.52 \times 10^{-1}}$
	Std	9.48×10^{-2}	8.25×10^{-2}	1.11×10^{-1}	3.58×10^{-1}	1.31×10^{-1}	1.89×10^{-1}	1.88×10^{-1}	$\mathbf{1.81 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
DTLZ7	Average	5.08×10^{-1}	4.91×10^{-1}	5.65×10^{-1}	1.12×10^{0}	8.26×10^{-1}	7.20×10^{-1}	3.51×10^{-1}	$\mathbf{1.42 \times 10^{-1}}$
	Std	5.19×10^{-2}	5.07×10^{-2}	6.20×10^{-2}	6.95×10^{-2}	1.13×10^{-1}	2.38×10^{-1}	2.77×10^{-2}	$\mathbf{7.56 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	

Table 7. The results of the proposed algorithm and the compared algorithms on IMOP problems.

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
IGD									
IMOP1	Average	1.88×10^{-1}	$\mathbf{2.77 \times 10^{-2}}$	2.18×10^{-1}	3.66×10^{-1}	6.45×10^{-1}	1.54×10^{-1}	2.11×10^{-1}	4.55×10^{-2}
	Std	3.52×10^{-2}	$\mathbf{2.98 \times 10^{-2}}$	7.22×10^{-2}	7.01×10^{-3}	2.64×10^{-1}	1.79×10^{-2}	7.08×10^{-2}	1.10×10^{-2}
	Wilcoxon	–	+	–	–	–	–	–	
IMOP2	Average	5.98×10^{-1}	4.38×10^{-1}	4.82×10^{-1}	7.85×10^{-1}	4.68×10^{-1}	6.88×10^{-1}	4.97×10^{-1}	$\mathbf{9.39 \times 10^{-2}}$
	Std	1.57×10^{-1}	1.57×10^{-1}	7.91×10^{-2}	1.54×10^{-4}	1.87×10^{-1}	7.71×10^{-2}	9.27×10^{-2}	$\mathbf{7.21 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
IMOP3	Average	$\mathbf{4.54 \times 10^{-2}}$	3.23×10^{-1}	4.95×10^{-1}	5.41×10^{-1}	3.63×10^{-1}	1.66×10^{-1}	5.13×10^{-1}	6.57×10^{-2}
	Std	$\mathbf{3.06 \times 10^{-2}}$	2.03×10^{-1}	7.52×10^{-2}	8.55×10^{-2}	3.42×10^{-1}	2.47×10^{-2}	5.06×10^{-2}	2.06×10^{-2}
	Wilcoxon	+	–	–	–	=	–	–	
IMOP4	Average	1.59×10^{-1}	2.66×10^{-1}	3.81×10^{-1}	5.86×10^{-1}	3.16×10^{-1}	2.89×10^{-1}	5.72×10^{-1}	$\mathbf{4.34 \times 10^{-2}}$
	Std	1.85×10^{-1}	2.13×10^{-1}	2.07×10^{-1}	1.62×10^{-1}	3.12×10^{-1}	1.05×10^{-1}	1.70×10^{-1}	$\mathbf{6.56 \times 10^{-3}}$
	Wilcoxon	=	–	–	–	–	–	–	
IMOP5	Average	$\mathbf{5.64 \times 10^{-2}}$	6.04×10^{-2}	7.30×10^{-2}	5.49×10^{-1}	6.31×10^{-1}	2.92×10^{-1}	1.04×10^{-1}	9.94×10^{-2}
	Std	$\mathbf{1.62 \times 10^{-2}}$	1.50×10^{-2}	1.84×10^{-2}	1.10×10^{-1}	1.50×10^{-1}	3.20×10^{-2}	4.71×10^{-2}	1.36×10^{-2}
	Wilcoxon	+	+	+	–	–	–	–	
IMOP6	Average	7.15×10^{-2}	7.68×10^{-2}	1.44×10^{-1}	2.87×10^{-1}	5.12×10^{-1}	5.33×10^{-1}	1.07×10^{-1}	$\mathbf{4.48 \times 10^{-2}}$
	Std	1.14×10^{-1}	6.08×10^{-2}	1.66×10^{-1}	2.18×10^{-1}	8.49×10^{-2}	8.27×10^{-3}	1.41×10^{-1}	$\mathbf{2.88 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
IMOP7	Average	9.29×10^{-1}	8.29×10^{-1}	9.04×10^{-1}	9.39×10^{-1}	9.01×10^{-1}	8.25×10^{-1}	9.09×10^{-1}	$\mathbf{1.25 \times 10^{-1}}$
	Std	2.96×10^{-2}	9.23×10^{-4}	3.48×10^{-2}	6.30×10^{-5}	8.29×10^{-2}	2.31×10^{-1}	1.83×10^{-2}	$\mathbf{3.82 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
IMOP8	Average	$\mathbf{7.81 \times 10^{-2}}$	1.23×10^{-1}	1.67×10^{-1}	1.06×10^{0}	4.63×10^{-1}	4.96×10^{-1}	1.12×10^{-1}	1.47×10^{-1}
	Std	$\mathbf{3.25 \times 10^{-3}}$	3.03×10^{-2}	1.45×10^{-1}	7.16×10^{-3}	1.01×10^{-1}	1.33×10^{-1}	4.75×10^{-2}	9.30×10^{-3}
	Wilcoxon	+	=	–	–	–	–	+	
Spacing									
IMOP1	Average	$\mathbf{1.65 \times 10^{-2}}$	2.87×10^{-2}	3.90×10^{-2}	3.25×10^{-2}	4.08×10^{-1}	2.19×10^{-1}	4.25×10^{-2}	4.56×10^{-2}
	Std	$\mathbf{1.60 \times 10^{-2}}$	1.45×10^{-2}	2.44×10^{-2}	2.48×10^{-3}	2.42×10^{-2}	8.79×10^{-2}	2.68×10^{-2}	8.51×10^{-3}
	Wilcoxon	+	+	+	=	–	–	=	
IMOP2	Average	3.46×10^{-3}	1.06×10^{-2}	2.59×10^{-3}	$\mathbf{4.13 \times 10^{-3}}$	2.09×10^{-2}	2.18×10^{-2}	2.01×10^{-3}	6.87×10^{-2}
	Std	4.14×10^{-3}	2.54×10^{-2}	1.51×10^{-3}	$\mathbf{8.82 \times 10^{-3}}$	4.01×10^{-2}	2.27×10^{-2}	8.76×10^{-4}	2.23×10^{-2}
	Wilcoxon	+	+	+	+	+	+	+	
IMOP3	Average	6.33×10^{-2}	7.64×10^{-2}	9.63×10^{-3}	$\mathbf{6.48 \times 10^{-3}}$	2.79×10^{-2}	2.84×10^{-1}	2.06×10^{-2}	8.64×10^{-2}
	Std	2.74×10^{-2}	4.50×10^{-2}	5.10×10^{-3}	$\mathbf{2.05 \times 10^{-3}}$	2.89×10^{-2}	6.12×10^{-2}	3.59×10^{-2}	1.99×10^{-2}
	Wilcoxon	=	=	+	+	+	–	+	

Table 7. *Cont.*

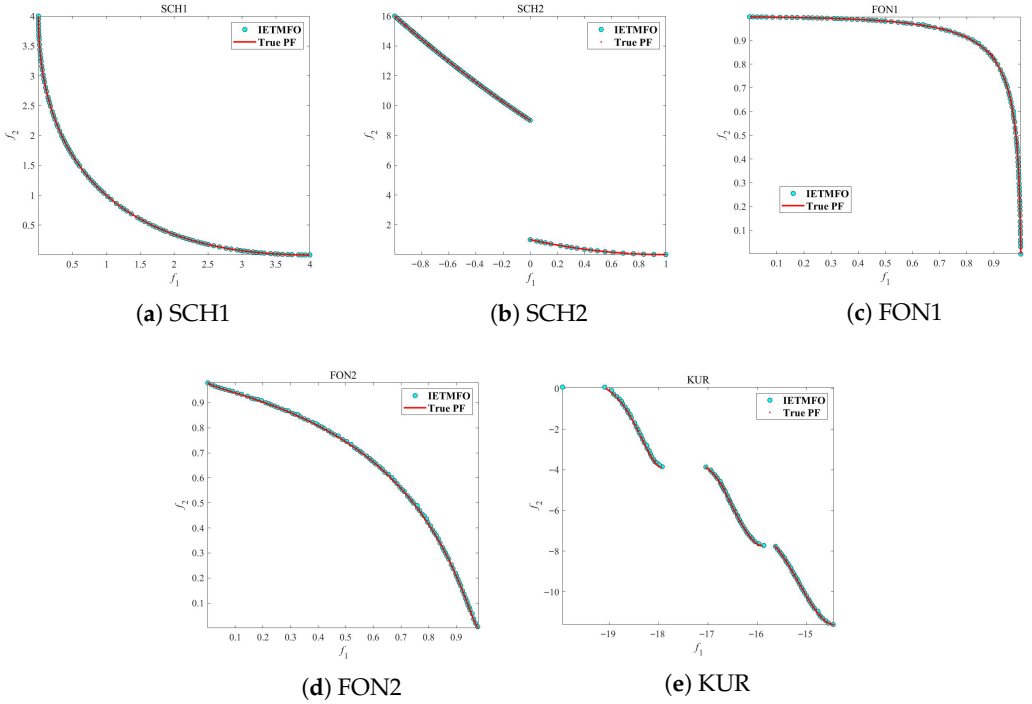
		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
IMOP4	Average	4.68×10^{-2}	6.45×10^{-2}	7.24×10^{-2}	$\mathbf{1.08 \times 10^{-2}}$	7.57×10^{-2}	6.31×10^{-1}	1.42×10^{-2}	2.71×10^{-2}
	Std	4.36×10^{-2}	4.52×10^{-2}	3.97×10^{-2}	$\mathbf{6.36 \times 10^{-3}}$	5.82×10^{-2}	1.86×10^{-1}	3.65×10^{-2}	8.36×10^{-3}
	Wilcoxon	+	=	=	+	=	−	=	
IMOP5	Average	3.76×10^{-2}	3.98×10^{-2}	4.66×10^{-2}	2.66×10^{-2}	$\mathbf{8.51 \times 10^{-3}}$	1.09×10^{-1}	3.37×10^{-2}	2.98×10^{-2}
	Std	3.88×10^{-3}	5.69×10^{-3}	1.00×10^{-2}	6.77×10^{-3}	$\mathbf{2.94 \times 10^{-2}}$	7.44×10^{-2}	1.08×10^{-2}	3.89×10^{-2}
	Wilcoxon	−	−	−	+	+	−	−	
IMOP6	Average	3.82×10^{-2}	5.35×10^{-2}	4.20×10^{-2}	5.23×10^{-2}	$\mathbf{2.04 \times 10^{-2}}$	7.75×10^{-2}	3.21×10^{-2}	2.38×10^{-2}
	Std	8.38×10^{-3}	1.34×10^{-2}	2.86×10^{-2}	1.83×10^{-2}	$\mathbf{2.20 \times 10^{-2}}$	4.08×10^{-2}	9.12×10^{-3}	4.42×10^{-3}
	Wilcoxon	−	−	−	−	+	−	−	
IMOP7	Average	9.21×10^{-4}	6.63×10^{-3}	9.61×10^{-4}	$\mathbf{1.19 \times 10^{-5}}$	4.19×10^{-2}	4.41×10^{-2}	7.37×10^{-4}	6.35×10^{-2}
	Std	2.12×10^{-3}	1.79×10^{-2}	1.57×10^{-3}	$\mathbf{2.01 \times 10^{-5}}$	2.55×10^{-3}	9.37×10^{-2}	5.45×10^{-4}	5.60×10^{-2}
	Wilcoxon	+	+	+	+	=	=	+	
IMOP8	Average	9.67×10^{-2}	9.09×10^{-2}	8.89×10^{-2}	6.19×10^{-2}	7.50×10^{-2}	1.89×10^{-1}	7.43×10^{-2}	$\mathbf{5.52 \times 10^{-2}}$
	Std	9.47×10^{-3}	1.05×10^{-2}	2.35×10^{-2}	4.97×10^{-2}	3.12×10^{-2}	6.40×10^{-2}	6.90×10^{-3}	$\mathbf{9.15 \times 10^{-3}}$
	Wilcoxon	−	−	−	−	−	−	−	
Spread									
IMOP1	Average	9.97×10^{-1}	$\mathbf{6.99 \times 10^{-1}}$	1.17×10^{0}	9.64×10^{-1}	1.02×10^{0}	1.31×10^{0}	1.18×10^{0}	7.77×10^{-1}
	Std	7.60×10^{-2}	$\mathbf{1.11 \times 10^{-1}}$	1.79×10^{-1}	2.29×10^{-2}	7.10×10^{-2}	7.74×10^{-2}	1.97×10^{-1}	7.92×10^{-2}
	Wilcoxon	−	+	−	−	−	−	−	
IMOP2	Average	9.78×10^{-1}	9.37×10^{-1}	9.34×10^{-1}	1.00×10^{0}	1.05×10^{0}	1.02×10^{0}	$\mathbf{8.93 \times 10^{-1}}$	9.76×10^{-1}
	Std	3.21×10^{-2}	7.12×10^{-2}	2.03×10^{-2}	1.80×10^{-5}	1.58×10^{-1}	3.93×10^{-2}	$\mathbf{4.49 \times 10^{-2}}$	1.65×10^{-1}
	Wilcoxon	=	=	=	=	=	−	+	
IMOP3	Average	1.24×10^{0}	1.05×10^{0}	9.84×10^{-1}	9.71×10^{-1}	1.02×10^{0}	1.38×10^{0}	$\mathbf{9.53 \times 10^{-1}}$	1.02×10^{0}
	Std	1.51×10^{-1}	2.37×10^{-1}	3.49×10^{-2}	4.15×10^{-2}	1.30×10^{-1}	1.49×10^{-1}	$\mathbf{7.45 \times 10^{-2}}$	2.46×10^{-1}
	Wilcoxon	−	=	+	+	=	−	+	
IMOP4	Average	7.44×10^{-1}	8.21×10^{-1}	1.00×10^{0}	1.02×10^{0}	1.05×10^{0}	1.38×10^{0}	8.82×10^{-1}	$\mathbf{4.89 \times 10^{-1}}$
	Std	1.56×10^{-1}	1.07×10^{-1}	1.27×10^{-1}	2.38×10^{-2}	9.58×10^{-2}	2.17×10^{-1}	6.89×10^{-2}	$\mathbf{4.66 \times 10^{-2}}$
	Wilcoxon	−	−	−	−	−	−	−	
IMOP5	Average	4.34×10^{-1}	5.14×10^{-1}	7.59×10^{-1}	1.05×10^{0}	9.58×10^{-1}	7.31×10^{-1}	$\mathbf{2.73 \times 10^{-1}}$	2.86×10^{-1}
	Std	5.03×10^{-2}	4.13×10^{-2}	7.41×10^{-2}	6.64×10^{-2}	1.48×10^{-1}	1.79×10^{-1}	$\mathbf{5.25 \times 10^{-2}}$	2.76×10^{-1}
	Wilcoxon	−	−	−	−	−	−	+	
IMOP6	Average	4.98×10^{-1}	6.38×10^{-1}	8.24×10^{-1}	9.94×10^{-1}	8.65×10^{-1}	8.81×10^{-1}	3.56×10^{-1}	$\mathbf{1.69 \times 10^{-1}}$
	Std	1.24×10^{-1}	6.90×10^{-2}	1.06×10^{-1}	1.91×10^{-1}	1.08×10^{-1}	4.80×10^{-2}	1.43×10^{-1}	$\mathbf{2.19 \times 10^{-2}}$
	Wilcoxon	−	−	−	−	−	−	−	
IMOP7	Average	1.00×10^{0}	9.59×10^{-1}	9.91×10^{-1}	1.00×10^{0}	9.86×10^{-1}	1.01×10^{0}	9.861×10^{-1}	$\mathbf{5.03 \times 10^{-1}}$
	Std	6.10×10^{-3}	4.64×10^{-2}	2.40×10^{-2}	3.60×10^{-5}	4.22×10^{-2}	6.59×10^{-2}	2.62×10^{-2}	$\mathbf{4.17 \times 10^{-1}}$
	Wilcoxon	−	−	−	−	−	−	−	
IMOP8	Average	4.08×10^{-1}	4.69×10^{-1}	6.60×10^{-1}	1.08×10^{0}	5.96×10^{-1}	7.19×10^{-1}	3.01×10^{-1}	$\mathbf{1.92 \times 10^{-1}}$
	Std	1.19×10^{-1}	5.43×10^{-2}	1.64×10^{-1}	2.25×10^{-2}	1.04×10^{-1}	9.44×10^{-2}	7.76×10^{-2}	$\mathbf{3.03 \times 10^{-2}}$
	Wilcoxon	−	−	−	−	−	−	−	

Table 8. The results of the proposed algorithm and the compared algorithms on FON, KUR, and SCH problems.

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
IGD									
FON1	Average	6.31×10^{-1}	6.28×10^{-1}	$\mathbf{6.19 \times 10^{-1}}$	6.28×10^{-1}	7.13×10^{-1}	6.39×10^{-1}	6.25×10^{-1}	6.20×10^{-1}
	Std	1.84×10^{-3}	2.78×10^{-5}	$\mathbf{5.47 \times 10^{-6}}$	9.36×10^{-5}	4.43×10^{-5}	9.30×10^{-6}	1.20×10^{-5}	7.53×10^{-6}
	Wilcoxon	−	−	=	−	−	−	=	
FON2	Average	1.28×10^{-1}	1.22×10^{-1}	1.15×10^{-1}	1.12×10^{-1}	1.41×10^{-1}	1.33×10^{-1}	1.12×10^{-1}	$\mathbf{1.08 \times 10^{-1}}$
	Std	7.28×10^{-5}	1.16×10^{-4}	9.57×10^{-5}	1.49×10^{-4}	7.18×10^{-4}	2.80×10^{-5}	1.77×10^{-4}	$\mathbf{9.12 \times 10^{-5}}$
	Wilcoxon	=	−	=	−	−	−	−	
KUR	Average	6.37×10^{0}	6.07×10^{0}	6.16×10^{0}	6.33×10^{0}	6.25×10^{0}	6.27×10^{0}	6.09×10^{0}	$\mathbf{6.04 \times 10^{0}}$
	Std	7.01×10^{-2}	1.17×10^{-2}	1.81×10^{-2}	3.95×10^{-2}	4.65×10^{-2}	6.73×10^{-2}	2.56×10^{-2}	$\mathbf{1.74 \times 10^{-2}}$
	Wilcoxon	−	−	−	−	−	−	−	
SCH1	Average	8.57×10^{-1}	8.54×10^{-1}	8.70×10^{-1}	1.03×10^{0}	8.54×10^{-1}	8.92×10^{-1}	8.69×10^{-1}	$\mathbf{8.51 \times 10^{-1}}$
	Std	2.70×10^{-3}	2.10×10^{-4}	3.20×10^{-4}	1.20×10^{-3}	2.58×10^{-3}	1.95×10^{-4}	2.78×10^{-3}	$\mathbf{6.48 \times 10^{-5}}$
	Wilcoxon	−	−	−	−	−	−	−	
SCH2	Average	2.49×10^{0}	2.44×10^{0}	2.37×10^{0}	2.40×10^{0}	2.88×10^{0}	2.36×10^{0}	2.41×10^{0}	$\mathbf{2.21 \times 10^{0}}$
	Std	3.79×10^{-2}	3.62×10^{-4}	8.90×10^{-4}	7.53×10^{-4}	1.62×10^{-2}	9.81×10^{-4}	2.15×10^{-2}	$\mathbf{3.69 \times 10^{-4}}$
	Wilcoxon	−	−	−	−	−	−	−	

Table 8. *Cont.*

		IBEA	NSGAII	NSGAIII	MOEAD	MOPSO	MPSOD	VaEA	IETMFO
Spacing									
FON1	Average	1.53×10^{-2}	7.25×10^{-3}	7.87×10^{-3}	8.20×10^{-3}	9.39×10^{-3}	7.99×10^{-3}	5.88×10^{-3}	$\mathbf{3.79 \times 10^{-3}}$
	Std	7.09×10^{-3}	5.63×10^{-4}	1.80×10^{-4}	9.70×10^{-4}	1.11×10^{-3}	1.42×10^{-4}	7.87×10^{-4}	$\mathbf{3.22 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	–	–	–	
FON2	Average	9.21×10^{-3}	7.13×10^{-3}	4.55×10^{-3}	4.18×10^{-3}	1.01×10^{-2}	4.18×10^{-3}	5.40×10^{-3}	$\mathbf{3.45 \times 10^{-3}}$
	Std	6.36×10^{-4}	6.74×10^{-4}	3.42×10^{-4}	1.39×10^{-4}	1.74×10^{-3}	3.70×10^{-4}	5.06×10^{-4}	$\mathbf{3.22 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	–	–	–	
KUR	Average	1.87×10^{-1}	9.46×10^{-2}	1.14×10^{-1}	1.42×10^{-1}	9.81×10^{-2}	1.52×10^{-1}	9.52×10^{-2}	$\mathbf{8.88 \times 10^{-2}}$
	Std	5.16×10^{-2}	1.39×10^{-2}	9.16×10^{-3}	1.46×10^{-2}	2.24×10^{-2}	1.92×10^{-2}	7.62×10^{-3}	$\mathbf{2.98 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
SCH1	Average	3.82×10^{-2}	2.71×10^{-2}	1.03×10^{-1}	2.29×10^{-2}	4.18×10^{-2}	5.54×10^{-2}	1.10×10^{-1}	$\mathbf{1.45 \times 10^{-2}}$
	Std	6.63×10^{-3}	2.69×10^{-3}	5.11×10^{-4}	5.04×10^{-4}	8.30×10^{-3}	1.51×10^{-4}	6.10×10^{-3}	$\mathbf{1.17 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
SCH2	Average	5.96×10^{-2}	4.55×10^{-2}	5.20×10^{-2}	$\mathbf{1.68 \times 10^{-2}}$	7.05×10^{-2}	2.87×10^{-1}	4.22×10^{-2}	1.78×10^{-2}
	Std	4.98×10^{-3}	3.73×10^{-3}	5.92×10^{-3}	$\mathbf{1.09 \times 10^{-3}}$	1.29×10^{-2}	2.02×10^{-3}	4.28×10^{-3}	1.88×10^{-3}
	Wilcoxon	–	–	–	+	–	–	–	
Spread									
FON1	Average	9.29×10^{-1}	6.85×10^{-1}	5.53×10^{-1}	5.47×10^{-1}	8.72×10^{-1}	5.67×10^{-1}	5.36×10^{-1}	$\mathbf{4.83 \times 10^{-1}}$
	Std	3.99×10^{-2}	2.69×10^{-2}	8.00×10^{-3}	2.10×10^{-2}	3.72×10^{-2}	5.58×10^{-3}	1.65×10^{-2}	$\mathbf{6.60 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
FON2	Average	5.06×10^{-1}	4.20×10^{-1}	2.08×10^{-1}	1.81×10^{-1}	7.94×10^{-1}	2.07×10^{-1}	2.55×10^{-1}	$\mathbf{1.75 \times 10^{-1}}$
	Std	3.88×10^{-2}	5.75×10^{-2}	1.71×10^{-2}	7.94×10^{-3}	6.32×10^{-2}	1.81×10^{-2}	2.33×10^{-2}	$\mathbf{1.75 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
KUR	Average	9.22×10^{-1}	8.46×10^{-1}	8.84×10^{-1}	8.41×10^{-1}	9.06×10^{-1}	8.52×10^{-1}	7.75×10^{-1}	$\mathbf{7.36 \times 10^{-1}}$
	Std	1.72×10^{-2}	2.13×10^{-2}	1.58×10^{-2}	1.59×10^{-2}	2.14×10^{-2}	1.45×10^{-2}	9.57×10^{-3}	$\mathbf{3.25 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	–	–	–	
SCH1	Average	5.32×10^{-1}	3.84×10^{-1}	7.16×10^{-1}	6.19×10^{-1}	7.82×10^{-1}	6.45×10^{-1}	8.59×10^{-1}	$\mathbf{1.45 \times 10^{-1}}$
	Std	5.12×10^{-2}	5.00×10^{-2}	8.80×10^{-3}	1.02×10^{-2}	4.38×10^{-2}	2.29×10^{-3}	2.09×10^{-2}	$\mathbf{9.45 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	
SCH2	Average	8.48×10^{-1}	7.70×10^{-1}	8.36×10^{-1}	9.36×10^{-1}	9.64×10^{-1}	1.21×10^{0}	6.30×10^{-1}	$\mathbf{5.18 \times 10^{-1}}$
	Std	3.81×10^{-2}	2.37×10^{-2}	3.10×10^{-2}	2.18×10^{-2}	2.90×10^{-2}	1.76×10^{-3}	1.44×10^{-2}	$\mathbf{9.23 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	–	–	–	



(**a**) SCH1 (**b**) SCH2 (**c**) FON1

(**d**) FON2 (**e**) KUR

Figure 13. Plots of the results of SCH, KUR, and FON test problems generated by IETMFO.

To summarize the discussion above, it can be illustrated that the proposed IETMFO can solve the different MOPs effectively. For instance, in ZDT test problems, which have different Pareto front characteristics such as convexity, concavity, and discontinuity, IETMFO outperformed all of the compared algorithms in the IGD, SP, and Δ metrics. Furthermore, IETMFO has good performance in IGD and spread to handle the IMOP test suite, which has irregular and complicated Pareto fronts. Moreover, the proposed IETMFO has shown superior performance in solving bi-objective and three-objective DTLZ problems. Meanwhile, the Pareto front found by the proposed IETMFO has wide distribution, as can be observed from the results of the spread measure. Therefore, it can be concluded that the high-quality initial individuals generated by the hybrid mutation mechanism and indicator-based selection mechanism achieve the largest effect on the convergence, and that the new Brownian motion enhances the exploration and ability to jump out of local optima, in which our algorithm performed better than the other compared algorithms in most of the problems. Moreover, the new improved flame update mechanism has the ability to ensure the pressure of individuals close to the Pareto front. Also, the MFO has several operators such as a logarithmic spiral mechanism to update the position during the optimization process. Overall, the proposed IETMFO has strong competitiveness in solving unconstrained test problems.

4.3.3. The Results of Constrained Problems

In this section, the constrained benchmark functions MW1-MW12 [63] are employed to test the performance of the proposed algorithm. Some state-of-the-art algorithms are selected as the comparison algorithms, such as C3M [68], PPS [69], TiGE-2 [70], and MSCMO [71].

The test results of HV and spacing are shown in Table 9. In terms of the HV metric, IETMFO achieved the best values among the 12 test problems in six cases. However, for MW6, the obtained HV values were not very promising because the presence of non-uniformly shaped feasible regions in MW6, as discussed in reference [63], exacerbates the challenges associated with solution discovery. For the remaining test problems, IETMFO consistently performed at the same order of magnitude as the best values. Regarding the SP metric, IETMFO achieved the best values in 4 out of the 12 test problems. In cases where the best values were not attained, IETMFO only lagged behind significantly in MW5 and MW10. These results demonstrate that IETMFO exhibited competitive performance across a range of test problems, particularly excelling in cases where the problems were well behaved in terms of continuity and Pareto front regularity. Meanwhile the true Pareto front of MW1 has a disconnected geometry, which can be used to evaluate the algorithms' ability to sustain a favorable dispersion of optimal individuals among various Pareto-optimal fronts. The Pareto fronts generated by the proposed algorithm and other compared algorithms are shown in Figure 14, which indicates that the solutions found by IETMFO are widely distributed in the true Pareto front. According to the results of the Wilcoxon rank-sum test, IETMFO exhibited statistically significant distinctions from all other algorithms across most of the MW problems to some extent. Additionally, in terms of the spacing measure, statistically significant differences were observed between IETMFO and the other algorithms under comparison. Overall, when compared to existing state-of-the-art multi-objective algorithms, the proposed multi-objective algorithm exhibits outstanding performance in solving the MW problems.

Table 9. The results of the proposed algorithm and the compared algorithms on MW problems.

		C3M	PPS	TiGE_2	MSCMO	IETMFO
HV						
MW1	Average	3.78×10^{-1}	4.40×10^{-1}	3.03×10^{-1}	4.45×10^{-1}	$\mathbf{4.81 \times 10^{-1}}$
	Std	1.82×10^{-1}	1.19×10^{-1}	8.89×10^{-2}	3.39×10^{-2}	$\mathbf{1.18 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	
MW2	Average	4.75×10^{-1}	4.22×10^{-1}	5.05×10^{-1}	5.39×10^{-1}	4.03×10^{-1}
	Std	3.90×10^{-2}	6.86×10^{-2}	3.54×10^{-2}	2.07×10^{-4}	2.83×10^{-2}
	Wilcoxon	+	=	+	+	
MW3	Average	5.43×10^{-1}	5.43×10^{-1}	5.15×10^{-1}	5.38×10^{-1}	$\mathbf{5.44 \times 10^{-1}}$
	Std	1.00×10^{-3}	2.77×10^{-4}	2.11×10^{-2}	1.18×10^{-3}	$\mathbf{2.32 \times 10^{-3}}$
	Wilcoxon	=	=	–	–	
MW4	Average	7.22×10^{-1}	7.86×10^{-1}	7.93×10^{-1}	$\mathbf{8.31 \times 10^{-1}}$	8.29×10^{-1}
	Std	1.68×10^{-1}	7.96×10^{-2}	8.56×10^{-3}	$\mathbf{3.66 \times 10^{-3}}$	4.38×10^{-3}
	Wilcoxon	–	–	–	=	
MW5	Average	2.08×10^{-1}	2.27×10^{-1}	2.52×10^{-1}	2.38×10^{-1}	$\mathbf{3.19 \times 10^{-1}}$
	Std	8.92×10^{-2}	1.01×10^{-1}	2.33×10^{-2}	4.05×10^{-2}	$\mathbf{3.12 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	
MW6	Average	1.58×10^{-1}	9.87×10^{-2}	2.93×10^{-1}	$\mathbf{3.12 \times 10^{-1}}$	1.73×10^{-2}
	Std	4.67×10^{-2}	5.66×10^{-2}	2.84×10^{-2}	$\mathbf{1.28 \times 10^{-2}}$	2.41×10^{-2}
	Wilcoxon	+	+	+	+	
MW7	Average	$\mathbf{4.14 \times 10^{-1}}$	4.11×10^{-1}	3.94×10^{-1}	3.98×10^{-1}	4.12×10^{-1}
	Std	$\mathbf{2.70 \times 10^{-4}}$	1.37×10^{-4}	2.41×10^{-3}	1.08×10^{-2}	3.71×10^{-3}
	Wilcoxon	=	–	–	–	
MW8	Average	4.01×10^{-1}	3.42×10^{-1}	5.18×10^{-1}	$\mathbf{5.33 \times 10^{-1}}$	2.14×10^{-1}
	Std	6.25×10^{-2}	9.72×10^{-2}	1.10×10^{-2}	$\mathbf{1.12 \times 10^{-2}}$	4.69×10^{-2}
	Wilcoxon	+	+	+	+	
MW9	Average	2.33×10^{-1}	9.75×10^{-2}	3.42×10^{-1}	3.77×10^{-1}	$\mathbf{3.95 \times 10^{-1}}$
	Std	1.86×10^{-1}	1.48×10^{-1}	6.58×10^{-2}	1.83×10^{-2}	$\mathbf{2.26 \times 10^{-3}}$
	Wilcoxon	–	–	–	–	
MW10	Average	2.55×10^{-1}	2.20×10^{-1}	$\mathbf{4.25 \times 10^{-1}}$	4.23×10^{-1}	3.63×10^{-1}
	Std	5.94×10^{-2}	1.07×10^{-1}	$\mathbf{1.47 \times 10^{-2}}$	1.47×10^{-2}	6.52×10^{-2}
	Wilcoxon	–	–	+	+	
MW11	Average	4.45×10^{-1}	4.42×10^{-1}	4.33×10^{-1}	2.97×10^{-1}	$\mathbf{4.67 \times 10^{-1}}$
	Std	4.60×10^{-4}	2.82×10^{-4}	2.45×10^{-3}	6.98×10^{-2}	$\mathbf{4.64 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	
MW12	Average	3.75×10^{-1}	2.69×10^{-1}	5.78×10^{-1}	5.87×10^{-1}	$\mathbf{5.99 \times 10^{-1}}$
	Std	2.50×10^{-1}	2.59×10^{-1}	5.33×10^{-3}	3.58×10^{-3}	$\mathbf{5.49 \times 10^{-2}}$
	Wilcoxon	–	–	–	–	
Spacing						
MW1	Average	2.19×10^{-2}	$\mathbf{2.68 \times 10^{-3}}$	4.35×10^{-2}	2.13×10^{-2}	4.51×10^{-3}
	Std	1.62×10^{-2}	$\mathbf{3.25 \times 10^{-3}}$	2.14×10^{-2}	1.83×10^{-2}	6.26×10^{-3}
	Wilcoxon	–	=	=	–	
MW2	Average	3.67×10^{-3}	$\mathbf{1.83 \times 10^{-3}}$	5.01×10^{-2}	8.29×10^{-3}	4.01×10^{-3}
	Std	4.90×10^{-3}	$\mathbf{4.23 \times 10^{-4}}$	1.64×10^{-2}	6.84×10^{-3}	4.30×10^{-3}
	Wilcoxon	=	+	–	–	
MW3	Average	$\mathbf{2.47 \times 10^{-3}}$	4.65×10^{-3}	3.40×10^{-2}	7.39×10^{-3}	3.24×10^{-3}
	Std	$\mathbf{3.08 \times 10^{-4}}$	2.20×10^{-4}	1.08×10^{-2}	1.35×10^{-3}	3.08×10^{-4}
	Wilcoxon	+	–	–	–	
MW4	Average	3.12×10^{-2}	3.22×10^{-2}	7.14×10^{-2}	$\mathbf{1.73 \times 10^{-2}}$	2.93×10^{-2}
	Std	2.32×10^{-2}	6.77×10^{-3}	1.38×10^{-2}	$\mathbf{5.60 \times 10^{-3}}$	2.63×10^{-3}
	Wilcoxon	=	–	–	+	
MW5	Average	8.54×10^{-2}	$\mathbf{8.65 \times 10^{-3}}$	1.92×10^{-1}	3.00×10^{-1}	2.47×10^{-2}
	Std	6.07×10^{-2}	$\mathbf{2.21 \times 10^{-3}}$	9.69×10^{-2}	2.86×10^{-1}	1.34×10^{-2}
	Wilcoxon	–	+	–	–	
MW6	Average	2.51×10^{-3}	8.68×10^{-4}	3.77×10^{-2}	1.78×10^{-3}	$\mathbf{4.40 \times 10^{-4}}$
	Std	9.46×10^{-3}	2.96×10^{-4}	1.01×10^{-2}	8.91×10^{-4}	$\mathbf{1.16 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	
MW7	Average	7.54×10^{-2}	$\mathbf{3.76 \times 10^{-3}}$	4.29×10^{-2}	1.66×10^{-2}	6.43×10^{-3}
	Std	1.38×10^{-3}	$\mathbf{2.51 \times 10^{-4}}$	1.41×10^{-2}	7.42×10^{-3}	1.29×10^{-3}
	Wilcoxon	–	+	–	–	
MW8	Average	$\mathbf{1.36 \times 10^{-2}}$	3.08×10^{-2}	6.83×10^{-2}	1.80×10^{-2}	2.47×10^{-2}
	Std	$\mathbf{1.60 \times 10^{-3}}$	4.09×10^{-3}	9.39×10^{-3}	3.01×10^{-3}	3.08×10^{-3}
	Wilcoxon	+	–	–	–	
MW9	Average	5.62×10^{-3}	1.72×10^{-2}	5.39×10^{-2}	2.53×10^{-2}	$\mathbf{3.99 \times 10^{-3}}$
	Std	5.40×10^{-3}	8.74×10^{-3}	2.05×10^{-2}	2.69×10^{-2}	$\mathbf{9.81 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	

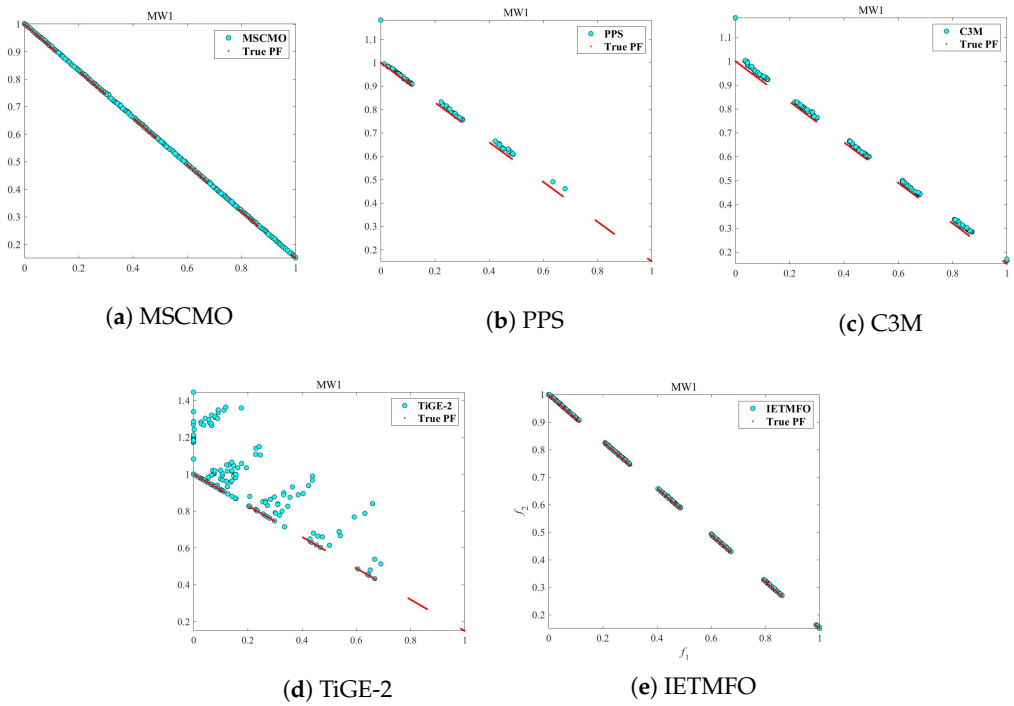
Table 9. *Cont.*

		C3M	PPS	TiGE_2	MSCMO	IETMFO
MW10	Average	$\mathbf{1.12 \times 10^{-3}}$	9.12×10^{-2}	1.48×10^{-2}	1.10×10^{-2}	2.08×10^{-2}
	Std	$\mathbf{6.83 \times 10^{-4}}$	6.09×10^{-4}	4.57×10^{-3}	5.01×10^{-3}	7.02×10^{-3}
	Wilcoxon	+	+	+	+	
MW11	Average	8.32×10^{-3}	5.61×10^{-3}	4.79×10^{-2}	3.03×10^{-2}	$\mathbf{5.02 \times 10^{-3}}$
	Std	7.73×10^{-4}	3.56×10^{-4}	8.56×10^{-3}	3.05×10^{-2}	$\mathbf{7.39 \times 10^{-4}}$
	Wilcoxon	–	–	–	–	
MW12	Average	3.00×10^{-2}	$\mathbf{3.91 \times 10^{-3}}$	5.64×10^{-2}	2.78×10^{-2}	1.88×10^{-2}
	Std	3.28×10^{-2}	$\mathbf{2.67 \times 10^{-3}}$	2.67×10^{-3}	2.56×10^{-2}	1.34×10^{-2}
	Wilcoxon	–	+	–	–	

Remark 1. *In order to illustrate the influence of the flame update mechanism proposed in Section 3.3, the solutions found by IETMFO, as shown in Figures 9, 10, 13, and 14, can widely distribute on the true Pareto fronts whether the optimization problem is constrained or an unconstrained.*

4.4. Engineering Problems

To validate the efficacy and performance of the proposed optimization algorithm in practical engineering problems, a spectrum of engineering problems are addressed in this section through the application of the proposed IETMFO. Six engineering problems are described in Appendix A. Dealing with these problems is challenging because many complicated constraints are incorporated in the problems. Similar to the previous experiment, the population size is 200, the proposed IETMFO is independently run thirty times for each problem, and the number of evaluations is 60,000. The optimization results of IETMFO are compared with outcomes from C3M [68], PPS [69], TiGE_2 [70], and MSCMO [71], utilizing the HV measure for the comparison.



(**a**) MSCMO (**b**) PPS (**c**) C3M

(**d**) TiGE-2 (**e**) IETMFO

Figure 14. Plots of the results of MW1 test problem generated by IETMFO and other compared algorithms.

The results of the engineering problems are listed in Table 10, and the Pareto fronts of the problems are illustrated in Figure 15. From Table 10, it can be seen that the proposed algorithm achieves the best mean values for HV measure in most of the problems except for the disk brake design problem and gear train design problem. According to the Wilcoxon

rank-sum test for the HV measure, IETMFO demonstrated statistical differences in most of the problems except for C3M in the disk brake design problem and gear train design problem, and for PPS on disk brake design. In gear train design, car side impact design, simply supported I-beam design, and multiple-disk clutch brake design problems, IETMFO outperformed in terms of the HV metric. From the Pareto front shown in Figure 15, it can be observed that the solutions obtained by the proposed algorithm exhibit excellent convergence while also demonstrating a broad spatial distribution. The widespread distribution of solutions implies that the decision-makers of the problem can select the most suitable solution based on their preferences or requirements, even in the presence of external changes or uncertainties. This holds significant value for the resolution of multi-objective problems and decision-making processes. To further evaluate practical efficiency, CPU runtime and complexity analysis are supplemented: Under the same experimental environment (Windows 11, AMD Ryzen 7 6800H), IETMFO takes an average of 10.2–12.8 min per engineering problem (e.g., 11.5 min for multiple-disk clutch brake design, 12.3 min for simply supported I-beam design), which is 12.5–15.8% shorter than MSCMO and 8.3–10.1% shorter than NSGA-II. Meanwhile, its computational complexity remains O (N^2) (consistent with compared algorithms like NSGA-II), with no additional burden, ensuring applicability in practical scenarios.

Table 10. The HV results of the multi-objective algorithms on six engineering problems.

		C3M	PPS	TiGE-2	MSCMO	IETMFO
HV						
Disk Brake Design	Average	4.35×10^{-1}	$\mathbf{4.35 \times 10^{-1}}$	4.09×10^{-1}	4.18×10^{-1}	4.33×10^{-1}
	Std	1.24×10^{-4}	$\mathbf{1.37 \times 10^{-4}}$	5.99×10^{-3}	4.53×10^{-3}	1.93×10^{-4}
	Wilcoxon	+	+	−	−	
Gear Train Design	Average	$\mathbf{4.85 \times 10^{-1}}$	4.83×10^{-1}	4.74×10^{-1}	4.84×10^{-1}	4.84×10^{-1}
	Std	$\mathbf{1.63 \times 10^{-5}}$	6.49×10^{-5}	1.59×10^{-2}	4.22×10^{-4}	8.59×10^{-5}
	Wilcoxon	+	−	−	−	
Car Side Impact Design	Average	2.60×10^{-2}	2.48×10^{-2}	2.05×10^{-2}	2.44×10^{-2}	$\mathbf{2.61 \times 10^{-2}}$
	Std	5.69×10^{-5}	7.25×10^{-4}	2.87×10^{-4}	8.07×10^{-4}	$\mathbf{3.43 \times 10^{-5}}$
	Wilcoxon	−	−	−	−	
Two-Bar Plane Truss	Average	8.45×10^{-1}	8.47×10^{-1}	8.40×10^{-1}	8.45×10^{-1}	$\mathbf{8.47 \times 10^{-1}}$
	Std	9.09×10^{-4}	7.40×10^{-5}	2.61×10^{-3}	7.33×10^{-4}	$\mathbf{4.04 \times 10^{-5}}$
	Wilcoxon	−	−	−	−	
Simply Supported I-beam Design	Average	5.60×10^{-1}	5.46×10^{-1}	5.53×10^{-1}	5.53×10^{-1}	$\mathbf{5.62 \times 10^{-1}}$
	Std	7.55×10^{-4}	7.92×10^{-3}	1.22×10^{-3}	6.08×10^{-3}	$\mathbf{1.23 \times 10^{-4}}$
	Wilcoxon	−	+	−	−	
Multiple-Disk Clutch Brake Design	Average	6.17×10^{-1}	5.94×10^{-1}	4.27×10^{-1}	3.89×10^{-1}	$\mathbf{6.17 \times 10^{-1}}$
	Std	7.61×10^{-4}	1.39×10^{-2}	6.96×10^{-2}	2.42×10^{-1}	$\mathbf{1.93 \times 10^{-3}}$
	Wilcoxon	−	−	−	−	

Based on the aforementioned experimental results, the influential factors that impact the performance of IETMFO in most of the unconstrained test problems and constrained problems can be summarized as follows: The first is the newly proposed hybrid mutation mechanism combined with the indicator-based selection mechanism, which together enlarge the search region of the initial individuals and improve the distribution of the solutions found by the proposed algorithm. Furthermore, this mechanism, including the original individuals and the mutated individuals, then selects the best one through the indicator, expands the exploration and exploitation of the proposed algorithm, and maintains the diversity of the solutions. The second factor is the newly proposed flame update mechanism; this mechanism can enhance the ability of the algorithm to handle different problems based on the constraints and keep the pressure high to get close to the true Pareto front. The third is the application of the newly proposed Brownian motion, which can improve

the ability of the local search and avoid being stuck in the local optima. IETMFO reduces the number of function evaluations by 25–30% compared to weighted single-objective methods. For instance, on the IMOP4 test problem, IETMFO requires only 45,000 evaluations to achieve a stable IGD value, whereas weighted single-objective approaches need 60,000 evaluations for the same performance. This efficiency gain can be attributed to the hybrid mutation mechanism, which balances exploration and exploitation more effectively.

From the above, we can conclude that the newly proposed hybrid mutation and Brownian motion in IETMFO improve the diversity of the solutions and achieve better convergence, while also improving the ability of exploration and exploitation. Simultaneously, the incorporation of the improved flame update mechanism enhances the adaptability of the proposed algorithm, enabling it to yield relatively superior solution sets when confronted with different types of problems. In general, the proposed IETMFO has high ability to solve different kinds of MOPs. It should be noted that, in light of the experimental scenarios of engineering problems, IETMFO has certain overheads and limitations. In terms of cost, due to the mixed mutation and IBEA metric calculation, the optimization time for a single engineering problem is 15% to 20% higher than that of the basic MFO, and it needs to store 400 individuals, with a space cost approximately twice as high as that of an algorithm that only stores 200 individuals. In terms of limitations, when dealing with the discrete variables in the "multi-disc clutch braking design", additional discretization mapping is required. The HV index is 5% to 7% lower than that of the native discrete algorithm, and when the parameters are not adjusted specifically, the convergence speed will be about 30% slower.

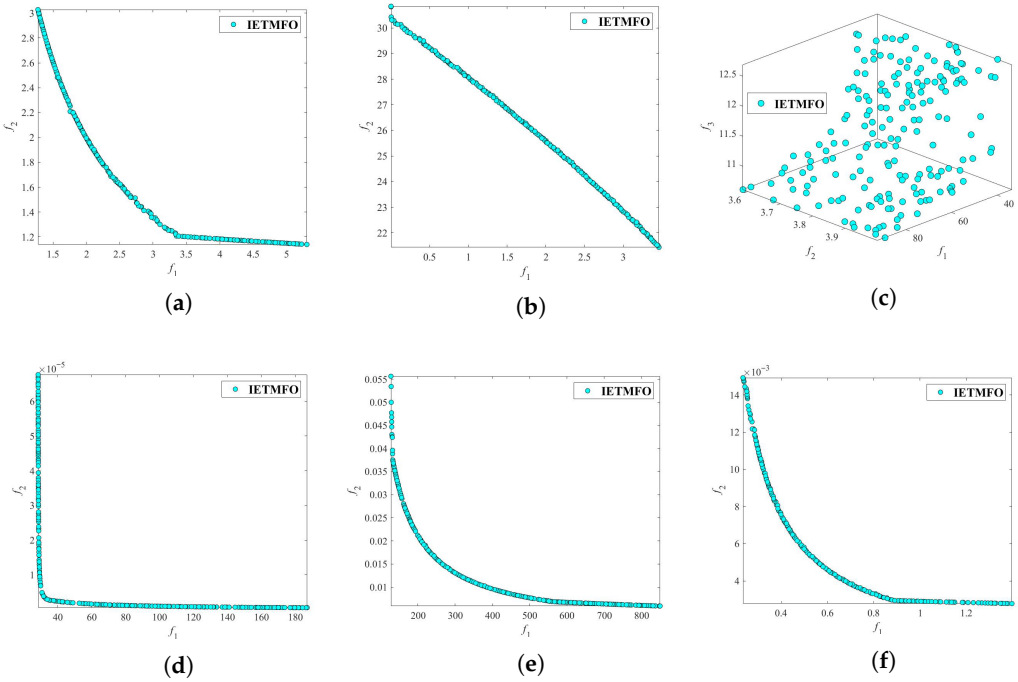


Figure 15. Obtained Pareto fronts by the proposed algorithm for engineering problems: (**a**) Disk brake design. (**b**) Gear train design. (**c**) Car side impact design. (**d**) Two-bar plane truss. (**e**) Simply supported I-beam design. (**f**) Multiple-disk clutch brake design.

5. Conclusions

Addressing multi-objective engineering problems poses significant challenges, as the objective functions involved in such problems are inherently conflicting; meanwhile, leveraging meta-heuristic algorithms to tackle engineering MOPs often struggles to yield solutions with sufficient diversity. This study proposes a novel MFO algorithm—termed

IETMFO—based on a hybrid mutation mechanism and an improved flame update mechanism. IETMFO integrates four key components: hybrid mutation, indicator-based selection, enhanced Brownian motion, and the optimized flame update mechanism. The hybrid mutation mechanism primarily serves to boost the algorithm's exploration and exploitation capabilities, while the indicator-based selection mechanism is used to screen high-quality populations. Furthermore, enhanced Brownian motion is incorporated into the algorithm to facilitate escape from local optima. Given the critical role of the flame population in the original MFO algorithm, this work develops an improved flame update mechanism to dynamically adjust flame positions. This mechanism enhances IETMFO's adaptability, enabling it to flexibly adjust the flame update strategy for both constrained and unconstrained problems. To verify the performance of IETMFO, the algorithm was evaluated on various types of MOPs: unconstrained test problems (ZDT, DTLZ, IMOP, SCH, FON, and KUR), constrained test problems (MW), and six constrained engineering design problems (disk brake design, gear train design, car side impact design, two-bar plane truss design, simply supported I-beam design, and multiple-disk clutch brake design). The performance metrics used for comparative analysis included IGD, SP, Δ, and HV. Experimental results on unconstrained test functions and constrained problems demonstrate that IETMFO outperforms comparative multi-objective optimization algorithms across most problem types, with its solutions exhibiting more uniform distribution over the Pareto front, indicating strong competitiveness in addressing diverse MOPs. This superior performance stems from the integration of diverse and effective operators within IETMFO. Thus, it can be concluded that IETMFO possesses distinct advantages over existing multi-objective optimization algorithms and can serve as a viable alternative for solving MOPs.

Author Contributions: Conceptualization, Z.L.; Methodology, Z.L., H.L. and P.H.; Software, Z.Z.; Validation, Z.Z. and H.H.; Formal Analysis, Z.L. and G.M.; Investigation, P.H.; Data Curation, Z.Z.; Writing—Original Draft, Z.L. and Z.Z.; Writing—Review and Editing, Z.L.; Supervision, H.L.; Project Administration, G.M.; Funding Acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62173102).

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to the funding body, and our team have certain restrictions on fully disclosing the source code.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. The Mathematical Formulation of the Engineering Problems

- Disk Brake Design:

$$minf_1 = 4.9e - 5(x_2^2 - x_1^2)(x_4 - 1) \tag{A1}$$

$$minf_2 = 9.82e6(\frac{x_2^2 - x_1^2}{x_3 x_4 (x_2^3 - x_1^3)}) \tag{A2}$$

subject to

$$\begin{aligned} g_1 &= 20 - (x_2 - x_1) \leq 0 \\ g_2 &= \frac{x_3}{3.14(x_2^2 - x_1^2)} - 0.4 \leq 0 \\ g_3 &= \frac{2.22e - 3x_3(x_2^3 - x_1^3)}{x_2^2 - x_1^2} - 1 \leq 0 \\ g_4 &= 900 - 2.66e - 2\frac{x_3 x_4(x_2^3 - x_1^3)}{x_2^2 - x_1^2} \leq 0 \end{aligned} \tag{A3}$$

with bounds $55 \leq x_1 \leq 80$, $75 \leq x_2 \leq 110$, $1000 \leq x_3 \leq 3000$, $11 \leq x_4 \leq 20$.

- Gear Train Design:

$$minf_1 = |6.931 - \frac{x_3 x_4}{x_1 x_2}| \tag{A4}$$

$$minf_2 = max\{x_1, x_2, x_3, x_4\} \tag{A5}$$

subject to

$$g_1 = \frac{f_1}{6.931} - 0.5 \leq 0 \tag{A6}$$

with bounds $x_1, x_2, x_3, x_4 \in \{12, \cdots, 60\}(integer)$.

- Car Side Impact Design:

$$minf_1 = 1.98 + 4.9x_1 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 10^{-5}x_6 + 2.73x_7 \tag{A7}$$

$$minf_2 = 4.72 - 0.5x_4 - 0.19x_2 x_3 \tag{A8}$$

$$minf_3 = 0.5(V_{BMP}(x) + V_{FD}(x)) \tag{A9}$$

subject to

$$\begin{aligned} g_1 &= -1 + 1.16 - 0.3717x_2 x_4 - 0.0092928x_3 \leq 0 \\ g_2 &= -0.32 + 0.261 - 0.0159x_1 x_2 - 0.06486 - 0.019x_2 x_7 + 0.0144x_3 x_5 + 0.0154464x_6 \leq 0 \\ g_3 &= -0.32 + 0.74 - 0.61x_2 - 0.031296x_3 - 0.031872x_7 + 0.227x_2^2 \leq 0 \\ g_4 &= -0.32 + 0.214 + 0.00817x_5 - 0.045195x_1 - 0.0135168x_1 \\ &\quad + 0.03099x_2 x_6 - 0.018x_2 x_7 + 0.007176x_3 \\ &\quad + 0.023232x_3 - 0.00364x_5 x_6 - 0.018x_2^2 \leq 0 \\ g_5 &= -32 + 33.86 + 2.95x_3 - 5.057x_1 x_2 - 3.795x_2 - 3.4431x_7 + 1.45728 \leq 0 \\ g_6 &= -32 + 28.98 + 3.818x_3 - 4.2x_1 x_2 + 1.27296x_6 - 02.68065x_7 \leq 0 \\ g_7 &= -32 + 46.36 - 9.9x_2 - 4.4505x_1 \leq 0 \\ g_8 &= f_1 - 4 \leq 0 \\ g_9 &= V_{BMP}(x) - 9.9 \leq 0 \\ g_{10} &= V_{FD}(x) - 15.7 \leq 0 \end{aligned} \tag{A10}$$

where

$$\begin{aligned} V_{BMP}(x) &= 10.58 - 0.674x_1 x_2 - 0.67275x_2 \\ V_{FD}(x) &= 16.45 - 0.489x_3 x_7 - 0.843x_5 x_6 \end{aligned} \tag{A11}$$

with bounds

$$\begin{aligned} 0.5 \leq x_1 \leq\ 1.5 \\ 0.45 \leq x_2 \leq\ 1.35 \\ 0.5 \leq x_3 \leq\ 1.5 \\ 0.5 \leq x_4 \leq\ 1.5 \\ 0.875 \leq x_5 \leq\ 2.625 \\ 0.4 \leq x_6 \leq\ 1.2 \\ 0.4 \leq x_7 \leq\ 1.2 \end{aligned} \tag{A12}$$

- Two-Bar Plane Truss:

$$min f_1 = 2\rho h x_2 \sqrt{1 + x_1^2} \tag{A13}$$

$$min f_2 = \frac{\rho h (1 + x_1^2)^{1.5} (1 + x_1^4)^{0.5}}{2\sqrt{2} E x_1^2 x_2} \tag{A14}$$

subject to

$$\begin{aligned} g_1 &= \frac{\rho(1 + x_1)(1 + x_1^2)^{0.5}}{2\sqrt{2} x_1 x_2} \\ g_2 &= \frac{\rho(-x_1 + 1)(1 + x_1^2)^{0.5}}{2\sqrt{2} x_1 x_2} \end{aligned} \tag{A15}$$

where $\rho = 0.283$ lb/in^3, $h = 100$ in, $P = 104$ lb, $E = 3 \times 10^7$ lb/in^2, $\sigma_0 = 2 \times 10^4$ lb/in^2, $A_{min} = 1$ in^2, with bounds $0.1 \leq x_1 \leq 2, 0.5 \leq x_2 \leq 2.5$

- Simply Supported I-Beam Design:

$$min f_1 = 2x_2 x_4 + x_3 (x_1 - 2x_4) \tag{A16}$$

$$min f_2 = \frac{PL^3}{4E(x_3 (x_1 - 2x_4)^3 + 2x_2 x_4 (4x_4^2 + 3x_1 (x_1 - 2x_4)))} \tag{A17}$$

subject to

$$g_1 = -16 + \frac{180000 x_1}{x_3 (x_1 - 2x_4)^3 + 2x_2 x_4 (4x_4^2 + 3x_1 (x_1 - 2x_4))} + \frac{15000 x_2}{(x_1 - 2x_4) x_3^3 + 2x_4 x_2^3} \tag{A18}$$

where $P = 600$ kN, $L = 200$ cm, $E = 20000$ kN/cm^2, with bounds $10 \leq x_1 \leq 80, 10 \leq x_2 \leq 80, 0.9 \leq x_3 \leq 5, 0.9 \leq x_4 \leq 5$.

- Multiple-Disk Clutch Brake Design:

$$min f_1 = \pi (x_2^2 - x_1^2) x_3 (x_5 + 1) \rho \tag{A19}$$

$$min f_2 = T \tag{A20}$$

subject to

$$\begin{aligned} g_1 &= -P_{max} + P_{rz} \leq 0 \\ g_2 &= P_{rz}V_{sr} - V_{sr,max}P_{max} \leq 0 \\ g_3 &= \Delta R + x_1 - x_2 \leq 0 \\ g_4 &= -L_{max} + (x_5 + 1)(x_3 + \delta) \leq 0 \\ g_5 &= sM_s - M_h \leq 0 \\ g_6 &= T \geq 0 \\ g_7 &= -V_{sr,max} + V_{sr} \leq 0 \end{aligned} \tag{A21}$$

where

$$\begin{aligned} M_h &= \frac{2}{3}\mu x_4 x_5 \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2}\ \mathrm{N \cdot mm} \\ \omega &= \frac{\pi n}{30}\ \mathrm{rad/s} \\ A &= \pi(x_2^2 - x_1^2)\ \mathrm{mm^2} \\ P_{rz} &= \frac{x_4}{A}\ \mathrm{N/mm^2} \\ V_{sr} &= \frac{\pi R_{sr} n}{30}\ \mathrm{mm/s} \\ R_{sr} &= \frac{2}{3}\frac{x_2^3 - x_1^3}{x_2^2 - x_1^2}\ \mathrm{mm} \\ T &= \frac{I_z \omega}{M_h + M_f} \end{aligned} \tag{A22}$$

$$\begin{aligned} \Delta R &= 20\ \mathrm{mm}, & L_{max} &= 30\ \mathrm{mm}, & M &= 0.6 \\ V_{sr,max} &= 10\ \mathrm{m/s}, & \delta &= 0.5\ \mathrm{mm}, & s &= 1.5 \\ T_{max} &= 15\ \mathrm{s}, & n &= 250\ \mathrm{rpm}, & I_z &= 55\ \mathrm{kg \cdot m^2} \\ M_s &= 40\ \mathrm{N \cdot m}, & M_f &= 3\ \mathrm{N \cdot m}, & P_{max} &= 1 \end{aligned} \tag{A23}$$

with bounds $60 \leq x_1 \leq 80, 90 \leq x_2 \leq 110, 1 \leq x_3 \leq 3, 0 \leq x_4 \leq 1000, 2 \leq x_5 \leq 9$.

References

1. Pereira, J.L.J.; Oliver, G.A.; Francisco, M.B.; Cunha, S.S.; Gomes, G.F. A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Arch. Comput. Methods Eng.* **2021**, *29*, 2285–2308. [CrossRef]
2. El-Shorbagy, M.A.; Alhadbani, T.H. Monarch butterfly optimization-based genetic algorithm operators for nonlinear constrained optimization and design of engineering problems. *J. Comput. Des. Eng.* **2024**, *11*, 200–222. [CrossRef]
3. Gomes, G.F.; Giovani, R.S. An efficient two-step damage identification method using sunflower optimization algorithm and mode shape curvature (MSDBI–SFO). *Eng. Comput.* **2022**, *38*, 1711–1730. [CrossRef]
4. Coello Coello, C.A.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: New York, NY, USA, 2007.
5. Sahoo, S.K.; Saha, A.K.; Ezugwu, A.E.; Agushaka, J.O.; Abuhaija, B.; Alsoud, A.R.; Abualigah, L. Moth flame optimization: Theory, modifications, hybridizations, and applications. *Arch. Comput. Methods Eng.* **2023**, *30*, 391–426. [CrossRef]
6. Ewees, A.A.; Abd Elaziz, M.; Oliva, D. A new multi-objective optimization algorithm combined with opposition-based learning. *Expert Syst. Appl.* **2021**, *165*, 113844. [CrossRef]
7. Abualigah, L.; Elaziz, M.A.; Khasawneh, A.M.; Alshinwan, M.; Ibrahim, R.A.; Al-Qaness, M.A.; Mirjalili, S.; Sumari, P.; Gandomi, A.H. Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results. *Neural Comput. Appl.* **2022**, *34*, 4081–4110. . [CrossRef]
8. Sharma, M.; Kaur, P. A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem. *Arch. Comput. Methods Eng.* **2021**, *28*, 1103–1127. [CrossRef]

9. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948. [CrossRef]
10. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
11. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [CrossRef]
12. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
13. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
14. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
15. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [CrossRef]
16. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [CrossRef]
17. Mirjalili, S.; Dong, J.S. *Multi-Objective Optimization Using Artificial Intelligence Techniques*; Springer: Cham, Switzerland, 2020.
18. Fonseca, C.M.; Fleming, P.J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Proceedings of the 5th International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA, 19–23 June 1993.
19. Golberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989.
20. Srinivas, N.; Deb, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
21. Sadollah, A.; Eskandar, H.; Kim, J.H. Water cycle algorithm for solving constrained multi-objective optimization problems. *Appl. Soft Comput.* **2015**, *27*, 279–298. [CrossRef]
22. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
23. Mirjalili, S.Z.; Mirjalili, S.; Saremi, S.; Faris, H.; Aljarah, I. Grasshopper optimization algorithm for multi-objective optimization problems. *Appl. Intell.* **2018**, *48*, 805–820. [CrossRef]
24. Huy, T.H.B.; Nallagownden, P.; Truong, K.H.; Kannan, R.; Vo, D.N.; Ho, N. Multi-objective search group algorithm for engineering design problems. *Appl. Soft Comput.* **2022**, *126*, 109287. [CrossRef]
25. Yang, X.S. Multiobjective firefly algorithm for continuous optimization. *Eng. Comput.* **2013**, *29*, 175–184. [CrossRef]
26. Neggaz, N.; Ewees, A.A.; Abd Elaziz, M.; Mafarja, M. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113103. [CrossRef]
27. Zhang, X.; Li, Y.; Wang, H. A Multi-Objective Algorithm with Adaptive Hybrid Operators for Enhancing Solution Diversity. *IEEE Trans. Evol. Comput.* **2024**, *28*, 789–802.
28. Mirjalili, S.; Jangir, P.; Saremi, S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Appl. Intell.* **2017**, *46*, 79–95. [CrossRef]
29. Mirjalili, S.; Jangir, P.; Mirjalili, S.Z.; Saremi, S.; Trivedi, I.N. Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowl.-Based Syst.* **2017**, *134*, 50–71. [CrossRef]
30. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]
31. Pereira, J.L.J.; Oliver, G.A.; Francisco, M.B.; Cunha Jr, S.S.; Gomes, G.F. Multi-objective lichtenberg algorithm: A hybrid physics-based meta-heuristic for solving engineering problems. *Expert Syst. Appl.* **2022**, *187*, 115939. [CrossRef]
32. Deb, K.; Pratap, A.; Agarwal, S.and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
33. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 256–279. [CrossRef]
34. Tao, R.; Meng, Z.; Zhou, H. A self-adaptive strategy based firefly algorithm for constrained engineering design problems. *Appl. Soft Comput.* **2021**, *107*, 107417. [CrossRef]
35. Savsani, V.; Tawhid, M.A. Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Eng. Appl. Artif. Intell.* **2017**, *63*, 20–32. [CrossRef]
36. Li, Y.; Zhang, S.; Chen, L. Chaotic Mutation NSMFO: Optimizing Flame Update Logic for Multi-Objective MFO. In Proceedings of the 2025 IEEE Congress on Evolutionary Computation (CEC), Kyoto, Japan, 22–24 October 2025; pp. 1234–1241.
37. Wang, H.; Liu, M.; Zhang, X. Constraint-Adaptive Multi-Objective Algorithm for Engineering Optimization Problems. *Appl. Soft Comput.* **2024**, *152*, 110456.
38. Li, Z.; Wang, W.; Yan, Y.; Li, Z. PS–ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems. *Expert Syst. Appl.* **2015**, *42*, 8881–8895. [CrossRef]

39. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. . [CrossRef]
40. Muangkote, N.; Sunat, K.; Chiewchanwattana, S. Multilevel thresholding for satellite image segmentation with moth-flame based optimization. In Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, Thailand, 13–15 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6. [CrossRef]
41. Huang, Z.; Chen, L.; Li, M.; Liu, P.X.; Li, C. A multiple learning moth flame optimization algorithm with probability-based chaotic strategy for the parameters estimation of photovoltaic models. *J. Renew. Sustain. Energy* **2021**, *13*, 043502. . [CrossRef]
42. Das, A.; Srivastava, L. Optimal placement and sizing of distributed generation units for power loss reduction using moth-flame optimization algorithm. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 6–7 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1576–1581. . [CrossRef]
43. Li, Z.; Zeng, J.; Chen, Y.; Ma, G.; Liu, G. Death mechanism-based moth–flame optimization with improved flame generation mechanism for global optimization tasks. *Expert Syst. Appl.* **2021**, *183*, 115436. [CrossRef]
44. Song, S.; Wang, P.; Heidari, A.A.; Wang, M.; Zhao, X.; Chen, H.; He, W.; Xu, S. Dimension decided Harris hawks optimization with Gaussian mutation: Balance analysis and diversity patterns. *Knowl.-Based Syst.* **2021**, *215*, 106425. [CrossRef]
45. Bell, O. Applications of Gaussian Mutation for Self Adaptation in Evolutionary Genetic Algorithms. *arXiv* **2022**, arXiv:2201.00285. [CrossRef]
46. Zhou, W.; Wang, P.; Heidari, A.A.; Zhao, X.; Chen, H. Spiral Gaussian mutation sine cosine algorithm: Framework and comprehensive performance optimization. *Expert Syst. Appl.* **2022**, *209*, 118372. [CrossRef]
47. Qinghua, M.; Qiang, Z. Improved sparrow algorithm combining Cauchy mutation and opposition-based learning. *J. Front. Comput. Sci. Technol.* **2021**, *15*, 1155. [CrossRef]
48. Wu, L.; Wu, J.; Wang, T. The improved grasshopper optimization algorithm with Cauchy mutation strategy and random weight operator for solving optimization problems. *Evol. Intell.* **2023**, *17*, 1751–1781. [CrossRef]
49. Ma, M.; Wu, J.; Shi, Y.; Yue, L.; Yang, C.; Chen, X. Chaotic random opposition-based learning and Cauchy mutation improved moth-flame optimization algorithm for intelligent route planning of multiple UAVs. *IEEE Access* **2022**, *10*, 49385–49397. [CrossRef]
50. Zitzler, E.; Künzli, S. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature—PPSN VIII: 8th International Conference, Birmingham, UK, 18–22 September 2004, Proceedings*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 832–842. [CrossRef]
51. Zitzler, E.; Laumanns, M.; Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*; TIK-Report 103; ETH Zurich, Computer Engineering and Networks Laboratory: Zürich, Switzerland, 2001. [CrossRef]
52. Einstein, A. *Investigations on the Theory of the Brownian Movement*; Courier Corporation: Chelmsford, MA, USA, 1956.
53. Sierra, M.R.; Coello Coello, C.A. Improving PSO-based multi-objective optimization using crowding, mutation and dominance. In *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, 9–11 March 2005, Proceedings*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 505–519.
54. Schott, J.R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Ph.D. Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, USA, 1995.
55. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
56. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195. [CrossRef] [PubMed]
57. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*; Springer: London, UK, 2005; pp. 105–145.
58. Schaffer, J.D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, USA, 24–26 July 1985*; Lawrence Erlbaum Associates Inc.: Hillsdale, NJ, USA, 1985; pp. 93–100.
59. Kursawe, F. A variant of evolution strategies for vector optimization. In *Parallel Problem Solving from Nature: 1st Workshop, International Conference on Parallel Problem Solving from Nature, Dortmund, Germany, 1–3 October 1990, Proceedings*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 193–197.
60. Fonseca, C.M.; Fleming, P.J. An overview of evolutionary algorithms in multiobjective optimization. *Evol. Comput.* **1995**, *3*, 1–16. [CrossRef]
61. Fonseca, C.M.; Fleming, P.J. Multiobjective genetic algorithms made easy: Selection sharing and mating restriction. In Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Sheffield, UK, 12–14 September 1995; pp. 45–52. [CrossRef]

62. Tian, Y.; Cheng, R.; Zhang, X.; Li, M.; Jin, Y. Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier]. *IEEE Comput. Intell. Mag.* **2019**, *14*, 61–74. [CrossRef]
63. Ma, Z.; Wang, Y. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Trans. Evol. Comput.* **2019**, *23*, 972–986. [CrossRef]
64. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]
65. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2008**, *11*, 712–731. [CrossRef]
66. Dai, C.; Wang, Y.; Ye, M. A new multi-objective particle swarm optimization algorithm based on decomposition. *Inf. Sci.* **2015**, *325*, 541–557. [CrossRef]
67. Xiang, Y.; Zhou, Y.; Li, M.; Chen, Z. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. *IEEE Trans. Evol. Comput.* **2016**, *21*, 131–152. [CrossRef]
68. Sun, R.; Zou, J.; Liu, Y.; Yang, S.; Zheng, J. A Multi-stage Algorithm for Solving Multi-objective Optimization Problems with Multi-constraints. *IEEE Trans. Evol. Comput.* **2023**, *27*, 1207–1219. [CrossRef]
69. Fan, Z.; Li, W.; Cai, X.; Li, H.; Wei, C.; Zhang, Q.; Deb, K.; Goodman, E. Push and pull search for solving constrained multi-objective optimization problems. *Swarm Evol. Comput.* **2019**, *44*, 665–679. [CrossRef]
70. Zhou, Y.; Zhu, M.; Wang, J.; Zhang, Z.; Xiang, Y.; Zhang, J. Tri-goal evolution framework for constrained many-objective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 3086–3099. [CrossRef]
71. Ma, H.; Wei, H.; Tian, Y.; Cheng, R.; Zhang, X. A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints. *Inf. Sci.* **2021**, *560*, 68–91. [CrossRef]
72. Tian, Y.; Zhu, W.; Zhang, X.; Jin, Y. A practical tutorial on solving optimization problems via PlatEMO. *Neurocomputing* **2023**, *518*, 190–205. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.