mathematics

MDPI

Article

A Multi-Strategy Enhanced Farthest Better or Nearest Worse Optimizer for Complex Optimization Problems

Xiaojie Tang [1,*], Chengfen Jia [1], Pengju Qu [2,3] and Pan Zhang [1]

[1] School of Intelligent Manufacturing, Sichuan University Jinjiang College, Meishan 620860, China; jiachengfen@scujj.edu.cn (C.J.); zhangpan@scujj.edu.cn (P.Z.)
[2] Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University, Guiyang 550025, China; 20140497@git.edu.cn
[3] Engineering Training Center, Guizhou Institute of Technology, Guiyang 550003, China
* Correspondence: tangxiaojie@scujj.edu.cn

Abstract

An enhanced Farthest Better or Nearest Worse Optimizer (EFNO) is developed to overcome the limitations of the original FNO, including insufficient population diversity, and slow and premature convergence. To achieve a more effective balance between exploration and exploitation, three complementary strategies are incorporated: an adaptive weighted Euclidean distance mechanism to improve solution selection, an elite-guided archive reinforcement strategy for stage-wise differentiated population guidance, and a nearest-worse-individual-based reverse-reinforcement mechanism to enhance diversity. The effectiveness of EFNO is evaluated on the CEC2017 and CEC2022 benchmark suites under different dimensional settings. Comparative experiments with eleven State-of-the-Art algorithms, including a benchmark-winning method, are conducted to assess convergence behavior and optimization accuracy. The results indicate that EFNO consistently achieves superior performance and obtains the best overall ranking. To further demonstrate its applicability, EFNO is applied to two real-world engineering optimization problems and a robot path-planning task. In the path-planning experiments, six random grid-based environments with different sizes (20×20, 30×30, and 50×50) and obstacle densities (20% and 40%) are constructed. Experimental results show that EFNO outperforms all competing algorithms in terms of solution quality and robustness, confirming its effectiveness and strong generalization capability.

Keywords: farthest better or nearest worse optimizer; metaheuristic; path planning; engineering optimization problem; elite-guided archive reinforcement strategy

MSC: 68T20; 90C59; 68T40

Check for updates

Academic Editor: Hsien-Chung Wu

Received: 14 April 2026
Revised: 3 May 2026
Accepted: 5 May 2026
Published: 7 May 2026

Copyright: © 2026 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

With the rapid development of intelligent manufacturing, unmanned systems, and complex engineering applications, large-scale, nonlinear, and constrained optimization problems have become increasingly prevalent in fields such as robotic path planning, structural design, and resource scheduling [1]. These problems are typically high-dimensional and multimodal, making it difficult to locate global optima while being prone to getting trapped in local optima. Traditional gradient-based optimization methods often struggle to address such challenges effectively due to their reliance on derivative information and

problem structure. As a result, metaheuristic algorithms, particularly swarm intelligence-based methods, have attracted significant attention owing to their gradient-free nature and strong global search capabilities [2,3].

In recent decades, a wide range of bio-inspired optimization algorithms have been proposed, including Particle Swarm Optimization (PSO, 1995) [4], Differential Evolution (DE, 1995) [5], Gravitational Search Algorithm (GSA, 2009) [6], Grey Wolf Optimizer (GWO, 2014) [7], Whale Optimization Algorithm (WOA, 2016) [8], Sine Cosine Algorithm (SCA, 2016) [9], and Spherical Evolution (SE, 2019) [10]. These methods have been successfully applied to various engineering optimization problems. Zhao et al. [11] proposed a leader-driven particle swarm optimizer (LDPSO). This algorithm has been proven effective for solving flexible job shop scheduling problems. Zhu et al. [12] applied a self-adaptive updating strategy differential evolution algorithm (SaUSDE) to simulate spherical multi-aircraft route planning. Ge et al. [13] proposed an improved adaptive guided differential evolution (AGDE) algorithm and applied it to parameter identification of proton exchange membrane fuel cells (PEMFCs). Nithya et al. [14] proposed a method utilizing a gravity search algorithm (GSA-Blend) combined with hybrid crossover to address real-time trajectory optimization in dynamically complex driving scenarios. Shaikh et al. [15] proposed an improved chaotic grey wolf optimizer (ICGWO) for wireless sensor network coverage problems. Experimental results showed that ICGWO can effectively improve coverage and connectivity. Chen et al. [16] applied an improved whale optimization algorithm to ship supply path planning. Agrawal et al. [17] introduced an adaptive random sine cosine algorithm (ARSCA) to solve the optimal placement and tuning of flexible AC transmission system (FACTS) devices in the IEEE 30-bus system, including thyristor-controlled series capacitors (TCSCs), thyristor-controlled phase shifters (TCPSs), and static VAR compensators (SVCs). However, despite their effectiveness, most existing algorithms still face a common challenge: how to achieve a proper balance between exploration and exploitation. In complex and high-dimensional search spaces, insufficient population diversity and inappropriate search strategies often lead to premature convergence or stagnation [18,19].

To address these issues, numerous improvement strategies have been developed in recent studies. Existing works can be broadly categorized into three directions: (1) hybridization strategies that combine multiple algorithms to enhance exploration capability [20,21]; (2) adaptive mechanisms that dynamically adjust parameters or search behaviors to improve convergence performance [22,23]; and (3) multi-strategy frameworks that integrate diverse search operators to balance global and local search [24,25]. Although these approaches have achieved promising results, they still suffer from several limitations, such as unstable guidance, insufficient utilization of neighborhood information, and limited adaptability in complex and dynamic search environments [26,27].

Farthest Better or Nearest Worse Optimizer (FNO, 2026) [28] is a recently proposed metaheuristic algorithm. Its core idea is to build the search mechanism based on the relative positions among individuals. In each iteration, FNO selects two key neighbors for each individual, and then it proceeds as follows: (1) the farthest individual among all solutions better than the current one (farthest better, FB); (2) the nearest individual among all solutions worse than the current one (nearest worse, NW). Based on this mechanism, FNO constructs a unique search pattern called the Dynamic Focusing Strategy (DFS). In this strategy, FB serves as the global guiding direction, while NW is used as a local geometric reference to adjust the step size. When NW lies on the path toward FB, the algorithm directly crosses NW to avoid local interference. When NW is located in the opposite direction, its distance is used to adjust the search step size. In this way, a dynamic balance between exploration and exploitation is achieved.

Although FNO has advantages such as a simple structure, few parameters, and a clear search mechanism, it still has some limitations: (1) FNO uses the standard Euclidean distance to determine relationships among individuals. In high-dimensional spaces, the phenomenon of distance concentration may occur. This weakens the ability to distinguish between the farthest better (FB) and the nearest worse (NW), and reduces the effectiveness of the search direction. (2) As the population gradually converges, the candidate NW individuals become closer in space. This makes the avoidance mechanism less effective and limits the ability of the algorithm to escape from local optima. (3) The original algorithm uses the same update strategy for individuals with different quality. It ignores the functional differences between high-quality and low-quality individuals during the search process, which reduces the efficiency of cooperative evolution.

To address the above issues, this paper proposes an Enhanced Farthest Better or Nearest Worse Optimizer (EFNO). The proposed method optimizes the original algorithm from three aspects, namely distance measurement, the NW mechanism, and search strategy, to improve overall performance:

- Adaptive weighted Euclidean distance mechanism: Dynamic weights are assigned to each dimension to improve the accuracy of neighbor selection and direction determination;
- Elite-guided archive-reinforcement strategy: An external archive is used to expand the NW candidate set, and different guidance strategies are applied to different types of individuals, improving exploration ability and convergence accuracy;
- NW-based reverse-reinforcement mechanism: When stagnation or low diversity occurs, a reverse-perturbation strategy away from NW is constructed. This changes NW from a passive reference into an active driving factor and improves the ability to escape from local optima.

To evaluate the performance of EFNO, comprehensive experiments are conducted on the CEC2017 and CEC2022 benchmark functions. Ablation studies, convergence analysis, and statistical tests are used to assess the effectiveness of each strategy. In addition, EFNO is applied to engineering design problems and robot path planning to verify its practical applicability.

The main contributions of this paper are as follows:

- A novel enhanced FNO algorithm, named EFNO, is designed to improve the search mechanism of FNO with multiple cooperative strategies to achieve an optimal balance between exploration and exploitation;
- A multi-level experimental framework is designed on the CEC2017 and CEC2022 benchmark suites. It includes ablation study, convergence behavior analysis, and statistical tests, which systematically verify the individual contributions and combined effects of the proposed strategies;
- In the typical engineering design problem and path-planning task for applications, it can demonstrate its generalization ability and application potential in real-world complex optimization scenarios.

The structure of this paper is organized as follows:

Section 1 introduces the research background and main contributions; Section 2 presents the basic principles of the FNO algorithm; Section 3 describes the proposed improved algorithm, EFNO; Section 4 provides the parameter sensitivity analysis; Section 5 presents the experimental results and analysis, including ablation studies on CEC2022, convergence analysis, and performance evaluation on CEC2017, as well as experiments on two real-world engineering problems and robot path planning in six map environments; and, finally, Section 6 concludes this paper and discusses future work.

2. Farthest Better or Nearest Worse Optimizer (FNO)

Inspired by the quality and distance relationships between agents in the search space, the FNO algorithm divides the search process into two stages. In the first stage, the search agents skip nearby regions with low potential to avoid local optima. In the second stage, they explore distant positions with higher potential in order to reach or search for the global optimum. These operations enhance population diversity and provide FNO with opportunities to discover high-quality regions while avoiding low-quality areas. At the same time, the proposed Dynamic Focusing Strategy (DFS) is used to control the exploration rate.

2.1. Preliminaries

In FNO, the vector $X_i = \left\{x_i^{d_1}, x_i^{d_2}, x_i^{d_3}, \cdots, x_i^{d_D}\right\}$ represents the ith solution in the search space, where D is the dimension of the problem, and x^d $(d = 1, 2, 3, \ldots, D)$ denotes the decision variable. Each solution, X_i, corresponds to a cost function value, f_i $(i = 1, 2, \ldots, nPop)$, which is used to measure the quality of that solution. The relationship between the population variable, the solution variable, the decision variables, and the cost function is as follows:

$$Population = \begin{bmatrix} x_1^{d_1} & x_1^{d_2} & x_1^{d_3} & \cdots & x_1^{d_D} \\ x_2^{d_1} & x_2^{d_2} & x_2^{d_3} & \cdots & x_2^{d_D} \\ x_3^{d_1} & x_3^{d_2} & x_3^{d_3} & \ldots & x_3^{d_D} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{nPop}^{d_1} & x_{nPop}^{d_2} & x_{nPop}^{d_3} & \cdots & x_{nPop}^{d_D} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{nPop} \end{bmatrix} \quad (1)$$

where $nPop$ denotes the population size.

2.2. The Farthest Better Region

For a solution, X_i, its farthest better solution, X_{FB_i}, is defined as the solution in the population that is of higher quality than X_i (i.e., $f(X_j) < f(X_i)$) and is farthest from X_i in terms of Euclidean distance. The mathematical expression for this is as follows:

$$X_{FB_i} \leftarrow \left\{X_j | f(X_j) < f(X_i) \; and \; Max(Dist_{i,j})\right\}, \quad where \; j \neq i \; and \; i, j \epsilon [1, \; nPop] \quad (2)$$

where X_{FB_i} is the farthest optimal candidate at position X_i; $f(X_i)$ is the objective function value at X_i; $f(X_j)$ is the objective function value at X_j; and $Max(Dist_{i,j})$ is the maximum distance between the position, X_i, and the candidate position, X_j.

This mechanism encourages each individual to explore distant but more promising regions. Consequently, it enhances the diversity of the population and helps to prevent premature convergence.

2.3. The Nearest Worse Region

The solution X_i's nearest worse solution, X_{NW_i}, is defined as the solution in the population that is of lower quality than X_i (i.e., $f(X_j) > f(X_i)$) and is closest to X_i in Euclidean distance. The mathematical expression for this is as follows:

$$X_{NW_i} \leftarrow \left\{X_j | f(X_j) > f(X_i) \; and \; Min(Dist_{i,j})\right\}, \quad where \; j \neq i \; and \; i, j \epsilon [1, \; nPop] \quad (3)$$

where X_{NW_i} denotes the nearest worst candidate at position X_i; $Min(Dist_{i,j})$ denotes the minimum distance between the position, X_i, and candidate position, X_j.

2.4. Dynamic Focus Strategy (DFS)

To adaptively balance global exploration and local exploitation, FNO introduces a Dynamic Focusing Strategy. At its core is a dynamically contracting search radius. First, an adaptive control parameter, α, is defined, which decays from 1 to 0. It is typically defined as follows:

$$\alpha \leftarrow \left[1 - \frac{FES}{MaxFES}\right]^{\frac{1}{2}} \tag{4}$$

where FES is the current number of function evaluations, and $MaxFES$ is the maximum number of function evaluations.

DFS is then defined as a random vector based on a uniform distribution:

$$\overrightarrow{DFS_i} \leftarrow U(\mu - \alpha, u + \alpha), \quad where\ u \leftarrow U(0,1) \tag{5}$$

When α is large at the beginning of the iteration, the range of variation in $\overrightarrow{DFS_i}$ is wide, and the algorithm tends to explore a broad region. As the iteration progresses, α decreases and the range of variation in $\overrightarrow{DFS_i}$ narrows. This enables the algorithm to refine its search in the vicinity of the current optimal solution.

2.5. Position-Update Rules

The position-update process is divided into two stages, selected probabilistically. The choice between them is determined by comparing the random number, r, with α.

2.5.1. Stage One: Avoiding the Nearest Poor-Quality Point

Executed with probability $\left(1 - \frac{FES}{MaxFES}\right)$, this stage aims to avoid neighbouring low-quality regions. First, a temporary position, X_i', is calculated:

$$\varphi \leftarrow Min(0, sign(FB_i - X_i) \times sign(NW_i - X_i)) \tag{6}$$

$$X_i' \leftarrow \varphi \times NW_i + (1 - \varphi) \times X_i + sign(FB_i - X_i) \times r \times |X_i - NW_i|, \quad where\ r \leftarrow U(0,1) \tag{7}$$

$$X_{new} \leftarrow X_i', \quad if\ r < \left(1 - \frac{FES}{MaxFES}\right) \tag{8}$$

where φ is a binary decision variable used to determine whether X_i lies between FB_i and NW_i; the new position, X_{new}, is set directly to X_i'; $sign(\cdot)$ returns the sign of a given value; and r is a random value in the range [0, 1].

2.5.2. Stage Two: Searching for the Farthest Better Point

This stage is executed with probability $\frac{FES}{MaxFES}$, employing DFS to search in the direction of the farthest better point. The new position, X_{new}, is calculated using the following formula:

$$Step \leftarrow DFS \times (FB_i - X_i) \tag{9}$$

$$W \leftarrow (2 + U(-1, +1) \times U([1,\ 2])) \tag{10}$$

$$X_{new} \leftarrow X_i' + W \times Step, \quad if\ r < \frac{FES}{MaxFES} \tag{11}$$

In particular, DFS is a Dynamic Focusing Strategy calculated based on Equations (4) and (5), and W is a random value used to determine the step size.

3. Enhanced Farthest Better or Nearest Worse Optimizer (EFNO)

Although FNO improves search performance to some extent through the farthest better and nearest worse mechanism and the Dynamic Focusing Strategy, it still suffers from several limitations, including weak discrimination ability of distance measures in high-dimensional spaces, degradation of the NW avoidance mechanism, and a lack of differentiated update strategies for individuals.

To address these issues, this paper proposes an enhanced FNO (EFNO). It introduces an adaptive weighted Euclidean distance, an elite-guided and archive-enhanced mechanism, and an NW reverse-enhancement mechanism. These strategies are integrated to improve the global exploration ability, convergence accuracy, and convergence speed of the algorithm. The overall schematic diagram of EFNO is shown in Figure 1.

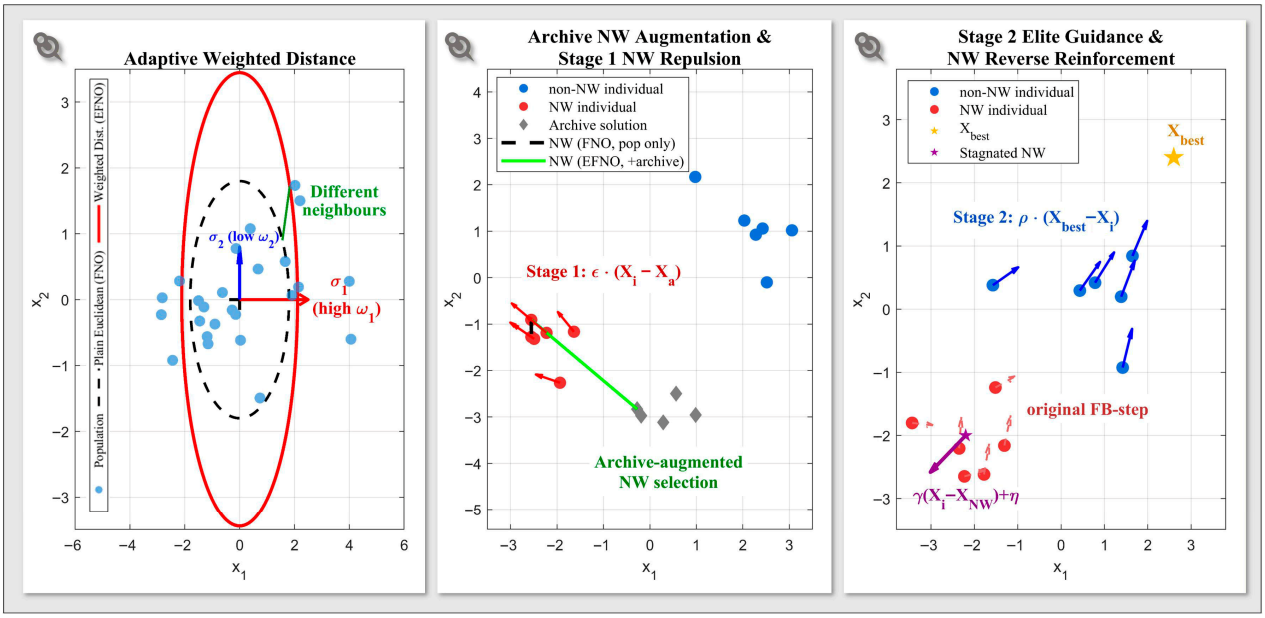


Figure 1. Schematic diagram illustrating the principle of EFNO.

As illustrated in Figure 1, the adaptive weighted distance is first employed to improve the selection of farthest better (FB) and nearest worse (NW) individuals, thereby enhancing the reliability of search directions. Then, within the two-stage update framework, an archive-based repulsion mechanism is applied to NW individuals in the early stage to promote exploration, while an elite-guided mechanism is applied to non-NW individuals in the later stage to accelerate convergence. In addition, when stagnation is detected, the NW-based reverse-reinforcement mechanism is activated to introduce perturbations, helping the algorithm escape from local optima. These components work collaboratively to achieve a dynamic balance between exploration and exploitation.

3.1. Adaptive Weighted Euclidean Distance Mechanism

The original FNO uses the pairwise distance, $Dist_{i,j}$, between individuals to select the farthest and nearest solutions, thereby determining X_{FB_i} and X_{NW_i}. However, the standard Euclidean distance suffers from the distance concentration phenomenon in high-dimensional spaces. As dimensionality increases, the distances between any two individuals tend to become similar. As a result, the discriminative ability of $Max(Dist_{i,j})$ and $Min(Dist_{i,j})$ decreases, and the selection of FB and NW degenerates into an almost random process.

In addition, different dimensions have different levels of importance and search activity during the optimization process. Dimensions with larger variance usually contain more useful search information and should be assigned higher weights. Therefore, an

adaptive weighted distance based on population statistical characteristics is introduced to enhance the discriminative ability of the distance metric.

Replace the Euclidean distance in Equations (2) and (3) with

$$Dist_{i,j} = \sqrt{\sum_{k=1}^{D} \omega_k(t) \times \left(x_i^k - x_j^k\right)^2} \tag{12}$$

$$\omega_k(t) = \frac{\sigma_k(t)}{\sum_{k=1}^{D} \sigma_k(t)} \tag{13}$$

where $\sigma_k(t)$ is the standard deviation of the population of the tth generation in the kth dimension, and $\omega_k(t)$ is the normalized weight in the kth dimension.

Although weighted distance metrics have been studied in clustering and optimization, the proposed mechanism is fundamentally different in its role and implementation. First, the adaptive weighted Euclidean distance is not used as a general similarity measure; instead, it is deeply embedded into the FB/NW selection mechanism, directly affecting the search direction of each individual. Second, the weights are dynamically determined based on the population distribution at each iteration, rather than being predefined or statically learned. Third, this mechanism specifically addresses the degradation of the farthest better and nearest worse selection in high-dimensional spaces, as such degradation has not been considered in existing FNO-related or distance-based optimization frameworks.

The proposed weighting scheme is based on the standard deviation of each dimension, which is used as a proxy to estimate its relative importance during the search process. The underlying intuition is that dimensions with larger population variance tend to exhibit more active search dynamics and may contain more useful information for guiding exploration.

However, it is important to note that variance is not always a perfect indicator of importance. In certain cases, high variance may be caused by noise, random fluctuations, or unstable search behavior, rather than meaningful structural information of the objective landscape. In such situations, assigning larger weights to these dimensions may not necessarily lead to improved performance.

To mitigate this potential issue, the proposed adaptive weighting mechanism operates in a dynamic and population-based manner. The weights are normalized across all dimensions and continuously updated during the search process, which reduces the influence of any single misleading or noisy dimension over time. Furthermore, when integrated with the other strategies in EFNO, the overall search process becomes more stable and robust.

Therefore, although variance-based weighting may have limitations in certain scenarios, it provides a simple yet effective mechanism to enhance the discriminative ability of distance measures in high-dimensional optimization problems.

3.2. Elite-Guided Archive Reinforcement Strategy

The original FNO has two main limitations. First, the nearest worse neighbor, X_{NW_i}, of an individual X_i is selected only from solutions worse than X_i in the current population. As the population converges, these worse candidates become closer to X_i in the search space. As a result, the NW-based avoidance mechanism gradually becomes ineffective.

Second, the original algorithm uses the same update strategy for both high-ranked individuals and low-ranked (NW) individuals. This ignores the fundamental functional difference between them during the search process. High-ranked individuals are already located in relatively good regions, and their main role is to refine convergence toward the global optimum. In contrast, NW individuals are located in low-quality regions, and their key role is to maintain population diversity and explore unknown areas.

To address these issues, this paper proposes a stage-based differentiated population guidance method, namely an elite-guided archive reinforcement strategy. Specifically,

- An external archive is introduced with a capacity of *2nPop* to store candidate solutions rejected by the greedy selection strategy [29]. The choice of *2nPop* aims to balance diversity enhancement and computational cost. In practice, the performance of the method is not highly sensitive to the exact archive size, as long as it remains proportional to the population size. A larger archive may slightly improve diversity but increases computational overhead, whereas a smaller archive may reduce its effectiveness in enhancing NW selection. The archive is updated only within the first 90% of function evaluations and is used to enhance NW selection.
- Based on the original two-stage structure of FNO, different guidance mechanisms are assigned to different stages. In Stage 1, an archive repulsion mechanism with a decay coefficient is applied to NW individuals to enhance exploration ability. In Stage 2, an elite-guided mechanism with a growth coefficient is applied to non-NW individuals to strengthen convergence toward the global optimum in the later search phase. The control coefficients of these two mechanisms change in opposite trends over time, forming a cooperative evolutionary pattern of early-stage NW dispersion and late-stage convergence of non-NW individuals.

3.2.1. Expansion of the NW Candidate Set

To enhance the discriminatory power of the *NW*, its candidate set is expanded to the union of the population and the reference set:

$$X_{NW_i} = \begin{cases} X_a, & if \ \min Dist(X_i, X_{arch}) < \min\limits_{x_j \epsilon W_i} Dist_{i,j} \\ & arg \min\limits_{x_j \epsilon W_i} Dist_{i,j}, \quad else \end{cases} \tag{14}$$

where W_i denotes the set of all individuals in the population that are inferior to X_i, and X_a denotes the individual closest to X_i after k individuals have been randomly sampled from the archive.

3.2.2. Stage One: Archive Repulsion Mechanism for NW Individuals

In Stage One, non-*NW* individuals follow the original update rule without any modification, while *NW* individuals are subject to a repulsion mechanism. This mechanism plays an important role in the early search stage. It can effectively push low-quality individuals away from poor regions and increase the overall diversity of the population. Therefore, Equation (8) is revised as follows:

$$X_{new} \leftarrow X_i' + \varepsilon \times (X_i - X_a), \quad if \ r < \left(1 - \tfrac{FES}{MaxFES}\right) \ and \ is \ NW \tag{15}$$

$$\varepsilon = r \times \left(1 - \tfrac{FES}{MaxFES}\right), \quad r \sim U(0,1) \tag{16}$$

where ε is the decay repulsion coefficient.

3.2.3. Stage Two: Elite-Guided Mechanism for Non-NW Individuals

In Stage Two, NW individuals keep the original update rule unchanged, while non-NW individuals are guided by an elite-based mechanism. This mechanism gradually becomes stronger in the later search stage. It is used to improve the convergence speed and solution accuracy of high-quality individuals. Therefore, Equation (11) is revised as follows:

$$X_{new} \leftarrow X_i' + W \times Step + \rho \times (X_{best} - X_i), \quad if \ r < \tfrac{FES}{MaxFES} \ and \ is \ not \ NW \tag{17}$$

$$\rho = r \times \frac{FES}{MaxFES}, \quad r \sim U(0,1) \tag{18}$$

where ρ is the incremental elite coefficient.

The proposed archive and elite-guided mechanisms provide the basic search guidance framework for the population, which serves as the foundation for the subsequent reverse-reinforcement operations introduced in Section 3.3.

Archive-based and elite-guided strategies have been widely used in evolutionary algorithms. However, the proposed mechanism differs in several key aspects. First, the archive is not merely used as a passive memory; it also actively participates in the NW selection and repulsion process, enhancing the effectiveness of the nearest worse mechanism. Second, unlike conventional elite strategies that uniformly guide the population, the proposed method introduces a functional differentiation between NW and non-NW individuals, assigning them distinct roles in exploration and exploitation. Third, a stage-wise dual-guidance scheme is designed, where archive-based repulsion dominates in the early stage, while elite-guided attraction gradually strengthens in the later stage. This cooperative and asymmetric guidance mechanism is fundamentally different from existing single-direction or static guidance strategies.

3.3. NW-Based Reverse-Reinforcement Mechanism

In FNO, NW individuals are only used as geometric references. Their role is limited to adjusting the step size, and they do not actively participate in constructing the search direction. As a result, when population diversity decreases or the algorithm stagnates, this mechanism cannot effectively help the algorithm escape from local optima.

To address this issue, an NW reverse-enhancement mechanism is introduced. It changes NW from a passive reference into an active source of perturbation, helping the algorithm escape local optima. This mechanism is triggered when stagnation occurs and population diversity becomes low. The corresponding position-update equation is given as follows:

$$X_{new} \leftarrow X_i + \gamma \times \left(X_i - X_{NW_i}\right) + \eta, \quad if\ \overline{\sigma}(t) < \delta_{div}\ and\ c_{stag} \geq T_{stag} \tag{19}$$

$$\overline{\sigma}(t) = \frac{1}{D}\sum_{k=1}^{D} \sigma_k(k) \tag{20}$$

$$\delta_{div} = \frac{0.05}{D}\sum_{k=1}^{D} (ub_k - lb_k) \tag{21}$$

$$\gamma = 0.5 \times (1 + r), \quad r \sim U(0,\ 1) \tag{22}$$

$$\eta^k \sim U(-0.1,\ 0.1) \cdot (ub_k - lb_k), \quad k = 1, \cdots, D \tag{23}$$

where $\overline{\sigma}(t)$ is the population mean standard deviation; δ_{div} is the diversity threshold; c_{stag} is the stagnation counter; T_{stag} is the stagnation threshold; γ denotes the inverse step size coefficient; r is a random number between 0 and 1; η denotes the random perturbation; and ub_k and lb_k are the upper and lower bounds of the kth-dimensional problem, respectively.

The parameters δ_{div} and T_{stag} are empirically set based on commonly adopted practices in the swarm intelligence literature, with $\delta_{div} = 0.05$ and $T_{stag} = 5$ used in this study. In practice, these parameters are not highly sensitive within a reasonable range. A smaller δ_{div} or T_{stag} may trigger the reverse mechanism more frequently, increasing exploration, while larger values may delay its activation and emphasize convergence. The chosen values provide a good balance between responsiveness and stability in our experiments.

The NW candidate information used in this mechanism is constructed based on the selection framework introduced in Section 3.2, which enhances the reliability of NW guidance without changing its original definition.

Compared with conventional opposition-based and reverse learning strategies, the proposed mechanism is guided by nearest worse individuals and is adaptively triggered based on the population state, making it a targeted escape strategy rather than a general perturbation operator.

Overall, the proposed NW-based reverse-reinforcement mechanism differs from conventional methods in regard to three aspects: (1) the reverse direction is guided by nearest worse individuals rather than random generation; (2) it is conditionally activated based on both population diversity and stagnation status; and (3) it is specifically designed to enhance the NW-based search structure rather than serving as a general-purpose diversity enhancement technique.

3.4. Pseudocode and Flowchart of EFNO

The flowchart of EFNO, which integrates the adaptive Euclidean distance, the elite-guided and archive-enhanced strategy, and the NW reverse-enhancement mechanism, is shown in Figure 2. The time complexity analysis is provided in Appendix A.1. The pseudocode of EFNO (Algorithm 1) is given as follows:

Algorithm 1. Pseudocode for EFNO.

Input: Population size, $nPop$; $MaxFES$; bounds lb and ub; dimension, D; objective function, f.
Output: Best solution, X^*; best fitness, f^*.
1: Initialize population X uniformly; evaluate $f(X_i)$ for all i.
2: Set $X^* \leftarrow argmin\ f(X)$; $archive\ A \leftarrow \varnothing$; $C_{Stag} \leftarrow 0$; $prev_best \leftarrow f^*$.
3: while FES < MaxFES do
4: sort X; $isNW(i) \leftarrow (rank(i) > N/2)$.
5: compute ω_k, $Dist_{i,j}$ by Equations (12) and (13); $\overline{\sigma}$ by Equation (20).
6: $C_{Stag} \leftarrow C_{Stag} + 1$ if $f^* \geq prev_best$, else ($C_{Stag} \leftarrow 0$; $prev_best \leftarrow f^*$).
7: stagnated $\leftarrow (\overline{\sigma} < \delta_{div})$ AND $(c_{stag} \geq T_{stag})$; if stagnated: $c_{stag} \leftarrow 0$.
8: for each i = 1 to N do
9: Determine FB_i by Equation (2); NW_i by Equation (14).
10: end for
11: for each i = 1 to N do
12: if stagnated then
13: $X_{new} \leftarrow$ Equation (19) using γ Equation (22), η Equation (23).
14: else
15: compute α Equation (4); DFS Equation (5); W Equation (10); X_i' by Equations (6) and (7).
16: if $r < 1 - FES/MaxFES$ then
17: $X_{new} \leftarrow$ Equations (15) and (16) if *isNW(i)* AND $|A| \geq 1$, else Equation (8).
18: else
19: $X_{new} \leftarrow$ Equations (17) and (18) if *~isNW(i)*, else Equation (11).
20: end if
21: end if
22: Clipping (X_{new}, lb, ub); evaluate f(X_{new}); FES ← FES + 1.
23: if f(X_{new}) < f(X_i): $X_i \leftarrow X_{new}$; update X^* if improved.
24: else if FES $\leq$ 0.9·MaxFES: add X_{new} to A (FIFO).
25: end for
26: end while
27: Return X^*, f^*.

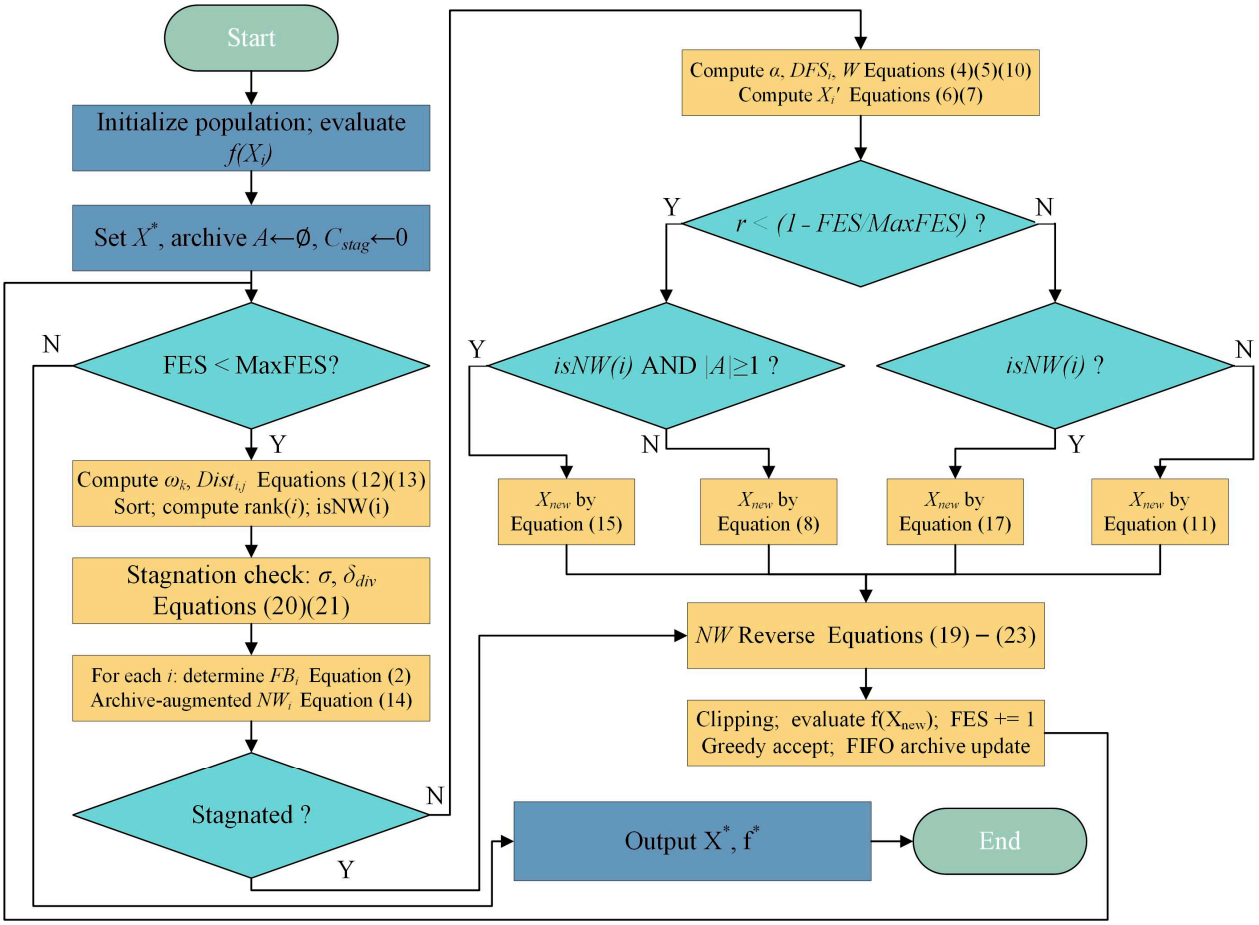


Figure 2. EFNO flowchart.

4. Parameter Sensitivity Analysis of EFNO

To investigate the influence of algorithm parameters on performance, this paper selects benchmark functions F1, F3–F5, F7, F8, F10, F12, F13, F15, F16, F21, F25, F26, and F30 from the CEC2017 test suite. Two key parameters are analyzed: the population size, $nPop$, and the archive sampling parameter, k.

During the parameter study, only one parameter is varied at a time, while all other parameters are kept unchanged. This ensures that the effect of each parameter can be evaluated independently. Based on the results under different parameter settings, the stability and robustness of the algorithm are assessed, and the optimal parameter configuration is determined.

Each experiment is run 30 independent times. The best value, mean value, variance, and Friedman rank of EFNO under different parameter settings are reported. The average Friedman ranking results of these two parameters are shown in Figure 3, and the detailed results are provided in Tables A3 and A4 in Appendix A.2.

According to the results, EFNO achieves the best performance when $nPop = 30$ and $k = 3$. Therefore, this parameter setting is used in all subsequent experiments.

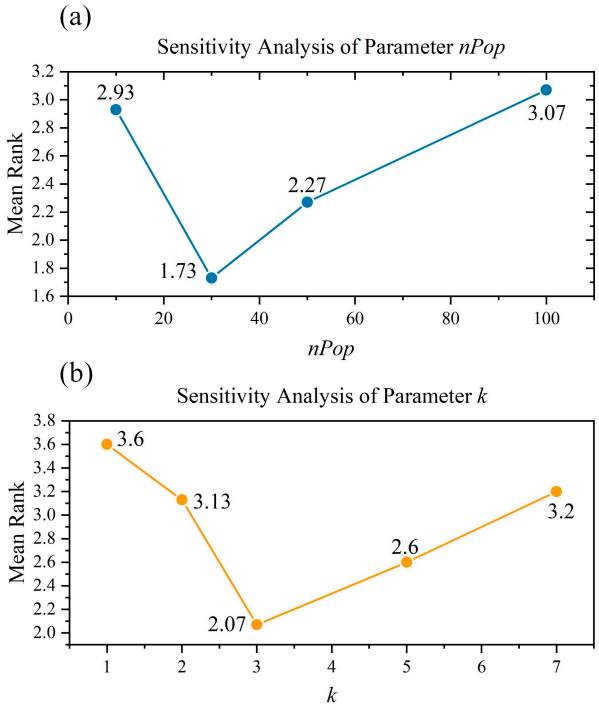


Figure 3. Average Friedman rankings from parameter sensitivity experiments on selected CEC2017 functions: (**a**) *nPop* and (**b**) *k*.

5. Experimental Results and Analysis

In this section, the optimization performance of EFNO is evaluated using benchmark test suites, two real-world engineering problems, and robot path-planning tasks.

5.1. Benchmark Test Suite Experiments

5.1.1. Experimental Configuration

To verify the optimization performance of EFNO, experiments are conducted on the CEC2017 and CEC2022 benchmark test suites. CEC2017 and CEC2022 contain 29 and 12 benchmark functions, respectively, covering various types of optimization problems. The detailed definitions of these functions are provided in Tables A1 and A2 in Appendix A.2.

Eleven representative algorithms are selected for comparison with EFNO, including (1) classical high-impact algorithms, i.e., PSO, DE, WOA, and SCA; (2) recent algorithms, i.e., DBO (Dung Beetle Optimizer, 2022) [30], GJO (Golden Jackal Optimization, 2022) [31], GGO (Greylag Goose Optimization, 2024) [32], and HBO (Hannibal Barca Optimizer, 2025) [33]; (3) classical variants, i.e., CLPSO (Comprehensive Learning Particle Swarm Optimizer, 2006) [34] and LSHADE (Success-History Based Adaptive Differential Evolution with Linear Population Size Reduction, 2014) [35]; and (4) the original FNO. The parameter settings of all algorithms follow the original literature and are summarized in Table 1.

The population size is set to 30. In the convergence behavior analysis and ablation study, the maximum number of iterations is set to 500. In other experiments, the termination condition is the maximum number of function evaluations, set to 1000D. Each algorithm is independently run 30 times to ensure fairness. The best value, mean value, and standard deviation are reported, along with statistical test metrics, including Friedman ranking and Wilcoxon rank-sum test results. All experiments are implemented in MATLAB R2021b.

Table 1. Parameter settings.

Algorithm	Parameters Setting
PSO	$C_1 = 1, C_2 = 1, v_{min} = -10,\ v_{max} = 10,\ \omega = [0.9,\ 0.4]$
DE	$F = 0.5, CR = 0.5$
WOA	$a = [2, 0]$
SCA	$a = 2$
DBO	$b = 0.3$
GJO	Parameter-free
GGO	$b = 1$
HBO	$\alpha = 0.63$
CL-PSO	$C = 1.2,\ F_{max} = 7,\ \omega_{max} = 0.9, \omega_{min} = 0.4$
LSHADE	$F = 0.5, CR = 0.5$
FNO	Parameter-free

5.1.2. Ablation Study on CEC2022

To evaluate the effectiveness of each proposed strategy, each improvement component is individually integrated into the original FNO to form corresponding variants. These variants are compared with the original FNO and EFNO on 20-dimensional functions from the CEC2022 benchmark suite. Each experiment is independently repeated 30 times, and the best value, mean value, variance, and Friedman ranking are recorded. The constructed variants include the following:

- FNO_A: FNO with the adaptive Euclidean distance;
- FNO_E: FNO with the elite-guided and archive-enhanced strategy;
- FNO_I: FNO with the NW reverse-enhancement mechanism;
- FNO_AE: FNO_A combined with the elite-guided and archive-enhanced mechanism.

The experimental results are presented in Table 2. The best values in the table are highlighted in bold and underlined. It can be observed that FNO_A and FNO_AE achieve better rankings than the original FNO, while FNO_E and FNO_I perform worse when applied individually. However, EFNO, which integrates all three strategies, achieves the best overall performance.

This phenomenon can be explained by the interaction and complementarity among the proposed strategies at the mechanism level. When applied individually, each strategy modifies only a specific component of the search process and may disrupt the original balance between exploration and exploitation. For example, the elite-guided and archive-enhanced strategy (FNO_E) strengthens convergence by biasing the search toward high-quality individuals, but it may lead to premature convergence without sufficient diversity support. Similarly, the NW-based reverse-enhancement mechanism (FNO_I) introduces perturbations to escape local optima, but it may weaken convergence stability when used alone.

However, when combined, these strategies form a coordinated search framework within the FB/NW mechanism. Specifically, the adaptive weighted Euclidean distance improves the discrimination ability of FB/NW selection, resulting in more reliable search directions. Based on this improved selection, the archive-based mechanism enhances the diversity of candidate neighbors, particularly in the early stage, while the elite-guided mechanism accelerates convergence in the later stage. In addition, the NW-based reverse-reinforcement mechanism is adaptively triggered when stagnation occurs, enabling the algorithm to escape local optima without continuously disrupting convergence.

Through this staged and functionally differentiated cooperation, EFNO achieves a dynamic and adaptive balance between exploration and exploitation. Therefore, the superior performance of EFNO is not simply due to the accumulation of individual strategies, but

rather the result of their coordinated interaction within the FB/NW framework, leading to a more stable and effective optimization process.

Table 2. Ablation study results.

Function	Index	FNO	FNO_A	FNO_E	FNO_I	FNO_AE	EFNO
F1	Best	$1.019 \times 10^{+3}$	$7.766 \times 10^{+2}$	$4.053 \times 10^{+2}$	$1.401 \times 10^{+3}$	$\mathbf{3.558 \times 10^{+2}}$	$4.258 \times 10^{+2}$
	Mean	$4.034 \times 10^{+3}$	$3.672 \times 10^{+3}$	$\mathbf{1.244 \times 10^{+3}}$	$4.629 \times 10^{+3}$	$1.464 \times 10^{+3}$	$1.826 \times 10^{+3}$
	Std	$2.078 \times 10^{+3}$	$2.205 \times 10^{+3}$	$1.106 \times 10^{+3}$	$2.139 \times 10^{+3}$	$\mathbf{8.116 \times 10^{+2}}$	$1.403 \times 10^{+3}$
	Rank	5	4	1	6	2	3
F2	Best	$4.449 \times 10^{+2}$	$4.078 \times 10^{+2}$	$4.038 \times 10^{+2}$	$4.044 \times 10^{+2}$	$4.449 \times 10^{+2}$	$\mathbf{4.035 \times 10^{+2}}$
	Mean	$4.559 \times 10^{+2}$	$4.525 \times 10^{+2}$	$\mathbf{4.501 \times 10^{+2}}$	$4.528 \times 10^{+2}$	$4.576 \times 10^{+2}$	$4.567 \times 10^{+2}$
	Std	$\mathbf{1.100 \times 10^{+1}}$	$1.501 \times 10^{+1}$	$1.969 \times 10^{+1}$	$1.311 \times 10^{+1}$	$1.198 \times 10^{+1}$	$1.531 \times 10^{+1}$
	Rank	5	3	1	6	2	4
F3	Best	$6.001 \times 10^{+2}$	$6.001 \times 10^{+2}$	$6.003 \times 10^{+2}$	$6.001 \times 10^{+2}$	$6.006 \times 10^{+2}$	$\mathbf{6.001 \times 10^{+2}}$
	Mean	$6.015 \times 10^{+2}$	$6.011 \times 10^{+2}$	$6.031 \times 10^{+2}$	$6.019 \times 10^{+2}$	$6.035 \times 10^{+2}$	$\mathbf{6.011 \times 10^{+2}}$
	Std	$1.847 \times 10^{+0}$	$1.350 \times 10^{+0}$	$2.353 \times 10^{+0}$	$1.114 \times 10^{+0}$	$2.565 \times 10^{+0}$	$\mathbf{9.416 \times 10^{-1}}$
	Rank	3	2	5	4	6	1
F4	Best	$8.129 \times 10^{+2}$	$8.169 \times 10^{+2}$	$8.119 \times 10^{+2}$	$8.139 \times 10^{+2}$	$8.139 \times 10^{+2}$	$\mathbf{8.090 \times 10^{+2}}$
	Mean	$8.293 \times 10^{+2}$	$8.329 \times 10^{+2}$	$8.321 \times 10^{+2}$	$8.305 \times 10^{+2}$	$8.287 \times 10^{+2}$	$\mathbf{8.262 \times 10^{+2}}$
	Std	$9.155 \times 10^{+0}$	$1.130 \times 10^{+1}$	$1.148 \times 10^{+1}$	$9.334 \times 10^{+0}$	$\mathbf{8.576 \times 10^{+0}}$	$9.308 \times 10^{+0}$
	Rank	4	6	5	2	3	1
F5	Best	$9.042 \times 10^{+2}$	$9.015 \times 10^{+2}$	$9.034 \times 10^{+2}$	$9.029 \times 10^{+2}$	$9.025 \times 10^{+2}$	$\mathbf{9.005 \times 10^{+2}}$
	Mean	$9.278 \times 10^{+2}$	$9.170 \times 10^{+2}$	$9.274 \times 10^{+2}$	$9.307 \times 10^{+2}$	$9.284 \times 10^{+2}$	$\mathbf{9.156 \times 10^{+2}}$
	Std	$4.181 \times 10^{+1}$	$1.156 \times 10^{+1}$	$2.586 \times 10^{+1}$	$3.030 \times 10^{+1}$	$3.253 \times 10^{+1}$	$1.535 \times 10^{+1}$
	Rank	5	2	4	6	3	1
F6	Best	$1.869 \times 10^{+3}$	$1.884 \times 10^{+3}$	$1.867 \times 10^{+3}$	$\mathbf{1.866 \times 10^{+3}}$	$2.133 \times 10^{+3}$	$1.889 \times 10^{+3}$
	Mean	$\mathbf{4.266 \times 10^{+3}}$	$5.274 \times 10^{+3}$	$4.708 \times 10^{+3}$	$5.823 \times 10^{+3}$	$5.035 \times 10^{+3}$	$4.568 \times 10^{+3}$
	Std	$3.319 \times 10^{+3}$	$3.726 \times 10^{+3}$	$2.919 \times 10^{+3}$	$3.961 \times 10^{+3}$	$\mathbf{2.873 \times 10^{+3}}$	$2.893 \times 10^{+3}$
	Rank	1	4	3	5	6	2
F7	Best	$2.023 \times 10^{+3}$	$2.024 \times 10^{+3}$	$2.027 \times 10^{+3}$	$2.025 \times 10^{+3}$	$\mathbf{2.021 \times 10^{+3}}$	$2.023 \times 10^{+3}$
	Mean	$2.044 \times 10^{+3}$	$2.046 \times 10^{+3}$	$2.051 \times 10^{+3}$	$2.048 \times 10^{+3}$	$2.052 \times 10^{+3}$	$\mathbf{2.039 \times 10^{+3}}$
	Std	$1.392 \times 10^{+1}$	$1.500 \times 10^{+1}$	$1.591 \times 10^{+1}$	$1.862 \times 10^{+1}$	$1.983 \times 10^{+1}$	$\mathbf{1.198 \times 10^{+1}}$
	Rank	2	4	6	3	5	1
F8	Best	$2.223 \times 10^{+3}$	$2.222 \times 10^{+3}$	$\mathbf{2.221 \times 10^{+3}}$	$2.222 \times 10^{+3}$	$2.221 \times 10^{+3}$	$2.221 \times 10^{+3}$
	Mean	$2.227 \times 10^{+3}$	$2.227 \times 10^{+3}$	$2.228 \times 10^{+3}$	$2.227 \times 10^{+3}$	$2.227 \times 10^{+3}$	$\mathbf{2.226 \times 10^{+3}}$
	Std	$2.650 \times 10^{+0}$	$\mathbf{2.591 \times 10^{+0}}$	$2.122 \times 10^{+1}$	$3.061 \times 10^{+0}$	$5.350 \times 10^{+0}$	$3.539 \times 10^{+0}$
	Rank	5	4	6	3	2	1
F9	Best	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$\mathbf{2.481 \times 10^{+3}}$
	Mean	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$2.481 \times 10^{+3}$	$\mathbf{2.481 \times 10^{+3}}$
	Std	2.271×10^{-2}	3.364×10^{-2}	$\mathbf{2.446 \times 10^{-03}}$	2.494×10^{-2}	6.741×10^{-3}	1.540×10^{-2}
	Rank	4	5	2	6	3	1
F10	Best	$2.500 \times 10^{+3}$	$2.500 \times 10^{+3}$	$2.501 \times 10^{+3}$	$2.500 \times 10^{+3}$	$2.500 \times 10^{+3}$	$\mathbf{2.500 \times 10^{+3}}$
	Mean	$2.575 \times 10^{+3}$	$2.554 \times 10^{+3}$	$2.689 \times 10^{+3}$	$2.696 \times 10^{+3}$	$2.668 \times 10^{+3}$	$\mathbf{2.546 \times 10^{+3}}$
	Std	$1.581 \times 10^{+2}$	$1.208 \times 10^{+2}$	$3.731 \times 10^{+2}$	$4.245 \times 10^{+2}$	$3.493 \times 10^{+2}$	$\mathbf{7.128 \times 10^{+1}}$
	Rank	4	2	6	5	3	1
F11	Best	$\mathbf{2.600 \times 10^{+3}}$	$\mathbf{2.600 \times 10^{+3}}$	$\mathbf{2.600 \times 10^{+3}}$	$\mathbf{2.600 \times 10^{+3}}$	$\mathbf{2.600 \times 10^{+3}}$	$\mathbf{2.600 \times 10^{+3}}$
	Mean	$2.903 \times 10^{+3}$	$\mathbf{2.880 \times 10^{+3}}$	$2.913 \times 10^{+3}$	$2.920 \times 10^{+3}$	$2.893 \times 10^{+3}$	$2.920 \times 10^{+3}$
	Std	$\mathbf{6.687 \times 10^{+1}}$	$9.965 \times 10^{+1}$	$7.303 \times 10^{+1}$	$7.611 \times 10^{+1}$	$1.081 \times 10^{+2}$	$7.611 \times 10^{+1}$
	Rank	5	1	3	6	2	4
F12	Best	$2.947 \times 10^{+3}$	$2.947 \times 10^{+3}$	$\mathbf{2.944 \times 10^{+3}}$	$2.953 \times 10^{+3}$	$2.954 \times 10^{+3}$	$2.953 \times 10^{+3}$
	Mean	$2.980 \times 10^{+3}$	$\mathbf{2.974 \times 10^{+3}}$	$2.991 \times 10^{+3}$	$2.980 \times 10^{+3}$	$2.992 \times 10^{+3}$	$2.980 \times 10^{+3}$
	Std	$2.051 \times 10^{+1}$	$1.749 \times 10^{+1}$	$3.000 \times 10^{+1}$	$2.013 \times 10^{+1}$	$2.662 \times 10^{+1}$	$\mathbf{1.311 \times 10^{+1}}$
	Rank	3	1	5	4	6	2
Mean Rank		3.83	3.17	3.92	4.67	3.58	**1.83**
Final Ranking		4	2	5	6	3	1

5.1.3. Convergence Behavior Analysis

To comprehensively evaluate the dynamic search behavior of the proposed EFNO algorithm during the optimization process, this section provides a systematic analysis from the perspective of convergence behavior. Compared with traditional performance evaluation methods that rely only on the best fitness value, convergence behavior analysis can more intuitively reveal the evolution of exploration and exploitation abilities during the search process, as well as the influence of different mechanisms on the optimization process.

In this study, several representative functions with different characteristics are selected from the CEC2017 benchmark suite for testing. The specific analysis includes the following steps: (1) Record the search trajectories of population individuals to illustrate the overall search movement trend of EFNO in the solution space. (2) Introduce the Population Dimension Diversity Index [36] to construct curves that reflect changes in exploration and exploitation abilities in order to analyze the search characteristics of the algorithm at different stages. The definition of this metric is given in Equations (24)–(26). (3) Select five representative individuals and plot their trajectory evolution in the first dimension to reflect individual-level search dynamics. And finally, (4) use the convergence curve of the best individual to evaluate the convergence speed and stability of the algorithm.

$$Div = \frac{1}{D}\sum_{j=1}^{D}\frac{1}{N}\left(\sum_{i=1}^{nPop}\left|median\left(X_j\right) - X_{i,j}\right|\right) \tag{24}$$

$$Exploration\% = \frac{Div}{Div_{max}} \tag{25}$$

$$Exploitation\% = \frac{\left|Div - Div_{max}\right|}{Div_{max}} \tag{26}$$

where Div_{max} denotes the population's dimensional diversity; $X_{i,j}$ denotes the position of the ith individual on the jth dimension; and $median\left(X_j\right)$ is the median of the positions of all individuals on the jth dimension.

The experimental results are presented in Figure 4. The first column shows the three-dimensional landscapes of the benchmark functions; the second column illustrates the population search trajectories, with the red stars representing the global optimum; the third column presents the exploration–exploitation curves; the fourth column shows the trajectories of the five best-performing search agents in the first dimension; and the fifth column displays the convergence curves of the best individual.

The results indicate that EFNO exhibits strong dynamic search behavior throughout the optimization process. The search trajectories demonstrate that the algorithm can effectively explore the search space in the early stage and gradually converge toward high-quality regions. The exploration–exploitation curves further confirm that the algorithm maintains relatively high diversity in the early stage and gradually strengthens its exploitation ability in the later stage.

The individual trajectory analysis shows that different search agents are well dispersed in the global search space and progressively gather toward the optimal region, demonstrating strong cooperative search ability. At the same time, the convergence curve of the best individual indicates that EFNO achieves fast convergence while maintaining stable convergence behavior.

Overall, the proposed multi-strategy cooperative mechanism effectively improves the balance between exploration and exploitation, thereby enhancing the global optimization performance of the algorithm.

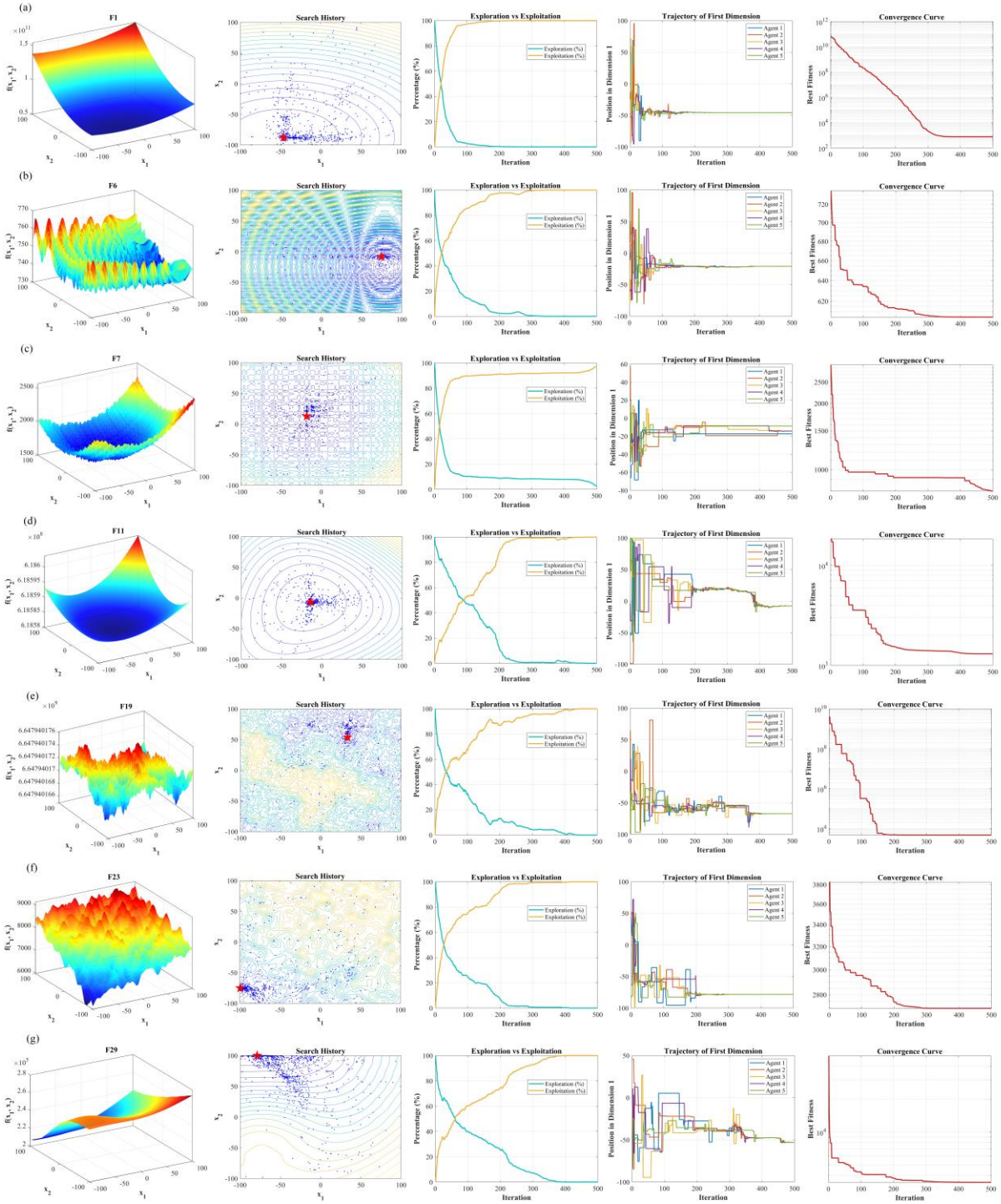


Figure 4. Results of the analysis of EFNO convergence behavior: (**a**) F1, (**b**) F6, (**c**) F7, (**d**) F11, (**e**) F19, (**f**) F23, and (**g**) F29.

5.1.4. Diversity Analysis

To quantitatively analyze the exploration–exploitation behavior of EFNO, the population diversity curves of EFNO and FNO are compared on representative functions from different categories of the CEC2017 benchmark suite, including unimodal (F1), simple multimodal (F8), hybrid (F18), and composition (F22) functions.

The diversity is measured using the same indicator as employed in the algorithm, i.e., the mean standard deviation-based diversity, δ_{div}, ensuring consistency between the

internal mechanism and external analysis. All experiments are conducted for 500 iterations and repeated 10 times.

The experimental results are shown in Figure 5. From the results, different diversity evolution patterns can be observed across function types. On unimodal and simple multimodal functions, EFNO exhibits a faster reduction in diversity compared to FNO, indicating stronger exploitation capability. On more complex hybrid and composition functions, EFNO tends to maintain relatively higher diversity in the early stage, followed by a gradual reduction, and may exhibit diversity re-increase in later iterations.

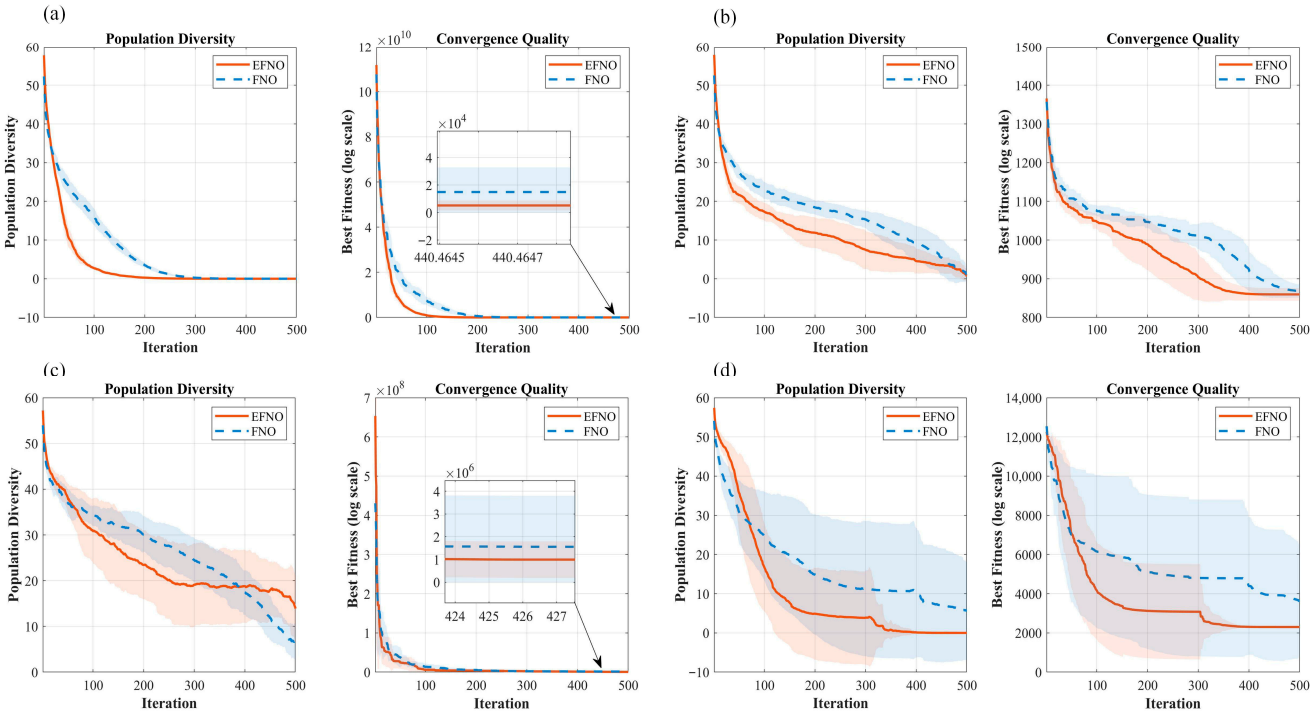


Figure 5. Results of the diversity analysis experiment: (**a**) F1, (**b**) F8, (**c**) F18, and (**d**) F22.

This behavior can be explained by the collaborative effect of the proposed strategies. The elite-guided mechanism accelerates convergence by guiding individuals toward high-quality regions, leading to a faster decrease in diversity on simpler landscapes. Meanwhile, the archive-enhanced NW selection improves the quality of candidate neighbors, enabling more effective exploration in complex landscapes. In addition, the NW-based reverse-reinforcement mechanism is triggered when stagnation occurs and population diversity becomes low. This mechanism can introduce perturbations to help the population escape local optima, which explains the observed increase in diversity in the later stage on hybrid and composition functions.

By jointly analyzing diversity and optimization performance, it can be observed that EFNO consistently achieves better solutions than FNO across all tested functions, despite not always maintaining higher diversity. This indicates that EFNO emphasizes effective and structured exploration rather than maintaining high population diversity, resulting in a more efficient exploration–exploitation balance.

5.1.5. Test Results and Analysis on CEC2017

Figure 6 summarizes the Friedman average ranking results of all algorithms on the 30-dimensional, 50-dimensional, and 100-dimensional CEC2017 benchmark functions. For detailed results, see Tables A5–A7 in Appendix A.2. In these tables, the best results are highlighted in bold and underlined. The symbols "+/=/−" are determined based on

the Wilcoxon rank-sum test results and the average fitness values, indicating that EFNO performs better than, equal to, or worse than the compared algorithm, respectively.

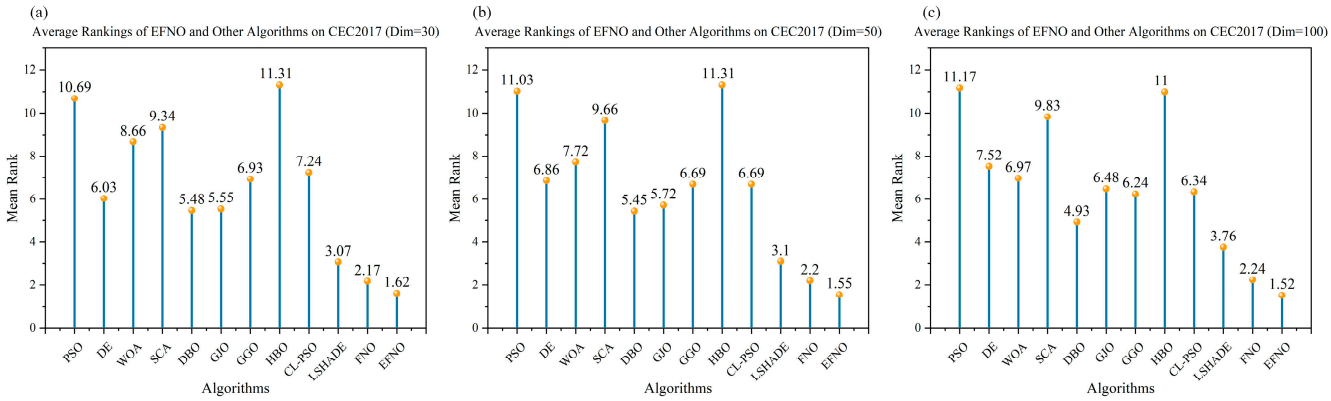


Figure 6. Average Friedman rankings of EFNO and other algorithms on CEC2017: (**a**) Dim = 30, (**b**) Dim = 50, and (**c**) Dim = 100.

From the experimental results, EFNO achieves superior performance on 19, 18, and 22 functions out of the total 29 benchmark functions in the 30D, 50D, and 100D cases, respectively. The average Friedman rankings are 1.62, 1.55, and 1.52, respectively, ranking first among all compared methods. This demonstrates that EFNO performs stably across low-, medium-, and high-dimensional problems, confirming its strong robustness. Since EFNO's core mechanisms are population-based and do not rely on problem-specific structures, the method can be extended to higher-dimensional scenarios. However, similar to most swarm intelligence algorithms, the computational cost increases with dimensionality, and further improvements may be required for very large-scale optimization problems.

Figures 7–12 present the convergence curves and boxplots of all algorithms under different dimensions on the CEC2017 benchmark suite. From the convergence curves, EFNO achieves faster convergence speed and higher solution accuracy than the original FNO on most test functions. This indicates that the proposed strategies effectively enhance the global search capability and improve the convergence efficiency of the algorithm.

5.1.6. Performance Analysis of Compared Algorithms

From the experimental results on the CEC2017 benchmark suite, the compared algorithms exhibit distinct optimization behaviors in terms of convergence speed, solution quality, and robustness.

Classical algorithms, including PSO, DE, WOA, and SCA, generally demonstrate stable but relatively limited performance. Although these methods are effective in simple or low-dimensional problems, they tend to suffer from premature convergence and insufficient exploration when dealing with high-dimensional or complex multimodal landscapes.

The recently proposed algorithms, such as DBO, GJO, GGO, and HBO, show certain improvements over classical methods due to enhanced exploration mechanisms. However, their performance is not consistently stable across all test functions. In particular, these methods may still experience performance degradation in high-dimensional scenarios or when the problem landscape becomes highly irregular.

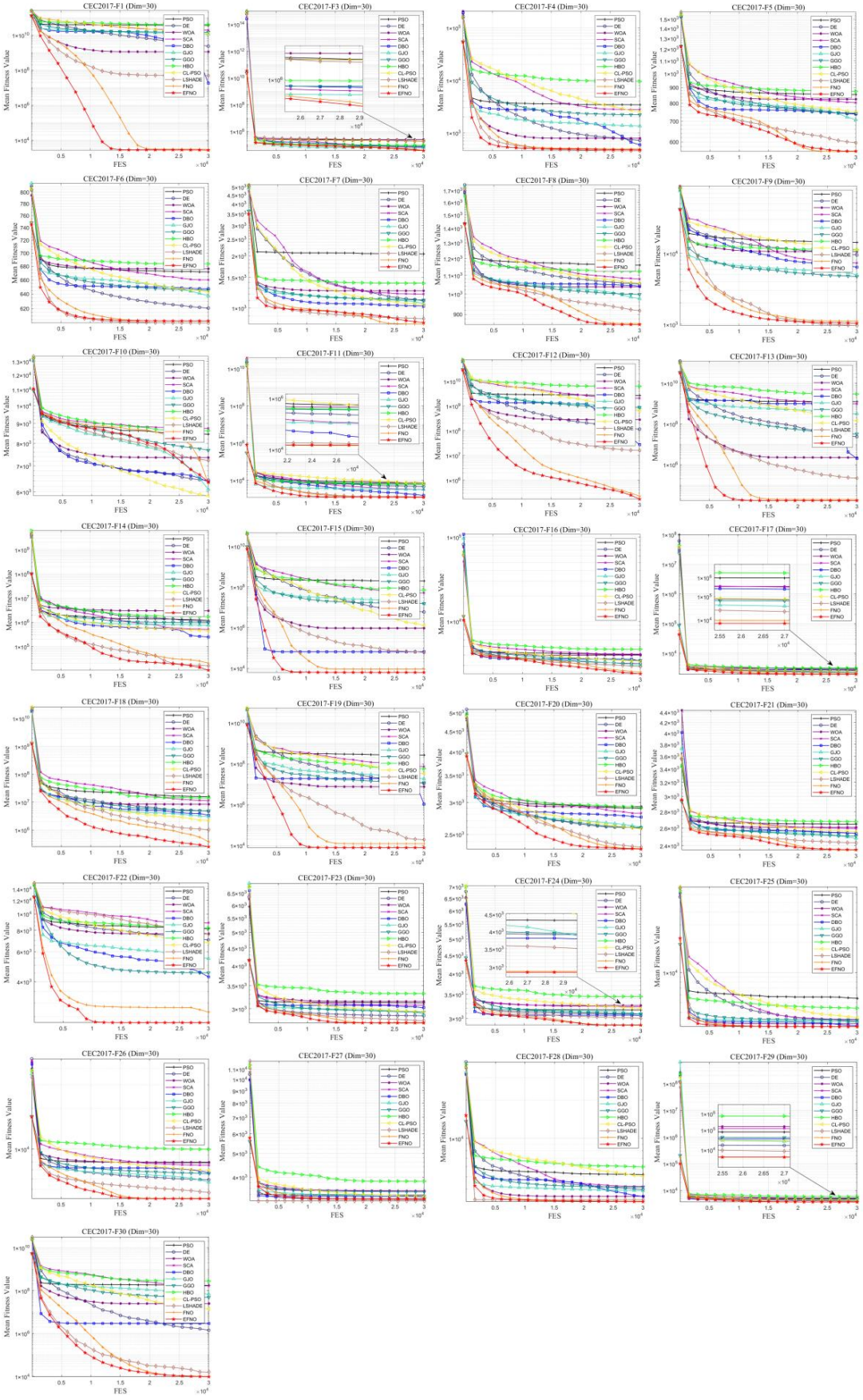


Figure 7. Iteration curves for EFNO and other algorithms on CEC2017 (Dim = 30).

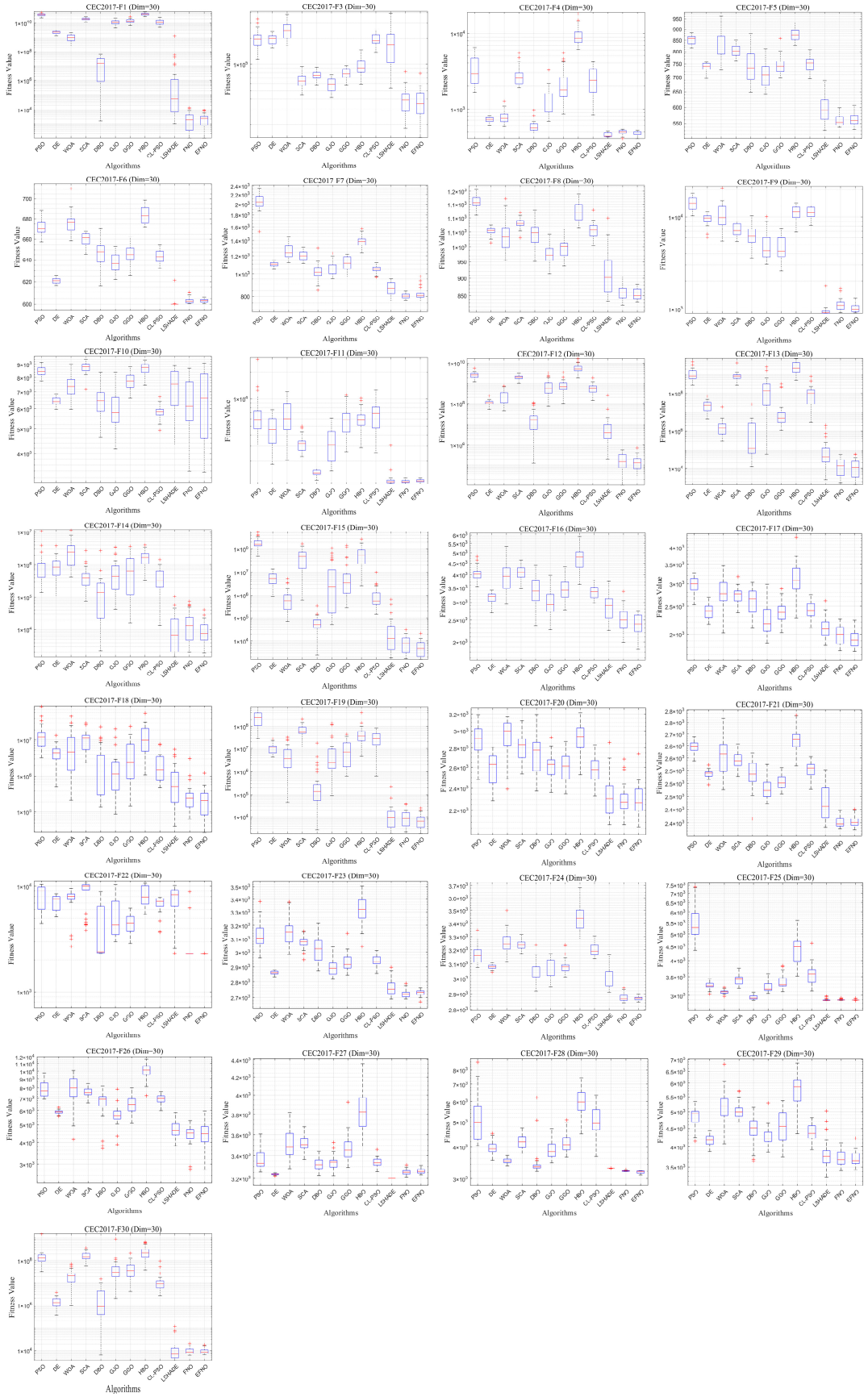


Figure 8. Box plots for EFNO and other algorithms on CEC2017 (Dim = 30).

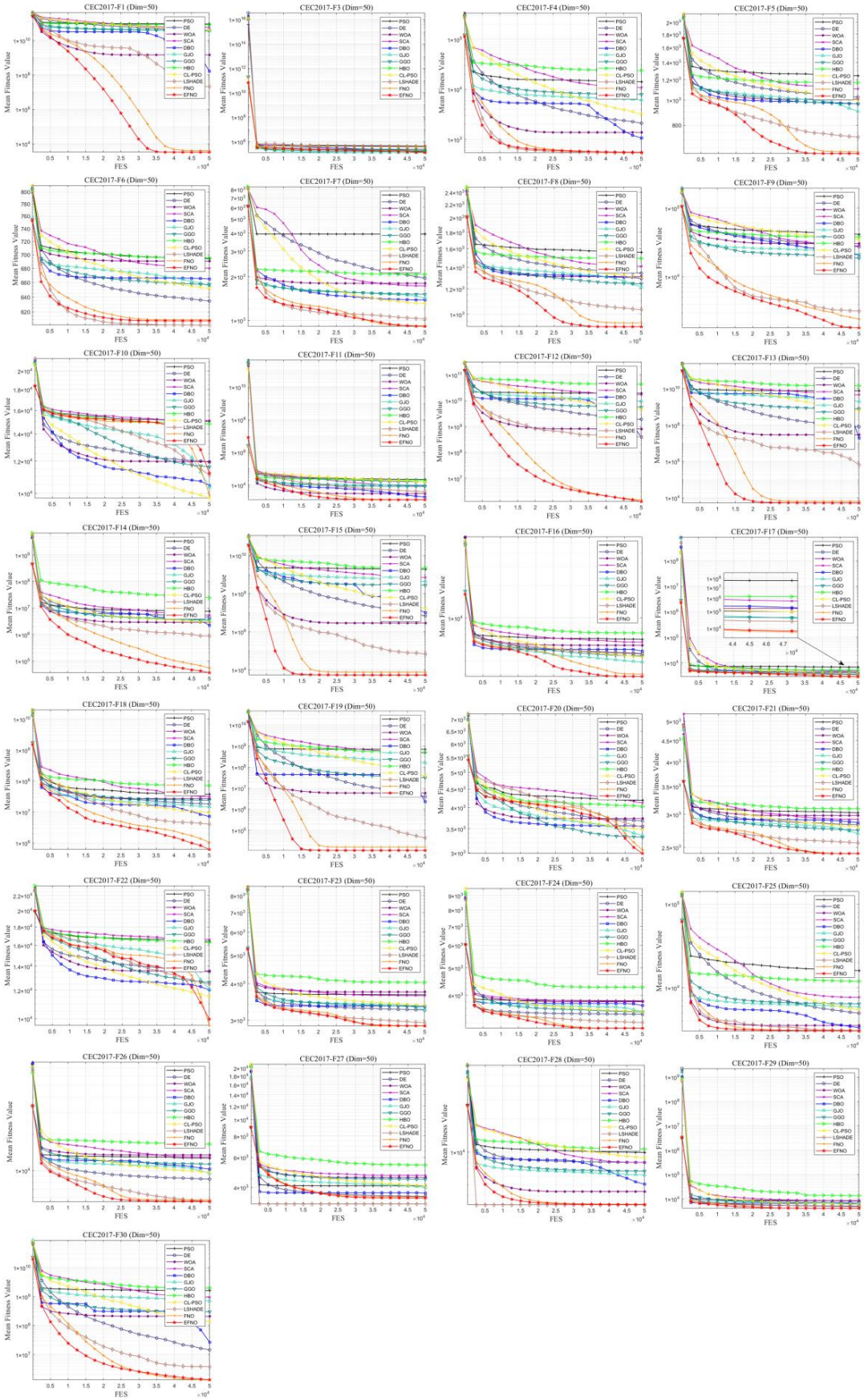


Figure 9. Iteration curves for EFNO and other algorithms on CEC2017 (Dim = 50).

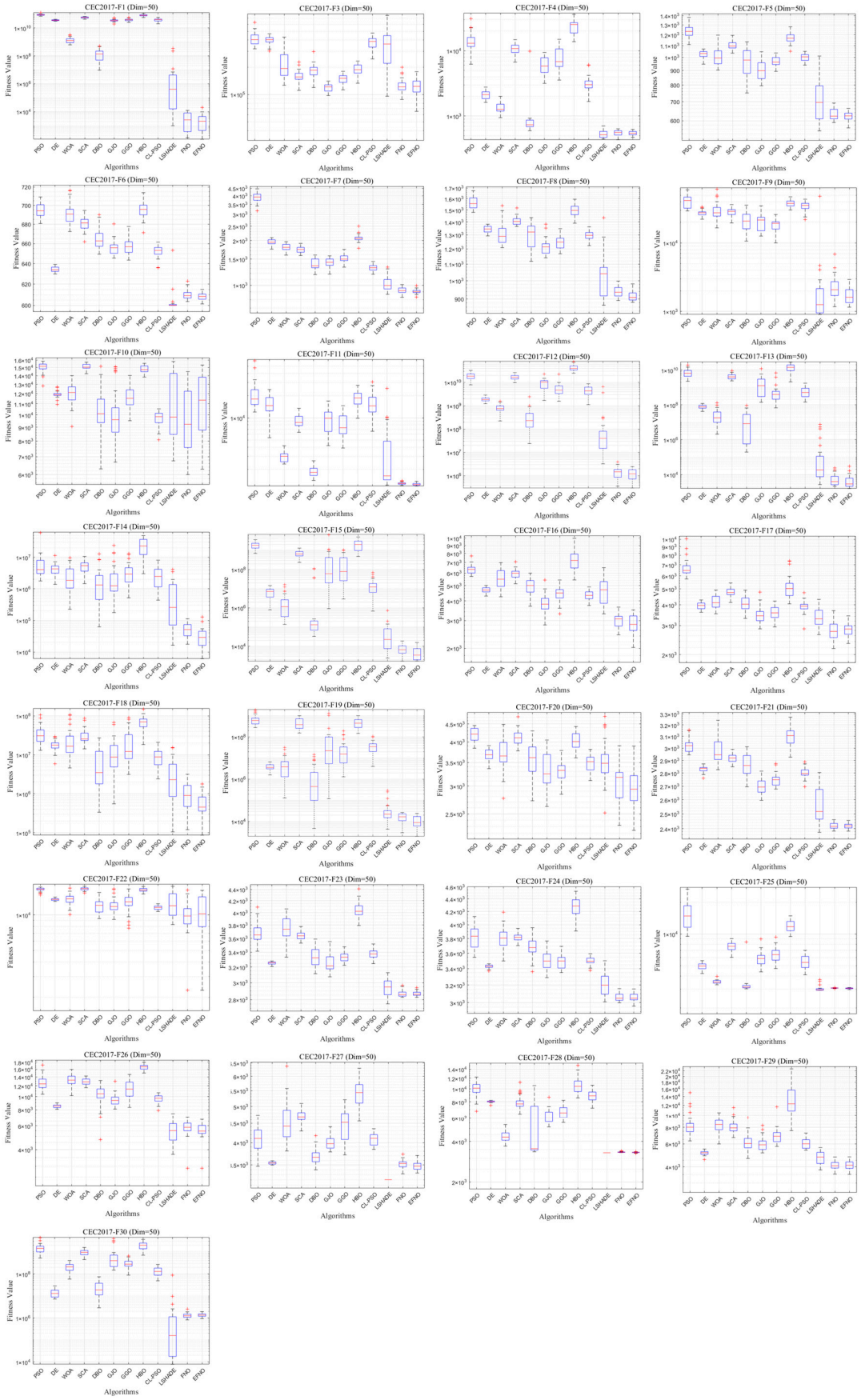


Figure 10. Box plots for EFNO and other algorithms on CEC2017 (Dim = 50).

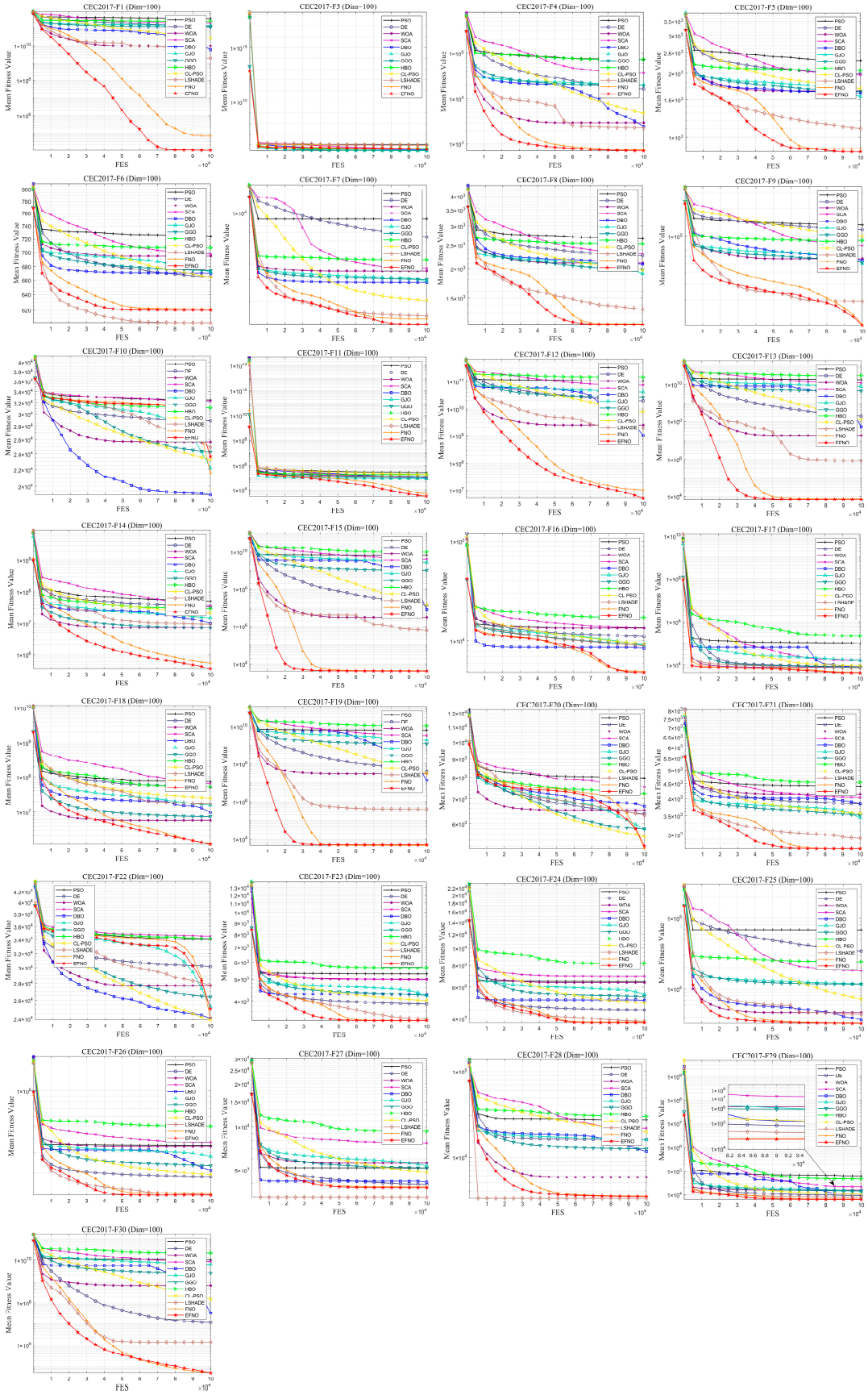


Figure 11. Iteration curves for EFNO and other algorithms on CEC2017 (Dim = 100).

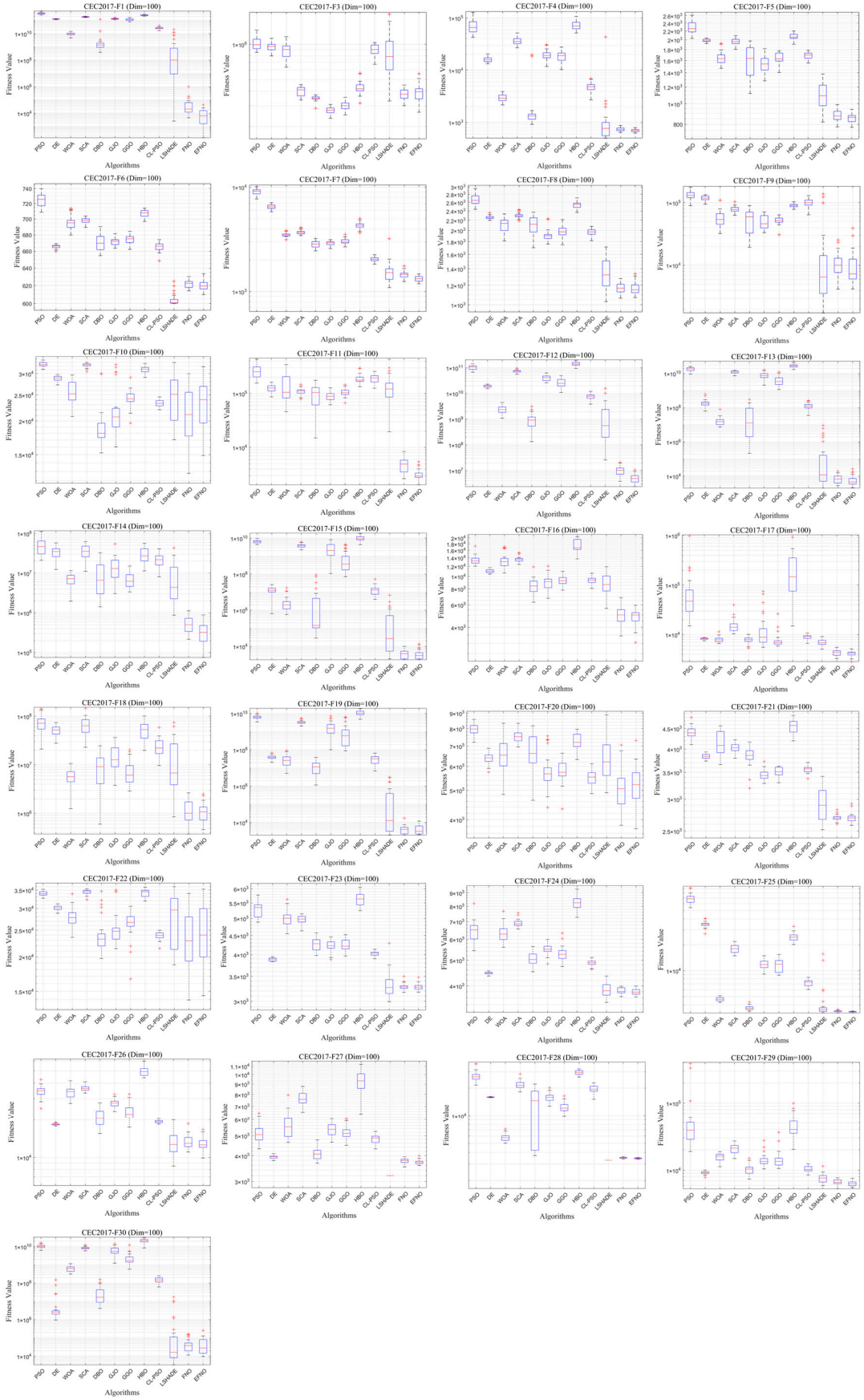


Figure 12. Box plots for EFNO and other algorithms on CEC2017 (Dim = 100).

Advanced variants, including CLPSO and LSHADE, exhibit stronger optimization capabilities. Among them, LSHADE is widely recognized as a highly competitive algorithm, especially in CEC benchmark competitions, due to its adaptive parameter control and population size reduction strategy. It achieves relatively stable and competitive results across different types of problems and therefore serves as a strong baseline in this study.

In addition to comparisons with other algorithms, it is important to analyze the performance of the original FNO. The original FNO benefits from the farthest better and nearest worse mechanism, which helps maintain population diversity and avoid low-quality regions. However, its performance degrades in high-dimensional spaces due to the distance concentration phenomenon, which weakens the discriminative ability of distance-based selection. Moreover, the effectiveness of the nearest worse avoidance mechanism diminishes as the population converges.

Compared with the above methods, the proposed EFNO demonstrates superior and more stable optimization performance across most benchmark functions. In particular, EFNO outperforms strong baselines such as LSHADE in terms of convergence accuracy and robustness in the majority of test cases. This improvement can be attributed to the adaptive weighted Euclidean distance mechanism, which enhances distance discrimination in high-dimensional spaces; the elite-guided archive reinforcement strategy, which introduces stage-wise and differentiated population guidance; and the NW-based reverse-reinforcement mechanism, which provides an effective way to escape local optima.

Overall, while existing algorithms exhibit strengths in specific scenarios, none of them consistently achieve superior performance across all problem types. In contrast, EFNO shows a more balanced and robust optimization capability, demonstrating its effectiveness and generalization ability.

5.2. Real-World Engineering Design Problems

To verify the effectiveness of the proposed algorithm in real-world applications, two classic engineering optimization problems [37,38] are selected: the spring design problem and the welded-beam design problem. By comparing the optimal solutions and stability of different algorithms, the performance of EFNO in constrained optimization problems is analyzed.

5.2.1. Experimental Configuration for Engineering Optimization Problems

The classical algorithms PSO and WOA, along with the recently well-performing algorithms in Section 5.1, including DBO, GJO, GGO, and the original FNO, are selected as comparative methods for evaluation against EFNO. The parameter settings of all algorithms are kept the same as in Section 5.1. The maximum number of function evaluations is set to 20,000. Each engineering problem is independently run 30 times, and the mean value, variance, median, best value, worst value, and Friedman ranking of the fitness results are reported.

5.2.2. Tension/Compression Spring Design Problem

The tension/compression spring design problem is a classical structural engineering optimization problem. Its objective is to minimize the weight of the spring while satisfying four constraints: deflection, shear stress, surge frequency, and outer diameter.

To solve this problem, three design variables are considered: wire diameter (d), mean coil diameter (D), and number of active coils (N). The schematic diagram of the spring and its parameters is shown in Figure 13. The mathematical model of this problem is formulated as follows:

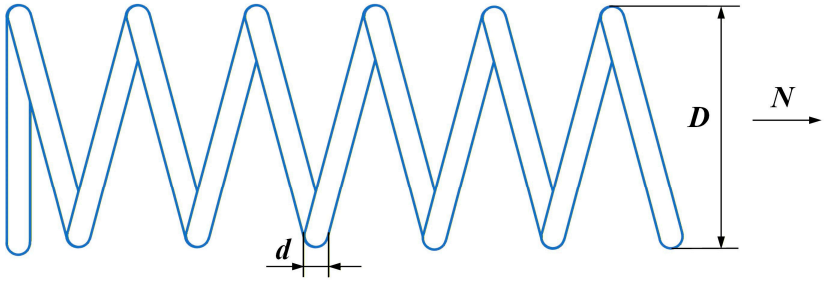


Figure 13. Schematic diagram of tension/compression spring design problem.

Consider

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} d & D & N \end{bmatrix}, \tag{27}$$

Minimize

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2, \tag{28}$$

Subject to

$$\begin{aligned} g_1(\vec{x}) &= 1 - \frac{x_2^2 x_3}{71{,}785x_1^4} \leq 0, \\ g_2(\vec{x}) &= \frac{4x_2^2 - x_1x_2}{12{,}566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\ g_3(\vec{x}) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\ g_4(\vec{x}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0. \end{aligned} \tag{29}$$

Parameter range: $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.30, 2 \leq x_3 \leq 15$.

The experimental results are shown in Tables 3 and 4, and the best results are highlighted in bold and underlined. Table 3 shows that when $d = 0.051830827$, $D = 0.360137849$, and $N = 11.0912945$, the EFNO algorithm yields an optimal spring weight of 0.012665655. Table 4 shows that, compared with other algorithms, the EFNO algorithm achieved the best results across all metrics and ranked first overall in the Friedman ranking.

Table 3. Optimal solutions for tension/compression spring design problem using EFNO and other algorithms.

Algorithms	x_1	x_2	x_3	Result
PSO	0.05	0.317333025	14.0454916	0.012729411
WOA	0.051337782	0.348315261	11.7995607	0.012668114
DBO	0.05	0.317425416	14.0277697	0.012719054
GJO	0.051327943	0.347845528	11.8438219	0.012686743
GGO	0.050118303	0.319823111	13.8707848	0.012749729
FNO	0.052209405	0.3693654	10.5842488	0.012670128
EFNO	**0.051830827**	**0.360137849**	**11.0912945**	**0.012665655**

Table 4. Statistical results for EFNO and other algorithms in the design of tension/compression spring.

Index	PSO	WOA	DBO	GJO	GGO	FNO	EFNO
Mean	0.013538	0.013841	0.014097	0.012871	0.013256	0.012876	**0.012778**
Std	0.001839	0.001164	0.001970	0.000253	0.000458	0.000300	**0.000106**
Median	0.012765	0.013540	0.013193	0.012755	0.013042	0.012778	**0.012748**
Best	0.012729	0.012668	0.012719	0.012687	0.012750	0.012670	**0.012666**
Worst	0.018348	0.017037	0.017784	0.013609	0.014345	0.014249	**0.013112**
Friedman Value	3.8	5.4	4.93	3	5.2	3.1	**2.56**

5.2.3. Welded-Beam Design Problem

The welded-beam design problem is a real-world engineering optimization problem with complex constraints. Its objective is to minimize the fabrication cost of the welded beam.

The schematic diagram of the problem is shown in Figure 14. During the optimization process, several constraints must be satisfied, including shear stress (τ), bending stress (σ), buckling load (P_c), beam deflection (δ), and side constraints.

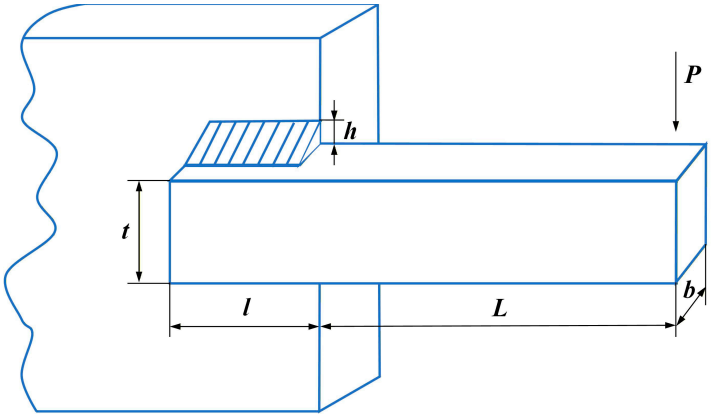


Figure 14. Schematic diagram of welded-beam design problem.

In addition, the fabrication cost of the welded beam is related to four key design variables: width (t), weld thickness (h), length (l), and thickness (b). The mathematical model of this problem is formulated as follows:

Consider

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} = \begin{bmatrix} h & l & t & b \end{bmatrix}, \tag{30}$$

Minimize

$$f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2), \tag{31}$$

Subject to

$$\begin{aligned} &g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0, \\ &g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0, \\ &g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0, \\ &g_4(\vec{x}) = x_1 - x_4 \le 0, \\ &g_5(\vec{x}) = p - p_c(\vec{x}) \le 0, \\ &g_6(\vec{x}) = 0.125 - x_1 \le 0, \\ &g_7(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0. \end{aligned} \tag{32}$$

Here,

$$\begin{aligned} &\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}, \\ &\tau' = \frac{P}{\sqrt{2} x_1 x_2}, \ \tau'' = \frac{MR}{J}, \\ &M = P\left(L + \frac{x_2}{2}\right), \\ &R = \sqrt{\frac{X_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \\ &J = 2\left\{ \sqrt{2} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\}, \\ &\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}, \ \delta(\vec{x}) = \frac{4PL^3}{E x_3^3 x_4}, \\ &P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \\ &P = 6000lb, \ L = 14in, \ \delta_{max} = 0.25in, \\ &E = 30 \times 10^6 psi, \ G = 12 \times 10^6 psi, \\ &\tau_{max} = 13,600 psi, \ \sigma_{max} = 30,000 psi. \end{aligned} \tag{33}$$

Parameter range: $0.1 \le x_1, x_4 \le 2, 0.1 \le x_2, x_3 \le 10$.

The experimental results are shown in Tables 5 and 6, and the best results are highlighted in bold and underlined. Table 5 shows that when the parameters are set to $h = 0.2057, l = 3.2349, t = 9.0366$, and $b = 0.2057$, the minimum manufacturing cost of 1.692768 is obtained for the optimization of the welded-beam problem. Although both FNO and EFNO achieved the same minimum manufacturing cost, Table 6 shows that EFNO obtained the optimal results across all metrics, ranking first among all algorithms. This indicates that ENO continues to perform robustly in welded-beam design problems, and that EFNO demonstrates superior performance in solving real-world engineering problems.

Table 5. Optimal solutions for welded-beam design problem using EFNO and other algorithms.

Algorithms	x_1	x_2	x_3	x_4	Result
PSO	0.201110132	3.38355152	9.14854279	0.205398751	1.72271144
WOA	0.202459838	3.25117478	9.22759272	0.204795888	1.71564458
DBO	0.205729713	3.23491909	9.036621	0.205729772	1.69276884
GJO	0.204736856	3.25639641	9.03629628	0.205835335	1.69496442
GGO	0.202640161	3.32812957	8.97477887	0.208651804	1.71208108
FNO	**0.20572964**	**3.23491931**	**9.03662391**	**0.20572964**	**1.69276826**
EFNO	**0.20572964**	**3.23491931**	**9.03662391**	**0.20572964**	**1.69276826**

Table 6. Statistical results comparing EFNO with other algorithms in the design of welded beam.

Index	PSO	WOA	DBO	GJO	GGO	FNO	EFNO
Mean	1.792598	2.642857	1.734264	1.700728	1.937068	1.704986	**1.693289**
Std	0.028715	0.979292	0.039094	0.005235	0.283902	0.042672	**0.001186**
Median	1.800423	2.259409	1.735427	1.698697	1.822548	1.692839	**1.692768**
Best	1.722711	1.715645	1.692769	1.694964	1.712081	**1.692768**	**1.692768**
Worst	1.837339	4.925633	1.778588	1.716047	2.728656	1.911971	**1.697432**
Friedman Value	5.23	6.7	3.63	3.2	5.67	2.2	**1.37**

5.3. Path-Planning Experiments

To further verify the applicability of the proposed EFNO algorithm in real-world optimization problems, it is applied to a two-dimensional path-planning problem on grid maps. Multiple environments with different map sizes and obstacle densities are constructed to evaluate the search performance and path optimization capability of the algorithm under complex constraints.

5.3.1. Mathematical Modeling

In the grid map environment, the path-planning problem of a mobile robot can be formulated as an optimization problem to minimize its length under a goal orientation condition [39,40]. Assuming there is a path composed of multiple connected nodes sequentially,

$$P = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \tag{34}$$

Then, the total path length is given by

$$d = \sum_{k=1}^{n} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \tag{35}$$

where n represents the number of path nodes, (x_k, y_k) denotes the coordinates of the kth node in the path, and d is the total path length.

Therefore, the path-planning problem aims to find the shortest feasible path from the starting point to the goal node under specified constraints. EFNO performs optimization

efficiently in complex discrete spaces through a combination of global exploration and local exploitation, enabling it to obtain optimal or near-optimal solutions.

5.3.2. Experimental Configuration for Path Planning

To conduct a comprehensive evaluation of the proposed algorithm in path-planning tasks, six random grid-based environments with different complexity levels were constructed, as illustrated in Figure 15. The map sizes are 20×20, 30×30, and 50×50, with obstacle densities of 20% (simple environments, Maps 1–3) and 40% (complex environments, Maps 4–6).

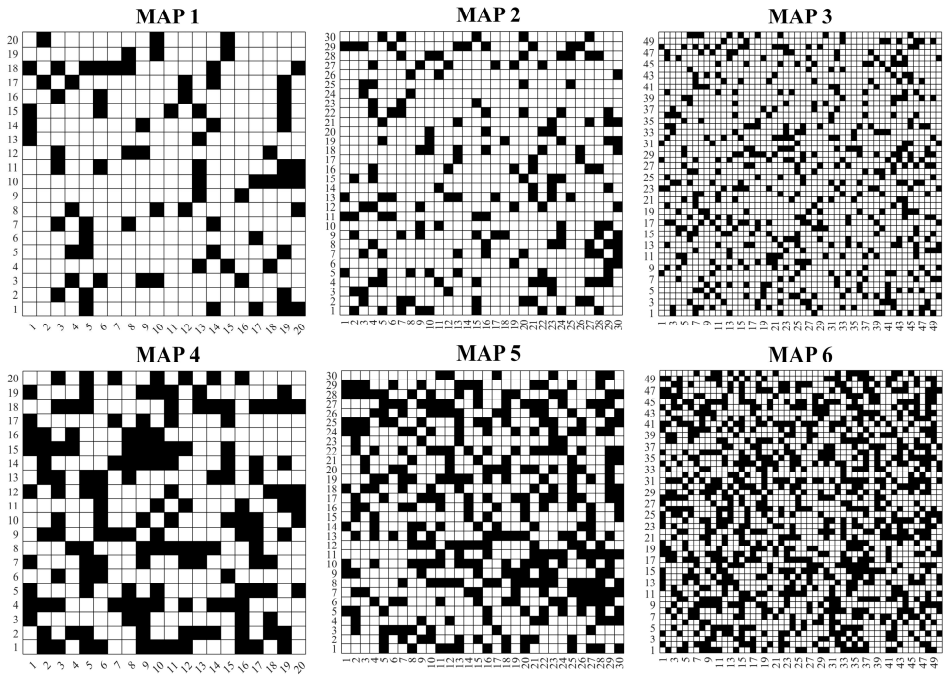


Figure 15. Path-planning simulation maps.

Consistent with Section 5.2, the proposed method is compared with PSO, WOA, DBO, GJO, GGO, and the original FNO under the same parameter settings. Each algorithm is independently executed 30 times in each environment. The best fitness, mean fitness, standard deviation, and Friedman ranking are recorded to evaluate performance.

5.3.3. Path Planning in Simple Environments

To analyze the performance of the algorithms in low-obstacle-density environments, Maps 1–3 shown in Figure 15 were used in the experiments. The corresponding results are presented in Table 7. The best values in the table are highlighted in bold and underlined. Figure 16 illustrates the average convergence curves and visualizations of the optimal path-planning results obtained by all algorithms in these simple environments.

According to Table 7 and Figure 16, EFNO achieves the shortest path distance and the lowest average path length across all map scales, ranking first among all compared algorithms. Compared with the original FNO, EFNO exhibits a steeper decline in the convergence curves during the early stage and converges more rapidly. In terms of path quality, EFNO generates smoother paths with fewer redundant nodes.

Overall, in low-complexity environments, EFNO demonstrates excellent convergence speed and strong path optimization capability.

Table 7. Path-planning results for EFNO and other algorithms in simple environments.

Map	Index	PSO	WOA	DBO	GJO	GGO	FNO	EFNO
1	Best	30.970563	34.384776	31.556349	32.970563	31.798990	29.213203	**28.627417**
	Mean	33.711984	35.517297	35.364437	34.310022	35.392051	31.108634	**29.842782**
	Std	1.649468	**0.213899**	0.741028	0.908963	0.694665	1.134728	0.937748
	Rank	3	7	5	4	6	2	1
2	Best	46.526912	48.526912	48.526912	48.526912	48.526912	45.355339	**43.941126**
	Mean	50.882309	56.114661	53.981328	52.771277	53.244644	48.645219	**48.095709**
	Std	2.726041	2.179723	3.083867	2.545090	2.856724	**1.931777**	2.200368
	Rank	3	7	6	4	5	2	1
3	Best	79.396970	83.396970	83.154329	83.154329	83.3969670	75.154329	**74.568543**
	Mean	85.347391	83.396970	83.380794	83.299913	83.3969670	86.855072	**81.976392**
	Std	3.273195	$\mathbf{2.8908 \times 10^{-14}}$	0.061560	0.120901	$\mathbf{2.8908 \times 10^{-14}}$	4.352370	3.344619
	Rank	5	4	3	2	5	6	1
Mean Rank		3.67	6	4.67	3.33	5.33	3.33	**1**
Final Ranking		4	7	5	2	6	2	1

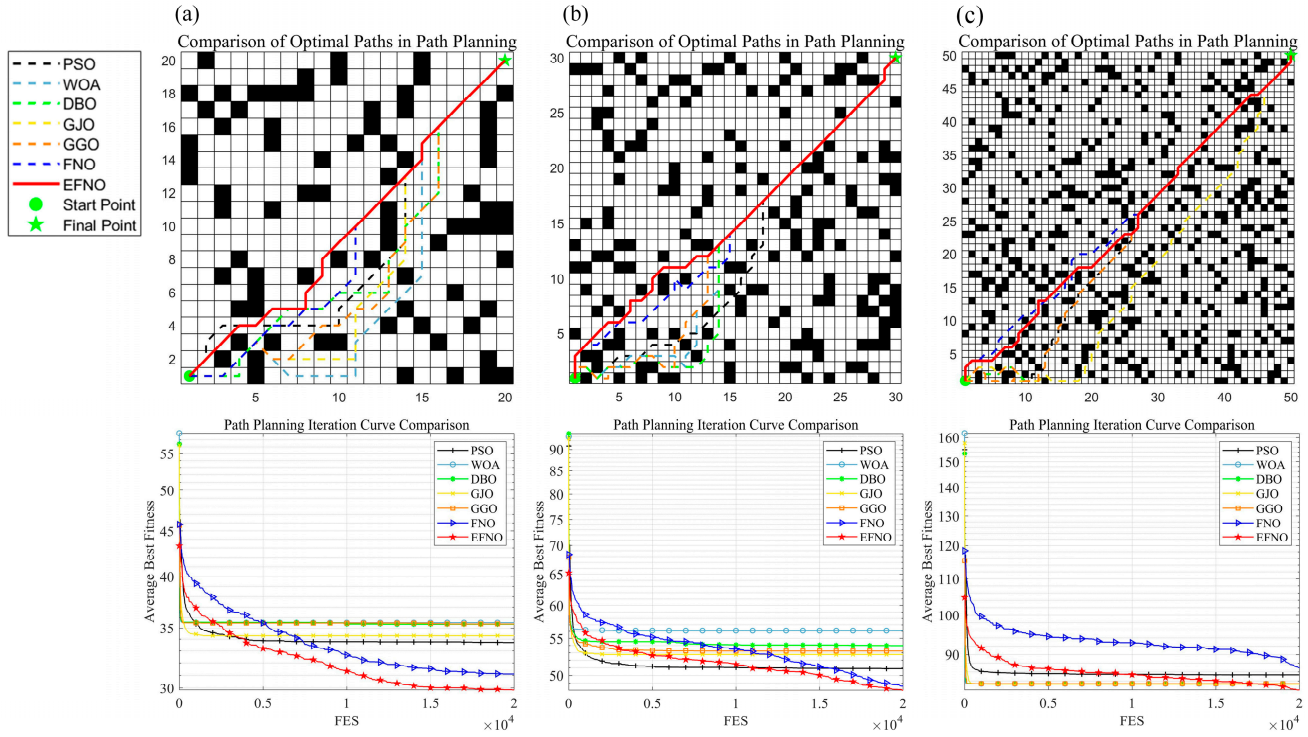


Figure 16. Optimal path results and average iteration curves for EFNO and other algorithms in simple environments: (**a**) Map 1, (**b**) Map 2, and (**c**) Map 3.

5.3.4. Path Planning in Complex Environments

Furthermore, all algorithms are tested in complex environments with a 40% obstacle ratio (Maps 4 to 6) to evaluate their performance under highly constrained conditions. The results are presented in Table 8. Figure 17 shows the average convergence curves and the best path-planning results in complex environments.

The results indicate that EFNO achieves the best performance on all evaluation metrics in the 30 × 30 and 50 × 50 maps. In the 20 × 20 map, it obtains the best average path length. Overall, EFNO ranks first among all compared algorithms. Compared with other methods, EFNO can more effectively bypass dense obstacle regions and avoid being trapped in local optima. The convergence curves show that EFNO maintains high diversity while still achieving stable convergence. Compared with the original FNO, both path quality and stability are significantly improved.

In summary, in complex environments, EFNO demonstrates stronger global search capability and robustness, and it can effectively handle the difficulties caused by high obstacle density.

Table 8. Path-planning results for EFNO and other algorithms in complex environments.

Map	Index	PSO	WOA	DBO	GJO	GGO	FNO	EFNO
4	Best	30.384776	32.384776	31.798990	33.213203	31.798990	**29.798990**	31.213203
	Mean	35.609040	38.797027	37.231342	37.620717	38.141799	33.961087	**33.416316**
	Std	2.141990	**4.083728**	3.313875	3.834821	4.530061	1.874958	0.6892041
	Rank	3	7	6	5	4	2	1
5	Best	47.941126	50.284271	48.284271	53.941126	47.112698	**47.112698**	**47.112698**
	Mean	52.387453	57.442107	56.013105	58.410568	55.385491	48.363189	**47.429856**
	Std	2.617846	4.218027	4.347761	2.295208	5.432239	1.313195	**0.435578**
	Rank	3	6	5	7	4	2	1
6	Best	78.911688	89.254834	81.740115	94.325902	84.083261	82.426407	**77.740115**
	Mean	89.554091	109.065837	103.686513	102.849998	104.44746	88.605731	**85.047559**
	Std	7.747546	6.336858	10.075351	4.515949	10.220065	5.072855	**3.430896**
	Rank	2	7	5	4	6	3	1
Mean Rank		2.67	6.67	5.33	5.33	4.67	2.33	**1**
Final Ranking		3	7	4	4	6	2	1

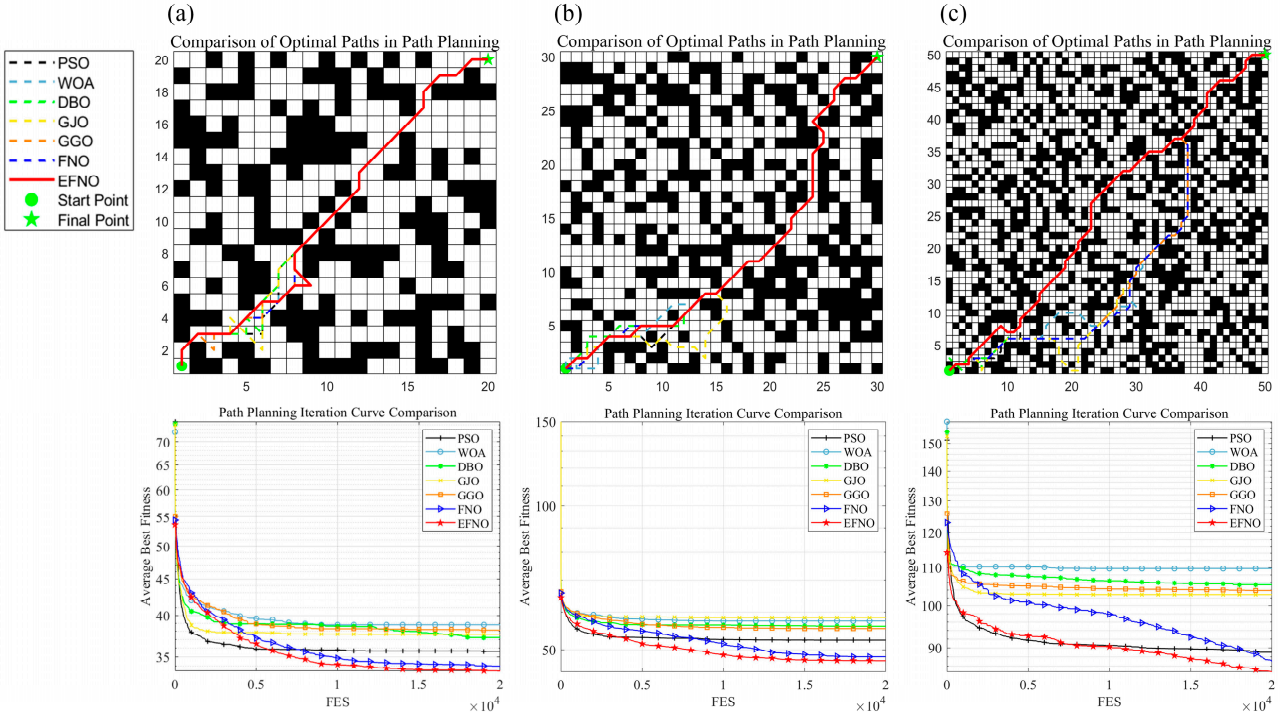


Figure 17. Optimal path results and average iteration curves for EFNO and other algorithms in complex environments: (**a**) Map 4, (**b**) Map 5, and (**c**) Map 6.

6. Conclusions

In this paper, an Enhanced Farthest Better or Nearest Worse Optimizer (EFNO) is proposed to improve the performance of the original FNO algorithm. By analyzing the intrinsic characteristics of the search mechanism, three effective strategies are introduced, namely an adaptive weighted Euclidean distance mechanism, an elite-guided archive reinforcement strategy for stage-based differentiated population guidance, and an NW-based reverse-reinforcement mechanism, to enhance the balance between exploration and exploitation.

To validate the effectiveness of the proposed EFNO, extensive experiments are conducted on the CEC2017 and CEC2022 benchmark functions. Parameter sensitivity analysis and ablation studies confirm the contribution of each strategy, while statistical tests demonstrate the superior performance and robustness of EFNO compared with 11 classical and State-of-the-Art algorithms. Furthermore, convergence behavior analysis reveals that EFNO achieves a better balance between population diversity and convergence speed.

Lastly, the proposed method was applied to two classical engineering design problems and a robot path-planning task. In the path-planning experiments, six grid-based maps with different sizes and complexity levels were constructed. The results show that EFNO can obtain high-quality solutions with strong stability, demonstrating its effectiveness and applicability in solving practical optimization problems. In addition, the proposed EFNO is independent of the specific encoding scheme used in the application. Although a grid-based representation is adopted in this study, the algorithm can be extended to other representations, such as continuous or graph-based encodings, provided that the solution can be expressed in vector form.

Further improvements in the future may include an adaptive strategy selection mechanism and extension to multi-objective or dynamic optimization problems with EFNO. Furthermore, in the future, it is necessary to conduct research on dynamic and uncertain environment obstacles' distribution changes with time, which requires more reliable adaptive optimization ability.

Author Contributions: Conceptualization, X.T.; methodology, X.T. and C.J.; software, X.T.; validation, C.J. and P.Q.; data curation, P.Z. and P.Q.; writing—original draft preparation, X.T.; writing—review and editing, X.T. and C.J.; visualization, P.Z.; funding acquisition, X.T. and P.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 52565031), the Science and Technology Program Projects of Meishan Municipal Science and Technology Bureau (No. 2024KJZD162), and Guizhou Provincial Department of Science and Technology Projects (No. [2023] G094, No. [2023] G125, and No. [2024] K154).

Data Availability Statement: The original contributions presented in this study are included in this article. Further inquiries can be directed to the corresponding author.

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

Appendix A

Appendix A.1

Time Complexity Analysis of EFNO

Let the number of iterations be T, the population size be $nPop$, and the problem dimension be D. Let $f(\cdot)$ denote the computational cost of a single fitness evaluation. Then, the time complexity of the original FNO algorithm is given by $O\left(T\cdot\left(nPop^2\cdot D + N\cdot f(\cdot)\right)\right)$.

Compared with FNO, EFNO introduces several additional computational costs, which are analyzed as follows:

- Adaptive Euclidean distance mechanism:
  This introduces an additional preprocessing cost of $O(nPop \cdot D)$.
- NW archive reinforcement mechanism and stage-wise differentiated guidance:
  (1) Each individual sample's k solutions from the archive to compute a weighted distance, resulting in a total cost of $O(nPop \cdot D)$;
  (2) Additional vector operations are performed for approximately N/2 NW individuals and non-NW individuals, leading to an additional cost of $O(nPop \cdot D)$.
- NW Reverse-Reinforcement Mechanism:

(1) Diversity checking in each iteration cost, $O(D)$;
(2) When triggered, the mechanism incurs a cost of $O(nPop \cdot D + N \cdot f(\cdot))$. However, since the triggering condition is strict, the amortized cost is negligible.

Overall, EFNO introduces an additional computational cost of $O(nPop \cdot D)$ per iteration. When N = 30, we have $nPop \cdot D \ll \cdot nPop^2 \cdot D$, meaning that the linear overhead introduced by the improved mechanisms is negligible compared with the main loop. Therefore, the overall time complexity of EFNO remains $O\left(T \cdot \left(nPop^2 \cdot D + N \cdot f(\cdot)\right)\right)$.

Appendix A.2

Table A1. CEC2022 benchmark functions.

Type	No.	Functions	$F_i^* = F_i(X^*)$
Unimodal Functions	1	Shifted and full Rotated Zakharov Function	300
Basic Functions	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid Functions	6	Hybrid Function 1 (N = 3)	1800
	7	Hybrid Function 2(N = 3)	2000
	8	Hybrid Function 3 (N = 3)	2200
Composition Functions	9	Composition Function 1 (N = 3)	2300
	10	Composition Function 2 (N = 4)	2400
	11	Composition Function 3 (N = 5)	2600
	12	Composition Function 4 (N = 6)	2700
Search Range: $[-100, 100]^D$			

Table A2. CEC2017 benchmark functions.

	No.	Functions	$F_i^* = F_i(X^*)$
Unimodal Functions	1	Shifted and Rotated Bent Cigar Function	100
	3	Shifted and Rotated Zakharov Function	200
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock's Function	300
	5	Shifted and Rotated Rastrigin's Function	400
	6	Shifted and Rotated Expanded Scaffer's F6 Function	500
	7	Shifted and Rotated Lunacek Bi_Rastrigin Function	600
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	700
	9	Shifted and Rotated Levy Function	800
	10	Shifted and Rotated Schwefel's Function	900
Hybrid Functions	11	Hybrid Function 1 (N = 3)	1000
	12	Hybrid Function 2 (N = 3)	1100
	13	Hybrid Function 3 (N = 3)	1200
	14	Hybrid Function 4 (N = 4)	1300
	15	Hybrid Function 5 (N = 4)	1400
	16	Hybrid Function 6 (N = 4)	1500
	17	Hybrid Function 6 (N = 5)	1600
	18	Hybrid Function 6 (N = 5)	1700
	19	Hybrid Function 6 (N = 5)	1800
	20	Hybrid Function 6 (N = 6)	1900

Table A2. *Cont.*

	No.	Functions	$F_i^* = F_i(X^*)$
Composition Functions	21	Composition Function 1 (N = 3)	2000
	22	Composition Function 2 (N = 3)	2100
	23	Composition Function 3 (N = 4)	2200
	24	Composition Function 4 (N = 4)	2300
	25	Composition Function 5 (N = 5)	2400
	26	Composition Function 6 (N = 5)	2500
	27	Composition Function 7 (N = 6)	2600
	28	Composition Function 8 (N = 6)	2700
	29	Composition Function 9 (N = 3)	2800
	30	Composition Function 10 (N = 3)	2900
Search Range: $[-100, 100]^D$			

Table A3. Sensitivity analysis results for parameter *nPop* on selected CEC2017 benchmark functions.

Function	Index	*nPop*			
		10	30	50	100
F1	Best	$1.000 \times 10^{+2}$	$1.037 \times 10^{+2}$	$1.005 \times 10^{+2}$	$1.003 \times 10^{+2}$
	Mean	$3.212 \times 10^{+3}$	$2.397 \times 10^{+3}$	$2.833 \times 10^{+3}$	$2.938 \times 10^{+3}$
	Std	$4.446 \times 10^{+3}$	$2.678 \times 10^{+3}$	$2.397 \times 10^{+3}$	$2.450 \times 10^{+3}$
	Rank	4	1	2	3
F3	Best	$2.034 \times 10^{+3}$	$1.488 \times 10^{+4}$	$2.990 \times 10^{+4}$	$3.228 \times 10^{+4}$
	Mean	$1.361 \times 10^{+4}$	$3.775 \times 10^{+4}$	$4.673 \times 10^{+4}$	$5.253 \times 10^{+4}$
	Std	$8.896 \times 10^{+3}$	$1.539 \times 10^{+4}$	$1.059 \times 10^{+4}$	$1.061 \times 10^{+4}$
	Rank	1	2	3	4
F4	Best	$4.041 \times 10^{+2}$	$4.651 \times 10^{+2}$	$4.000 \times 10^{+2}$	$4.685 \times 10^{+2}$
	Mean	$4.882 \times 10^{+2}$	$4.905 \times 10^{+2}$	$4.908 \times 10^{+2}$	$4.992 \times 10^{+2}$
	Std	$3.304 \times 10^{+1}$	$1.833 \times 10^{+1}$	$2.580 \times 10^{+1}$	$1.943 \times 10^{+1}$
	Rank	1	2	3	4
F5	Best	$5.557 \times 10^{+2}$	$5.338 \times 10^{+2}$	$5.304 \times 10^{+2}$	$5.251 \times 10^{+2}$
	Mean	$6.316 \times 10^{+2}$	$5.617 \times 10^{+2}$	$5.504 \times 10^{+2}$	$5.659 \times 10^{+2}$
	Std	$4.039 \times 10^{+1}$	$1.562 \times 10^{+1}$	$1.538 \times 10^{+1}$	$3.579 \times 10^{+1}$
	Rank	4	2	1	3
F7	Best	$8.091 \times 10^{+2}$	$7.592 \times 10^{+2}$	$7.529 \times 10^{+2}$	$8.028 \times 10^{+2}$
	Mean	$1.081 \times 10^{+3}$	$8.093 \times 10^{+2}$	$8.186 \times 10^{+2}$	$8.759 \times 10^{+2}$
	Std	$1.823 \times 10^{+2}$	$4.218 \times 10^{+1}$	$5.938 \times 10^{+1}$	$4.611 \times 10^{+1}$
	Rank	4	1	2	3
F8	Best	$8.706 \times 10^{+2}$	$8.308 \times 10^{+2}$	$8.269 \times 10^{+2}$	$8.204 \times 10^{+2}$
	Mean	$9.208 \times 10^{+2}$	$8.561 \times 10^{+2}$	$8.517 \times 10^{+2}$	$8.846 \times 10^{+2}$
	Std	$3.211 \times 10^{+1}$	$1.322 \times 10^{+1}$	$1.900 \times 10^{+1}$	$4.794 \times 10^{+1}$
	Rank	4	2	1	3
F10	Best	$3.119 \times 10^{+3}$	$3.025 \times 10^{+3}$	$3.945 \times 10^{+3}$	$6.849 \times 10^{+3}$
	Mean	$4.968 \times 10^{+3}$	$7.203 \times 10^{+3}$	$7.861 \times 10^{+3}$	$8.434 \times 10^{+3}$
	Std	$1.431 \times 10^{+3}$	$1.616 \times 10^{+3}$	$1.204 \times 10^{+3}$	$4.401 \times 10^{+2}$
	Rank	1	2	3	4
F12	Best	$3.512 \times 10^{+4}$	$1.765 \times 10^{+4}$	$2.675 \times 10^{+4}$	$5.104 \times 10^{+4}$
	Mean	$2.119 \times 10^{+5}$	$1.574 \times 10^{+5}$	$3.820 \times 10^{+5}$	$6.387 \times 10^{+5}$
	Std	$2.018 \times 10^{+5}$	$1.504 \times 10^{+5}$	$2.677 \times 10^{+5}$	$5.718 \times 10^{+5}$
	Rank	2	1	3	4

Table A3. *Cont.*

Function	Index	*nPop*			
F13	Best	$1.898 \times 10^{+3}$	$1.477 \times 10^{+3}$	$1.417 \times 10^{+3}$	$1.416 \times 10^{+3}$
	Mean	$1.495 \times 10^{+4}$	$1.133 \times 10^{+4}$	$1.316 \times 10^{+4}$	$1.030 \times 10^{+4}$
	Std	$1.466 \times 10^{+4}$	$9.242 \times 10^{+3}$	$1.097 \times 10^{+4}$	$7.550 \times 10^{+3}$
	Rank	4	2	3	1
F15	Best	$1.681 \times 10^{+3}$	$1.568 \times 10^{+3}$	$1.515 \times 10^{+3}$	$1.548 \times 10^{+3}$
	Mean	$8.629 \times 10^{+3}$	$5.862 \times 10^{+3}$	$6.830 \times 10^{+3}$	$7.973 \times 10^{+3}$
	Std	$8.964 \times 10^{+3}$	$5.633 \times 10^{+3}$	$8.403 \times 10^{+3}$	$8.193 \times 10^{+3}$
	Rank	4	1	2	3
F16	Best	$2.023 \times 10^{+3}$	$1.863 \times 10^{+3}$	$1.974 \times 10^{+3}$	$1.936 \times 10^{+3}$
	Mean	$2.516 \times 10^{+3}$	$2.471 \times 10^{+3}$	$2.690 \times 10^{+3}$	$2.820 \times 10^{+3}$
	Std	$2.872 \times 10^{+2}$	$3.537 \times 10^{+2}$	$3.862 \times 10^{+2}$	$5.281 \times 10^{+2}$
	Rank	2	1	3	4
F21	Best	$2.363 \times 10^{+3}$	$2.332 \times 10^{+3}$	$2.327 \times 10^{+3}$	$2.327 \times 10^{+3}$
	Mean	$2.422 \times 10^{+3}$	$2.357 \times 10^{+3}$	$2.352 \times 10^{+3}$	$2.375 \times 10^{+3}$
	Std	$3.532 \times 10^{+1}$	$1.475 \times 10^{+1}$	$2.221 \times 10^{+1}$	$3.440 \times 10^{+1}$
	Rank	4	2	1	3
F25	Best	$2.884 \times 10^{+3}$	$2.884 \times 10^{+3}$	$2.884 \times 10^{+3}$	$2.886 \times 10^{+3}$
	Mean	$2.904 \times 10^{+3}$	$2.896 \times 10^{+3}$	$2.893 \times 10^{+3}$	$2.890 \times 10^{+3}$
	Std	$2.526 \times 10^{+1}$	$1.812 \times 10^{+1}$	$1.256 \times 10^{+1}$	$7.431 \times 10^{+0}$
	Rank	4	2	3	1
F26	Best	$2.800 \times 10^{+3}$	$2.800 \times 10^{+3}$	$2.800 \times 10^{+3}$	$2.900 \times 10^{+3}$
	Mean	$5.860 \times 10^{+3}$	$4.343 \times 10^{+3}$	$4.106 \times 10^{+3}$	$4.140 \times 10^{+3}$
	Std	$1.555 \times 10^{+3}$	$9.798 \times 10^{+2}$	$5.940 \times 10^{+2}$	$3.998 \times 10^{+2}$
	Rank	4	3	1	2
F30	Best	$6.197 \times 10^{+3}$	$6.168 \times 10^{+3}$	$6.582 \times 10^{+3}$	$7.852 \times 10^{+3}$
	Mean	$9.936 \times 10^{+3}$	$1.036 \times 10^{+4}$	$1.138 \times 10^{+4}$	$2.354 \times 10^{+4}$
	Std	$3.961 \times 10^{+3}$	$3.181 \times 10^{+3}$	$3.217 \times 10^{+3}$	$1.744 \times 10^{+4}$
	Rank	1	2	3	4
Mean Rank		2.93	1.73	2.27	3.07

Table A4. Sensitivity analysis results for parameter k on selected CEC2017 benchmark functions.

Function	Index	*k*				
		1	2	3	5	7
F1	Best	$1.169 \times 10^{+2}$	$1.019 \times 10^{+2}$	$1.139 \times 10^{+2}$	$1.000 \times 10^{+2}$	$1.801 \times 10^{+2}$
	Mean	$3.892 \times 10^{+3}$	$4.150 \times 10^{+3}$	$2.321 \times 10^{+3}$	$2.010 \times 10^{+3}$	$3.152 \times 10^{+3}$
	Std	$4.059 \times 10^{+3}$	$3.710 \times 10^{+3}$	$2.165 \times 10^{+3}$	$2.526 \times 10^{+3}$	$2.777 \times 10^{+3}$
	Rank	5	3	2	1	4
F3	Best	$3.461 \times 10^{+4}$	$2.100 \times 10^{+4}$	$1.898 \times 10^{+4}$	$2.243 \times 10^{+4}$	$2.069 \times 10^{+4}$
	Mean	$4.942 \times 10^{+4}$	$4.813 \times 10^{+4}$	$4.372 \times 10^{+4}$	$4.554 \times 10^{+4}$	$4.324 \times 10^{+4}$
	Std	$1.118 \times 10^{+4}$	$1.168 \times 10^{+4}$	$1.190 \times 10^{+4}$	$1.395 \times 10^{+4}$	$1.036 \times 10^{+4}$
	Rank	5	4	1	2	2
F4	Best	$4.005 \times 10^{+2}$	$4.003 \times 10^{+2}$	$4.000 \times 10^{+2}$	$4.001 \times 10^{+2}$	$4.002 \times 10^{+2}$
	Mean	$4.865 \times 10^{+2}$	$4.761 \times 10^{+2}$	$4.718 \times 10^{+2}$	$4.798 \times 10^{+2}$	$4.786 \times 10^{+2}$
	Std	$2.381 \times 10^{+1}$	$3.537 \times 10^{+1}$	$3.323 \times 10^{+1}$	$2.822 \times 10^{+1}$	$4.126 \times 10^{+1}$
	Rank	5	3	1	2	4
F5	Best	$5.309 \times 10^{+2}$	$5.229 \times 10^{+2}$	$5.269 \times 10^{+2}$	$5.249 \times 10^{+2}$	$5.279 \times 10^{+2}$
	Mean	$5.466 \times 10^{+2}$	$5.491 \times 10^{+2}$	$5.470 \times 10^{+2}$	$5.520 \times 10^{+2}$	$5.536 \times 10^{+2}$
	Std	$1.075 \times 10^{+1}$	$1.335 \times 10^{+1}$	$1.031 \times 10^{+1}$	$1.347 \times 10^{+1}$	$1.377 \times 10^{+1}$
	Rank	1	4	1	3	5

Table A4. *Cont.*

Function	Index	k				
F7	Best	$7.628 \times 10^{+2}$	$7.659 \times 10^{+2}$	$7.604 \times 10^{+2}$	$7.563 \times 10^{+2}$	$7.611 \times 10^{+2}$
	Mean	$7.888 \times 10^{+2}$	$7.876 \times 10^{+2}$	$7.821 \times 10^{+2}$	$7.929 \times 10^{+2}$	$7.890 \times 10^{+2}$
	Std	$1.856 \times 10^{+1}$	$1.401 \times 10^{+1}$	$1.496 \times 10^{+1}$	$3.418 \times 10^{+1}$	$1.752 \times 10^{+1}$
	Rank	3	4	1	2	4
F8	Best	$8.279 \times 10^{+2}$	$8.259 \times 10^{+2}$	$8.239 \times 10^{+2}$	$8.308 \times 10^{+2}$	$8.269 \times 10^{+2}$
	Mean	$8.494 \times 10^{+2}$	$8.462 \times 10^{+2}$	$8.480 \times 10^{+2}$	$8.464 \times 10^{+2}$	$8.447 \times 10^{+2}$
	Std	$1.336 \times 10^{+1}$	$1.300 \times 10^{+1}$	$1.407 \times 10^{+1}$	$1.085 \times 10^{+1}$	$1.172 \times 10^{+1}$
	Rank	5	3	4	2	1
F10	Best	$6.185 \times 10^{+3}$	$4.456 \times 10^{+3}$	$3.559 \times 10^{+3}$	$3.626 \times 10^{+3}$	$3.968 \times 10^{+3}$
	Mean	$7.884 \times 10^{+3}$	$7.049 \times 10^{+3}$	$6.589 \times 10^{+3}$	$6.401 \times 10^{+3}$	$7.116 \times 10^{+3}$
	Std	$8.045 \times 10^{+2}$	$1.307 \times 10^{+3}$	$1.612 \times 10^{+3}$	$1.545 \times 10^{+3}$	$1.230 \times 10^{+3}$
	Rank	5	4	2	1	3
F12	Best	$5.096 \times 10^{+4}$	$4.261 \times 10^{+4}$	$2.755 \times 10^{+4}$	$4.307 \times 10^{+4}$	$3.692 \times 10^{+4}$
	Mean	$2.769 \times 10^{+5}$	$2.040 \times 10^{+5}$	$2.435 \times 10^{+5}$	$2.484 \times 10^{+5}$	$2.327 \times 10^{+5}$
	Std	$2.108 \times 10^{+5}$	$1.892 \times 10^{+5}$	$1.865 \times 10^{+5}$	$2.226 \times 10^{+5}$	$1.751 \times 10^{+5}$
	Rank	5	1	3	2	4
F13	Best	$1.488 \times 10^{+3}$	$1.502 \times 10^{+3}$	$1.351 \times 10^{+3}$	$1.491 \times 10^{+3}$	$1.536 \times 10^{+3}$
	Mean	$1.152 \times 10^{+4}$	$1.835 \times 10^{+4}$	$8.494 \times 10^{+3}$	$1.522 \times 10^{+4}$	$1.360 \times 10^{+4}$
	Std	$9.138 \times 10^{+3}$	$1.433 \times 10^{+4}$	$7.788 \times 10^{+3}$	$1.287 \times 10^{+4}$	$1.151 \times 10^{+4}$
	Rank	2	5	1	4	3
F15	Best	$1.549 \times 10^{+3}$	$1.563 \times 10^{+3}$	$1.552 \times 10^{+3}$	$1.557 \times 10^{+3}$	$1.528 \times 10^{+3}$
	Mean	$5.137 \times 10^{+3}$	$5.734 \times 10^{+3}$	$5.608 \times 10^{+3}$	$7.164 \times 10^{+3}$	$7.229 \times 10^{+3}$
	Std	$5.426 \times 10^{+3}$	$4.309 \times 10^{+3}$	$5.561 \times 10^{+3}$	$5.618 \times 10^{+3}$	$7.350 \times 10^{+3}$
	Rank	2	2	1	5	4
F16	Best	$1.645 \times 10^{+3}$	$1.944 \times 10^{+3}$	$1.861 \times 10^{+3}$	$1.958 \times 10^{+3}$	$1.742 \times 10^{+3}$
	Mean	$2.509 \times 10^{+3}$	$2.603 \times 10^{+3}$	$2.375 \times 10^{+3}$	$2.334 \times 10^{+3}$	$2.358 \times 10^{+3}$
	Std	$4.966 \times 10^{+2}$	$4.177 \times 10^{+2}$	$3.088 \times 10^{+2}$	$2.095 \times 10^{+2}$	$3.438 \times 10^{+2}$
	Rank	4	5	3	2	1
F21	Best	$2.327 \times 10^{+3}$	$2.328 \times 10^{+3}$	$2.327 \times 10^{+3}$	$2.321 \times 10^{+3}$	$2.322 \times 10^{+3}$
	Mean	$2.351 \times 10^{+3}$	$2.351 \times 10^{+3}$	$2.350 \times 10^{+3}$	$2.344 \times 10^{+3}$	$2.351 \times 10^{+3}$
	Std	$2.221 \times 10^{+1}$	$1.571 \times 10^{+1}$	$1.536 \times 10^{+1}$	$1.258 \times 10^{+1}$	$1.729 \times 10^{+1}$
	Rank	2	5	3	1	3
F25	Best	$2.884 \times 10^{+3}$	$2.883 \times 10^{+3}$	$2.884 \times 10^{+3}$	$2.883 \times 10^{+3}$	$2.884 \times 10^{+3}$
	Mean	$2.891 \times 10^{+3}$	$2.893 \times 10^{+3}$	$2.890 \times 10^{+3}$	$2.890 \times 10^{+3}$	$2.892 \times 10^{+3}$
	Std	$8.088 \times 10^{+0}$	$1.346 \times 10^{+1}$	$1.032 \times 10^{+1}$	$6.262 \times 10^{+0}$	$1.317 \times 10^{+1}$
	Rank	4	1	2	3	5
F26	Best	$2.800 \times 10^{+3}$	$2.800 \times 10^{+3}$	$3.851 \times 10^{+3}$	$2.900 \times 10^{+3}$	$2.900 \times 10^{+3}$
	Mean	$4.290 \times 10^{+3}$	$4.199 \times 10^{+3}$	$4.462 \times 10^{+3}$	$4.222 \times 10^{+3}$	$4.229 \times 10^{+3}$
	Std	$3.850 \times 10^{+2}$	$3.254 \times 10^{+2}$	$2.932 \times 10^{+2}$	$5.041 \times 10^{+2}$	$3.182 \times 10^{+2}$
	Rank	2	1	5	4	2
F30	Best	$6.434 \times 10^{+3}$	$6.698 \times 10^{+3}$	$6.390 \times 10^{+3}$	$5.393 \times 10^{+3}$	$6.294 \times 10^{+3}$
	Mean	$9.206 \times 10^{+3}$	$9.821 \times 10^{+3}$	$9.436 \times 10^{+3}$	$9.693 \times 10^{+3}$	$9.773 \times 10^{+3}$
	Std	$2.496 \times 10^{+3}$	$4.596 \times 10^{+3}$	$2.744 \times 10^{+3}$	$2.567 \times 10^{+3}$	$3.088 \times 10^{+3}$
	Rank	4	2	1	5	3
Mean Rank		3.6	3.13	2.07	2.6	3.2

Table A5. Comparison results of EFNO with other algorithms on CEC2017 (Dim=30).

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F1	Best	$2.287 \times 10^{+10}$	$1.403 \times 10^{+9}$	$2.710 \times 10^{+8}$	$1.159 \times 10^{+10}$	$1.486 \times 10^{+3}$	$4.707 \times 10^{+9}$	$6.867 \times 10^{+9}$	$2.675 \times 10^{+10}$	$5.216 \times 10^{+9}$	$9.790 \times 10^{+2}$	$1.013 \times 10^{+2}$	**$1.008 \times 10^{+2}$**
	Mean	$3.574 \times 10^{+10}$	$2.302 \times 10^{+9}$	$1.113 \times 10^{+9}$	$1.890 \times 10^{+10}$	$1.790 \times 10^{+7}$	$1.129 \times 10^{+10}$	$1.439 \times 10^{+10}$	$4.031 \times 10^{+10}$	$1.218 \times 10^{+10}$	$4.782 \times 10^{+7}$	$2.963 \times 10^{+3}$	**$2.630 \times 10^{+3}$**
	Std	$6.606 \times 10^{+9}$	$4.187 \times 10^{+8}$	$5.569 \times 10^{+8}$	$3.645 \times 10^{+9}$	$1.941 \times 10^{+7}$	$4.203 \times 10^{+9}$	$4.684 \times 10^{+9}$	$9.151 \times 10^{+9}$	$4.840 \times 10^{+9}$	$2.374 \times 10^{+8}$	$3.565 \times 10^{+3}$	**$2.506 \times 10^{+3}$**
	Rank	11	6	5	10	4	7	9	12	8	3	2	1
F3	Best	$1.138 \times 10^{+5}$	$1.563 \times 10^{+5}$	$1.662 \times 10^{+5}$	$4.409 \times 10^{+4}$	$5.803 \times 10^{+4}$	$4.155 \times 10^{+4}$	$5.799 \times 10^{+4}$	$5.944 \times 10^{+4}$	$1.381 \times 10^{+5}$	$5.322 \times 10^{+4}$	$1.804 \times 10^{+4}$	**$1.410 \times 10^{+4}$**
	Mean	$2.008 \times 10^{+5}$	$1.981 \times 10^{+5}$	$2.499 \times 10^{+5}$	$6.666 \times 10^{+4}$	$7.498 \times 10^{+4}$	$5.880 \times 10^{+4}$	$7.903 \times 10^{+4}$	$9.580 \times 10^{+4}$	$1.976 \times 10^{+5}$	$1.796 \times 10^{+5}$	$3.927 \times 10^{+4}$	**$3.662 \times 10^{+4}$**
	Std	$4.824 \times 10^{+4}$	$2.487 \times 10^{+4}$	$5.594 \times 10^{+4}$	$1.358 \times 10^{+4}$	**$8.990 \times 10^{+3}$**	$9.560 \times 10^{+3}$	$1.112 \times 10^{+4}$	$2.244 \times 10^{+4}$	$3.104 \times 10^{+4}$	$9.087 \times 10^{+4}$	$1.407 \times 10^{+4}$	$1.488 \times 10^{+4}$
	Rank	11	9	12	4	5	3	6	7	10	8	2	1
F4	Best	$1.665 \times 10^{+3}$	$5.977 \times 10^{+2}$	$5.774 \times 10^{+2}$	$1.899 \times 10^{+3}$	$4.753 \times 10^{+2}$	$6.801 \times 10^{+2}$	$8.579 \times 10^{+2}$	$6.085 \times 10^{+3}$	$8.271 \times 10^{+2}$	$4.232 \times 10^{+2}$	$4.049 \times 10^{+2}$	**$4.040 \times 10^{+2}$**
	Mean	$3.402 \times 10^{+3}$	$7.134 \times 10^{+2}$	$7.826 \times 10^{+2}$	$2.759 \times 10^{+3}$	$5.869 \times 10^{+2}$	$1.341 \times 10^{+3}$	$2.196 \times 10^{+3}$	$9.487 \times 10^{+3}$	$2.521 \times 10^{+3}$	**$4.529 \times 10^{+2}$**	$4.912 \times 10^{+2}$	$4.686 \times 10^{+2}$
	Std	$1.447 \times 10^{+3}$	$5.584 \times 10^{+1}$	$1.540 \times 10^{+2}$	$8.672 \times 10^{+2}$	$1.022 \times 10^{+2}$	$5.494 \times 10^{+2}$	$1.163 \times 10^{+3}$	$2.942 \times 10^{+3}$	$9.870 \times 10^{+2}$	$3.149 \times 10^{+1}$	**$3.004 \times 10^{+1}$**	$3.682 \times 10^{+1}$
	Rank	11	5	6	10	4	7	8	12	9	1	3	2
F5	Best	$8.154 \times 10^{+2}$	$6.959 \times 10^{+2}$	$7.267 \times 10^{+2}$	$7.621 \times 10^{+2}$	$6.492 \times 10^{+2}$	$6.407 \times 10^{+2}$	$6.978 \times 10^{+2}$	$8.299 \times 10^{+2}$	$6.946 \times 10^{+2}$	**$5.304 \times 10^{+2}$**	$5.388 \times 10^{+2}$	$5.328 \times 10^{+2}$
	Mean	$8.541 \times 10^{+2}$	$7.385 \times 10^{+2}$	$8.256 \times 10^{+2}$	$8.047 \times 10^{+2}$	$7.408 \times 10^{+2}$	$7.080 \times 10^{+2}$	$7.433 \times 10^{+2}$	$8.756 \times 10^{+2}$	$7.500 \times 10^{+2}$	$5.964 \times 10^{+2}$	**$5.592 \times 10^{+2}$**	$5.606 \times 10^{+2}$
	Std	$1.894 \times 10^{+1}$	**$1.557 \times 10^{+1}$**	$5.596 \times 10^{+1}$	$2.481 \times 10^{+1}$	$5.723 \times 10^{+1}$	$4.284 \times 10^{+1}$	$3.377 \times 10^{+1}$	$2.833 \times 10^{+1}$	$3.107 \times 10^{+1}$	$4.139 \times 10^{+1}$	$1.788 \times 10^{+1}$	$1.640 \times 10^{+1}$
	Rank	11	5	10	9	7	4	6	12	8	3	1	2
F6	Best	$6.576 \times 10^{+2}$	$6.163 \times 10^{+2}$	$6.585 \times 10^{+2}$	$6.455 \times 10^{+2}$	$6.161 \times 10^{+2}$	$6.218 \times 10^{+2}$	$6.258 \times 10^{+2}$	$6.718 \times 10^{+2}$	$6.321 \times 10^{+2}$	**$6.000 \times 10^{+2}$**	$6.004 \times 10^{+2}$	$6.005 \times 10^{+2}$
	Mean	$6.719 \times 10^{+2}$	$6.207 \times 10^{+2}$	$6.764 \times 10^{+2}$	$6.602 \times 10^{+2}$	$6.474 \times 10^{+2}$	$6.374 \times 10^{+2}$	$6.454 \times 10^{+2}$	$6.838 \times 10^{+2}$	$6.433 \times 10^{+2}$	**$6.007 \times 10^{+2}$**	$6.033 \times 10^{+2}$	$6.033 \times 10^{+2}$
	Std	$7.587 \times 10^{+0}$	$2.447 \times 10^{+0}$	$1.040 \times 10^{+1}$	$6.094 \times 10^{+0}$	$1.318 \times 10^{+1}$	$8.092 \times 10^{+0}$	$7.950 \times 10^{+0}$	$7.828 \times 10^{+0}$	$6.243 \times 10^{+0}$	$3.851 \times 10^{+0}$	$2.779 \times 10^{+0}$	**$1.561 \times 10^{+0}$**
	Rank	10	4	11	9	8	5	7	12	6	1	2	**3**
F7	Best	$1.525 \times 10^{+3}$	$1.048 \times 10^{+3}$	$1.117 \times 10^{+3}$	$1.107 \times 10^{+3}$	$8.532 \times 10^{+2}$	$9.544 \times 10^{+2}$	$9.718 \times 10^{+2}$	$1.245 \times 10^{+3}$	$9.666 \times 10^{+2}$	$7.686 \times 10^{+2}$	$7.745 \times 10^{+2}$	**$7.675 \times 10^{+2}$**
	Mean	$2.053 \times 10^{+3}$	$1.099 \times 10^{+3}$	$1.256 \times 10^{+3}$	$1.203 \times 10^{+3}$	$1.024 \times 10^{+3}$	$1.060 \times 10^{+3}$	$1.112 \times 10^{+3}$	$1.389 \times 10^{+3}$	$1.046 \times 10^{+3}$	$8.670 \times 10^{+2}$	**$8.047 \times 10^{+2}$**	$8.223 \times 10^{+2}$
	Std	$1.631 \times 10^{+2}$	$2.439 \times 10^{+1}$	$9.504 \times 10^{+1}$	$5.485 \times 10^{+1}$	$8.544 \times 10^{+1}$	$6.699 \times 10^{+1}$	$7.584 \times 10^{+1}$	$7.614 \times 10^{+1}$	$3.873 \times 10^{+1}$	$5.547 \times 10^{+1}$	**$1.884 \times 10^{+1}$**	$4.772 \times 10^{+1}$
	Rank	12	7	10	9	4	6	8	11	5	3	1	2
F8	Best	$1.109 \times 10^{+3}$	$1.014 \times 10^{+3}$	$9.551 \times 10^{+2}$	$1.030 \times 10^{+3}$	$9.512 \times 10^{+2}$	$9.136 \times 10^{+2}$	$9.375 \times 10^{+2}$	$1.065 \times 10^{+3}$	$1.004 \times 10^{+3}$	$8.342 \times 10^{+2}$	**$8.249 \times 10^{+2}$**	$8.328 \times 10^{+2}$
	Mean	$1.158 \times 10^{+3}$	$1.053 \times 10^{+3}$	$1.036 \times 10^{+3}$	$1.078 \times 10^{+3}$	$1.041 \times 10^{+3}$	$9.755 \times 10^{+2}$	$9.968 \times 10^{+2}$	$1.122 \times 10^{+3}$	$1.056 \times 10^{+3}$	$9.175 \times 10^{+2}$	$8.588 \times 10^{+2}$	**$8.544 \times 10^{+2}$**
	Std	$2.094 \times 10^{+1}$	$1.548 \times 10^{+1}$	$5.331 \times 10^{+1}$	$1.979 \times 10^{+1}$	$4.262 \times 10^{+1}$	$2.954 \times 10^{+1}$	$3.110 \times 10^{+1}$	$3.690 \times 10^{+1}$	$2.740 \times 10^{+1}$	$6.557 \times 10^{+1}$	$1.648 \times 10^{+1}$	**$1.545 \times 10^{+1}$**
	Rank	12	8	6	10	7	4	5	11	9	3	2	1
F9	Best	$1.030 \times 10^{+4}$	$6.029 \times 10^{+3}$	$5.494 \times 10^{+3}$	$5.460 \times 10^{+3}$	$3.566 \times 10^{+3}$	$3.037 \times 10^{+3}$	$2.564 \times 10^{+3}$	$6.888 \times 10^{+3}$	$8.139 \times 10^{+3}$	**$9.000 \times 10^{+2}$**	$9.176 \times 10^{+2}$	$9.085 \times 10^{+2}$
	Mean	$1.420 \times 10^{+4}$	$9.487 \times 10^{+3}$	$1.059 \times 10^{+4}$	$7.301 \times 10^{+3}$	$6.383 \times 10^{+3}$	$4.967 \times 10^{+3}$	$4.702 \times 10^{+3}$	$1.115 \times 10^{+4}$	$1.128 \times 10^{+4}$	**$9.608 \times 10^{+2}$**	$1.113 \times 10^{+3}$	$1.037 \times 10^{+3}$
	Std	$2.277 \times 10^{+3}$	$1.256 \times 10^{+3}$	$3.414 \times 10^{+3}$	$1.090 \times 10^{+3}$	$1.499 \times 10^{+3}$	$1.771 \times 10^{+3}$	$1.408 \times 10^{+3}$	$1.879 \times 10^{+3}$	$1.502 \times 10^{+3}$	$1.595 \times 10^{+2}$	$1.735 \times 10^{+2}$	**$1.218 \times 10^{+2}$**
	Rank	12	8	9	7	6	5	4	10	11	1	3	**2**
F10	Best	$7.707 \times 10^{+3}$	$5.956 \times 10^{+3}$	$5.963 \times 10^{+3}$	$7.176 \times 10^{+3}$	$4.631 \times 10^{+3}$	$4.181 \times 10^{+3}$	$6.593 \times 10^{+3}$	$7.462 \times 10^{+3}$	$4.929 \times 10^{+3}$	$4.350 \times 10^{+3}$	$3.397 \times 10^{+3}$	**$3.369 \times 10^{+3}$**
	Mean	$8.442 \times 10^{+3}$	$6.433 \times 10^{+3}$	$7.349 \times 10^{+3}$	$8.738 \times 10^{+3}$	$6.363 \times 10^{+3}$	$6.092 \times 10^{+3}$	$7.681 \times 10^{+3}$	$8.620 \times 10^{+3}$	**$5.864 \times 10^{+3}$**	$7.229 \times 10^{+3}$	$6.304 \times 10^{+3}$	$6.372 \times 10^{+3}$
	Std	$3.697 \times 10^{+2}$	**$2.319 \times 10^{+2}$**	$7.575 \times 10^{+2}$	$4.460 \times 10^{+2}$	$8.563 \times 10^{+2}$	$1.104 \times 10^{+3}$	$6.373 \times 10^{+2}$	$5.031 \times 10^{+2}$	$3.660 \times 10^{+2}$	$1.386 \times 10^{+3}$	$1.551 \times 10^{+3}$	$1.946 \times 10^{+3}$
	Rank	10	4	8	12	3	2	9	11	1	7	5	6
F11	Best	$3.138 \times 10^{+3}$	$1.855 \times 10^{+3}$	$2.061 \times 10^{+3}$	$2.281 \times 10^{+3}$	$1.239 \times 10^{+3}$	$1.562 \times 10^{+3}$	$2.617 \times 10^{+3}$	$2.883 \times 10^{+3}$	$2.525 \times 10^{+3}$	$1.132 \times 10^{+3}$	$1.126 \times 10^{+3}$	**$1.121 \times 10^{+3}$**
	Mean	$7.075 \times 10^{+3}$	$4.604 \times 10^{+3}$	$6.725 \times 10^{+3}$	$3.232 \times 10^{+3}$	$1.505 \times 10^{+3}$	$3.181 \times 10^{+3}$	$5.929 \times 10^{+3}$	$6.020 \times 10^{+3}$	$6.740 \times 10^{+3}$	$1.268 \times 10^{+3}$	**$1.197 \times 10^{+3}$**	$1.215 \times 10^{+3}$
	Std	$4.719 \times 10^{+3}$	$1.615 \times 10^{+3}$	$2.954 \times 10^{+3}$	$7.086 \times 10^{+2}$	$1.011 \times 10^{+2}$	$1.336 \times 10^{+3}$	$1.995 \times 10^{+3}$	$1.735 \times 10^{+3}$	$2.310 \times 10^{+3}$	$3.496 \times 10^{+2}$	$4.439 \times 10^{+1}$	**$4.436 \times 10^{+1}$**
	Rank	10	7	11	6	4	5	8	9	12	1	2	3
F12	Best	$1.202 \times 10^{+9}$	$5.314 \times 10^{+7}$	$4.704 \times 10^{+7}$	$1.051 \times 10^{+9}$	$1.243 \times 10^{+5}$	$7.321 \times 10^{+7}$	$1.043 \times 10^{+8}$	$1.972 \times 10^{+9}$	$1.444 \times 10^{+8}$	$2.022 \times 10^{+5}$	**$1.082 \times 10^{+4}$**	$2.965 \times 10^{+4}$
	Mean	$2.682 \times 10^{+9}$	$1.160 \times 10^{+8}$	$2.804 \times 10^{+8}$	$2.081 \times 10^{+9}$	$2.722 \times 10^{+7}$	$8.616 \times 10^{+8}$	$8.730 \times 10^{+8}$	$6.279 \times 10^{+9}$	$6.466 \times 10^{+8}$	$1.592 \times 10^{+7}$	$2.240 \times 10^{+5}$	**$1.691 \times 10^{+5}$**
	Std	$8.678 \times 10^{+8}$	$3.530 \times 10^{+7}$	$1.998 \times 10^{+8}$	$5.146 \times 10^{+8}$	$3.241 \times 10^{+7}$	$7.940 \times 10^{+8}$	$6.722 \times 10^{+8}$	$3.049 \times 10^{+9}$	$3.741 \times 10^{+8}$	$4.821 \times 10^{+7}$	$1.744 \times 10^{+5}$	**$1.515 \times 10^{+5}$**
	Rank	11	5	6	10	4	9	8	12	7	3	2	1
F13	Best	$2.957 \times 10^{+8}$	$4.578 \times 10^{+6}$	$3.029 \times 10^{+5}$	$2.853 \times 10^{+8}$	$1.260 \times 10^{+4}$	$5.905 \times 10^{+4}$	$1.074 \times 10^{+6}$	$5.364 \times 10^{+8}$	$2.904 \times 10^{+6}$	$2.532 \times 10^{+3}$	$1.771 \times 10^{+3}$	**$1.372 \times 10^{+3}$**
	Mean	$1.238 \times 10^{+9}$	$2.354 \times 10^{+7}$	$2.229 \times 10^{+6}$	$9.661 \times 10^{+8}$	$1.989 \times 10^{+6}$	$3.131 \times 10^{+8}$	$3.187 \times 10^{+7}$	$2.902 \times 10^{+9}$	$1.422 \times 10^{+8}$	$2.140 \times 10^{+5}$	$1.932 \times 10^{+4}$	**$1.693 \times 10^{+4}$**
	Std	$9.715 \times 10^{+8}$	$1.458 \times 10^{+7}$	$3.354 \times 10^{+6}$	$6.875 \times 10^{+8}$	$4.853 \times 10^{+6}$	$5.452 \times 10^{+8}$	$8.136 \times 10^{+7}$	$1.782 \times 10^{+9}$	$1.757 \times 10^{+8}$	$4.575 \times 10^{+5}$	$1.718 \times 10^{+4}$	**$1.555 \times 10^{+4}$**
	Rank	11	7	5	10	4	8	6	12	9	3	2	1

Table A5. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F14	Best	$1.440 \times 10^{+5}$	$1.113 \times 10^{+5}$	$4.305 \times 10^{+5}$	$7.574 \times 10^{+4}$	$2.281 \times 10^{+3}$	$4.195 \times 10^{+4}$	$1.601 \times 10^{+4}$	$3.301 \times 10^{+5}$	$1.330 \times 10^{+4}$	**$1.480 \times 10^{+3}$**	$1.995 \times 10^{+3}$	$1.929 \times 10^{+3}$
	Mean	$1.179 \times 10^{+6}$	$1.049 \times 10^{+6}$	$2.964 \times 10^{+6}$	$5.054 \times 10^{+5}$	$2.462 \times 10^{+5}$	$7.520 \times 10^{+5}$	$9.545 \times 10^{+5}$	$1.682 \times 10^{+6}$	$4.411 \times 10^{+5}$	$1.581 \times 10^{+4}$	$2.065 \times 10^{+4}$	**$1.039 \times 10^{+4}$**
	Std	$1.949 \times 10^{+6}$	$7.449 \times 10^{+5}$	$2.545 \times 10^{+6}$	$5.063 \times 10^{+5}$	$4.864 \times 10^{+5}$	$9.158 \times 10^{+5}$	$9.599 \times 10^{+5}$	$9.393 \times 10^{+5}$	$3.267 \times 10^{+5}$	$2.259 \times 10^{+4}$	$2.120 \times 10^{+4}$	**$8.624 \times 10^{+3}$**
	Rank	9	10	12	6	4	7	8	11	5	2	3	1
F15	Best	$4.812 \times 10^{+7}$	$8.747 \times 10^{+5}$	$6.816 \times 10^{+4}$	$5.987 \times 10^{+5}$	$2.445 \times 10^{+3}$	$5.162 \times 10^{+4}$	$2.765 \times 10^{+5}$	$2.655 \times 10^{+6}$	$1.406 \times 10^{+5}$	$1.903 \times 10^{+3}$	$1.772 \times 10^{+3}$	**$1.637 \times 10^{+3}$**
	Mean	$1.955 \times 10^{+8}$	$5.788 \times 10^{+6}$	$9.287 \times 10^{+5}$	$5.084 \times 10^{+7}$	$6.393 \times 10^{+4}$	$1.474 \times 10^{+7}$	$1.523 \times 10^{+7}$	$6.947 \times 10^{+7}$	$1.273 \times 10^{+6}$	$5.857 \times 10^{+4}$	$9.001 \times 10^{+3}$	**$6.113 \times 10^{+3}$**
	Std	$1.089 \times 10^{+8}$	$3.352 \times 10^{+6}$	$1.186 \times 10^{+6}$	$4.197 \times 10^{+7}$	$5.944 \times 10^{+4}$	$2.567 \times 10^{+7}$	$3.399 \times 10^{+7}$	$6.687 \times 10^{+7}$	$1.861 \times 10^{+6}$	$1.326 \times 10^{+5}$	$6.814 \times 10^{+3}$	**$5.292 \times 10^{+3}$**
	Rank	12	9	5	10	4	7	8	11	6	3	2	1
F16	Best	$3.538 \times 10^{+3}$	$2.685 \times 10^{+3}$	$2.935 \times 10^{+3}$	$3.473 \times 10^{+3}$	$2.361 \times 10^{+3}$	$2.261 \times 10^{+3}$	$2.775 \times 10^{+3}$	$3.625 \times 10^{+3}$	$2.974 \times 10^{+3}$	$2.244 \times 10^{+3}$	$1.992 \times 10^{+3}$	**$1.858 \times 10^{+3}$**
	Mean	$4.043 \times 10^{+3}$	$3.153 \times 10^{+3}$	$3.963 \times 10^{+3}$	$4.088 \times 10^{+3}$	$3.397 \times 10^{+3}$	$3.012 \times 10^{+3}$	$3.440 \times 10^{+3}$	$4.664 \times 10^{+3}$	$3.341 \times 10^{+3}$	$2.903 \times 10^{+3}$	$2.512 \times 10^{+3}$	**$2.381 \times 10^{+3}$**
	Std	$3.311 \times 10^{+2}$	**$1.718 \times 10^{+2}$**	$5.957 \times 10^{+2}$	$2.812 \times 10^{+2}$	$5.447 \times 10^{+2}$	$4.155 \times 10^{+2}$	$3.901 \times 10^{+2}$	$5.649 \times 10^{+2}$	$2.086 \times 10^{+2}$	$3.950 \times 10^{+2}$	$2.997 \times 10^{+2}$	$2.440 \times 10^{+2}$
	Rank	10	5	9	11	7	4	8	12	6	3	2	1
F17	Best	$2.524 \times 10^{+3}$	$2.164 \times 10^{+3}$	$2.021 \times 10^{+3}$	$2.369 \times 10^{+3}$	$2.100 \times 10^{+3}$	$1.868 \times 10^{+3}$	$2.027 \times 10^{+3}$	$2.279 \times 10^{+3}$	$2.105 \times 10^{+3}$	$1.830 \times 10^{+3}$	$1.759 \times 10^{+3}$	**$1.752 \times 10^{+3}$**
	Mean	$2.975 \times 10^{+3}$	$2.410 \times 10^{+3}$	$2.752 \times 10^{+3}$	$2.729 \times 10^{+3}$	$2.594 \times 10^{+3}$	$2.235 \times 10^{+3}$	$2.386 \times 10^{+3}$	$3.130 \times 10^{+3}$	$2.420 \times 10^{+3}$	$2.120 \times 10^{+3}$	$1.983 \times 10^{+3}$	**$1.930 \times 10^{+3}$**
	Std	$1.905 \times 10^{+2}$	$1.332 \times 10^{+2}$	$3.471 \times 10^{+2}$	$1.890 \times 10^{+2}$	$2.732 \times 10^{+2}$	$2.722 \times 10^{+2}$	$2.094 \times 10^{+2}$	$4.231 \times 10^{+2}$	$1.668 \times 10^{+2}$	$1.904 \times 10^{+2}$	$1.472 \times 10^{+2}$	**$1.210 \times 10^{+2}$**
	Rank	11	5	9	10	8	4	7	12	6	3	2	**1**
F18	Best	$3.241 \times 10^{+6}$	$5.220 \times 10^{+5}$	$2.221 \times 10^{+5}$	$2.338 \times 10^{+6}$	$1.405 \times 10^{+5}$	$8.703 \times 10^{+4}$	$1.509 \times 10^{+5}$	$1.085 \times 10^{+6}$	$4.821 \times 10^{+5}$	$4.148 \times 10^{+4}$	$6.471 \times 10^{+4}$	**$2.780 \times 10^{+4}$**
	Mean	$1.578 \times 10^{+7}$	$4.714 \times 10^{+6}$	$8.374 \times 10^{+6}$	$1.148 \times 10^{+7}$	$3.382 \times 10^{+6}$	$3.044 \times 10^{+6}$	$4.854 \times 10^{+6}$	$1.417 \times 10^{+7}$	$2.193 \times 10^{+6}$	$1.021 \times 10^{+6}$	$3.798 \times 10^{+5}$	**$2.484 \times 10^{+5}$**
	Std	$1.642 \times 10^{+7}$	$2.623 \times 10^{+6}$	$1.041 \times 10^{+7}$	$6.448 \times 10^{+6}$	$5.192 \times 10^{+6}$	$5.522 \times 10^{+6}$	$5.806 \times 10^{+6}$	$1.256 \times 10^{+7}$	$1.848 \times 10^{+6}$	$1.301 \times 10^{+6}$	$5.507 \times 10^{+5}$	**$2.370 \times 10^{+5}$**
	Rank	12	8	9	10	5	4	7	11	6	3	2	1
F19	Best	$2.833 \times 10^{+7}$	$4.360 \times 10^{+6}$	$4.711 \times 10^{+4}$	$1.178 \times 10^{+7}$	$2.716 \times 10^{+3}$	$8.980 \times 10^{+4}$	$6.471 \times 10^{+5}$	$4.525 \times 10^{+6}$	$6.501 \times 10^{+5}$	$1.948 \times 10^{+3}$	$2.300 \times 10^{+3}$	**$1.932 \times 10^{+3}$**
	Mean	$2.580 \times 10^{+8}$	$1.094 \times 10^{+7}$	$7.379 \times 10^{+6}$	$7.269 \times 10^{+7}$	$1.056 \times 10^{+6}$	$1.799 \times 10^{+7}$	$1.168 \times 10^{+7}$	$5.670 \times 10^{+7}$	$3.294 \times 10^{+7}$	$1.901 \times 10^{+4}$	$1.209 \times 10^{+4}$	**$7.718 \times 10^{+3}$**
	Std	$1.709 \times 10^{+8}$	$5.473 \times 10^{+6}$	$9.082 \times 10^{+6}$	$3.926 \times 10^{+7}$	$3.290 \times 10^{+6}$	$3.715 \times 10^{+7}$	$1.265 \times 10^{+7}$	$7.313 \times 10^{+7}$	$2.097 \times 10^{+7}$	$4.098 \times 10^{+4}$	$1.030 \times 10^{+4}$	**$5.880 \times 10^{+3}$**
	Rank	12	8	5	11	4	6	7	10	9	3	2	1
F20	Best	$2.482 \times 10^{+3}$	$2.288 \times 10^{+3}$	$2.394 \times 10^{+3}$	$2.535 \times 10^{+3}$	$2.375 \times 10^{+3}$	$2.364 \times 10^{+3}$	$2.352 \times 10^{+3}$	$2.527 \times 10^{+3}$	$2.327 \times 10^{+3}$	$2.082 \times 10^{+3}$	$2.081 \times 10^{+3}$	**$2.060 \times 10^{+3}$**
	Mean	$2.897 \times 10^{+3}$	$2.590 \times 10^{+3}$	$2.927 \times 10^{+3}$	$2.819 \times 10^{+3}$	$2.755 \times 10^{+3}$	$2.609 \times 10^{+3}$	$2.599 \times 10^{+3}$	$2.910 \times 10^{+3}$	$2.581 \times 10^{+3}$	$2.323 \times 10^{+3}$	$2.304 \times 10^{+3}$	**$2.297 \times 10^{+3}$**
	Std	$1.587 \times 10^{+2}$	$1.461 \times 10^{+2}$	$2.058 \times 10^{+2}$	$1.332 \times 10^{+2}$	$2.176 \times 10^{+2}$	**$1.326 \times 10^{+2}$**	$1.606 \times 10^{+2}$	$1.575 \times 10^{+2}$	$1.344 \times 10^{+2}$	$1.836 \times 10^{+2}$	$1.354 \times 10^{+2}$	$1.462 \times 10^{+2}$
	Rank	10	7	12	9	8	6	5	11	4	3	2	1
F21	Best	$2.587 \times 10^{+3}$	$2.495 \times 10^{+3}$	$2.476 \times 10^{+3}$	$2.527 \times 10^{+3}$	$2.364 \times 10^{+3}$	$2.422 \times 10^{+3}$	$2.460 \times 10^{+3}$	$2.569 \times 10^{+3}$	$2.478 \times 10^{+3}$	$2.332 \times 10^{+3}$	$2.327 \times 10^{+3}$	**$2.324 \times 10^{+3}$**
	Mean	$2.644 \times 10^{+3}$	$2.537 \times 10^{+3}$	$2.607 \times 10^{+3}$	$2.592 \times 10^{+3}$	$2.536 \times 10^{+3}$	$2.479 \times 10^{+3}$	$2.504 \times 10^{+3}$	$2.675 \times 10^{+3}$	$2.555 \times 10^{+3}$	$2.429 \times 10^{+3}$	**$2.351 \times 10^{+3}$**	$2.355 \times 10^{+3}$
	Std	$2.505 \times 10^{+1}$	**$1.502 \times 10^{+1}$**	$7.099 \times 10^{+1}$	$2.820 \times 10^{+1}$	$5.562 \times 10^{+1}$	$4.162 \times 10^{+1}$	$2.674 \times 10^{+1}$	$5.211 \times 10^{+1}$	$2.926 \times 10^{+1}$	$6.707 \times 10^{+1}$	$1.625 \times 10^{+1}$	$1.988 \times 10^{+1}$
	Rank	11	6	10	9	7	4	5	12	8	3	1	2
F22	Best	$4.442 \times 10^{+3}$	$5.145 \times 10^{+3}$	$2.687 \times 10^{+3}$	$3.849 \times 10^{+3}$	$2.315 \times 10^{+3}$	$2.996 \times 10^{+3}$	$2.872 \times 10^{+3}$	$5.433 \times 10^{+3}$	$3.692 \times 10^{+3}$	$2.300 \times 10^{+3}$	$2.300 \times 10^{+3}$	**$2.300 \times 10^{+3}$**
	Mean	$8.183 \times 10^{+3}$	$7.026 \times 10^{+3}$	$7.634 \times 10^{+3}$	$8.851 \times 10^{+3}$	$4.270 \times 10^{+3}$	$5.468 \times 10^{+3}$	$4.504 \times 10^{+3}$	$8.229 \times 10^{+3}$	$6.776 \times 10^{+3}$	$7.624 \times 10^{+3}$	$2.651 \times 10^{+3}$	**$2.301 \times 10^{+3}$**
	Std	$1.938 \times 10^{+3}$	$1.162 \times 10^{+3}$	$1.639 \times 10^{+3}$	$2.437 \times 10^{+3}$	$2.409 \times 10^{+3}$	$2.423 \times 10^{+3}$	$8.280 \times 10^{+2}$	$1.594 \times 10^{+3}$	$1.269 \times 10^{+3}$	$2.139 \times 10^{+3}$	$1.377 \times 10^{+3}$	**$1.360 \times 10^{+0}$**
	Rank	11	7	8	12	4	5	3	10	6	9	2	1
F23	Best	$2.959 \times 10^{+3}$	$2.832 \times 10^{+3}$	$2.983 \times 10^{+3}$	$2.948 \times 10^{+3}$	$2.870 \times 10^{+3}$	$2.816 \times 10^{+3}$	$2.843 \times 10^{+3}$	$3.041 \times 10^{+3}$	$2.854 \times 10^{+3}$	$2.689 \times 10^{+3}$	$2.690 \times 10^{+3}$	**$2.673 \times 10^{+3}$**
	Mean	$3.119 \times 10^{+3}$	$2.859 \times 10^{+3}$	$3.149 \times 10^{+3}$	$3.071 \times 10^{+3}$	$3.018 \times 10^{+3}$	$2.897 \times 10^{+3}$	$2.929 \times 10^{+3}$	$3.318 \times 10^{+3}$	$2.939 \times 10^{+3}$	$2.767 \times 10^{+3}$	**$2.723 \times 10^{+3}$**	$2.728 \times 10^{+3}$
	Std	$9.749 \times 10^{+1}$	**$1.268 \times 10^{+1}$**	$1.028 \times 10^{+2}$	$4.290 \times 10^{+1}$	$9.143 \times 10^{+1}$	$5.650 \times 10^{+1}$	$6.295 \times 10^{+1}$	$1.123 \times 10^{+2}$	$3.642 \times 10^{+1}$	$5.694 \times 10^{+1}$	$2.455 \times 10^{+1}$	$1.782 \times 10^{+1}$
	Rank	10	4	11	9	8	5	6	12	7	3	1	2
F24	Best	$3.077 \times 10^{+3}$	$3.042 \times 10^{+3}$	$3.114 \times 10^{+3}$	$3.171 \times 10^{+3}$	$2.917 \times 10^{+3}$	$2.944 \times 10^{+3}$	$3.010 \times 10^{+3}$	$3.284 \times 10^{+3}$	$3.136 \times 10^{+3}$	$2.911 \times 10^{+3}$	**$2.840 \times 10^{+3}$**	$2.848 \times 10^{+3}$
	Mean	$3.164 \times 10^{+3}$	$3.079 \times 10^{+3}$	$3.256 \times 10^{+3}$	$3.234 \times 10^{+3}$	$3.056 \times 10^{+3}$	$3.070 \times 10^{+3}$	$3.086 \times 10^{+3}$	$3.451 \times 10^{+3}$	$3.200 \times 10^{+3}$	$3.001 \times 10^{+3}$	$2.876 \times 10^{+3}$	**$2.871 \times 10^{+3}$**
	Std	$6.525 \times 10^{+1}$	**$1.594 \times 10^{+1}$**	$8.166 \times 10^{+1}$	$3.571 \times 10^{+1}$	$6.767 \times 10^{+1}$	$6.260 \times 10^{+1}$	$5.311 \times 10^{+1}$	$1.058 \times 10^{+2}$	$4.621 \times 10^{+1}$	$6.049 \times 10^{+1}$	$2.304 \times 10^{+1}$	$2.080 \times 10^{+1}$
	Rank	8	6	11	10	4	5	7	12	9	3	2	1
F25	Best	$4.350 \times 10^{+3}$	$3.031 \times 10^{+3}$	$2.976 \times 10^{+3}$	$3.177 \times 10^{+3}$	$2.884 \times 10^{+3}$	$3.053 \times 10^{+3}$	$3.096 \times 10^{+3}$	$3.519 \times 10^{+3}$	$3.110 \times 10^{+3}$	**$2.878 \times 10^{+3}$**	$2.884 \times 10^{+3}$	$2.884 \times 10^{+3}$
	Mean	$5.535 \times 10^{+3}$	$3.256 \times 10^{+3}$	$3.088 \times 10^{+3}$	$3.415 \times 10^{+3}$	$2.949 \times 10^{+3}$	$3.224 \times 10^{+3}$	$3.366 \times 10^{+3}$	$4.421 \times 10^{+3}$	$3.564 \times 10^{+3}$	$2.894 \times 10^{+3}$	$2.894 \times 10^{+3}$	**$2.890 \times 10^{+3}$**
	Std	$8.281 \times 10^{+2}$	$9.006 \times 10^{+1}$	$4.522 \times 10^{+1}$	$1.567 \times 10^{+2}$	$4.911 \times 10^{+1}$	$1.505 \times 10^{+2}$	$1.875 \times 10^{+2}$	$4.553 \times 10^{+2}$	$3.106 \times 10^{+2}$	$2.666 \times 10^{+1}$	$1.314 \times 10^{+1}$	**$8.202 \times 10^{+0}$**
	Rank	12	7	5	9	4	6	8	11	10	2	3	1

Table A5. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F26	Best	$6.931 \times 10^{+3}$	$5.572 \times 10^{+3}$	$4.163 \times 10^{+3}$	$6.606 \times 10^{+3}$	$3.703 \times 10^{+3}$	$3.888 \times 10^{+3}$	$5.100 \times 10^{+3}$	$7.262 \times 10^{+3}$	$5.994 \times 10^{+3}$	$3.821 \times 10^{+3}$	$2.800 \times 10^{+3}$	**$2.800 \times 10^{+3}$**
	Mean	$7.909 \times 10^{+3}$	$5.896 \times 10^{+3}$	$7.978 \times 10^{+3}$	$7.570 \times 10^{+3}$	$6.646 \times 10^{+3}$	$5.659 \times 10^{+3}$	$6.550 \times 10^{+3}$	$9.953 \times 10^{+3}$	$6.926 \times 10^{+3}$	$4.742 \times 10^{+3}$	$4.290 \times 10^{+3}$	**$4.265 \times 10^{+3}$**
	Std	$7.810 \times 10^{+2}$	**$1.524 \times 10^{+2}$**	$1.321 \times 10^{+3}$	$4.530 \times 10^{+2}$	$1.112 \times 10^{+3}$	$6.930 \times 10^{+2}$	$7.296 \times 10^{+2}$	$9.574 \times 10^{+2}$	$3.778 \times 10^{+2}$	$5.549 \times 10^{+2}$	$7.296 \times 10^{+2}$	$9.083 \times 10^{+2}$
	Rank	11	5	10	9	7	4	6	12	8	3	2	1
F27	Best	$3.254 \times 10^{+3}$	$3.221 \times 10^{+3}$	$3.285 \times 10^{+3}$	$3.366 \times 10^{+3}$	$3.224 \times 10^{+3}$	$3.223 \times 10^{+3}$	$3.294 \times 10^{+3}$	$3.493 \times 10^{+3}$	$3.258 \times 10^{+3}$	**$3.200 \times 10^{+3}$**	$3.212 \times 10^{+3}$	$3.230 \times 10^{+3}$
	Mean	$3.375 \times 10^{+3}$	$3.236 \times 10^{+3}$	$3.511 \times 10^{+3}$	$3.518 \times 10^{+3}$	$3.321 \times 10^{+3}$	$3.340 \times 10^{+3}$	$3.484 \times 10^{+3}$	$3.846 \times 10^{+3}$	$3.340 \times 10^{+3}$	**$3.200 \times 10^{+3}$**	$3.254 \times 10^{+3}$	$3.262 \times 10^{+3}$
	Std	$9.380 \times 10^{+1}$	$6.769 \times 10^{+0}$	$1.355 \times 10^{+2}$	$6.654 \times 10^{+1}$	$5.693 \times 10^{+1}$	$7.006 \times 10^{+1}$	$1.324 \times 10^{+2}$	$2.024 \times 10^{+2}$	$4.166 \times 10^{+1}$	**2.139×10^{-04}**	$2.345 \times 10^{+1}$	$1.979 \times 10^{+1}$
	Rank	8	2	10	11	5	6	9	12	7	1	3	4
F28	Best	$4.046 \times 10^{+3}$	$3.539 \times 10^{+3}$	$3.364 \times 10^{+3}$	$3.757 \times 10^{+3}$	$3.200 \times 10^{+3}$	$3.450 \times 10^{+3}$	$3.635 \times 10^{+3}$	$4.502 \times 10^{+3}$	$3.654 \times 10^{+3}$	$3.289 \times 10^{+3}$	$3.196 \times 10^{+3}$	**$3.101 \times 10^{+3}$**
	Mean	$5.201 \times 10^{+3}$	$3.972 \times 10^{+3}$	$3.514 \times 10^{+3}$	$4.194 \times 10^{+3}$	$3.505 \times 10^{+3}$	$3.887 \times 10^{+3}$	$4.146 \times 10^{+3}$	$5.989 \times 10^{+3}$	$5.054 \times 10^{+3}$	$3.299 \times 10^{+3}$	$3.213 \times 10^{+3}$	**$3.182 \times 10^{+3}$**
	Std	$1.090 \times 10^{+3}$	$2.430 \times 10^{+2}$	$8.108 \times 10^{+1}$	$2.514 \times 10^{+2}$	$6.057 \times 10^{+2}$	$3.327 \times 10^{+2}$	$3.654 \times 10^{+2}$	$6.826 \times 10^{+2}$	$6.212 \times 10^{+2}$	**$2.697 \times 10^{+0}$**	$1.619 \times 10^{+1}$	$3.808 \times 10^{+1}$
	Rank	11	7	5	9	4	6	8	12	10	**3**	2	1
F29	Best	$4.148 \times 10^{+3}$	$3.873 \times 10^{+3}$	$4.079 \times 10^{+3}$	$4.681 \times 10^{+3}$	$3.636 \times 10^{+3}$	$3.857 \times 10^{+3}$	$3.732 \times 10^{+3}$	$4.350 \times 10^{+3}$	$3.921 \times 10^{+3}$	**$3.288 \times 10^{+3}$**	$3.419 \times 10^{+3}$	$3.439 \times 10^{+3}$
	Mean	$4.841 \times 10^{+3}$	$4.186 \times 10^{+3}$	$5.193 \times 10^{+3}$	$5.058 \times 10^{+3}$	$4.470 \times 10^{+3}$	$4.299 \times 10^{+3}$	$4.561 \times 10^{+3}$	$5.823 \times 10^{+3}$	$4.388 \times 10^{+3}$	$3.857 \times 10^{+3}$	**$3.678 \times 10^{+3}$**	$3.684 \times 10^{+3}$
	Std	$2.952 \times 10^{+2}$	**$1.380 \times 10^{+2}$**	$5.126 \times 10^{+2}$	$2.636 \times 10^{+2}$	$3.635 \times 10^{+2}$	$2.691 \times 10^{+2}$	$4.688 \times 10^{+2}$	$5.697 \times 10^{+2}$	$2.193 \times 10^{+2}$	$4.005 \times 10^{+2}$	$1.836 \times 10^{+2}$	$1.707 \times 10^{+2}$
	Rank	9	4	11	10	7	5	8	12	6	3	2	1
F30	Best	$3.263 \times 10^{+7}$	$3.519 \times 10^{+5}$	$9.525 \times 10^{+5}$	$5.748 \times 10^{+7}$	$6.151 \times 10^{+3}$	$1.925 \times 10^{+6}$	$4.004 \times 10^{+6}$	$3.782 \times 10^{+7}$	$2.688 \times 10^{+6}$	**$3.405 \times 10^{+3}$**	$6.034 \times 10^{+3}$	$6.476 \times 10^{+3}$
	Mean	$1.757 \times 10^{+8}$	$1.420 \times 10^{+6}$	$2.470 \times 10^{+7}$	$1.671 \times 10^{+8}$	$2.921 \times 10^{+6}$	$6.841 \times 10^{+7}$	$4.850 \times 10^{+7}$	$2.747 \times 10^{+8}$	$1.407 \times 10^{+7}$	$1.564 \times 10^{+4}$	$9.809 \times 10^{+3}$	**$9.555 \times 10^{+3}$**
	Std	$2.480 \times 10^{+8}$	$7.713 \times 10^{+5}$	$1.903 \times 10^{+7}$	$7.153 \times 10^{+7}$	$3.736 \times 10^{+6}$	$1.609 \times 10^{+8}$	$4.363 \times 10^{+7}$	$1.873 \times 10^{+8}$	$1.786 \times 10^{+7}$	$2.535 \times 10^{+4}$	$3.396 \times 10^{+3}$	**$3.066 \times 10^{+3}$**
	Rank	10	5	7	11	4	9	8	12	6	2	3	1
Mean Rank		10.69	6.03	8.66	9.34	5.48	5.55	6.93	11.31	7.24	3.07	2.17	**1.62**
+/=/−		29/0/0	27/1/1	27/2/0	29/0/0	27/1/1	27/2/0	28/1/0	29/0/0	27/1/1	25/0/4	20/8/1	~

Table A6. Comparison results of EFNO with other algorithms on CEC2017 (Dim = 50).

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F1	Best	$6.960 \times 10^{+10}$	$3.096 \times 10^{+10}$	$6.227 \times 10^{+8}$	$4.208 \times 10^{+10}$	$9.327 \times 10^{+6}$	$1.960 \times 10^{+10}$	$2.481 \times 10^{+10}$	$5.565 \times 10^{+10}$	$1.988 \times 10^{+10}$	$9.957 \times 10^{+2}$	$1.380 \times 10^{+2}$	**$1.066 \times 10^{+2}$**
	Mean	$8.774 \times 10^{+10}$	$3.575 \times 10^{+10}$	$1.430 \times 10^{+9}$	$5.542 \times 10^{+10}$	$1.606 \times 10^{+8}$	$3.450 \times 10^{+10}$	$3.769 \times 10^{+10}$	$8.019 \times 10^{+10}$	$3.837 \times 10^{+10}$	$2.092 \times 10^{+7}$	$4.061 \times 10^{+3}$	**$3.270 \times 10^{+3}$**
	Std	$1.138 \times 10^{+10}$	$3.188 \times 10^{+9}$	$7.347 \times 10^{+8}$	$6.882 \times 10^{+9}$	$1.342 \times 10^{+8}$	$6.648 \times 10^{+9}$	$7.791 \times 10^{+9}$	$1.176 \times 10^{+10}$	$8.874 \times 10^{+9}$	$7.437 \times 10^{+7}$	**$3.847 \times 10^{+3}$**	$4.052 \times 10^{+3}$
	Rank	12	7	5	10	4	6	8	11	9	3	2	1
F3	Best	$3.202 \times 10^{+5}$	$3.030 \times 10^{+5}$	$1.265 \times 10^{+5}$	$1.112 \times 10^{+5}$	$1.195 \times 10^{+5}$	$9.755 \times 10^{+4}$	$1.127 \times 10^{+5}$	$1.334 \times 10^{+5}$	$2.471 \times 10^{+5}$	$9.571 \times 10^{+4}$	$8.832 \times 10^{+4}$	**$6.513 \times 10^{+4}$**
	Mean	$4.187 \times 10^{+5}$	$4.007 \times 10^{+5}$	$2.239 \times 10^{+5}$	$1.597 \times 10^{+5}$	$1.836 \times 10^{+5}$	**$1.199 \times 10^{+5}$**	$1.478 \times 10^{+5}$	$1.897 \times 10^{+5}$	$3.756 \times 10^{+5}$	$3.492 \times 10^{+5}$	$1.266 \times 10^{+5}$	$1.252 \times 10^{+5}$
	Std	$7.138 \times 10^{+4}$	$3.976 \times 10^{+4}$	$8.014 \times 10^{+4}$	$2.388 \times 10^{+4}$	$3.360 \times 10^{+4}$	**$1.169 \times 10^{+4}$**	$1.890 \times 10^{+4}$	$2.741 \times 10^{+4}$	$5.648 \times 10^{+4}$	$1.587 \times 10^{+5}$	$2.510 \times 10^{+4}$	$2.819 \times 10^{+4}$
	Rank	12	11	8	5	6	1	4	7	10	9	2	3
F4	Best	$6.202 \times 10^{+3}$	$1.618 \times 10^{+3}$	$9.452 \times 10^{+2}$	$6.687 \times 10^{+3}$	$5.970 \times 10^{+2}$	$3.168 \times 10^{+3}$	$3.550 \times 10^{+3}$	$1.375 \times 10^{+4}$	$1.670 \times 10^{+3}$	$4.454 \times 10^{+2}$	**$4.330 \times 10^{+2}$**	$4.710 \times 10^{+2}$
	Mean	$1.412 \times 10^{+4}$	$2.095 \times 10^{+3}$	$1.368 \times 10^{+3}$	$1.088 \times 10^{+4}$	$1.060 \times 10^{+3}$	$6.159 \times 10^{+3}$	$7.904 \times 10^{+3}$	$2.377 \times 10^{+4}$	$3.197 \times 10^{+3}$	**$5.371 \times 10^{+2}$**	$5.503 \times 10^{+2}$	$5.479 \times 10^{+2}$
	Std	$5.255 \times 10^{+3}$	$3.321 \times 10^{+2}$	$2.827 \times 10^{+2}$	$2.001 \times 10^{+3}$	$1.664 \times 10^{+3}$	$1.777 \times 10^{+3}$	$2.936 \times 10^{+3}$	$5.811 \times 10^{+3}$	$9.675 \times 10^{+2}$	$7.583 \times 10^{+1}$	$5.650 \times 10^{+1}$	**$4.491 \times 10^{+1}$**
	Rank	11	6	5	10	4	8	9	12	7	1	3	2
F5	Best	$1.109 \times 10^{+3}$	$9.469 \times 10^{+2}$	$9.023 \times 10^{+2}$	$1.035 \times 10^{+3}$	$7.507 \times 10^{+2}$	$7.962 \times 10^{+2}$	$8.939 \times 10^{+2}$	$1.051 \times 10^{+3}$	$9.398 \times 10^{+2}$	**$5.532 \times 10^{+2}$**	$5.916 \times 10^{+2}$	$5.667 \times 10^{+2}$
	Mean	$1.242 \times 10^{+3}$	$1.026 \times 10^{+3}$	$1.009 \times 10^{+3}$	$1.105 \times 10^{+3}$	$9.673 \times 10^{+2}$	$9.057 \times 10^{+2}$	$9.672 \times 10^{+2}$	$1.169 \times 10^{+3}$	$1.001 \times 10^{+3}$	$7.196 \times 10^{+2}$	$6.327 \times 10^{+2}$	**$6.218 \times 10^{+2}$**
	Std	$5.985 \times 10^{+1}$	$3.165 \times 10^{+1}$	$7.584 \times 10^{+1}$	$3.960 \times 10^{+1}$	$1.005 \times 10^{+2}$	$6.997 \times 10^{+1}$	$4.164 \times 10^{+1}$	$4.671 \times 10^{+1}$	$2.962 \times 10^{+1}$	$1.269 \times 10^{+2}$	$3.040 \times 10^{+1}$	**$2.240 \times 10^{+1}$**
	Rank	12	9	7	10	6	4	5	11	8	3	2	1

Table A6. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F6	Best	$6.807 \times 10^{+2}$	$6.298 \times 10^{+2}$	$6.724 \times 10^{+2}$	$6.619 \times 10^{+2}$	$6.493 \times 10^{+2}$	$6.453 \times 10^{+2}$	$6.434 \times 10^{+2}$	$6.714 \times 10^{+2}$	$6.360 \times 10^{+2}$	**$6.000 \times 10^{+2}$**	$6.036 \times 10^{+2}$	$6.014 \times 10^{+2}$
	Mean	$6.941 \times 10^{+2}$	$6.345 \times 10^{+2}$	$6.907 \times 10^{+2}$	$6.804 \times 10^{+2}$	$6.650 \times 10^{+2}$	$6.553 \times 10^{+2}$	$6.573 \times 10^{+2}$	$6.954 \times 10^{+2}$	$6.519 \times 10^{+2}$	**$6.028 \times 10^{+2}$**	$6.098 \times 10^{+2}$	$6.083 \times 10^{+2}$
	Std	$7.682 \times 10^{+0}$	$2.852 \times 10^{+0}$	$1.142 \times 10^{+1}$	$7.233 \times 10^{+0}$	$1.102 \times 10^{+1}$	$7.371 \times 10^{+0}$	$7.890 \times 10^{+0}$	$8.913 \times 10^{+0}$	$6.082 \times 10^{+0}$	$9.958 \times 10^{+0}$	$4.348 \times 10^{+0}$	$3.345 \times 10^{+0}$
	Rank	11	4	10	9	8	6	7	12	5	1	3	2
F7	Best	$3.167 \times 10^{+3}$	$1.752 \times 10^{+3}$	$1.597 \times 10^{+3}$	$1.584 \times 10^{+3}$	$1.179 \times 10^{+3}$	$1.188 \times 10^{+3}$	$1.323 \times 10^{+3}$	$1.774 \times 10^{+3}$	$1.184 \times 10^{+3}$	$8.677 \times 10^{+2}$	**$8.295 \times 10^{+2}$**	$8.333 \times 10^{+2}$
	Mean	$3.926 \times 10^{+3}$	$1.955 \times 10^{+3}$	$1.801 \times 10^{+3}$	$1.732 \times 10^{+3}$	$1.384 \times 10^{+3}$	$1.424 \times 10^{+3}$	$1.513 \times 10^{+3}$	$2.076 \times 10^{+3}$	$1.311 \times 10^{+3}$	$1.028 \times 10^{+3}$	$9.202 \times 10^{+2}$	**$9.085 \times 10^{+2}$**
	Std	$3.068 \times 10^{+2}$	$8.499 \times 10^{+1}$	$1.030 \times 10^{+2}$	$8.514 \times 10^{+1}$	$1.328 \times 10^{+2}$	$9.527 \times 10^{+1}$	$1.047 \times 10^{+2}$	$1.222 \times 10^{+2}$	$6.245 \times 10^{+1}$	$1.149 \times 10^{+2}$	$4.951 \times 10^{+1}$	**$2.986 \times 10^{+1}$**
	Rank	12	10	9	8	5	6	7	11	4	3	2	1
F8	Best	$1.479 \times 10^{+3}$	$1.292 \times 10^{+3}$	$1.209 \times 10^{+3}$	$1.364 \times 10^{+3}$	$1.118 \times 10^{+3}$	$1.140 \times 10^{+3}$	$1.167 \times 10^{+3}$	$1.390 \times 10^{+3}$	$1.223 \times 10^{+3}$	**$8.691 \times 10^{+2}$**	$8.925 \times 10^{+2}$	$8.836 \times 10^{+2}$
	Mean	$1.560 \times 10^{+3}$	$1.342 \times 10^{+3}$	$1.311 \times 10^{+3}$	$1.409 \times 10^{+3}$	$1.291 \times 10^{+3}$	$1.214 \times 10^{+3}$	$1.244 \times 10^{+3}$	$1.493 \times 10^{+3}$	$1.298 \times 10^{+3}$	$1.033 \times 10^{+3}$	$9.394 \times 10^{+2}$	**$9.147 \times 10^{+2}$**
	Std	$5.374 \times 10^{+1}$	$2.433 \times 10^{+1}$	$8.052 \times 10^{+1}$	$3.342 \times 10^{+1}$	$9.900 \times 10^{+1}$	$5.449 \times 10^{+1}$	$4.437 \times 10^{+1}$	$4.779 \times 10^{+1}$	$3.505 \times 10^{+1}$	$1.298 \times 10^{+2}$	$2.957 \times 10^{+1}$	**$2.308 \times 10^{+1}$**
	Rank	12	9	8	10	7	4	5	11	6	3	2	1
F9	Best	$2.930 \times 10^{+4}$	$2.226 \times 10^{+4}$	$1.663 \times 10^{+4}$	$1.968 \times 10^{+4}$	$1.066 \times 10^{+4}$	$1.262 \times 10^{+4}$	$1.003 \times 10^{+4}$	$3.036 \times 10^{+4}$	$2.198 \times 10^{+4}$	**$9.051 \times 10^{+2}$**	$1.174 \times 10^{+3}$	$1.157 \times 10^{+3}$
	Mean	$4.143 \times 10^{+4}$	$2.746 \times 10^{+4}$	$3.038 \times 10^{+4}$	$2.812 \times 10^{+4}$	$2.084 \times 10^{+4}$	$2.103 \times 10^{+4}$	$1.861 \times 10^{+4}$	$3.815 \times 10^{+4}$	$3.479 \times 10^{+4}$	$3.162 \times 10^{+3}$	$2.407 \times 10^{+3}$	**$1.801 \times 10^{+3}$**
	Std	$8.519 \times 10^{+3}$	$2.742 \times 10^{+3}$	$9.322 \times 10^{+3}$	$3.918 \times 10^{+3}$	$5.806 \times 10^{+3}$	$6.380 \times 10^{+3}$	$3.943 \times 10^{+3}$	$4.549 \times 10^{+3}$	$5.010 \times 10^{+3}$	$8.578 \times 10^{+3}$	$1.141 \times 10^{+3}$	**$5.371 \times 10^{+2}$**
	Rank	12	7	9	8	6	5	4	11	10	2	3	1
F10	Best	$1.282 \times 10^{+4}$	$1.096 \times 10^{+4}$	$9.068 \times 10^{+3}$	$1.424 \times 10^{+4}$	$6.302 \times 10^{+3}$	$6.675 \times 10^{+3}$	$9.520 \times 10^{+3}$	$1.384 \times 10^{+4}$	$8.084 \times 10^{+3}$	$6.749 \times 10^{+3}$	**$6.002 \times 10^{+3}$**	$6.293 \times 10^{+3}$
	Mean	$1.503 \times 10^{+4}$	$1.192 \times 10^{+4}$	$1.196 \times 10^{+4}$	$1.510 \times 10^{+4}$	$1.044 \times 10^{+4}$	$1.035 \times 10^{+4}$	$1.160 \times 10^{+4}$	$1.480 \times 10^{+4}$	**$9.718 \times 10^{+3}$**	$1.108 \times 10^{+4}$	$9.888 \times 10^{+3}$	$1.120 \times 10^{+4}$
	Std	$6.357 \times 10^{+2}$	**$3.306 \times 10^{+2}$**	$1.116 \times 10^{+3}$	$4.308 \times 10^{+2}$	$1.851 \times 10^{+3}$	$2.509 \times 10^{+3}$	$1.046 \times 10^{+3}$	$4.723 \times 10^{+2}$	$5.933 \times 10^{+2}$	$2.949 \times 10^{+3}$	$2.803 \times 10^{+3}$	$2.732 \times 10^{+3}$
	Rank	12	8	9	11	4	3	7	10	1	5	2	6
F11	Best	$1.199 \times 10^{+4}$	$5.377 \times 10^{+3}$	$2.365 \times 10^{+3}$	$6.350 \times 10^{+3}$	$1.417 \times 10^{+3}$	$4.204 \times 10^{+3}$	$3.891 \times 10^{+3}$	$9.859 \times 10^{+3}$	$6.571 \times 10^{+3}$	$1.222 \times 10^{+3}$	$1.204 \times 10^{+3}$	**$1.185 \times 10^{+3}$**
	Mean	$2.175 \times 10^{+4}$	$1.544 \times 10^{+4}$	$2.991 \times 10^{+3}$	$9.164 \times 10^{+3}$	$1.871 \times 10^{+3}$	$9.849 \times 10^{+3}$	$8.162 \times 10^{+3}$	$1.894 \times 10^{+4}$	$1.541 \times 10^{+4}$	$3.949 \times 10^{+3}$	$1.284 \times 10^{+3}$	**$1.252 \times 10^{+3}$**
	Std	$1.029 \times 10^{+4}$	$4.383 \times 10^{+3}$	$4.727 \times 10^{+2}$	$1.860 \times 10^{+3}$	$2.936 \times 10^{+2}$	$3.241 \times 10^{+3}$	$2.748 \times 10^{+3}$	$4.452 \times 10^{+3}$	$5.309 \times 10^{+3}$	$4.826 \times 10^{+3}$	$4.765 \times 10^{+1}$	**$4.501 \times 10^{+1}$**
	Rank	12	10	5	7	3	8	6	11	9	4	2	1
F12	Best	$7.882 \times 10^{+9}$	$1.226 \times 10^{+9}$	$2.221 \times 10^{+8}$	$9.744 \times 10^{+9}$	$2.397 \times 10^{+7}$	$1.691 \times 10^{+9}$	$1.554 \times 10^{+9}$	$2.491 \times 10^{+10}$	$1.121 \times 10^{+9}$	$3.249 \times 10^{+6}$	$3.591 \times 10^{+5}$	**$2.996 \times 10^{+5}$**
	Mean	$1.927 \times 10^{+10}$	$1.892 \times 10^{+9}$	$8.100 \times 10^{+8}$	$1.696 \times 10^{+10}$	$3.832 \times 10^{+8}$	$9.483 \times 10^{+9}$	$5.610 \times 10^{+9}$	$4.355 \times 10^{+10}$	$4.632 \times 10^{+9}$	$4.006 \times 10^{+8}$	$1.484 \times 10^{+6}$	**$1.268 \times 10^{+6}$**
	Std	$5.706 \times 10^{+9}$	$3.666 \times 10^{+8}$	$3.267 \times 10^{+8}$	$4.034 \times 10^{+9}$	$4.681 \times 10^{+8}$	$4.420 \times 10^{+9}$	$3.937 \times 10^{+9}$	$1.302 \times 10^{+10}$	$2.129 \times 10^{+9}$	$1.330 \times 10^{+9}$	$7.934 \times 10^{+5}$	**$6.241 \times 10^{+5}$**
	Rank	11	6	5	10	4	9	8	12	7	3	2	1
F13	Best	$2.274 \times 10^{+9}$	$4.226 \times 10^{+7}$	$2.113 \times 10^{+6}$	$2.412 \times 10^{+9}$	$1.965 \times 10^{+5}$	$1.431 \times 10^{+8}$	$6.796 \times 10^{+7}$	$2.119 \times 10^{+9}$	$1.475 \times 10^{+8}$	$2.662 \times 10^{+3}$	$2.139 \times 10^{+3}$	**$1.581 \times 10^{+3}$**
	Mean	$7.546 \times 10^{+9}$	$7.986 \times 10^{+7}$	$2.858 \times 10^{+7}$	$4.571 \times 10^{+9}$	$1.893 \times 10^{+7}$	$1.998 \times 10^{+9}$	$7.647 \times 10^{+8}$	$1.428 \times 10^{+10}$	$6.503 \times 10^{+8}$	$6.608 \times 10^{+5}$	$6.411 \times 10^{+3}$	**$5.182 \times 10^{+3}$**
	Std	$3.963 \times 10^{+9}$	$2.038 \times 10^{+7}$	$3.132 \times 10^{+7}$	$1.709 \times 10^{+9}$	$2.531 \times 10^{+7}$	$2.561 \times 10^{+9}$	$1.368 \times 10^{+9}$	$7.017 \times 10^{+9}$	$4.324 \times 10^{+8}$	$1.788 \times 10^{+6}$	**$5.575 \times 10^{+3}$**	$5.741 \times 10^{+3}$
	Rank	11	6	5	10	4	9	7	12	8	3	2	1
F14	Best	$1.780 \times 10^{+6}$	$1.384 \times 10^{+6}$	$2.252 \times 10^{+5}$	$1.469 \times 10^{+6}$	$6.254 \times 10^{+4}$	$1.768 \times 10^{+5}$	$5.188 \times 10^{+5}$	$3.030 \times 10^{+6}$	$4.378 \times 10^{+5}$	$1.630 \times 10^{+4}$	$1.760 \times 10^{+4}$	**$6.093 \times 10^{+3}$**
	Mean	$7.509 \times 10^{+6}$	$4.593 \times 10^{+6}$	$2.880 \times 10^{+6}$	$5.343 \times 10^{+6}$	$2.241 \times 10^{+6}$	$3.282 \times 10^{+6}$	$3.646 \times 10^{+6}$	$2.461 \times 10^{+7}$	$2.974 \times 10^{+6}$	$8.820 \times 10^{+5}$	$5.410 \times 10^{+4}$	**$3.617 \times 10^{+4}$**
	Std	$1.062 \times 10^{+7}$	$2.308 \times 10^{+6}$	$2.421 \times 10^{+6}$	$2.313 \times 10^{+6}$	$2.873 \times 10^{+6}$	$5.179 \times 10^{+6}$	$3.153 \times 10^{+6}$	$1.395 \times 10^{+7}$	$2.246 \times 10^{+6}$	$1.119 \times 10^{+6}$	**$2.547 \times 10^{+4}$**	$3.052 \times 10^{+4}$
	Rank	10	9	7	11	4	5	8	12	6	3	**2**	1
F15	Best	$6.976 \times 10^{+8}$	$7.917 \times 10^{+5}$	$1.367 \times 10^{+5}$	$2.378 \times 10^{+8}$	$3.205 \times 10^{+4}$	$9.153 \times 10^{+5}$	$2.891 \times 10^{+6}$	$4.860 \times 10^{+8}$	$6.807 \times 10^{+5}$	$2.403 \times 10^{+3}$	**$1.643 \times 10^{+3}$**	$1.671 \times 10^{+3}$
	Mean	$1.889 \times 10^{+9}$	$6.727 \times 10^{+6}$	$2.828 \times 10^{+6}$	$6.999 \times 10^{+8}$	$9.874 \times 10^{+6}$	$4.373 \times 10^{+8}$	$2.719 \times 10^{+8}$	$2.215 \times 10^{+9}$	$1.600 \times 10^{+7}$	$6.701 \times 10^{+4}$	$7.504 \times 10^{+3}$	**$5.318 \times 10^{+3}$**
	Std	$8.086 \times 10^{+8}$	$3.558 \times 10^{+6}$	$4.183 \times 10^{+6}$	$2.545 \times 10^{+8}$	$2.869 \times 10^{+7}$	$1.204 \times 10^{+9}$	$3.568 \times 10^{+8}$	$1.377 \times 10^{+9}$	$1.579 \times 10^{+7}$	$1.346 \times 10^{+5}$	$4.445 \times 10^{+3}$	**$4.296 \times 10^{+3}$**
	Rank	11	6	5	10	4	8	9	12	7	3	2	1
F16	Best	$5.721 \times 10^{+3}$	$4.291 \times 10^{+3}$	$4.244 \times 10^{+3}$	$5.140 \times 10^{+3}$	$3.700 \times 10^{+3}$	$2.814 \times 10^{+3}$	$3.313 \times 10^{+3}$	$5.438 \times 10^{+3}$	$3.739 \times 10^{+3}$	$3.305 \times 10^{+3}$	$2.434 \times 10^{+3}$	**$2.023 \times 10^{+3}$**
	Mean	$6.343 \times 10^{+3}$	$4.680 \times 10^{+3}$	$5.583 \times 10^{+3}$	$5.978 \times 10^{+3}$	$4.928 \times 10^{+3}$	$3.903 \times 10^{+3}$	$4.432 \times 10^{+3}$	$7.270 \times 10^{+3}$	$4.365 \times 10^{+3}$	$4.626 \times 10^{+3}$	$3.018 \times 10^{+3}$	**$2.867 \times 10^{+3}$**
	Std	$4.544 \times 10^{+2}$	**$1.973 \times 10^{+2}$**	$8.004 \times 10^{+2}$	$4.045 \times 10^{+2}$	$6.004 \times 10^{+2}$	$5.256 \times 10^{+2}$	$4.903 \times 10^{+2}$	$1.035 \times 10^{+3}$	$2.974 \times 10^{+2}$	$9.407 \times 10^{+2}$	$3.149 \times 10^{+2}$	$3.909 \times 10^{+2}$
	Rank	11	7	9	10	8	3	5	12	4	6	2	1
F17	Best	$5.742 \times 10^{+3}$	$3.612 \times 10^{+3}$	$3.534 \times 10^{+3}$	$4.140 \times 10^{+3}$	$3.313 \times 10^{+3}$	$2.862 \times 10^{+3}$	$2.941 \times 10^{+3}$	$4.060 \times 10^{+3}$	$2.865 \times 10^{+3}$	$2.641 \times 10^{+3}$	**$2.176 \times 10^{+3}$**	$2.337 \times 10^{+3}$
	Mean	$6.736 \times 10^{+3}$	$3.941 \times 10^{+3}$	$4.203 \times 10^{+3}$	$4.747 \times 10^{+3}$	$4.096 \times 10^{+3}$	$3.494 \times 10^{+3}$	$3.598 \times 10^{+3}$	$5.137 \times 10^{+3}$	$3.917 \times 10^{+3}$	$3.373 \times 10^{+3}$	**$2.796 \times 10^{+3}$**	$2.838 \times 10^{+3}$
	Std	$9.123 \times 10^{+2}$	**$1.897 \times 10^{+2}$**	$4.095 \times 10^{+2}$	$3.439 \times 10^{+2}$	$3.871 \times 10^{+2}$	$4.215 \times 10^{+2}$	$3.730 \times 10^{+2}$	$8.702 \times 10^{+2}$	$3.291 \times 10^{+2}$	$4.183 \times 10^{+2}$	$3.342 \times 10^{+2}$	$2.573 \times 10^{+2}$
	Rank	12	7	9	10	8	4	5	11	6	3	2	1

Table A6. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F18	Best	$1.309 \times 10^{+7}$	$5.898 \times 10^{+6}$	$4.620 \times 10^{+6}$	$1.426 \times 10^{+7}$	$3.366 \times 10^{+5}$	$5.549 \times 10^{+5}$	$3.147 \times 10^{+6}$	$1.831 \times 10^{+7}$	$2.425 \times 10^{+6}$	$1.057 \times 10^{+5}$	$1.210 \times 10^{+5}$	**$8.724 \times 10^{+4}$**
	Mean	$3.744 \times 10^{+7}$	$1.815 \times 10^{+7}$	$2.687 \times 10^{+7}$	$3.186 \times 10^{+7}$	$7.151 \times 10^{+6}$	$1.469 \times 10^{+7}$	$2.383 \times 10^{+7}$	$7.192 \times 10^{+7}$	$9.474 \times 10^{+6}$	$4.044 \times 10^{+6}$	$1.050 \times 10^{+6}$	**$6.163 \times 10^{+5}$**
	Std	$2.213 \times 10^{+7}$	$5.591 \times 10^{+6}$	$2.704 \times 10^{+7}$	$1.641 \times 10^{+7}$	$7.371 \times 10^{+6}$	$1.586 \times 10^{+7}$	$2.388 \times 10^{+7}$	$2.599 \times 10^{+7}$	$4.860 \times 10^{+6}$	$4.272 \times 10^{+6}$	$7.370 \times 10^{+5}$	**$4.101 \times 10^{+5}$**
	Rank	11	8	9	10	4	6	7	12	5	3	2	1
F19	Best	$2.784 \times 10^{+8}$	$1.588 \times 10^{+6}$	$1.302 \times 10^{+5}$	$1.538 \times 10^{+8}$	$4.652 \times 10^{+3}$	$1.199 \times 10^{+5}$	$1.297 \times 10^{+6}$	$1.465 \times 10^{+8}$	$4.044 \times 10^{+6}$	$4.280 \times 10^{+3}$	$2.930 \times 10^{+3}$	**$1.986 \times 10^{+3}$**
	Mean	$6.971 \times 10^{+8}$	$3.952 \times 10^{+6}$	$5.838 \times 10^{+6}$	$4.835 \times 10^{+8}$	$2.261 \times 10^{+6}$	$1.627 \times 10^{+8}$	$3.147 \times 10^{+7}$	$5.075 \times 10^{+8}$	$3.796 \times 10^{+7}$	$4.316 \times 10^{+4}$	$1.601 \times 10^{+4}$	**$1.113 \times 10^{+4}$**
	Std	$4.099 \times 10^{+8}$	$1.375 \times 10^{+6}$	$7.404 \times 10^{+6}$	$2.693 \times 10^{+8}$	$3.874 \times 10^{+6}$	$3.640 \times 10^{+8}$	$4.400 \times 10^{+7}$	$2.686 \times 10^{+8}$	$2.468 \times 10^{+7}$	$6.657 \times 10^{+4}$	$7.634 \times 10^{+3}$	**$6.572 \times 10^{+3}$**
	Rank	12	6	5	10	4	8	7	11	9	3	2	1
F20	Best	$3.840 \times 10^{+3}$	$3.360 \times 10^{+3}$	$2.770 \times 10^{+3}$	$3.773 \times 10^{+3}$	$2.725 \times 10^{+3}$	$2.624 \times 10^{+3}$	$2.843 \times 10^{+3}$	$3.606 \times 10^{+3}$	$3.109 \times 10^{+3}$	$2.516 \times 10^{+3}$	$2.320 \times 10^{+3}$	$2.245 \times 10^{+3}$
	Mean	$4.189 \times 10^{+3}$	$3.676 \times 10^{+3}$	$3.732 \times 10^{+3}$	$4.123 \times 10^{+3}$	$3.552 \times 10^{+3}$	$3.348 \times 10^{+3}$	$3.312 \times 10^{+3}$	$4.029 \times 10^{+3}$	$3.482 \times 10^{+3}$	$3.536 \times 10^{+3}$	$3.053 \times 10^{+3}$	**$2.989 \times 10^{+3}$**
	Std	$1.854 \times 10^{+2}$	**$1.503 \times 10^{+2}$**	$4.081 \times 10^{+2}$	$2.162 \times 10^{+2}$	$3.987 \times 10^{+2}$	$4.270 \times 10^{+2}$	$2.462 \times 10^{+2}$	$2.354 \times 10^{+2}$	$2.006 \times 10^{+2}$	$4.890 \times 10^{+2}$	$3.695 \times 10^{+2}$	$4.382 \times 10^{+2}$
	Rank	12	8	9	11	7	4	3	10	5	6	2	1
F21	Best	$2.945 \times 10^{+3}$	$2.760 \times 10^{+3}$	$2.826 \times 10^{+3}$	$2.849 \times 10^{+3}$	$2.695 \times 10^{+3}$	$2.596 \times 10^{+3}$	$2.679 \times 10^{+3}$	$2.926 \times 10^{+3}$	$2.697 \times 10^{+3}$	**$2.376 \times 10^{+3}$**	$2.384 \times 10^{+3}$	$2.383 \times 10^{+3}$
	Mean	$3.026 \times 10^{+3}$	$2.832 \times 10^{+3}$	$2.979 \times 10^{+3}$	$2.917 \times 10^{+3}$	$2.865 \times 10^{+3}$	$2.698 \times 10^{+3}$	$2.747 \times 10^{+3}$	$3.094 \times 10^{+3}$	$2.801 \times 10^{+3}$	$2.561 \times 10^{+3}$	**$2.418 \times 10^{+3}$**	**$2.418 \times 10^{+3}$**
	Std	$5.582 \times 10^{+1}$	$2.439 \times 10^{+1}$	$1.090 \times 10^{+2}$	$3.265 \times 10^{+1}$	$8.914 \times 10^{+1}$	$5.406 \times 10^{+1}$	$4.882 \times 10^{+1}$	$7.494 \times 10^{+1}$	$4.333 \times 10^{+1}$	$1.227 \times 10^{+2}$	$2.054 \times 10^{+1}$	**$1.696 \times 10^{+1}$**
	Rank	11	7	10	9	8	4	5	12	6	3	2	1
F22	Best	$1.473 \times 10^{+4}$	$1.299 \times 10^{+4}$	$1.007 \times 10^{+4}$	$1.530 \times 10^{+4}$	$9.303 \times 10^{+3}$	$9.154 \times 10^{+3}$	$7.739 \times 10^{+3}$	$1.500 \times 10^{+4}$	$1.068 \times 10^{+4}$	$8.292 \times 10^{+3}$	$2.300 \times 10^{+3}$	**$2.300 \times 10^{+3}$**
	Mean	$1.641 \times 10^{+4}$	$1.352 \times 10^{+4}$	$1.358 \times 10^{+4}$	$1.660 \times 10^{+4}$	$1.202 \times 10^{+4}$	$1.209 \times 10^{+4}$	$1.263 \times 10^{+4}$	$1.628 \times 10^{+4}$	$1.155 \times 10^{+4}$	$1.248 \times 10^{+4}$	**$9.581 \times 10^{+3}$**	$9.978 \times 10^{+3}$
	Std	$6.018 \times 10^{+2}$	**$3.104 \times 10^{+2}$**	$1.437 \times 10^{+3}$	$5.160 \times 10^{+2}$	$1.724 \times 10^{+3}$	$1.788 \times 10^{+3}$	$2.024 \times 10^{+3}$	$6.043 \times 10^{+2}$	$4.741 \times 10^{+2}$	$2.925 \times 10^{+3}$	$3.205 \times 10^{+3}$	$4.411 \times 10^{+3}$
	Rank	11	9	8	12	4	5	7	10	3	6	1	2
F23	Best	$3.415 \times 10^{+3}$	$3.207 \times 10^{+3}$	$3.332 \times 10^{+3}$	$3.530 \times 10^{+3}$	$3.113 \times 10^{+3}$	$3.074 \times 10^{+3}$	$3.223 \times 10^{+3}$	$3.794 \times 10^{+3}$	$3.245 \times 10^{+3}$	**$2.752 \times 10^{+3}$**	$2.825 \times 10^{+3}$	$2.826 \times 10^{+3}$
	Mean	$3.666 \times 10^{+3}$	$3.251 \times 10^{+3}$	$3.746 \times 10^{+3}$	$3.643 \times 10^{+3}$	$3.340 \times 10^{+3}$	$3.251 \times 10^{+3}$	$3.336 \times 10^{+3}$	$4.045 \times 10^{+3}$	$3.375 \times 10^{+3}$	$2.942 \times 10^{+3}$	**$2.867 \times 10^{+3}$**	$2.869 \times 10^{+3}$
	Std	$1.555 \times 10^{+2}$	**$2.003 \times 10^{+1}$**	$1.873 \times 10^{+2}$	$6.873 \times 10^{+1}$	$1.247 \times 10^{+2}$	$1.122 \times 10^{+2}$	$6.499 \times 10^{+1}$	$1.410 \times 10^{+2}$	$5.554 \times 10^{+1}$	$1.097 \times 10^{+2}$	$3.430 \times 10^{+1}$	$2.935 \times 10^{+1}$
	Rank	10	4	11	9	6	5	7	12	8	3	2	1
F24	Best	$3.546 \times 10^{+3}$	$3.371 \times 10^{+3}$	$3.499 \times 10^{+3}$	$3.704 \times 10^{+3}$	$3.365 \times 10^{+3}$	$3.290 \times 10^{+3}$	$3.352 \times 10^{+3}$	$3.908 \times 10^{+3}$	$3.381 \times 10^{+3}$	$3.009 \times 10^{+3}$	$2.998 \times 10^{+3}$	**$2.961 \times 10^{+3}$**
	Mean	$3.824 \times 10^{+3}$	$3.429 \times 10^{+3}$	$3.797 \times 10^{+3}$	$3.814 \times 10^{+3}$	$3.682 \times 10^{+3}$	$3.506 \times 10^{+3}$	$3.495 \times 10^{+3}$	$4.273 \times 10^{+3}$	$3.502 \times 10^{+3}$	$3.204 \times 10^{+3}$	**$3.063 \times 10^{+3}$**	**$3.063 \times 10^{+3}$**
	Std	$1.487 \times 10^{+2}$	**$2.265 \times 10^{+1}$**	$1.584 \times 10^{+2}$	$5.857 \times 10^{+1}$	$1.349 \times 10^{+2}$	$1.165 \times 10^{+2}$	$8.411 \times 10^{+1}$	$1.466 \times 10^{+2}$	$5.060 \times 10^{+1}$	$1.328 \times 10^{+2}$	$4.721 \times 10^{+1}$	$4.779 \times 10^{+1}$
	Rank	10	4	9	11	8	6	5	12	7	3	2	1
F25	Best	$9.505 \times 10^{+3}$	$4.222 \times 10^{+3}$	$3.304 \times 10^{+3}$	$6.009 \times 10^{+3}$	$3.023 \times 10^{+3}$	$4.382 \times 10^{+3}$	$4.706 \times 10^{+3}$	$9.462 \times 10^{+3}$	$4.101 \times 10^{+3}$	**$2.931 \times 10^{+3}$**	$3.023 \times 10^{+3}$	$2.975 \times 10^{+3}$
	Mean	$1.603 \times 10^{+4}$	$4.965 \times 10^{+3}$	$3.535 \times 10^{+3}$	$7.691 \times 10^{+3}$	$3.344 \times 10^{+3}$	$5.911 \times 10^{+3}$	$6.422 \times 10^{+3}$	$1.196 \times 10^{+4}$	$5.433 \times 10^{+3}$	**$3.030 \times 10^{+3}$**	$3.075 \times 10^{+3}$	$3.060 \times 10^{+3}$
	Std	$5.051 \times 10^{+3}$	$3.412 \times 10^{+2}$	$1.382 \times 10^{+2}$	$6.786 \times 10^{+2}$	$9.561 \times 10^{+2}$	$9.632 \times 10^{+2}$	$1.038 \times 10^{+3}$	$1.562 \times 10^{+3}$	$8.239 \times 10^{+2}$	$1.803 \times 10^{+2}$	**$2.624 \times 10^{+1}$**	$3.082 \times 10^{+1}$
	Rank	12	6	5	10	4	8	9	11	7	1	3	2
F26	Best	$1.042 \times 10^{+4}$	$8.055 \times 10^{+3}$	$1.017 \times 10^{+4}$	$1.165 \times 10^{+4}$	$4.736 \times 10^{+3}$	$8.082 \times 10^{+3}$	$8.323 \times 10^{+3}$	$1.496 \times 10^{+4}$	$7.902 \times 10^{+3}$	$3.688 \times 10^{+3}$	$2.903 \times 10^{+3}$	**$2.902 \times 10^{+3}$**
	Mean	$1.265 \times 10^{+4}$	$8.492 \times 10^{+3}$	$1.342 \times 10^{+4}$	$1.292 \times 10^{+4}$	$1.027 \times 10^{+4}$	$9.468 \times 10^{+3}$	$1.128 \times 10^{+4}$	$1.650 \times 10^{+4}$	$9.659 \times 10^{+3}$	**$5.524 \times 10^{+3}$**	$5.740 \times 10^{+3}$	$5.581 \times 10^{+3}$
	Std	$1.411 \times 10^{+3}$	**$2.458 \times 10^{+2}$**	$1.513 \times 10^{+3}$	$6.440 \times 10^{+2}$	$1.818 \times 10^{+3}$	$9.777 \times 10^{+2}$	$1.635 \times 10^{+3}$	$7.588 \times 10^{+2}$	$6.647 \times 10^{+2}$	$9.396 \times 10^{+2}$	$9.282 \times 10^{+2}$	$7.247 \times 10^{+2}$
	Rank	9	4	11	10	7	5	8	12	6	1	3	2
F27	Best	$3.470 \times 10^{+3}$	$3.495 \times 10^{+3}$	$3.805 \times 10^{+3}$	$4.291 \times 10^{+3}$	$3.397 \times 10^{+3}$	$3.785 \times 10^{+3}$	$3.716 \times 10^{+3}$	$4.563 \times 10^{+3}$	$3.844 \times 10^{+3}$	$3.200 \times 10^{+3}$	$3.315 \times 10^{+3}$	$3.333 \times 10^{+3}$
	Mean	$4.087 \times 10^{+3}$	$3.542 \times 10^{+3}$	$4.579 \times 10^{+3}$	$4.709 \times 10^{+3}$	$3.696 \times 10^{+3}$	$4.001 \times 10^{+3}$	$4.458 \times 10^{+3}$	$5.408 \times 10^{+3}$	$4.092 \times 10^{+3}$	**$3.200 \times 10^{+3}$**	$3.523 \times 10^{+3}$	$3.469 \times 10^{+3}$
	Std	$3.006 \times 10^{+2}$	$2.795 \times 10^{+1}$	$6.171 \times 10^{+2}$	$2.019 \times 10^{+2}$	$1.755 \times 10^{+2}$	$1.604 \times 10^{+2}$	$3.940 \times 10^{+2}$	$4.242 \times 10^{+2}$	$1.432 \times 10^{+2}$	**2.736×10^{-4}**	$8.152 \times 10^{+1}$	$8.278 \times 10^{+1}$
	Rank	7	4	10	11	5	6	9	12	8	1	3	2
F28	Best	$8.646 \times 10^{+3}$	$7.099 \times 10^{+3}$	$3.804 \times 10^{+3}$	$6.745 \times 10^{+3}$	$3.410 \times 10^{+3}$	$5.369 \times 10^{+3}$	$5.287 \times 10^{+3}$	$9.196 \times 10^{+3}$	$5.631 \times 10^{+3}$	$3.300 \times 10^{+3}$	$3.308 \times 10^{+3}$	**$3.283 \times 10^{+3}$**
	Mean	$1.044 \times 10^{+4}$	$7.993 \times 10^{+3}$	$4.337 \times 10^{+3}$	$7.848 \times 10^{+3}$	$4.963 \times 10^{+3}$	$6.257 \times 10^{+3}$	$6.747 \times 10^{+3}$	$1.057 \times 10^{+4}$	$8.842 \times 10^{+3}$	**$3.300 \times 10^{+3}$**	$3.339 \times 10^{+3}$	$3.324 \times 10^{+3}$
	Std	$1.040 \times 10^{+3}$	$2.298 \times 10^{+2}$	$2.893 \times 10^{+2}$	$8.807 \times 10^{+2}$	$2.338 \times 10^{+3}$	$5.833 \times 10^{+2}$	$7.777 \times 10^{+2}$	$9.516 \times 10^{+2}$	$1.279 \times 10^{+3}$	**4.363×10^{-4}**	$2.795 \times 10^{+1}$	$2.937 \times 10^{+1}$
	Rank	11	9	4	8	5	6	7	12	10	1	3	2
F29	Best	$6.138 \times 10^{+3}$	$4.687 \times 10^{+3}$	$7.225 \times 10^{+3}$	$6.251 \times 10^{+3}$	$4.968 \times 10^{+3}$	$4.644 \times 10^{+3}$	$5.859 \times 10^{+3}$	$7.951 \times 10^{+3}$	$5.153 \times 10^{+3}$	$3.647 \times 10^{+3}$	$3.563 \times 10^{+3}$	**$3.556 \times 10^{+3}$**
	Mean	$8.233 \times 10^{+3}$	$5.113 \times 10^{+3}$	$8.553 \times 10^{+3}$	$8.100 \times 10^{+3}$	$6.038 \times 10^{+3}$	$5.812 \times 10^{+3}$	$7.357 \times 10^{+3}$	$1.568 \times 10^{+4}$	$6.068 \times 10^{+3}$	$4.497 \times 10^{+3}$	$4.193 \times 10^{+3}$	**$4.112 \times 10^{+3}$**
	Std	$9.857 \times 10^{+2}$	**$1.925 \times 10^{+2}$**	$1.066 \times 10^{+3}$	$7.043 \times 10^{+2}$	$8.666 \times 10^{+2}$	$6.004 \times 10^{+2}$	$9.367 \times 10^{+2}$	$7.773 \times 10^{+3}$	$3.657 \times 10^{+2}$	$6.886 \times 10^{+2}$	$2.297 \times 10^{+2}$	$3.429 \times 10^{+2}$
	Rank	9	4	11	10	6	5	8	12	7	3	2	1

Table A6. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F30	Best	$5.260 \times 10^{+8}$	$7.212 \times 10^{+6}$	$5.796 \times 10^{+7}$	$4.457 \times 10^{+8}$	$2.870 \times 10^{+6}$	$1.475 \times 10^{+8}$	$8.789 \times 10^{+7}$	$7.060 \times 10^{+8}$	$4.824 \times 10^{+7}$	**$7.861 \times 10^{+3}$**	$8.108 \times 10^{+5}$	$9.260 \times 10^{+5}$
	Mean	$1.606 \times 10^{+9}$	$1.447 \times 10^{+7}$	$2.078 \times 10^{+8}$	$9.498 \times 10^{+8}$	$2.637 \times 10^{+7}$	$7.102 \times 10^{+8}$	$2.959 \times 10^{+8}$	$1.976 \times 10^{+9}$	$1.374 \times 10^{+8}$	$3.772 \times 10^{+6}$	**$1.315 \times 10^{+6}$**	$1.347 \times 10^{+6}$
	Std	$8.562 \times 10^{+8}$	$5.991 \times 10^{+6}$	$8.939 \times 10^{+7}$	$3.002 \times 10^{+8}$	$2.030 \times 10^{+7}$	$9.401 \times 10^{+8}$	$1.318 \times 10^{+8}$	$6.976 \times 10^{+8}$	$5.829 \times 10^{+7}$	$1.600 \times 10^{+7}$	$3.563 \times 10^{+5}$	**$2.757 \times 10^{+5}$**
	Rank	11	4	7	10	5	9	8	12	6	1	2	3
Mean Rank		11.03	6.86	7.72	9.66	5.45	5.72	6.69	11.31	6.69	3.10	2.20	**1.55**
+/=/−		29/0/0	28/1/0	28/1/0	29/0/0	27/2/0	26/3/0	27/1/1	27/1/1	27/1/1	21/5/3	17/6/6	~

Table A7. Comparison results of EFNO with other algorithms on CEC2017 (Dim = 100).

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F1	Best	$2.530 \times 10^{+11}$	$1.141 \times 10^{+11}$	$4.854 \times 10^{+9}$	$1.650 \times 10^{+11}$	$3.946 \times 10^{+8}$	$1.103 \times 10^{+11}$	$7.818 \times 10^{+10}$	$2.078 \times 10^{+11}$	$1.688 \times 10^{+10}$	$2.604 \times 10^{+3}$	$4.595 \times 10^{+3}$	**$1.435 \times 10^{+2}$**
	Mean	$3.368 \times 10^{+11}$	$1.251 \times 10^{+11}$	$9.621 \times 10^{+9}$	$1.837 \times 10^{+11}$	$5.902 \times 10^{+9}$	$1.334 \times 10^{+11}$	$1.155 \times 10^{+11}$	$2.461 \times 10^{+11}$	$2.666 \times 10^{+10}$	$1.953 \times 10^{+9}$	$7.038 \times 10^{+4}$	**$1.006 \times 10^{+4}$**
	Std	$5.163 \times 10^{+10}$	$5.718 \times 10^{+9}$	$2.075 \times 10^{+9}$	$1.061 \times 10^{+10}$	$2.178 \times 10^{+10}$	$1.154 \times 10^{+10}$	$2.330 \times 10^{+10}$	$2.380 \times 10^{+10}$	$4.864 \times 10^{+9}$	$4.626 \times 10^{+9}$	$1.819 \times 10^{+5}$	**$1.010 \times 10^{+4}$**
	Rank	12	8	5	10	4	9	7	11	6	3	2	1
F3	Best	$8.517 \times 10^{+5}$	$7.960 \times 10^{+5}$	$6.391 \times 10^{+5}$	$3.362 \times 10^{+5}$	$2.865 \times 10^{+5}$	**$2.349 \times 10^{+5}$**	$2.510 \times 10^{+5}$	$3.169 \times 10^{+5}$	$6.761 \times 10^{+5}$	$3.300 \times 10^{+5}$	$3.008 \times 10^{+5}$	$2.649 \times 10^{+5}$
	Mean	$1.017 \times 10^{+6}$	$9.489 \times 10^{+5}$	$8.996 \times 10^{+5}$	$4.008 \times 10^{+5}$	$3.482 \times 10^{+5}$	**$2.743 \times 10^{+5}$**	$3.001 \times 10^{+5}$	$4.271 \times 10^{+5}$	$8.966 \times 10^{+5}$	$8.612 \times 10^{+5}$	$3.784 \times 10^{+5}$	$3.915 \times 10^{+5}$
	Std	$1.205 \times 10^{+5}$	$7.885 \times 10^{+4}$	$1.414 \times 10^{+5}$	$3.589 \times 10^{+4}$	**$1.579 \times 10^{+4}$**	$1.792 \times 10^{+4}$	$2.555 \times 10^{+4}$	$5.190 \times 10^{+4}$	$9.243 \times 10^{+4}$	$3.834 \times 10^{+5}$	$3.594 \times 10^{+4}$	$6.945 \times 10^{+4}$
	Rank	12	11	10	6	3	1	2	7	9	8	4	5
F4	Best	$4.255 \times 10^{+4}$	$1.325 \times 10^{+4}$	$2.155 \times 10^{+3}$	$2.687 \times 10^{+4}$	$9.158 \times 10^{+2}$	$1.177 \times 10^{+4}$	$1.026 \times 10^{+4}$	$5.156 \times 10^{+4}$	$2.696 \times 10^{+3}$	**$4.969 \times 10^{+2}$**	$6.399 \times 10^{+2}$	$6.221 \times 10^{+2}$
	Mean	$7.224 \times 10^{+4}$	$1.600 \times 10^{+4}$	$2.975 \times 10^{+3}$	$3.670 \times 10^{+4}$	$2.478 \times 10^{+3}$	$1.985 \times 10^{+4}$	$1.895 \times 10^{+4}$	$7.329 \times 10^{+4}$	$4.795 \times 10^{+3}$	$2.308 \times 10^{+3}$	$7.378 \times 10^{+2}$	**$6.989 \times 10^{+2}$**
	Std	$2.264 \times 10^{+4}$	$1.806 \times 10^{+3}$	$4.791 \times 10^{+2}$	$5.766 \times 10^{+3}$	$4.521 \times 10^{+3}$	$4.350 \times 10^{+3}$	$4.683 \times 10^{+3}$	$1.330 \times 10^{+4}$	$1.021 \times 10^{+3}$	$7.733 \times 10^{+3}$	$5.858 \times 10^{+1}$	**$4.175 \times 10^{+1}$**
	Rank	11	7	5	10	4	9	8	12	6	3	2	1
F5	Best	$2.031 \times 10^{+3}$	$1.920 \times 10^{+3}$	$1.469 \times 10^{+3}$	$1.806 \times 10^{+3}$	$1.118 \times 10^{+3}$	$1.283 \times 10^{+3}$	$1.403 \times 10^{+3}$	$1.906 \times 10^{+3}$	$1.555 \times 10^{+3}$	$8.206 \times 10^{+2}$	$7.787 \times 10^{+2}$	**$7.746 \times 10^{+2}$**
	Mean	$2.293 \times 10^{+3}$	$1.986 \times 10^{+3}$	$1.647 \times 10^{+3}$	$1.962 \times 10^{+3}$	$1.606 \times 10^{+3}$	$1.551 \times 10^{+3}$	$1.634 \times 10^{+3}$	$2.067 \times 10^{+3}$	$1.686 \times 10^{+3}$	$1.101 \times 10^{+3}$	$8.784 \times 10^{+2}$	**$8.541 \times 10^{+2}$**
	Std	$1.487 \times 10^{+2}$	**$3.240 \times 10^{+1}$**	$1.089 \times 10^{+2}$	$6.313 \times 10^{+1}$	$2.649 \times 10^{+2}$	$1.319 \times 10^{+2}$	$1.008 \times 10^{+2}$	$6.972 \times 10^{+1}$	$6.122 \times 10^{+1}$	$1.458 \times 10^{+2}$	$5.585 \times 10^{+1}$	$4.153 \times 10^{+1}$
	Rank	12	10	7	9	5	4	6	11	8	3	2	1
F6	Best	$7.089 \times 10^{+2}$	$6.600 \times 10^{+2}$	$6.798 \times 10^{+2}$	$6.895 \times 10^{+2}$	$6.547 \times 10^{+2}$	$6.636 \times 10^{+2}$	$6.622 \times 10^{+2}$	$6.969 \times 10^{+2}$	$6.486 \times 10^{+2}$	**$6.000 \times 10^{+2}$**	$6.140 \times 10^{+2}$	$6.098 \times 10^{+2}$
	Mean	$7.245 \times 10^{+2}$	$6.658 \times 10^{+2}$	$6.951 \times 10^{+2}$	$6.979 \times 10^{+2}$	$6.704 \times 10^{+2}$	$6.720 \times 10^{+2}$	$6.741 \times 10^{+2}$	$7.071 \times 10^{+2}$	$6.652 \times 10^{+2}$	**$6.036 \times 10^{+2}$**	$6.218 \times 10^{+2}$	$6.201 \times 10^{+2}$
	Std	$8.052 \times 10^{+0}$	**$2.314 \times 10^{+0}$**	$8.930 \times 10^{+0}$	$3.755 \times 10^{+0}$	$1.048 \times 10^{+1}$	$4.658 \times 10^{+0}$	$5.815 \times 10^{+0}$	$4.594 \times 10^{+0}$	$5.241 \times 10^{+0}$	$6.177 \times 10^{+0}$	$4.051 \times 10^{+0}$	$5.378 \times 10^{+0}$
	Rank	12	4	9	10	6	7	8	11	5	1	3	2
F7	Best	$7.721 \times 10^{+3}$	$5.826 \times 10^{+3}$	$3.148 \times 10^{+3}$	$3.436 \times 10^{+3}$	$2.455 \times 10^{+3}$	$2.584 \times 10^{+3}$	$2.681 \times 10^{+3}$	$3.672 \times 10^{+3}$	$1.829 \times 10^{+3}$	**$1.093 \times 10^{+3}$**	$1.248 \times 10^{+3}$	$1.189 \times 10^{+3}$
	Mean	$9.027 \times 10^{+3}$	$6.485 \times 10^{+3}$	$3.506 \times 10^{+3}$	$3.709 \times 10^{+3}$	$2.847 \times 10^{+3}$	$2.913 \times 10^{+3}$	$3.008 \times 10^{+3}$	$4.310 \times 10^{+3}$	$2.046 \times 10^{+3}$	$1.552 \times 10^{+3}$	$1.463 \times 10^{+3}$	**$1.329 \times 10^{+3}$**
	Std	$6.082 \times 10^{+2}$	$3.256 \times 10^{+2}$	$1.370 \times 10^{+2}$	$1.743 \times 10^{+2}$	$1.982 \times 10^{+2}$	$1.404 \times 10^{+2}$	$1.924 \times 10^{+2}$	$2.896 \times 10^{+2}$	$1.029 \times 10^{+2}$	$3.937 \times 10^{+2}$	$1.103 \times 10^{+2}$	**$8.110 \times 10^{+1}$**
	Rank	12	11	8	9	5	6	7	10	4	3	2	1
F8	Best	$2.434 \times 10^{+3}$	$2.194 \times 10^{+3}$	$1.813 \times 10^{+3}$	$2.190 \times 10^{+3}$	$1.702 \times 10^{+3}$	$1.759 \times 10^{+3}$	$1.751 \times 10^{+3}$	$2.364 \times 10^{+3}$	$1.817 \times 10^{+3}$	**$1.031 \times 10^{+3}$**	$1.065 \times 10^{+3}$	$1.070 \times 10^{+3}$
	Mean	$2.662 \times 10^{+3}$	$2.261 \times 10^{+3}$	$2.107 \times 10^{+3}$	$2.302 \times 10^{+3}$	$2.084 \times 10^{+3}$	$1.906 \times 10^{+3}$	$1.978 \times 10^{+3}$	$2.531 \times 10^{+3}$	$1.969 \times 10^{+3}$	$1.348 \times 10^{+3}$	$1.172 \times 10^{+3}$	**$1.165 \times 10^{+3}$**
	Std	$1.301 \times 10^{+2}$	**$3.831 \times 10^{+1}$**	$1.215 \times 10^{+2}$	$5.895 \times 10^{+1}$	$2.015 \times 10^{+2}$	$1.091 \times 10^{+2}$	$1.062 \times 10^{+2}$	$8.200 \times 10^{+1}$	$6.313 \times 10^{+1}$	$1.940 \times 10^{+2}$	$5.904 \times 10^{+1}$	$7.013 \times 10^{+1}$
	Rank	12	9	8	10	7	4	6	11	5	3	2	1
F9	Best	$8.887 \times 10^{+4}$	$9.422 \times 10^{+4}$	$3.211 \times 10^{+4}$	$6.284 \times 10^{+4}$	$1.942 \times 10^{+4}$	$3.304 \times 10^{+4}$	$3.061 \times 10^{+4}$	$7.801 \times 10^{+4}$	$6.388 \times 10^{+4}$	**$1.739 \times 10^{+3}$**	$4.162 \times 10^{+3}$	$4.182 \times 10^{+3}$
	Mean	$1.324 \times 10^{+5}$	$1.172 \times 10^{+5}$	$5.683 \times 10^{+4}$	$7.882 \times 10^{+4}$	$5.481 \times 10^{+4}$	$5.037 \times 10^{+4}$	$5.207 \times 10^{+4}$	$8.978 \times 10^{+4}$	$9.962 \times 10^{+4}$	$1.920 \times 10^{+4}$	$1.088 \times 10^{+4}$	**$1.057 \times 10^{+4}$**
	Std	$2.245 \times 10^{+4}$	$1.047 \times 10^{+4}$	$1.561 \times 10^{+4}$	$8.514 \times 10^{+3}$	$2.070 \times 10^{+4}$	$1.197 \times 10^{+4}$	$6.328 \times 10^{+3}$	$5.486 \times 10^{+3}$	$1.484 \times 10^{+4}$	$3.488 \times 10^{+4}$	**$4.947 \times 10^{+3}$**	$7.705 \times 10^{+3}$
	Rank	12	11	7	8	6	5	4	9	10	3	2	1

Table A7. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F10	Best	$3.111 \times 10^{+4}$	$2.723 \times 10^{+4}$	$2.079 \times 10^{+4}$	$3.053 \times 10^{+4}$	$1.524 \times 10^{+4}$	$1.603 \times 10^{+4}$	$1.967 \times 10^{+4}$	$2.902 \times 10^{+4}$	$2.202 \times 10^{+4}$	$1.706 \times 10^{+4}$	$1.279 \times 10^{+4}$	**$1.490 \times 10^{+4}$**
	Mean	$3.245 \times 10^{+4}$	$2.883 \times 10^{+4}$	$2.566 \times 10^{+4}$	$3.228 \times 10^{+4}$	**$1.909 \times 10^{+4}$**	$2.221 \times 10^{+4}$	$2.429 \times 10^{+4}$	$3.110 \times 10^{+4}$	$2.325 \times 10^{+4}$	$2.488 \times 10^{+4}$	$2.158 \times 10^{+4}$	$2.368 \times 10^{+4}$
	Std	$6.886 \times 10^{+2}$	$5.934 \times 10^{+2}$	$2.391 \times 10^{+3}$	**$5.764 \times 10^{+2}$**	$3.330 \times 10^{+3}$	$4.846 \times 10^{+3}$	$1.694 \times 10^{+3}$	$9.238 \times 10^{+2}$	$6.647 \times 10^{+2}$	$4.900 \times 10^{+3}$	$4.704 \times 10^{+3}$	$5.123 \times 10^{+3}$
	Rank	12	9	8	11	1	3	6	10	4	7	2	5
F11	Best	$1.569 \times 10^{+5}$	$8.690 \times 10^{+4}$	$4.620 \times 10^{+4}$	$7.885 \times 10^{+4}$	$1.498 \times 10^{+4}$	$6.178 \times 10^{+4}$	$6.716 \times 10^{+4}$	$1.346 \times 10^{+5}$	$1.257 \times 10^{+5}$	$1.968 \times 10^{+4}$	$2.637 \times 10^{+3}$	**$2.022 \times 10^{+3}$**
	Mean	$2.693 \times 10^{+5}$	$1.257 \times 10^{+5}$	$1.366 \times 10^{+5}$	$1.112 \times 10^{+5}$	$9.733 \times 10^{+4}$	$8.901 \times 10^{+4}$	$1.063 \times 10^{+5}$	$1.879 \times 10^{+5}$	$1.920 \times 10^{+5}$	$1.339 \times 10^{+5}$	$5.012 \times 10^{+3}$	**$3.168 \times 10^{+3}$**
	Std	$7.375 \times 10^{+4}$	$2.126 \times 10^{+4}$	$7.866 \times 10^{+4}$	$1.570 \times 10^{+4}$	$4.310 \times 10^{+4}$	$1.754 \times 10^{+4}$	$1.814 \times 10^{+4}$	$3.708 \times 10^{+4}$	$3.491 \times 10^{+4}$	$8.664 \times 10^{+4}$	$1.610 \times 10^{+3}$	**$7.819 \times 10^{+2}$**
	Rank	12	9	7	6	4	3	5	10	11	8	2	1
F12	Best	$6.739 \times 10^{+10}$	$1.546 \times 10^{+10}$	$1.090 \times 10^{+9}$	$5.625 \times 10^{+10}$	$1.341 \times 10^{+8}$	$2.612 \times 10^{+10}$	$1.094 \times 10^{+10}$	$9.473 \times 10^{+10}$	$3.859 \times 10^{+9}$	$2.608 \times 10^{+7}$	$3.807 \times 10^{+6}$	**$2.370 \times 10^{+6}$**
	Mean	$1.048 \times 10^{+11}$	$1.959 \times 10^{+10}$	$2.443 \times 10^{+9}$	$7.487 \times 10^{+10}$	$1.016 \times 10^{+9}$	$4.106 \times 10^{+10}$	$2.657 \times 10^{+10}$	$1.408 \times 10^{+11}$	$7.901 \times 10^{+9}$	$1.881 \times 10^{+9}$	$1.025 \times 10^{+7}$	**$5.224 \times 10^{+6}$**
	Std	$2.118 \times 10^{+10}$	$2.091 \times 10^{+9}$	$8.226 \times 10^{+8}$	$9.049 \times 10^{+9}$	$6.913 \times 10^{+8}$	$1.061 \times 10^{+10}$	$1.133 \times 10^{+10}$	$2.558 \times 10^{+10}$	$1.961 \times 10^{+9}$	$3.369 \times 10^{+9}$	$4.344 \times 10^{+6}$	**$2.133 \times 10^{+6}$**
	Rank	11	7	5	10	3	9	8	12	6	4	2	1
F13	Best	$9.170 \times 10^{+9}$	$6.272 \times 10^{+7}$	$7.349 \times 10^{+6}$	$7.447 \times 10^{+9}$	$2.127 \times 10^{+5}$	$2.057 \times 10^{+9}$	$1.158 \times 10^{+9}$	$1.720 \times 10^{+10}$	$3.551 \times 10^{+7}$	**$1.990 \times 10^{+3}$**	$2.733 \times 10^{+3}$	$2.111 \times 10^{+3}$
	Mean	$1.790 \times 10^{+10}$	$2.019 \times 10^{+8}$	$1.785 \times 10^{+7}$	$1.206 \times 10^{+10}$	$5.333 \times 10^{+7}$	$7.992 \times 10^{+9}$	$4.483 \times 10^{+9}$	$2.912 \times 10^{+10}$	$1.317 \times 10^{+8}$	$8.353 \times 10^{+5}$	$7.615 \times 10^{+3}$	**$6.864 \times 10^{+3}$**
	Std	$4.454 \times 10^{+9}$	$1.113 \times 10^{+8}$	$1.394 \times 10^{+7}$	$2.245 \times 10^{+9}$	$7.448 \times 10^{+7}$	$3.223 \times 10^{+9}$	$3.108 \times 10^{+9}$	$7.203 \times 10^{+9}$	$5.234 \times 10^{+7}$	$2.076 \times 10^{+6}$	**$4.158 \times 10^{+3}$**	$5.547 \times 10^{+3}$
	Rank	11	7	4	10	5	9	8	12	6	3	2	1
F14	Best	$2.108 \times 10^{+7}$	$1.235 \times 10^{+7}$	$1.968 \times 10^{+6}$	$1.111 \times 10^{+7}$	$1.415 \times 10^{+6}$	$3.063 \times 10^{+6}$	$3.354 \times 10^{+6}$	$1.140 \times 10^{+7}$	$8.100 \times 10^{+6}$	$8.695 \times 10^{+5}$	$2.157 \times 10^{+5}$	**$7.456 \times 10^{+4}$**
	Mean	$5.065 \times 10^{+7}$	$3.377 \times 10^{+7}$	$7.043 \times 10^{+6}$	$3.650 \times 10^{+7}$	$1.001 \times 10^{+7}$	$1.541 \times 10^{+7}$	$7.082 \times 10^{+6}$	$3.009 \times 10^{+7}$	$2.210 \times 10^{+7}$	$9.380 \times 10^{+6}$	$5.424 \times 10^{+5}$	**$3.600 \times 10^{+5}$**
	Std	$2.268 \times 10^{+7}$	$1.095 \times 10^{+7}$	$2.608 \times 10^{+6}$	$1.355 \times 10^{+7}$	$8.367 \times 10^{+6}$	$1.025 \times 10^{+7}$	$3.159 \times 10^{+6}$	$1.265 \times 10^{+7}$	$7.823 \times 10^{+6}$	$1.025 \times 10^{+7}$	$2.571 \times 10^{+5}$	**$2.156 \times 10^{+5}$**
	Rank	12	10	3	11	6	7	4	9	8	5	2	1
F15	Best	$4.412 \times 10^{+9}$	$6.580 \times 10^{+5}$	$5.846 \times 10^{+5}$	$2.283 \times 10^{+9}$	$2.873 \times 10^{+4}$	$1.103 \times 10^{+8}$	$7.354 \times 10^{+7}$	$4.384 \times 10^{+9}$	$4.023 \times 10^{+6}$	$2.041 \times 10^{+3}$	**$1.775 \times 10^{+3}$**	$1.812 \times 10^{+3}$
	Mean	$6.456 \times 10^{+9}$	$1.378 \times 10^{+7}$	$3.192 \times 10^{+6}$	$3.915 \times 10^{+9}$	$8.200 \times 10^{+6}$	$2.739 \times 10^{+9}$	$1.014 \times 10^{+9}$	$1.015 \times 10^{+10}$	$1.464 \times 10^{+7}$	$5.832 \times 10^{+5}$	$4.253 \times 10^{+3}$	**$4.065 \times 10^{+3}$**
	Std	$1.342 \times 10^{+9}$	$5.957 \times 10^{+6}$	$3.867 \times 10^{+6}$	$8.689 \times 10^{+8}$	$2.169 \times 10^{+7}$	$2.412 \times 10^{+9}$	$1.357 \times 10^{+9}$	$3.085 \times 10^{+9}$	$9.634 \times 10^{+6}$	$1.356 \times 10^{+6}$	**$2.459 \times 10^{+3}$**	$2.983 \times 10^{+3}$
	Rank	11	6	5	10	4	9	8	12	7	3	2	1
F16	Best	$1.204 \times 10^{+4}$	$1.041 \times 10^{+4}$	$1.063 \times 10^{+4}$	$1.241 \times 10^{+4}$	$6.299 \times 10^{+3}$	$6.759 \times 10^{+3}$	$7.812 \times 10^{+3}$	$1.371 \times 10^{+4}$	$8.073 \times 10^{+3}$	$5.633 \times 10^{+3}$	$3.435 \times 10^{+3}$	**$3.045 \times 10^{+3}$**
	Mean	$1.340 \times 10^{+4}$	$1.102 \times 10^{+4}$	$1.329 \times 10^{+4}$	$1.360 \times 10^{+4}$	$8.462 \times 10^{+3}$	$8.994 \times 10^{+3}$	$9.284 \times 10^{+3}$	$1.733 \times 10^{+4}$	$9.319 \times 10^{+3}$	$8.882 \times 10^{+3}$	$4.982 \times 10^{+3}$	**$4.840 \times 10^{+3}$**
	Std	$1.086 \times 10^{+3}$	**$3.516 \times 10^{+2}$**	$1.845 \times 10^{+3}$	$6.937 \times 10^{+2}$	$1.132 \times 10^{+3}$	$1.156 \times 10^{+3}$	$7.255 \times 10^{+2}$	$1.782 \times 10^{+3}$	$5.370 \times 10^{+2}$	$1.695 \times 10^{+3}$	$7.740 \times 10^{+2}$	$6.268 \times 10^{+2}$
	Rank	10	8	9	11	3	4	6	12	7	5	2	1
F17	Best	$1.498 \times 10^{+4}$	$7.482 \times 10^{+3}$	$6.596 \times 10^{+3}$	$1.028 \times 10^{+4}$	$5.314 \times 10^{+3}$	$5.528 \times 10^{+3}$	$5.856 \times 10^{+3}$	$1.498 \times 10^{+4}$	$6.670 \times 10^{+3}$	$5.056 \times 10^{+3}$	$3.284 \times 10^{+3}$	**$2.838 \times 10^{+3}$**
	Mean	$9.651 \times 10^{+4}$	$8.254 \times 10^{+3}$	$7.961 \times 10^{+3}$	$1.576 \times 10^{+4}$	$7.956 \times 10^{+3}$	$1.566 \times 10^{+4}$	$8.002 \times 10^{+3}$	$2.153 \times 10^{+5}$	$8.914 \times 10^{+3}$	$6.978 \times 10^{+3}$	$4.315 \times 10^{+3}$	**$4.202 \times 10^{+3}$**
	Std	$1.779 \times 10^{+5}$	**$3.430 \times 10^{+2}$**	$1.058 \times 10^{+3}$	$6.001 \times 10^{+3}$	$1.203 \times 10^{+3}$	$1.712 \times 10^{+4}$	$3.859 \times 10^{+3}$	$2.127 \times 10^{+5}$	$9.067 \times 10^{+2}$	$9.836 \times 10^{+2}$	$5.508 \times 10^{+2}$	$5.345 \times 10^{+2}$
	Rank	11	7	5	10	6	8	4	12	9	3	2	1
F18	Best	$2.117 \times 10^{+7}$	$2.792 \times 10^{+7}$	$1.247 \times 10^{+6}$	$2.300 \times 10^{+7}$	$5.986 \times 10^{+5}$	$3.780 \times 10^{+6}$	$2.875 \times 10^{+6}$	$1.982 \times 10^{+7}$	$9.699 \times 10^{+6}$	$8.468 \times 10^{+5}$	**$3.747 \times 10^{+5}$**	$4.619 \times 10^{+5}$
	Mean	$7.408 \times 10^{+7}$	$5.203 \times 10^{+7}$	$5.736 \times 10^{+6}$	$6.618 \times 10^{+7}$	$9.849 \times 10^{+6}$	$1.567 \times 10^{+7}$	$7.499 \times 10^{+6}$	$5.292 \times 10^{+7}$	$2.459 \times 10^{+7}$	$1.693 \times 10^{+7}$	$1.201 \times 10^{+6}$	**$1.129 \times 10^{+6}$**
	Std	$2.945 \times 10^{+7}$	$1.180 \times 10^{+7}$	$2.477 \times 10^{+6}$	$2.698 \times 10^{+7}$	$6.535 \times 10^{+6}$	$9.196 \times 10^{+6}$	$4.528 \times 10^{+6}$	$2.067 \times 10^{+7}$	$1.065 \times 10^{+7}$	$1.910 \times 10^{+7}$	$6.295 \times 10^{+5}$	**$5.087 \times 10^{+5}$**
	Rank	12	10	3	11	5	7	4	9	8	6	2	1
F19	Best	$3.513 \times 10^{+9}$	$2.063 \times 10^{+7}$	$5.034 \times 10^{+6}$	$2.057 \times 10^{+9}$	$1.162 \times 10^{+6}$	$1.009 \times 10^{+8}$	$8.865 \times 10^{+7}$	$4.856 \times 10^{+9}$	$6.781 \times 10^{+6}$	$2.082 \times 10^{+3}$	**$2.015 \times 10^{+3}$**	$2.139 \times 10^{+3}$
	Mean	$6.512 \times 10^{+9}$	$4.019 \times 10^{+7}$	$2.956 \times 10^{+7}$	$3.366 \times 10^{+9}$	$1.382 \times 10^{+7}$	$1.984 \times 10^{+9}$	$1.265 \times 10^{+9}$	$1.090 \times 10^{+10}$	$3.131 \times 10^{+7}$	$4.010 \times 10^{+5}$	**$4.560 \times 10^{+3}$**	$4.858 \times 10^{+3}$
	Std	$1.665 \times 10^{+9}$	$1.044 \times 10^{+7}$	$2.142 \times 10^{+7}$	$7.925 \times 10^{+8}$	$1.063 \times 10^{+7}$	$1.721 \times 10^{+9}$	$1.838 \times 10^{+9}$	$3.383 \times 10^{+9}$	$1.568 \times 10^{+7}$	$8.465 \times 10^{+5}$	**$2.954 \times 10^{+3}$**	$3.168 \times 10^{+3}$
	Rank	11	7	5	10	4	9	8	12	6	3	2	1
F20	Best	$7.208 \times 10^{+3}$	$5.742 \times 10^{+3}$	$4.846 \times 10^{+3}$	$6.962 \times 10^{+3}$	$4.639 \times 10^{+3}$	$4.390 \times 10^{+3}$	$4.345 \times 10^{+3}$	$6.322 \times 10^{+3}$	$4.877 \times 10^{+3}$	$4.913 \times 10^{+3}$	$3.833 \times 10^{+3}$	**$3.736 \times 10^{+3}$**
	Mean	$7.965 \times 10^{+3}$	$6.386 \times 10^{+3}$	$6.527 \times 10^{+3}$	$7.540 \times 10^{+3}$	$6.687 \times 10^{+3}$	$5.812 \times 10^{+3}$	$5.799 \times 10^{+3}$	$7.238 \times 10^{+3}$	$5.517 \times 10^{+3}$	$6.321 \times 10^{+3}$	**$5.123 \times 10^{+3}$**	$5.198 \times 10^{+3}$
	Std	$2.862 \times 10^{+2}$	**$2.683 \times 10^{+2}$**	$8.361 \times 10^{+2}$	$3.244 \times 10^{+2}$	$8.891 \times 10^{+2}$	$8.056 \times 10^{+2}$	$4.596 \times 10^{+2}$	$4.457 \times 10^{+2}$	$2.922 \times 10^{+2}$	$1.010 \times 10^{+3}$	$7.933 \times 10^{+2}$	$7.999 \times 10^{+2}$
	Rank	12	7	8	11	9	4	5	10	3	6	1	2
F21	Best	$4.103 \times 10^{+3}$	$3.733 \times 10^{+3}$	$3.658 \times 10^{+3}$	$3.794 \times 10^{+3}$	$3.195 \times 10^{+3}$	$3.284 \times 10^{+3}$	$3.298 \times 10^{+3}$	$4.191 \times 10^{+3}$	$3.376 \times 10^{+3}$	**$2.518 \times 10^{+3}$**	$2.602 \times 10^{+3}$	$2.582 \times 10^{+3}$
	Mean	$4.402 \times 10^{+3}$	$3.836 \times 10^{+3}$	$4.137 \times 10^{+3}$	$4.035 \times 10^{+3}$	$3.848 \times 10^{+3}$	$3.452 \times 10^{+3}$	$3.520 \times 10^{+3}$	$4.540 \times 10^{+3}$	$3.561 \times 10^{+3}$	$2.922 \times 10^{+3}$	**$2.693 \times 10^{+3}$**	$2.694 \times 10^{+3}$
	Std	$1.443 \times 10^{+2}$	$4.995 \times 10^{+1}$	$2.719 \times 10^{+2}$	$9.970 \times 10^{+1}$	$2.044 \times 10^{+2}$	$9.819 \times 10^{+1}$	$8.243 \times 10^{+1}$	$1.771 \times 10^{+2}$	$6.812 \times 10^{+1}$	$2.703 \times 10^{+2}$	**$4.962 \times 10^{+1}$**	$7.268 \times 10^{+1}$
	Rank	11	8	10	9	7	4	5	12	6	3	2	1

Table A7. *Cont.*

Func-tion	Index	PSO	DE	WOA	SCA	DBO	GJO	GGO	HBO	CL-PSO	LSHADE	FNO	EFNO
F22	Best	$3.267 \times 10^{+4}$	$2.882 \times 10^{+4}$	$2.356 \times 10^{+4}$	$3.229 \times 10^{+4}$	$1.975 \times 10^{+4}$	$2.143 \times 10^{+4}$	$1.663 \times 10^{+4}$	$3.192 \times 10^{+4}$	$2.148 \times 10^{+4}$	$1.873 \times 10^{+4}$	**$1.395 \times 10^{+4}$**	$1.447 \times 10^{+4}$
	Mean	$3.397 \times 10^{+4}$	$3.016 \times 10^{+4}$	$2.770 \times 10^{+4}$	$3.448 \times 10^{+4}$	$2.406 \times 10^{+4}$	$2.533 \times 10^{+4}$	$2.636 \times 10^{+4}$	$3.414 \times 10^{+4}$	**$2.385 \times 10^{+4}$**	$2.774 \times 10^{+4}$	$2.410 \times 10^{+4}$	$2.505 \times 10^{+4}$
	Std	$6.047 \times 10^{+2}$	**$5.906 \times 10^{+2}$**	$2.291 \times 10^{+3}$	$6.870 \times 10^{+2}$	$3.661 \times 10^{+3}$	$3.540 \times 10^{+3}$	$2.682 \times 10^{+3}$	$1.116 \times 10^{+3}$	$7.672 \times 10^{+2}$	$5.669 \times 10^{+3}$	$5.954 \times 10^{+3}$	$6.148 \times 10^{+3}$
	Rank	10	9	7	12	1	5	6	11	2	8	3	4
F23	Best	$4.886 \times 10^{+3}$	$3.829 \times 10^{+3}$	$4.545 \times 10^{+3}$	$4.633 \times 10^{+3}$	$3.980 \times 10^{+3}$	$3.884 \times 10^{+3}$	$3.966 \times 10^{+3}$	$5.245 \times 10^{+3}$	$3.898 \times 10^{+3}$	**$2.995 \times 10^{+3}$**	$3.169 \times 10^{+3}$	$3.175 \times 10^{+3}$
	Mean	$5.284 \times 10^{+3}$	$3.882 \times 10^{+3}$	$4.999 \times 10^{+3}$	$4.975 \times 10^{+3}$	$4.275 \times 10^{+3}$	$4.233 \times 10^{+3}$	$4.240 \times 10^{+3}$	$5.650 \times 10^{+3}$	$4.023 \times 10^{+3}$	$3.325 \times 10^{+3}$	$3.282 \times 10^{+3}$	**$3.277 \times 10^{+3}$**
	Std	$2.282 \times 10^{+2}$	**$3.452 \times 10^{+1}$**	$2.407 \times 10^{+2}$	$1.167 \times 10^{+2}$	$1.718 \times 10^{+2}$	$1.493 \times 10^{+2}$	$1.436 \times 10^{+2}$	$2.167 \times 10^{+2}$	$6.759 \times 10^{+1}$	$2.591 \times 10^{+2}$	$6.726 \times 10^{+1}$	$6.710 \times 10^{+1}$
	Rank	11	4	10	9	8	6	7	12	5	3	2	1
F24	Best	$5.436 \times 10^{+3}$	$4.357 \times 10^{+3}$	$5.600 \times 10^{+3}$	$6.559 \times 10^{+3}$	$4.524 \times 10^{+3}$	$4.848 \times 10^{+3}$	$4.732 \times 10^{+3}$	$7.267 \times 10^{+3}$	$4.624 \times 10^{+3}$	**$3.464 \times 10^{+3}$**	$3.641 \times 10^{+3}$	$3.631 \times 10^{+3}$
	Mean	$6.450 \times 10^{+3}$	$4.467 \times 10^{+3}$	$6.368 \times 10^{+3}$	$6.903 \times 10^{+3}$	$5.074 \times 10^{+3}$	$5.488 \times 10^{+3}$	$5.310 \times 10^{+3}$	$8.234 \times 10^{+3}$	$4.889 \times 10^{+3}$	$3.873 \times 10^{+3}$	$3.835 \times 10^{+3}$	**$3.798 \times 10^{+3}$**
	Std	$5.705 \times 10^{+2}$	**$4.859 \times 10^{+1}$**	$4.892 \times 10^{+2}$	$2.276 \times 10^{+2}$	$2.840 \times 10^{+2}$	$2.289 \times 10^{+2}$	$3.243 \times 10^{+2}$	$4.840 \times 10^{+2}$	$1.162 \times 10^{+2}$	$2.433 \times 10^{+2}$	$9.643 \times 10^{+1}$	$9.402 \times 10^{+1}$
	Rank	9	4	10	11	6	8	7	12	5	3	2	1
F25	Best	$5.407 \times 10^{+4}$	$2.682 \times 10^{+4}$	$4.275 \times 10^{+3}$	$1.491 \times 10^{+4}$	$3.379 \times 10^{+3}$	$9.102 \times 10^{+3}$	$8.854 \times 10^{+3}$	$2.012 \times 10^{+4}$	$6.036 \times 10^{+3}$	$3.232 \times 10^{+3}$	**$3.216 \times 10^{+3}$**	$3.217 \times 10^{+3}$
	Mean	$6.842 \times 10^{+4}$	$3.452 \times 10^{+4}$	$4.654 \times 10^{+3}$	$1.807 \times 10^{+4}$	$3.708 \times 10^{+3}$	$1.174 \times 10^{+4}$	$1.165 \times 10^{+4}$	$2.451 \times 10^{+4}$	$7.220 \times 10^{+3}$	$4.398 \times 10^{+3}$	$3.369 \times 10^{+3}$	**$3.320 \times 10^{+3}$**
	Std	$9.461 \times 10^{+3}$	$2.556 \times 10^{+3}$	$2.376 \times 10^{+2}$	$2.003 \times 10^{+3}$	$2.131 \times 10^{+2}$	$1.633 \times 10^{+3}$	$2.088 \times 10^{+3}$	$2.728 \times 10^{+3}$	$6.563 \times 10^{+2}$	$2.808 \times 10^{+3}$	$6.828 \times 10^{+1}$	**$5.026 \times 10^{+1}$**
	Rank	12	11	5	9	4	7	8	10	6	3	2	1
F26	Best	$2.490 \times 10^{+4}$	$1.780 \times 10^{+4}$	$2.736 \times 10^{+4}$	$3.319 \times 10^{+4}$	$1.553 \times 10^{+4}$	$2.357 \times 10^{+4}$	$1.747 \times 10^{+4}$	$4.403 \times 10^{+4}$	$1.844 \times 10^{+4}$	**$8.533 \times 10^{+3}$**	$1.104 \times 10^{+4}$	$9.885 \times 10^{+3}$
	Mean	$3.448 \times 10^{+4}$	$1.839 \times 10^{+4}$	$3.346 \times 10^{+4}$	$3.642 \times 10^{+4}$	$2.111 \times 10^{+4}$	$2.768 \times 10^{+4}$	$2.315 \times 10^{+4}$	$4.965 \times 10^{+4}$	$1.937 \times 10^{+4}$	$1.300 \times 10^{+4}$	$1.341 \times 10^{+4}$	**$1.290 \times 10^{+4}$**
	Std	$3.347 \times 10^{+3}$	**$3.001 \times 10^{+2}$**	$3.110 \times 10^{+3}$	$2.020 \times 10^{+3}$	$3.495 \times 10^{+3}$	$1.906 \times 10^{+3}$	$3.595 \times 10^{+3}$	$3.967 \times 10^{+3}$	$5.664 \times 10^{+2}$	$2.834 \times 10^{+3}$	$1.758 \times 10^{+3}$	$1.506 \times 10^{+3}$
	Rank	10	4	9	11	6	8	7	12	5	2	3	1
F27	Best	$4.321 \times 10^{+3}$	$3.788 \times 10^{+3}$	$4.640 \times 10^{+3}$	$6.548 \times 10^{+3}$	$3.675 \times 10^{+3}$	$4.652 \times 10^{+3}$	$4.486 \times 10^{+3}$	$6.409 \times 10^{+3}$	$4.313 \times 10^{+3}$	**$3.200 \times 10^{+3}$**	$3.527 \times 10^{+3}$	$3.583 \times 10^{+3}$
	Mean	$5.162 \times 10^{+3}$	$3.938 \times 10^{+3}$	$5.597 \times 10^{+3}$	$7.701 \times 10^{+3}$	$4.085 \times 10^{+3}$	$5.362 \times 10^{+3}$	$5.180 \times 10^{+3}$	$9.290 \times 10^{+3}$	$4.815 \times 10^{+3}$	**$3.200 \times 10^{+3}$**	$3.773 \times 10^{+3}$	$3.727 \times 10^{+3}$
	Std	$5.144 \times 10^{+2}$	$6.840 \times 10^{+1}$	$7.429 \times 10^{+2}$	$5.071 \times 10^{+2}$	$2.635 \times 10^{+2}$	$3.580 \times 10^{+2}$	$3.989 \times 10^{+2}$	$1.060 \times 10^{+3}$	$2.373 \times 10^{+2}$	**4.892×10^{-4}**	$9.301 \times 10^{+1}$	$8.335 \times 10^{+1}$
	Rank	8	4	10	11	5	9	7	12	6	1	3	2
F28	Best	$2.153 \times 10^{+4}$	$1.577 \times 10^{+4}$	$5.016 \times 10^{+3}$	$1.819 \times 10^{+4}$	$3.696 \times 10^{+3}$	$1.274 \times 10^{+4}$	$9.936 \times 10^{+3}$	$2.634 \times 10^{+4}$	$1.514 \times 10^{+4}$	**$3.300 \times 10^{+3}$**	$3.400 \times 10^{+3}$	$3.390 \times 10^{+3}$
	Mean	$2.712 \times 10^{+4}$	$1.596 \times 10^{+4}$	$5.775 \times 10^{+3}$	$2.179 \times 10^{+4}$	$1.174 \times 10^{+4}$	$1.587 \times 10^{+4}$	$1.251 \times 10^{+4}$	$2.958 \times 10^{+4}$	$1.963 \times 10^{+4}$	**$3.300 \times 10^{+3}$**	$3.500 \times 10^{+3}$	$3.455 \times 10^{+3}$
	Std	$3.549 \times 10^{+3}$	$1.246 \times 10^{+2}$	$4.934 \times 10^{+2}$	$2.212 \times 10^{+3}$	$6.889 \times 10^{+3}$	$1.616 \times 10^{+3}$	$1.543 \times 10^{+3}$	$1.694 \times 10^{+3}$	$1.624 \times 10^{+3}$	**5.945×10^{-4}**	$4.358 \times 10^{+1}$	$4.037 \times 10^{+1}$
	Rank	11	8	4	10	6	7	5	12	9	1	3	2
F29	Best	$1.887 \times 10^{+4}$	$7.871 \times 10^{+3}$	$1.126 \times 10^{+4}$	$1.485 \times 10^{+4}$	$7.417 \times 10^{+3}$	$1.048 \times 10^{+4}$	$1.060 \times 10^{+4}$	$2.023 \times 10^{+4}$	$8.483 \times 10^{+3}$	$5.950 \times 10^{+3}$	**$5.314 \times 10^{+3}$**	$5.563 \times 10^{+3}$
	Mean	$6.184 \times 10^{+4}$	$9.232 \times 10^{+3}$	$1.559 \times 10^{+4}$	$2.100 \times 10^{+4}$	$1.024 \times 10^{+4}$	$1.421 \times 10^{+4}$	$1.453 \times 10^{+4}$	$4.560 \times 10^{+4}$	$1.040 \times 10^{+4}$	$7.699 \times 10^{+3}$	$6.680 \times 10^{+3}$	**$6.446 \times 10^{+3}$**
	Std	$8.268 \times 10^{+4}$	**$5.024 \times 10^{+2}$**	$2.146 \times 10^{+3}$	$3.156 \times 10^{+3}$	$1.656 \times 10^{+3}$	$3.458 \times 10^{+3}$	$5.139 \times 10^{+3}$	$1.879 \times 10^{+4}$	$1.047 \times 10^{+3}$	$1.227 \times 10^{+3}$	$6.355 \times 10^{+2}$	$5.867 \times 10^{+2}$
	Rank	11	4	9	10	5	8	7	12	6	3	2	1
F30	Best	$6.277 \times 10^{+9}$	$9.609 \times 10^{+5}$	$3.135 \times 10^{+8}$	$5.832 \times 10^{+9}$	$4.102 \times 10^{+6}$	$1.233 \times 10^{+9}$	$5.853 \times 10^{+8}$	$8.358 \times 10^{+9}$	$6.196 \times 10^{+7}$	**$3.342 \times 10^{+3}$**	$1.141 \times 10^{+4}$	$9.614 \times 10^{+3}$
	Mean	$1.034 \times 10^{+10}$	$1.147 \times 10^{+7}$	$6.402 \times 10^{+8}$	$8.532 \times 10^{+9}$	$3.186 \times 10^{+7}$	$6.151 \times 10^{+9}$	$2.462 \times 10^{+9}$	$2.120 \times 10^{+10}$	$1.517 \times 10^{+8}$	$1.356 \times 10^{+6}$	$5.232 \times 10^{+4}$	**$5.010 \times 10^{+4}$**
	Std	$2.288 \times 10^{+9}$	$3.010 \times 10^{+7}$	$2.417 \times 10^{+8}$	$1.420 \times 10^{+9}$	$3.661 \times 10^{+7}$	$2.952 \times 10^{+9}$	$2.199 \times 10^{+9}$	$5.862 \times 10^{+9}$	$5.123 \times 10^{+7}$	$3.934 \times 10^{+6}$	**$4.429 \times 10^{+4}$**	$5.156 \times 10^{+4}$
	Rank	11	4	7	10	5	9	8	12	6	2	3	1
Mean Rank		11.17	7.52	6.97	9.83	4.93	6.48	6.24	11	6.34	3.76	2.24	1.52
+/=/−		29/0/0	29/0/0	28/1/0	28/1/0	26/2/1	26/2/1	26/2/1	29/0/0	27/2/0	19/7/3	18/10/1	~

References

1. Tomar, V.; Bansal, M.; Singh, P. Metaheuristic algorithms for optimization: A brief review. *Eng. Proc.* **2023**, *59*, 238. [CrossRef]
2. Khanduja, N.; Bhushan, B. Recent advances and application of metaheuristic algorithms: A survey (2014–2020). In *Metaheuristic and Evolutionary Computation: Algorithms and Applications*; Springer: Singapore, 2020; pp. 207–228. [CrossRef]
3. Rajwar, K.; Deep, K.; Das, S. An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. *Artif. Intell. Rev.* **2023**, *56*, 13187–13257. [CrossRef] [PubMed]
4. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995*; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948. [CrossRef]
5. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
6. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
7. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
8. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
9. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
10. Tang, D. Spherical evolution for solving continuous optimization problems. *Appl. Soft Comput.* **2019**, *81*, 105499. [CrossRef]
11. Zhao, S.; Zhou, H.; Zhou, H. Leaders-driven particle swarm optimizer. *Expert Syst. Appl.* **2025**, *292*, 128595. [CrossRef]
12. Zhu, L.; Zhou, Y.; Zhou, G.; Luo, Q.; Wei, Y. Solving spherical multi-aircraft path planning problem using self-adaptive update strategy differential evolution algorithm. *Swarm Evol. Comput.* **2025**, *96*, 102004. [CrossRef]
13. Ge, Y.; Zhang, C.; Liu, Q.; Zhang, X.; Chen, J.; Nazir, M.S.; Peng, T. A novel metaheuristic optimizer based on improved adaptive guided differential evolution algorithm for parameter identification of a PEMFC model. *Fuel* **2025**, *383*, 133869. [CrossRef]
14. Nithya, M.; Sundararaj, V. Dynamic lane-changing trajectory planning for autonomous vehicles: A novel approach using Gravitational Search Algorithm with Blend Crossover. *Eng. Appl. Artif. Intell.* **2025**, *160*, 112022. [CrossRef]
15. Shaikh, M.S.; Wang, C.; Xie, S.; Zheng, G.; Dong, X.; Qiu, S.; Ahmad, M.A.; Raj, S. Coverage and connectivity maximization for wireless sensor networks using improved chaotic grey wolf optimization. *Sci. Rep.* **2025**, *15*, 15706. [CrossRef]
16. Chen, Q.; Yao, G.; Yang, L.; Liu, T.; Sun, J.; Cai, S. Research on ship replenishment path planning based on the modified whale optimization algorithm. *Biomimetics* **2025**, *10*, 179. [CrossRef] [PubMed]
17. Agrawal, S.P.; Jangir, P.; Arpita; Pandya, S.B.; Parmar, A.; Hourani, A.O.; Trivedi, B.I.; Khishe, M. Exploring the effectiveness of adaptive randomized sine cosine algorithm in wind integrated scenario based power system optimization with FACTS devices. *Sci. Rep.* **2025**, *15*, 7090. [CrossRef] [PubMed]
18. Gharehchopogh, F.S. Quantum-inspired metaheuristic algorithms: Comprehensive survey and classification. *Artif. Intell. Rev.* **2023**, *56*, 5479–5543. [CrossRef]
19. Li, G.; Zhang, T.; Tsai, C.Y.; Yao, L.; Lu, Y.; Tang, J. Review of the metaheuristic algorithms in applications: Visual analysis based on bibliometrics. *Expert Syst. Appl.* **2024**, *255*, 124857. [CrossRef]
20. Shaikh, M.S.; Lin, H.; Xie, S.; Dong, X.; Lin, Y.; Shiva, C.K.; Mbasso, W.F. An intelligent hybrid grey wolf-particle swarm optimizer for optimization in complex engineering design problem. *Sci. Rep.* **2025**, *15*, 18313. [CrossRef]
21. Tang, C.; Sun, W.; Xue, M.; Zhang, X.; Tang, H.; Wu, W. A hybrid whale optimization algorithm with artificial bee colony. *Soft Comput.* **2022**, *26*, 2075–2097. [CrossRef]
22. Hu, G.; Zheng, Y.; Abualigah, L.; Hussien, A.G. DETDO: An adaptive hybrid dandelion optimizer for engineering optimization. *Adv. Eng. Inform.* **2023**, *57*, 102004. [CrossRef]
23. Hou, Y.; Gao, H.; Wang, Z.; Du, C. Improved grey wolf optimization algorithm and application. *Sensors* **2022**, *22*, 3810. [CrossRef]
24. Mostafa, R.R.; Gaheen, M.A.; Abd ElAziz, M.; Al-Betar, M.A.; Ewees, A.A. An improved gorilla troops optimizer for global optimization problems and feature selection. *Knowl.-Based Syst.* **2023**, *269*, 110462. [CrossRef]
25. Ewees, A.A.; Mostafa, R.R.; Ghoniem, R.M.; Gaheen, M.A. Improved seagull optimization algorithm using Lévy flight and mutation operator for feature selection. *Neural Comput. Appl.* **2022**, *34*, 7437–7472. [CrossRef]
26. Rajammal, R.R.; Mirjalili, S.; Ekambaram, G.; Palanisamy, N. Binary grey wolf optimizer with mutation and adaptive k-nearest neighbour for feature selection in Parkinson's disease diagnosis. *Knowl.-Based Syst.* **2022**, *246*, 108701. [CrossRef]
27. Li, W.; Liang, P.; Sun, B.; Sun, Y.; Huang, Y. Reinforcement learning-based particle swarm optimization with neighborhood differential mutation strategy. *Swarm Evol. Comput.* **2023**, *78*, 101274. [CrossRef]
28. Taheri, A.; RahimiZadeh, K.; Baumbach, J.; Beheshti, A.; Zolotareva, O.; Al-Betar, M.A.; Mirjalili, S.; Gandomi, A.H. Farthest better or nearest worse optimizer: A novel metaheuristic algorithm. *Artif. Intell. Rev.* **2026**, *59*, 79. [CrossRef]
29. Lin, R.; Xu, Z.; Yu, L.; Wei, T. EABC-AS: Elite-driven artificial bee colony algorithm with adaptive population scaling. *Swarm Evol. Comput.* **2025**, *94*, 101893. [CrossRef]
30. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [CrossRef]

31. Chopra, N.; Ansari, M.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [CrossRef]
32. El-Kenawy, E.S.M.; Khodadadi, N.; Mirjalili, S.; Abdelhamid, A.A.; Eid, M.M.; Ibrahim, A. Greylag goose optimization: Nature-inspired optimization algorithm. *Expert Syst. Appl.* **2024**, *238*, 122147. [CrossRef]
33. Ouertani, M.W.; Manita, G.; Korbaa, O. Hannibal Barca optimizer: The power of the pincer movement for global optimization and multilevel image thresholding. *Clust. Comput.* **2025**, *28*, 482. [CrossRef]
34. Shahzad, W.; Khan, F.A.; Siddiqui, A.B. Clustering in mobile ad hoc networks using comprehensive learning particle swarm optimization (CLPSO). In *Proceedings of the International Conference on Future Generation Communication and Net-Working, Jeju Island, Republic of Korea, 10–12 December 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 342–349. [CrossRef]
35. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014*; IEEE: New York, NY, USA, 2014; pp. 1658–1665. [CrossRef]
36. Cheng, S.; Shi, Y.; Qin, Q.; Zhang, Q.; Bai, R. Population diversity maintenance in brain storm optimization algorithm. *J. Artif. Intell. Soft Comput. Res.* **2014**, *4*, 83–97. [CrossRef]
37. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2020**, *56*, 100693. [CrossRef]
38. Biswas, S.; Singh, G.; Maiti, B.; Ezugwu, A.E.-S.; Saleem, K.; Smerat, A.; Abualigah, L.; Bera, U.K. Integrating Differential Evolution into Gazelle Optimization for advanced global optimization and engineering applications. *Comput. Methods Appl. Mech. Eng.* **2025**, *434*, 117588. [CrossRef]
39. Tang, X.; Qu, P.; He, Z.; Jia, C.; Zhang, Q. A Dynamic Multi-Niche Biogeography-Based Optimization Algorithm and Its Application to Robot Path Planning. *Biomimetics* **2026**, *11*, 221. [CrossRef]
40. Hu, W.; Zhang, Q.; Ye, S. An enhanced dung beetle optimizer with multiple strategies for robot path planning. *Sci. Rep.* **2025**, *15*, 4655. [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.