

Article

Symmetry-Enhanced, Improved Pathfinder Algorithm-Based Multi-Strategy Fusion for Engineering Optimization Problems

Xuedi Mao, Bing Wang *, Wenjian Ye and Yuxin Chai

School of Mathematical Sciences, Mudanjiang Normal University, Mudanjiang 157000, China;
1023321531@stu.mdjnu.edu.cn (X.M.); 1023323541@stu.mdjnu.edu.cn (W.Y.);
1023323548@stu.mdjnu.edu.cn (Y.C.)

* Correspondence: wbmsy2024@mdjnu.edu.cn

Abstract: The pathfinder algorithm (PFA) starts with a random search for the initial population, which is then partitioned into only a pathfinder phase and a follower phase. This approach often results in issues like poor solution accuracy, slow convergence, and susceptibility to local optima in the PFA. To address these challenges, a multi-strategy fusion approach is proposed in the symmetry-enhanced, improved pathfinder algorithm-based multi-strategy fusion for engineering optimization problems (IPFA) for function optimization problems. First, the elite opposition-based learning mechanism is incorporated to improve the population diversity and population quality, to enhance the solution accuracy of the algorithm; second, to enhance the convergence speed of the algorithm, the escape energy factor is embedded into the prey-hunting phase of the GWO and replaces the follower phase in the PFA, which increases the diversity of the algorithm and improves the search efficiency of the algorithm; lastly, to solve the problem of easily falling into the local optimum, the optimal individual position is perturbed using the dimension-by-dimension mutation method of t -distribution, which helps the individual to jump out of the local optimum rapidly and advance toward other regions. The IPFA is used for testing on 16 classical benchmark test functions and 29 complex CEC2017 function sets. The final optimization results of PFA and IPFA in pressure vessels are 5984.8222 and 5948.3597, respectively. The final optimization results in tension springs are 0.012719 and 0.012699, respectively, which are comparable with the original algorithm and other algorithms. A comparison between the original algorithm and other algorithms shows that the IPFA algorithm is significantly enhanced in terms of solution accuracy, and the lower engineering cost further verifies the robustness of the IPFA algorithm.



Citation: Mao, X.; Wang, B.; Ye, W.; Chai, Y. Symmetry-Enhanced, Improved Pathfinder Algorithm-Based Multi-Strategy Fusion for Engineering Optimization Problems. *Symmetry* **2024**, *16*, 324. <https://doi.org/10.3390/sym16030324>

Academic Editor: László T. Kóczy

Received: 22 January 2024

Revised: 13 February 2024

Accepted: 16 February 2024

Published: 7 March 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: pathfinder algorithm; grey wolf optimizer; elite opposition-based learning; dimension-by-dimension mutation; engineering optimization problems

1. Introduction

In recent years, the application of intelligent optimization algorithms in the domains of engineering, health, and economics is growing, specifically in problems such as image segmentation [1], intelligent transport system [2], optimal scheduling [3], and path planning [4], among others [5–9]. Intelligent optimization algorithms are largely derived from the behavioral and hunting patterns of organisms in the natural world. Scholars have proposed algorithms such as particle swarm optimization (PSO) [10], differential evolution (DE) [11], the whale optimization algorithm (WOA) [12], multi-verse optimizer (MVO) [13], and the sine cosine algorithm (SCA) [14] by studying the collective behavior of organisms or physical phenomena. In contrast to conventional optimization algorithms, these emerging algorithms possess the advantages of fewer parameters, simpler principles, and enhanced robustness. They have found successful applications across diverse scientific research domains.

The pathfinder algorithm (PFA) [15], a swarm intelligence optimization algorithm based on the concepts of biological evolution and swarm intelligence, was introduced

by Yapici et al. in 2019. In order to obtain the best answers, the algorithm simulates the processes of natural selection and biological evolution. The algorithm stands out for its adaptive parameter settings, diverse exploration strategies, robustness, and high efficiency. It has found widespread applications in various domains, including machine learning, signal processing, and image processing. However, PFA is not without its challenges, including slow convergence speed, suboptimal solution accuracy, and susceptibility to local optimization.

Sonali Priyadarshani et al. [16] employed the PFA to determine optimal parameters for the FOTID controller. This application aimed to enhance the generation control performance of multi-source power systems by integrating PFA with the AGC system and FOTID controller. Varaprasad Janamala [17] utilized the PFA to simulate the path search process for optimizing the configuration scheme of a solar PV system. The objective was to enhance the resilience and recovery of the system. Eid A. Gouda et al. [18] conducted an analysis and evaluation of the performance of fuel cells under dynamic loads and varying environments. They achieved this by applying the PFA to a fuel cell system. The study demonstrated how the fuel cell system adapts to diverse operating conditions and load demands, and further proposed improvement and optimization strategies. Zhi Yuan et al. [19] optimized the speed trajectory of a fuel cell-based locomotive by employing an enhanced PFA to achieve optimal hydrogen consumption.

To address the issue of low solution accuracy in the PFA, Tang [20] introduced a wizard mechanism to enhance the algorithm's performance. Pathfinder individuals equipped with the wizard mechanism collect valuable information from their surroundings and share this information with followers. Additionally, a novel variation probability, pcR , is defined to improve the algorithm's capability to escape local optima. A preceding researcher [21] integrated the teaching-learning algorithm with the PFA and introduced an exponential step operator to optimize individual followers during the follower phase. This enhancement notably increased the overall optimization accuracy and convergence speed of the algorithm. However, it did not completely address the issue of a too-restricted search range resulting from the random distribution of the initial population in the algorithm. The pathfinder serves as a pivotal entity in the PFA algorithm, Hu Rong [22] introduced a distance-based selection mechanism to broaden the search range of the pathfinder. Additionally, a self-learning search strategy was devised to facilitate a multi-neighborhood search on the updated pathfinder individuals, thereby enhancing the algorithm's capability for local exploration. However, this approach does not consider the impact on the follower group when pathfinder individuals encounter local optima. Lu Miao [23] enhanced the optimization performance of the grey wolf optimization algorithm by leveraging the distinctive updating pattern of the pathfinder and followers within the PFA. This fusion resulted in optimization algorithms characterized by high convergence accuracy. However, the inherent challenge of low optimization accuracy in the PFA itself remains to be universally resolved. In the work by Sun Zhezhong [24], a dynamic opposition-based learning strategy was employed to enhance the quality of the initial population. A novel leapfrog archive was introduced for preserving and generating new optimal individuals, guiding individuals trapped in local optima to escape their current positions. Furthermore, a two-jump model was proposed to harmonize the algorithm's global search and local exploitation capability. However, this approach did not address the challenge of the PFA's inefficiency in searching for the optimum on complex benchmark functions.

The pathfinder algorithm comprises only two stages: the pathfinder stage and the follower stage. However, when the pathfinder is trapped in a local region, the follower updates according to the pathfinder's position, resulting in slow convergence and a propensity to fall into local optima. Moreover, random initialization often yields lower-quality initial solutions, thereby diminishing the algorithm's solving accuracy. To address these challenges, this paper proposes the symmetry-enhanced, improved pathfinder algorithm-based multi-strategy fusion for engineering optimization problems (IPFA), which integrates multiple strategies. Firstly, the introduction of an elite opposition-based learning mech-

anism aims to inject learned superior individuals into the initial stage of the pathfinder optimization algorithm. This enhances both the diversity and quality of individuals within the entire population. By selecting individuals with superior performance in the search space and introducing them into the initial population, the algorithm is prompted to explore the solution space more comprehensively, thereby improving accuracy. Next, the escape energy factor is integrated into the prey-hunting phase of the grey wolf algorithm, replacing the follower phase in the pathfinder algorithm. This addition enables individuals to opt for escape during the search process, aiming to enhance algorithm diversity, search efficiency, and robustness, and ultimately improve convergence speed. Ultimately, employing the dimension-by-dimension mutation method to perturb the position of the optimal individual aims to facilitate a swift departure from local optimal solutions and steer toward alternative regions. This strategy proves effective by disrupting the local structure of the current solution through small random perturbations applied to each dimension of the optimal individual. Consequently, it guides the search process toward novel and potentially more optimal directions, thereby reducing the likelihood of the algorithm becoming trapped in local optimality. In summary, the amalgamation of these mechanisms into the pathfinder algorithm yields a comprehensive and robust optimization framework. This framework addresses aspects such as global optimality search, diversity maintenance, escape mechanisms, and perturbation strategies, thereby enhancing the algorithm's overall robustness and performance.

This paper's main contributions can be summed up as follows:

- In order to improve the quality of the initial solution of the PFA and enhance the accuracy of the algorithm, an elite reverse learning initialized population is introduced instead of a random initial population;
- In order to improve the convergence speed of PFA, the escape energy factor will be embedded into the prey-hunting phase of the grey wolf algorithm and replace the follower phase in the pathfinder algorithm;
- To ameliorate the problem of the PFA falling into a local optimum, the optimal individual position is perturbed using a t-distribution dimension-by-dimension variation method.

The paper's following sections are organized as follows: the second section outlines the standard pathfinder algorithm, presenting its pseudo-code and flowchart. In the third part, an enhanced pathfinder algorithm (IPFA) is introduced, amalgamating three improvements. The fourth section details the experimental simulations and result analysis, with applications to two engineering design problems. The fifth section concludes by summarizing the key findings of the paper.

2. PFA

2.1. Inspiration

The pathfinder algorithm draws inspiration from exploratory behavior in biology, particularly the behavior of animals searching for food, water, and safe shelter in unfamiliar environments. The algorithm leverages the innate exploratory instincts observed in animals, including birds during migration and insects while foraging for food.

In these biological behaviors, explorers make decisions by integrating information about the environment and internal drivers, dynamically adjusting their paths of action to secure optimal survival strategies. Taking cues from this behavioral pattern, the pathfinder algorithm relies on the cooperative efforts of multiple explorers within the search space. In this approach, each explorer continually adapts its movement direction and speed based on individual experience and information about the surrounding environment, with the ultimate goal of uncovering the globally optimal solution.

The inspiration for the pathfinder algorithm combines biological exploratory behavior with mathematical optimization algorithms aimed at finding optimal solutions in complex search spaces. The algorithm can strike a balance between local and global searches, better avoiding slipping into local optimal solutions by mimicking the behavior of explorers in

unfamiliar areas. Because of this motivation, the pathfinder algorithm has the potential to be applied to a wide range of problem domains, particularly for efficient global optimization searches in situations where there is a significant degree of uncertainty and complexity in the problem's solution space.

2.2. Mathematical Model

The population is split into two categories in PFA: pathfinders and followers. The pathfinder is the person with the highest fitness value, while the remaining people are classified as followers.

The position vector X in the PFA will consist of N individuals in the population of dimension d . Thus, the populations form an $N \times d$ dimensional matrix, i.e.,

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^d \end{bmatrix} \quad (1)$$

The person with the best fitness value is called the pathfinder in the population's iterative process, where N is the population size and d is the geographical dimension. All followers adjust their positions toward the pathfinder using the 'updating' method described in Equation (2), as follows:

$$x_p^{t+1} = x_p^t + 2r_1(x_p^t - x_p^{t-1}) + A \quad (2)$$

$$A = u_1 \times \exp\left\{\frac{-2t}{T_{\max}}\right\} \quad (3)$$

The expression provided indicates that t is the current iteration number, $T_{\max}$ is the maximum iteration number of the algorithm, x_p^t is the pathfinder's position at the t -th iteration, x_p^{t-1} is the pathfinder's position at the t -generation, and x_p^{t+1} is the pathfinder's position at the $t + 1$ generation. r_1 is the pathfinder's step factor in the range $[0,1]$, u_1 is a random number in the range $[-1,1]$, and A is the pathfinder's randomized step size, which is determined by the multi-directionality of the value of u_1 .

After the pathfinder is updated, the follower is updated according to the pathfinder position in Equation (4):

$$x_i^{t+1} = x_i^t + R_1(x_j^t - x_i^t) + R_2(x_p^t - x_i^t) + \varepsilon \quad (4)$$

$$\varepsilon = \left(1 - \frac{t}{T_{\max}}\right) u_2 D_{ij} \quad (5)$$

$$D_{ij} = \|x_i - x_j\| \quad (6)$$

$$R_1 = \alpha r_2 \quad (7)$$

$$R_2 = \beta r_3 \quad (8)$$

In this context, the positions of the i th and j th followers at the t -th iteration are shown by the symbols x_i^t and x_j^t , respectively. The updated position of the i th follower following the update is represented by x_i^{t+1} . Vectors R_1 and R_2 are random. The interaction coefficient between the pathfinder and followers is denoted by β , while the interaction coefficient among followers is represented by α . Vectors α and β have a uniform distribution within the range $[1,2]$. Random numbers with uniform distribution in the interval $[0,1]$ make up r_2 and r_3 . A random value in the interval $[-1,1]$ that determines the direction of the follower movement is denoted by u_2 , and ε stands for a factor that adds randomness to the follower

movement. The separation between the i th and j th followers is denoted by D_{ij} . Algorithm 1 is as below.

Algorithm 1 Pseudo-code of PFA.

```

1: Initialize IPFA parameters
2: Compute the initial value of population fitness
3: Individuals with the smallest fitness values were used as pathfinders
4: while  $t < \text{maximum iterations}$  do
5:   Update pathfinder the position using Equation (2) and check the bound
6:   if new pathfinder is better than old then
7:     Update pathfinder
8:   end if
9:   for  $i = 2$  to maximum iterations do
10:    Update follower the position using Equation (4) and check the bound
11:   end for
12:   Compute new fitness of members
13:   Find the best fitness
14:   if new best fitness  $<$  old best fitness then
15:     best member = new best member
16:     Fitness = best fitness
17:   end if
18:    $t = t + 1$ 
19: end while

```

2.3. PFA Steps

The pathfinder algorithm's pseudo-code and particular steps are as follows:

Step 1: Set the initialization parameters of the algorithm, including the population size N , the upper and lower bounds (ub and lb) of the search range, and the maximum iterations ($T_{\max}$);

Step 2: Initialize the population at random;

Step 3: The population's fitness values are computed and sorted based on the fitness function, and the pathfinder is the person with the lowest fitness value;

Step 4: Pathfinder phase. Pathfinder positions are updated according to Equation (2);

Step 5: Follower phase. Update the follower position according to Equation (4);

Step 6: The population's individuals' fitness values are updated, and the person with the lowest fitness value becomes the new pathfinder;

Step 7: If the iteration termination condition is met, repeat steps 4 through 7 continuously and output the global optimal solution.

3. IPFA Improvements

3.1. Elite Opposition-Based Learning Initialized Population

Opposition-based learning (OBL) [25,26], introduced by Tizhoosh in 2005 within the realm of intelligent computing, represents a novel concept aimed at enhancing the optimization process. Its primary objective is to incorporate high-quality individual solutions, thereby improving the algorithm's search efficiency and convergence performance while minimizing unnecessary explorations in the search space. This strategic integration enables the algorithm to converge to optimal solutions more swiftly. In this context, the opposition point is defined as follows:

Definition 1. *Opposite point.* Given a point $x = (x_1, x_2, \dots, x_d)$ in a d -dimensional space, where $x_i \in [lb, ub]$ with ub and lb representing the upper and lower bounds of the search range, for $i \in [1, d]$, the opposite point $x' = (x'_1, x'_2, \dots, x'_d)$ can be defined as follows:

$$x'_i = lb + ub - x_i \quad (9)$$

Definition 2. *Elite opposition solution.* Let the elite individuals in the population be denoted as $x_{i,j} = (x_{i,1}, x_{i,2}, \dots, x_{i,d}) (i = 1, 2, \dots, N)$, where d represents the dimensionality. The elite opposition-based solution, denoted as $x_{i,j}^* = (x_{i,1}^*, x_{i,2}^*, \dots, x_{i,d}^*)$, is defined as follows:

$$x_{i,j}^* = k \times (lb_j + ub_j) - x_{i,j} \quad (10)$$

where $k \in [0,1]$ of random numbers, $x_{i,j}^* \in [lb_j, ub_j]$, $lb_j = \min(x_{i,j})$, $ub_j = \max(x_{i,j})$. If $x_{i,j}^*$ crosses the boundary, the position is reset using the following equation:

$$x_{i,j}^* = rand \times (lb_j + ub_j) \quad (11)$$

Elite opposition-based learning (EOBL) [27] presents clear advantages over traditional reverse learning methods. This mechanism capitalizes on the valuable information carried by elite individuals in the population. It begins by forming a reverse population from these elite individuals and then selects the best individuals from both the reverse population and the current population to form a new population. EOBL has garnered widespread adoption among researchers. For instance, YuXin Guo [28] employed EOBL to optimize the Harris Hawks optimization algorithm, thereby enhancing population diversity and quality. Similarly, ChengWang Xie et al. [29] integrated EOBL into the fireworks explosion algorithm to bolster its global search capability.

3.2. Grey Wolf Optimizer

The grey wolf optimizer (GWO) [30] is an optimization algorithm based on the behavior of grey wolf packs in nature, which simulates the collaborative and competitive relationships between leaders and followers in a grey wolf pack. Due to its simple but effective search mechanism, GWO shows strong performance in solving optimization problems. The core idea of this algorithm is to mimic the behavioral style of a grey wolf pack, which includes synergy between the leader grey wolf, the deputy leader grey wolf, and the regular grey wolves. The leader grey wolf is responsible for guiding the entire pack to move in the direction of a more optimal solution, while the follower grey wolves search as the leader guides them. By simulating this collective intelligence and collaborative behavior of the grey wolf pack, GWO can efficiently search the solution space and find the global optimal solution or a near-optimal solution. The GWO algorithm is simple to understand, easy to implement, and shows good robustness and performance in solving various types of optimization problems. As a result, numerous scholars have devised adaptive weight factors for position updates to enhance the search speed and accuracy of the grey wolf optimizer (GWO). Xinming Zhang et al. [31] proposed a hybrid algorithm (HGWOP) that integrates particle swarm optimization (PSO) and GWO. They introduced a differential perturbation strategy into GWO, resulting in SDPGWO. Additionally, a stochastic mean example learning strategy was applied to PSO, yielding MELPSO. The efficient hybrid algorithm HGWOP was then created by seamlessly integrating SDPGWO and MELPSO. Yun Ou et al. [32] designed an adaptive weight factor for position updating to improve the speed and accuracy of the GWO search. In this paper, the use of the grey wolf algorithm in the (2) Hunting phase instead of the follower phase of the PFA can improve the convergence speed of the algorithm and reduce the probability of the overall algorithm being stagnant because the pathfinder falls into the local optimum; the following is the gray wolf optimizer's mathematical model:

The position vector X in the grey wolf optimizer will consist of N individuals in the population of dimension d . Each individual represents a "grey wolf", which has a social hierarchy, including " α " (the best individual), " β " (the second-best individual), " δ " (the third-best individual), and " ω " (the other individuals). The hierarchy is shown in Figure 1.

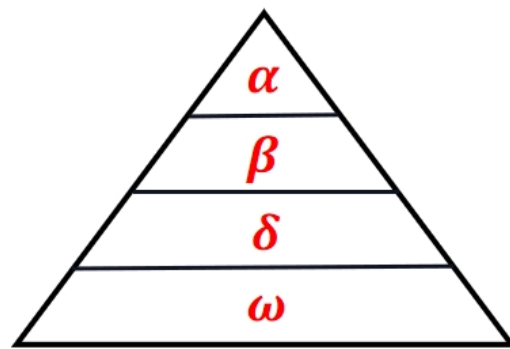


Figure 1. Grey wolf pyramid levels.

- (1) Encircling prey.

$$X^{(t+1)} = X_p^{(t)} - A \cdot D \quad (12)$$

$$D = \left| C \cdot X_p^{(t)} - X^{(t)} \right| \quad (13)$$

where $X_p^{(t)}$ is the current generation prey position, $X^{(t)}$ is the current generation grey wolf individual position, A and C are vector coefficients with formulas $A = 2a * r_1 - a$ and $C = 2r_2$, r_1 and r_2 are random numbers between the intervals $[0,1]$, and a is the convergence factor, which decreases linearly from 2 to 0 as the number of iterations increases, i.e.,

$$a = 2 - 2 \frac{t}{T_{\max}} \quad (14)$$

- (2) Hunting. The positions of the other grey wolves in group X are jointly determined based on the positions of α , β , and δ :

$$X_1^{(t)} = X_\alpha^{(t)} - A_1 \cdot D_\alpha, D_\alpha = \left| C_1 \cdot X_\alpha^{(t)} - X^{(t)} \right| \quad (15)$$

$$X_2^{(t)} = X_\beta^{(t)} - A_2 \cdot D_\beta, D_\beta = \left| C_2 \cdot X_\beta^{(t)} - X^{(t)} \right| \quad (16)$$

$$X_3^{(t)} = X_\delta^{(t)} - A_3 \cdot D_\delta, D_\delta = \left| C_3 \cdot X_\delta^{(t)} - X^{(t)} \right| \quad (17)$$

$$X^{(t+1)} = (X_1^{(t)} + X_2^{(t)} + X_3^{(t)}) / 3 \quad (18)$$

where A_1 , A_2 , and A_3 are defined the same as A and C_1 , C_2 and C_3 are defined the same as C . The way of updating the wolf position is shown in the following Figure 2.

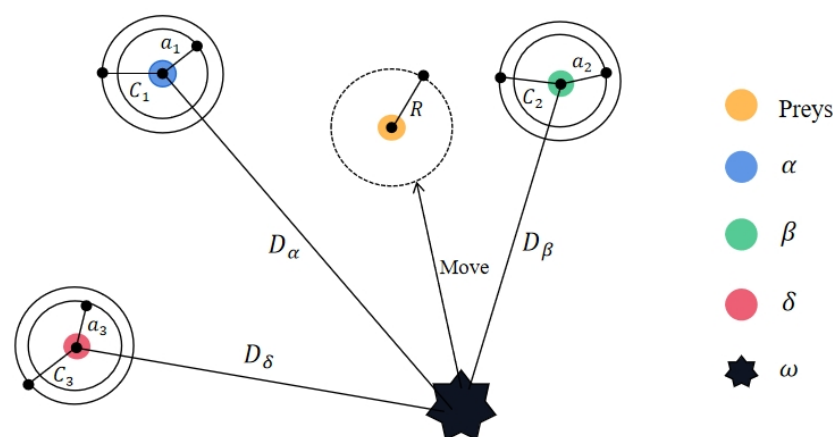


Figure 2. Schematic diagram of grey wolf location update.

3.3. Follower Update Formulation Based on GWO and Escape Energy

In highly complex multimodal problems, the PFA may fall into local optimal solutions; this is because the PFA is mainly based on competition and selection among individuals, and may miss the global optimal solution in some cases. GWO excels in global search, leveraging population behavior and alignment mechanisms to effectively explore the search space and identify either the global optimal solution or a near-optimal one. In light of these strengths, this paper integrates the PFA with the GWO algorithm to enhance the overall performance of PFA.

To strike a balance between global and local search, and to further improve the search efficiency and optimization performance of the algorithm, the paper introduces the concept of escape energy (E) from the Harris Hawks optimization (HHO) [33]. Equation (19) captures this concept and is presented as follows:

$$E = 2E_0 \left(1 - \frac{t}{T_{\max}} \right) \quad (19)$$

Here, E_0 represents a randomly generated number within the range of $[-1, 1]$. The escape energy (E) plays a crucial role in determining the algorithm's performance, influencing factors such as search speed, search quality, and convergence. When the escape energy $|E|$ is greater than or equal to 1, it encourages individuals to engage in global exploration more frequently. This is particularly beneficial when the entire search space has not been thoroughly explored, aiding the algorithm in swiftly identifying potential globally optimal solutions during the initial iterations. Conversely, when the escape energy $|E|$ is less than or equal to 1, individuals tend to prioritize local searches. They concentrate on delving deeper into regions near the currently identified superior solutions, facilitating the algorithm in optimizing those candidate solutions with a more detailed approach.

The follower update Equation (23) for the PFA, combining the GWO and the escape energy (E), is as follows:

$$x_1^t = x_\alpha^t - E \cdot A_1 \cdot D_\alpha, D_\alpha = |C_1 \cdot x_\alpha^t - x^t| \quad (20)$$

$$x_2^t = x_\beta^t - E \cdot A_2 \cdot D_\beta, D_\beta = |C_2 \cdot x_\beta^t - x^t| \quad (21)$$

$$x_3^t = x_\delta^t - E \cdot A_3 \cdot D_\delta, D_\delta = |C_3 \cdot x_\delta^t - x^t| \quad (22)$$

$$x^{t+1} = (x_1^t + x_2^t + x_3^t) / 3 \quad (23)$$

3.4. Dimension-by-Dimension Mutation

The t -distribution [34,35], also known as the student distribution, has the following probability density (Equation (24)):

$$p(x) = \frac{\Gamma\left(\frac{m+1}{2}\right)}{\sqrt{m\pi}\Gamma\left(\frac{m}{2}\right)} \left(1 + \frac{x^2}{2}\right)^{-\frac{m+1}{2}} \quad (24)$$

where $\Gamma\left(\frac{m+1}{2}\right) = \int_0^{+\infty} x^{\frac{m+1}{2}-1} e^{-x} dx$ is the Euler integral of type II. The t -distribution curve shape is related to the value of the degrees of freedom parameter m . When the parameter $m = 1$, the t -distribution is in the form of a Cauchy distribution, denoted as $t(m = 1) = C(0, 1)$. As the parameter " m " increases, the t -distribution gradually approaches the standard normal distribution. When the parameter " m " tends to infinity, the t -distribution gradually converges to the standard normal distribution $N(0, 1)$. Figure 3 shows the distribution of the t -distribution, Cauchy distribution, and normal distribution with parameter changes for different degrees of freedom.

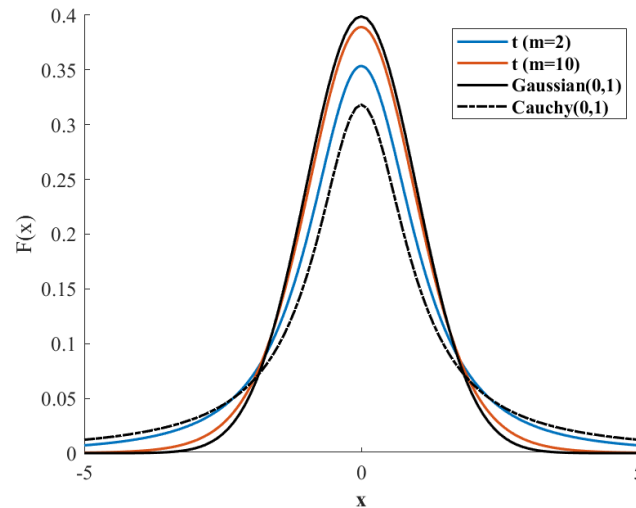


Figure 3. Gaussian, t -distribution, and Cauchy distribution density functions.

The output result of the mutation operation has uncertainty, and if the dimension-by-dimension mutation operation is performed on each individual in the population, it will inevitably lead to an increase in computational complexity and may reduce the search efficiency of the algorithm. Therefore, in this section, only the optimal individual in the current population is subjected to the dimension-by-dimension mutation operation, and the dimension-by-dimension mutation formula is as follows:

$$x_{newbest}^d = x_{best}^d + x_{best}^d \times t(iter) \quad (25)$$

where $t(iter)$ is the t -distribution of the PFA's iteration count for the degree of freedom parameter, and $x_{newbest}^d$ is the new solution following mutation. At the beginning of the iteration, $iter$ is smaller, representing a smaller degree of the freedom parameter; the model is close to the Cauchy distribution, $t(iter)$ has a significant role in the perturbation of the individual x_{best}^d , which effectively enhances the algorithm's ability to search globally, and helps to prevent the individual from falling into the local aggregation phenomenon. As the iteration of the algorithm proceeds, the increase in the degrees of freedom parameter causes the t distribution to gradually converge to a Gaussian distribution, which results in a gradual weakening of the perturbation of the distribution operator on the individual x_{best}^d . This evolution is particularly evident in the later stages of the algorithm (where $iter$ is larger), which in turn provides the algorithm with a higher degree of local exploitation and convergence accuracy.

Since the optimal individual position after mutation cannot be guaranteed to be better than the original position, an optimization-preserving strategy, i.e., greedy strategy, is added after the dimension-by-dimension mutation position perturbation. By comparing the fitness value of the mutated optimal individual with the original optimal individual, the individual with the best fitness value is selected as the global optimal solution for the current generation, then the mutated individual replaces the original individual, and the judgment formula is as follows:

$$x_{newbest}^d = \begin{cases} x_{best}^d & f(x_{best}^d) < f(x_{newbest}^d) \\ x_{newbest}^d & f(x_{best}^d) \geq f(x_{newbest}^d) \end{cases} \quad (26)$$

3.5. IPFA Steps

The IPFA algorithm's specific steps and Algorithm 2 are as follows, based on the enhancements in this section:

Step 1: The initialized algorithm parameters, including population size (N), upper and lower boundaries (ub and lb) of the search range, and the maximum iterations (T_{max});

Step 2: The population is initialized using the elite opposition-based learning strategy outlined in Equation (10);

Step 3: Computing the fitness values of each individual in the population based on a fitness function. Sort the individuals based on the fitness value and select the individual with the smallest fitness value as the pathfinder;

Step 4: Pathfinder phase. Update the pathfinder position according to Equation (2);

Step 5: Follower stage. Update the follower position according to Equation (23);

Step 6: Mutation stage. The fitness values of all individuals are computed and ranked, and then the individual with the optimal fitness value is picked for dimension-by-dimension mutation based on Equation (25);

Step 7: Determine whether to update the optimal individual according to the greedy strategy Equation (26);

Step 8: The population's individual fitness levels are updated, and the person with the lowest value is selected to be the new pathfinder;

Step 9: If the termination condition for iterations is met, output the optimal individual position x_{best}^t along with its corresponding fitness value. Otherwise, repeat Steps 4 through 9 until the iteration termination is satisfied.

Algorithm 2 Pseudo-code of IPFA.

```

1: Initialize IPFA parameters
2: Initialize population using EOBL by using Equations (9) and (10)
3: When the elite opposition solution crosses the boundary utilizing Equation (11) update
4: Compute the initial value of population fitness
5: Individuals with the smallest fitness values were used as pathfinders
6: while  $t < \text{maximum iterations}$  do
7:   Update pathfinder the position using Equation (2) and check the bound
8:   if new pathfinder is better than old then
9:     Update pathfinder
10:  end if
11:  for  $i = 2$  to maximum iterations do
12:    Using Equation (19) to calculate the value of the escape factor  $E$ 
13:    Individual population fitness values were ranked to determine  $\alpha$ ,  $\beta$ , and  $\delta$  wolf
14:    Update follower the position using Equations (20)–(23) and check the bound
15:  end for
16:  Compute new fitness of the population
17:  Find the best fitness
18:  Dimension-by-dimension mutation of new best member using Equation (25)
19:  if new best fitness  $>$  old best fitness then
20:    Get a new best member using Equation (26)
21:  else
22:    Get a new best member using Equation (26)
23:  end if
24:  Compute new fitness of members
25:  Find the best fitness
26:  if best fitness  $<$  fitness of pathfinder then
27:    pathfinder = best member
28:    fitness = best fitness
29:  end if
30:   $t = t + 1$ 
31: end while

```

3.6. Time Complexity Analysis

The total temporal complexity of the PFA is known to be $O(d + f(d))$, assuming a population size of N , spatial dimensions of d , and a maximum iteration count of $T_{\max}$.

In this scenario, $f(x)$ represents the fitness function. The time complexity of the IPFA is now analyzed.

For the IPFA, in the initialization population stage, assuming the time for initializing algorithm parameters is t_1 , and the time for sorting all individuals based on fitness values and selecting the pathfinder is t_2 , the time complexity of the initialization population stage can be expressed as follows:

$$T_1 = O(t_1 + N \times f(d) + t_2) + O(N \times d) = O(d + f(d))$$

Here, the time complexity for initializing the individuals in the population through elite opposition-based learning is $O(N \times d)$, and the time for calculating the fitness value for each individual is $f(d)$.

As the iteration progresses, during the pathfinder phase, the IPFA updates the pathfinder position like the standard PFA, without incurring additional time. Consequently, the time complexity of the pathfinder phase remains consistent with that of the standard PFA, as follows:

$$T_2 = O(d)$$

During the follower phase, the time required for computing the coefficient vectors A and C is denoted as t_3 . Additionally, the time for calculating the distances between α , β , and δ wolves for each individual in the population is represented by t_4 . The computation of $X_1^{(t)}$, $X_2^{(t)}$, and $X_3^{(t)}$ consumes time t_5 , while the calculation of $X^{(t+1)}$ requires time t_6 . The processing time for the boundaries of the follower's dimensions is denoted as t_7 , and the time for computing E is t_8 . Collectively, these components contribute to the time complexity of the follower phase:

$$T_3 = O((N - 1) \times (t_3 + 3(t_4 + t_5) + t_6 + d \times t_7 + f(d) + t_8)) = O(d + f(d))$$

During the mutation phase, the following time complexities are measured: t_9 is the time of sorting the population of individuals to find the optimal individual; t_{10} is the time of the optimal individual performing dimension-by-dimension mutation; $f(d)$ is the time of computing the fitness value of the optimal individual; t_{11} is the time of determining whether the new optimal individual replaces the old optimal individual by using the greedy strategy; t_{12} is the time of retaining the position of the optimal individual; and t_{13} is the time of processing the boundaries of the optimal individual in each dimension, so the time complexity of the mutation phase is as follows:

$$T_4 = O(1 \times d \times (t_{10} + t_{11} + t_{13}) + t_9 + t_{12} + 1 \times f(d)) = O(d + f(d))$$

To put it briefly, the IPFA's total temporal complexity is $T = T_1 + T_{\max}(T_2 + T_3 + T_4) = O(d + f(d))$. As a result, the IPFA's time complexity and execution efficiency are comparable to those of the regular PFA.

4. Experimental Simulation and Analysis of Results

4.1. Experimental Design

To assess the performance of the proposed IPFA in this study, IPFA is compared with particle swarm optimization (PSO) [10], differential evolution (DE) [11], whale optimization algorithm (WOA) [12], seagull optimization algorithm (SOA) [36], sine cosine algorithm (SCA) [14], grey wolf optimizer (GWO) [30], chimp optimization algorithm (ChOA) [37], and the standard PFA [15] on 45 benchmark test functions (function information as shown in Table 1). The functions f_1 – f_7 are unimodal (UN) functions, f_8 – f_{12} are multimodal (MN) functions, f_{13} – f_{16} are fixed-dimension multimodal (FM) functions, and F1–F29 are benchmark test functions from CEC2017. In CEC2017, F1–F9 involve functions with rotation and translation tests, F10–F19 are composite functions designed to assess the algorithm's capability in handling high-dimensional problems, and F20–F29 are hybrid functions designed to evaluate the algorithm's performance under various complex scenarios. To ensure the

fairness of the evaluation, each algorithm underwent 30 independent runs to minimize the possible effects of random samples. While maintaining uniformity, all algorithms shared a set of parameter settings, such as a population size of 30 and a maximum iteration of 1000. For the fairness of the experiment, all algorithms were subjected to optimization testing on a computer equipped with an Intel(R) Core(TM) i5—11260H CPU @ 2.60 GHz, running Windows 10, with 16 GB of memory and a 64-bit operating system. The optimization experiments were conducted using MATLAB R2021a software. Table 2 provides the parameter settings for each algorithm.

Table 1. Benchmark functions.

Functions	Function Name	Dim	Range	$f_{\min}$
f_1	Sphere	100	[−100, 100]	0
f_2	Schwefel's problem 2.22	100	[−10, 10]	0
f_3	Schwefel's problem 1.2	100	[−100, 100]	0
f_4	Schwefel's problem 2.21	100	[−100, 100]	0
f_5	Rosenbrock	100	[−30, 30]	0
f_6	Step	100	[−100, 100]	0
f_7	Noise	100	[−1.28, 1.28]	0
f_8	Rastrigin	100	[−5.12, 5.12]	0
f_9	Ackley	100	[−32, 32]	0
f_{10}	Griewank	100	[−600, 600]	0
f_{11}	Generalized Penalized function 1	100	[−50, 50]	0
f_{12}	Generalized Penalized function 2	100	[−50, 50]	0
f_{13}	Hartman 2	6	[0, 1]	−3.32
f_{14}	Shekel 1	4	[0, 10]	−10.1532
f_{15}	Shekel 2	4	[0, 10]	−10.4028
f_{16}	Shekel 3	4	[0, 10]	−10.5363
CEC2017 Test Functions				
F1	Shifted and Rotated Bent Cigar Function	30	[−100, 100]	100
F2	Shifted and Rotated Zakharov Function	30	[−100, 100]	200
F3	Shifted and Rotated Rosenbrock Function	30	[−100, 100]	300
F4	Shifted and Rotated Rastrigin's Function	30	[−100, 100]	400
F5	Shifted and Rotated Expanded Schaffer F6 Function	30	[−100, 100]	500
F6	Shifted and Rotated Lunacek Bi_ Rastrigin Function	30	[−100, 100]	600
F7	Shifted and Rotated Non-Continuous Rastrigin Function	30	[−100, 100]	700
F8	Shifted and Rotated Levy Function	30	[−100, 100]	800
F9	Shifted and Rotated Schwefel Function	30	[−100, 100]	900
F10	Hybrid Function 1 (N = 3)	30	[−100, 100]	1000
F11	Hybrid Function 2 (N = 3)	30	[−100, 100]	1100
F12	Hybrid Function 3 (N = 3)	30	[−100, 100]	1200
F13	Hybrid Function 4 (N = 4)	30	[−100, 100]	1300
F14	Hybrid Function 5 (N = 4)	30	[−100, 100]	1400
F15	Hybrid Function 6 (N = 4)	30	[−100, 100]	1500
F16	Hybrid Function 6 (N = 5)	30	[−100, 100]	1600
F17	Hybrid Function 6 (N = 5)	30	[−100, 100]	1700
F18	Hybrid Function 6 (N = 5)	30	[−100, 100]	1800
F19	Hybrid Function 6 (N = 6)	30	[−100, 100]	1900
F20	Composition Function 1 (N = 3)	30	[−100, 100]	2000
F21	Composition Function 2 (N = 3)	30	[−100, 100]	2100
F22	Composition Function 3 (N = 3)	30	[−100, 100]	2200
F23	Composition Function 4 (N = 3)	30	[−100, 100]	2300
F24	Composition Function 5 (N = 3)	30	[−100, 100]	2400
F25	Composition Function 6 (N = 3)	30	[−100, 100]	2500
F26	Composition Function 7 (N = 3)	30	[−100, 100]	2600
F27	Composition Function 8 (N = 3)	30	[−100, 100]	2700
F28	Composition Function 9 (N = 3)	30	[−100, 100]	2800
F29	Composition Function 10 (N = 3)	30	[−100, 100]	2900

Table 2. Experimental parameters of each algorithm.

Algorithms	Years	Parameters
IPFA	Present	$s = 0.5$
PFA	2019	-
PSO	1998	$w = 0.9, c_1 = c_2 = 2$
WOA	2016	$a_{\max} = 2, a_{\min} = 0$
DE	1995	$F = 0.85, PC_r = 0.8$
SOA	2019	$f_c = 2, u = v = 1$
SCA	2016	$a = 2$
ChOA	2020	$m = \text{chaos}(3, 1, 1)$
GWO	2014	$a = 2 - 2 \frac{t}{T_{\max}}$

4.2. Comparative Analysis of Optimization Performance

In this section, a comprehensive comparison is conducted between IPFA and nine other evolutionary algorithms, with summarized results presented in Table 3. Three key metrics are analyzed for each algorithm: mean, standard deviation (S.D), and Best. The “Mean” represents the average obtained by summing the results of 30 independent runs of a specific algorithm and dividing by the number of runs. Also, the algorithms are ranked according to their average values, with the rankings starting from 1, representing the best performing algorithm, followed by a rank of 2, and so on; if there are multiple algorithms with equal performance, they are ranked the same, while the rankings of the other algorithms are computed in the usual order. The sum of the algorithm rankings on the test functions and the overall rankings are shown in the last two rows. The better an algorithm performs on each test function, the smaller the value of the sum of the rankings. Thus, the best algorithm has a composite ranking of 1, followed closely by algorithms with a ranking of 2, and so on.

Regarding the single-peak functions, f_1 , f_2 , f_6 , and f_7 , the IPFA obtains a performance of Rank 1, in terms of the mean, S.D., and best, but regarding the functions, f_4 , f_5 , and f_8 , the IPFA is only ranked 2nd, 5th, and 3rd, which is less than ideal for finding the optimal results. Overall, the convergence performance of IPFA on single-peaked functions is slightly better than the other eight algorithms.

The IPFA secures the top ranking on multi-peak functions f_{11} and f_{12} . However, its performance is less impressive on other multi-peak and fixed-dimensional multi-peak functions, where it holds the 6th, 2nd, 4th, and 3rd positions for functions f_{10} , f_{13} , and f_{14} , respectively. It is important to remember that for these three functions, the ‘best’ is ideal.

Table 3. Comparison of benchmarking function optimization results.

Fun	Item	IPFA	PFA	PSO	DE	WOA	SCA	SOA	GWO	ChOA
f_1	Mean	1.1018×10^{-310}	2.0180×10^{-02}	$7.5848 \times 10^{+01}$	$6.2796 \times 10^{+02}$	6.0887×10^{-151}	7.1948×10^{-02}	2.1570×10^{-11}	1.9133×10^{-58}	4.0208×10^{-18}
	S.D	$0.0000 \times 10^{+00}$	1.7936×10^{-02}	$4.4935 \times 10^{+01}$	$1.6167 \times 10^{+03}$	3.2862×10^{-150}	1.9089×10^{-01}	5.6886×10^{-11}	8.6091×10^{-58}	1.0360×10^{-17}
	Best	$0.0000 \times 10^{+00}$	2.2859×10^{-03}	$5.4413 \times 10^{+00}$	8.4376×10^{-01}	5.8209×10^{-167}	4.3022×10^{-06}	1.6932×10^{-16}	4.9392×10^{-61}	2.6633×10^{-28}
	Rank	1	6	8	9	2	7	5	3	4
f_2	Mean	1.1775×10^{-174}	4.8973×10^{-01}	$1.1154 \times 10^{+01}$	$1.1288 \times 10^{+01}$	2.0410×10^{-102}	2.1208×10^{-05}	1.2090×10^{-11}	1.1068×10^{-34}	1.4656×10^{-12}
	S.D	$0.0000 \times 10^{+00}$	3.9074×10^{-01}	$8.9362 \times 10^{+00}$	$1.3686 \times 10^{+01}$	9.4371×10^{-102}	4.4957×10^{-05}	2.2189×10^{-11}	1.1958×10^{-34}	2.6385×10^{-12}
	Best	2.8139×10^{-188}	5.5184×10^{-02}	$2.9679 \times 10^{+00}$	2.8200×10^{-03}	6.7965×10^{-115}	4.0508×10^{-08}	6.6393×10^{-13}	6.9846×10^{-36}	1.1245×10^{-15}
	Rank	1	7	8	9	2	6	5	3	4
f_3	Mean	$4.6998 \times 10^{+02}$	$1.0129 \times 10^{+03}$	$6.4943 \times 10^{+03}$	$1.9042 \times 10^{+04}$	$2.1360 \times 10^{+04}$	$3.5608 \times 10^{+03}$	$5.2716 \times 10^{+02}$	2.4137×10^{-14}	3.6267×10^{-02}
	S.D	$4.6667 \times 10^{+02}$	$1.1733 \times 10^{+03}$	$5.1031 \times 10^{+03}$	$1.1123 \times 10^{+04}$	$9.6159 \times 10^{+03}$	$3.0275 \times 10^{+03}$	$1.3073 \times 10^{+03}$	1.2595×10^{-13}	7.5598×10^{-02}
	Best	$2.0910 \times 10^{+01}$	$1.5814 \times 10^{+01}$	$3.5172 \times 10^{+02}$	$2.5963 \times 10^{+03}$	$5.6903 \times 10^{+03}$	$3.3794 \times 10^{+02}$	9.9885×10^{-02}	2.0600×10^{-19}	9.7881×10^{-06}
	Rank	3	5	7	8	9	6	4	1	2
f_4	Mean	7.3219×10^{-08}	$3.1580 \times 10^{+00}$	$4.0909 \times 10^{+00}$	$6.9796 \times 10^{+01}$	$4.6767 \times 10^{+01}$	$2.1025 \times 10^{+01}$	1.8440×10^{-01}	2.2443×10^{-14}	2.2341×10^{-03}
	S.D	6.7708×10^{-08}	$1.4103 \times 10^{+00}$	$1.6530 \times 10^{+00}$	$9.6208 \times 10^{+00}$	$2.8508 \times 10^{+01}$	$9.4814 \times 10^{+00}$	3.0242×10^{-01}	2.3272×10^{-145}	1.0264×10^{-02}
	Best	2.2086×10^{-09}	5.9025×10^{-01}	$1.6779 \times 10^{+00}$	$5.0193 \times 10^{+01}$	2.7382×10^{-04}	$8.9974 \times 10^{+00}$	1.0988×10^{-02}	5.6557×10^{-16}	4.6896×10^{-07}
	Rank	2	5	6	9	8	7	4	1	3
f_5	Mean	$5.3791 \times 10^{+01}$	$5.6585 \times 10^{+01}$	$1.7017 \times 10^{+03}$	$5.7367 \times 10^{+06}$	$2.7392 \times 10^{+01}$	$5.0864 \times 10^{+02}$	$2.8595 \times 10^{+01}$	$2.6796 \times 10^{+01}$	$2.8837 \times 10^{+01}$
	S.D	$4.3437 \times 10^{+01}$	$3.6634 \times 10^{+01}$	$2.1718 \times 10^{+03}$	$8.6591 \times 10^{+06}$	6.6294×10^{-01}	$1.5720 \times 10^{+03}$	3.2503×10^{-01}	6.7781×10^{-01}	1.9958×10^{-01}
	Best	$1.7417 \times 10^{+01}$	$1.9217 \times 10^{+01}$	$7.8145 \times 10^{+01}$	$1.1467 \times 10^{+03}$	$2.6727 \times 10^{+01}$	$2.7940 \times 10^{+01}$	$2.8118 \times 10^{+01}$	$2.5252 \times 10^{+01}$	$2.8057 \times 10^{+01}$
	Rank	5	6	9	9	2	8	3	1	4
f_6	Mean	2.6384×10^{-05}	2.0423×10^{-02}	$7.3081 \times 10^{+01}$	$6.3670 \times 10^{+02}$	8.9580×10^{-02}	$4.4493 \times 10^{+00}$	$1.1827 \times 10^{+00}$	6.6292×10^{-01}	$3.2993 \times 10^{+00}$
	S.D	1.1385×10^{-05}	2.3094×10^{-02}	$4.3693 \times 10^{+01}$	$1.2204 \times 10^{+03}$	1.0282×10^{-01}	3.8119×10^{-01}	2.9815×10^{-01}	3.0481×10^{-01}	3.6342×10^{-01}
	Best	8.1525×10^{-06}	7.3713×10^{-04}	$1.3054 \times 10^{+01}$	4.5141×10^{-02}	9.0726×10^{-03}	$3.4658 \times 10^{+00}$	6.4310×10^{-01}	2.4684×10^{-01}	$2.4030 \times 10^{+00}$
	Rank	1	3	8	9	4	7	5	2	6
f_7	Mean	2.4457×10^{-04}	1.4076×10^{-02}	1.6285×10^{-01}	$7.8505 \times 10^{+00}$	1.3892×10^{-03}	3.7151×10^{-02}	3.4790×10^{-02}	7.8588×10^{-04}	8.1494×10^{-04}
	S.D	2.6070×10^{-04}	8.8628×10^{-03}	4.8588×10^{-01}	$7.4746 \times 10^{+00}$	1.3605×10^{-03}	5.3369×10^{-02}	2.1584×10^{-02}	4.0907×10^{-04}	1.1463×10^{-03}
	Best	1.4739×10^{-05}	1.6585×10^{-03}	1.5829×10^{-02}	3.1174×10^{-01}	3.7655×10^{-05}	2.0600×10^{-03}	1.2667×10^{-03}	1.3397×10^{-04}	5.7108×10^{-05}
	Rank	1	5	8	9	4	7	6	2	3
f_8	Mean	2.8422×10^{-14}	$1.0318 \times 10^{+01}$	$1.6881 \times 10^{+02}$	$1.4179 \times 10^{+02}$	$0.0000 \times 10^{+00}$	$1.2877 \times 10^{+01}$	4.4798×10^{-11}	7.5791×10^{-15}	$3.0970 \times 10^{+00}$
	S.D	3.8784×10^{-14}	$1.5015 \times 10^{+01}$	$3.3957 \times 10^{+01}$	$4.7013 \times 10^{+01}$	$0.0000 \times 10^{+00}$	$1.8824 \times 10^{+01}$	1.1829×10^{-10}	1.9653×10^{-14}	$6.2301 \times 10^{+00}$
	Best	$0.0000 \times 10^{+00}$	1.2513×10^{-03}	$9.8430 \times 10^{+01}$	$6.0802 \times 10^{+01}$	$0.0000 \times 10^{+00}$	1.7284×10^{-06}	5.6843×10^{-14}	$0.0000 \times 10^{+00}$	$0.0000 \times 10^{+00}$
	Rank	3	6	9	8	1	7	4	2	5

Table 3. Cont.

Fun	Item	IPFA	PFA	PSO	DE	WOA	SCA	SOA	GWO	ChOA
f_9	Mean	5.0981×10^{-14}	$2.9496 \times 10^{+00}$	$3.6389 \times 10^{+00}$	$1.9908 \times 10^{+01}$	3.8488×10^{-15}	$1.2234 \times 10^{+01}$	4.3185×10^{-07}	1.5810×10^{-14}	$1.9962 \times 10^{+01}$
	S.D	9.6667×10^{-15}	$1.0718 \times 10^{+00}$	7.3778×10^{-01}	3.3649×10^{-01}	2.3012×10^{-15}	$9.7507 \times 10^{+00}$	7.2421×10^{-07}	2.3603×10^{-15}	1.2530×10^{-03}
	Best	3.2863×10^{-14}	1.3302×10^{-02}	$2.2662 \times 10^{+00}$	$1.8128 \times 10^{+01}$	8.8818×10^{-16}	1.2084×10^{-04}	1.3093×10^{-08}	1.1546×10^{-14}	$1.9959 \times 10^{+01}$
	Rank	3	5	6	8	1	7	4	2	9
f_{10}	Mean	4.4107×10^{-02}	7.5884×10^{-02}	$1.5283 \times 10^{+00}$	$5.4793 \times 10^{+00}$	$0.0000 \times 10^{+00}$	2.1841×10^{-01}	4.6609×10^{-03}	3.6932×10^{-03}	7.0510×10^{-03}
	S.D	5.0855×10^{-02}	4.9250×10^{-02}	3.2849×10^{-01}	$6.8166 \times 10^{+00}$	$0.0000 \times 10^{+00}$	2.4852×10^{-01}	2.5529×10^{-02}	8.4703×10^{-03}	1.6606×10^{-02}
	Best	$0.0000 \times 10^{+00}$	1.3812×10^{-02}	$1.0446 \times 10^{+00}$	5.3830×10^{-02}	$0.0000 \times 10^{+00}$	5.7019×10^{-06}	2.5979×10^{-14}	$0.0000 \times 10^{+00}$	$0.0000 \times 10^{+00}$
	Rank	6	7	8	9	1	5	3	2	4
f_{11}	Mean	2.2248×10^{-03}	8.6189×10^{-01}	$2.2523 \times 10^{+00}$	$3.0666 \times 10^{+07}$	5.0271×10^{-03}	$2.4780 \times 10^{+00}$	7.5739×10^{-02}	3.6638×10^{-02}	4.6837×10^{-01}
	S.D	1.2174×10^{-02}	$1.1806 \times 10^{+00}$	$1.3740 \times 10^{+00}$	$7.8781 \times 10^{+07}$	4.8553×10^{-03}	$3.7241 \times 10^{+00}$	3.7425×10^{-02}	1.4697×10^{-02}	2.3966×10^{-01}
	Best	2.2561×10^{-07}	1.1358×10^{-04}	5.3460×10^{-01}	$4.8681 \times 10^{+02}$	8.0551×10^{-04}	4.8240×10^{-01}	2.7712×10^{-02}	1.3180×10^{-02}	2.0163×10^{-01}
	Rank	1	6	7	9	2	8	4	3	5
f_{12}	Mean	8.1691×10^{-02}	4.1812×10^{-01}	$7.5252 \times 10^{+00}$	$4.1546 \times 10^{+07}$	2.1369×10^{-01}	$2.5411 \times 10^{+01}$	$2.7503 \times 10^{+00}$	5.0768×10^{-01}	$2.8482 \times 10^{+00}$
	S.D	1.0356×10^{-01}	3.2602×10^{-01}	$3.2403 \times 10^{+00}$	$5.1088 \times 10^{+07}$	1.7601×10^{-01}	$1.0642 \times 10^{+02}$	9.5287×10^{-02}	1.7414×10^{-01}	1.4988×10^{-01}
	Best	7.3265×10^{-06}	3.0413×10^{-02}	$3.0930 \times 10^{+00}$	$9.1490 \times 10^{+03}$	1.7585×10^{-02}	$2.1445 \times 10^{+00}$	$2.5406 \times 10^{+00}$	1.9263×10^{-01}	$2.4480 \times 10^{+00}$
	Rank	1	2	7	9	3	8	5	4	6
f_{13}	Mean	$-3.2665 \times 10^{+00}$	$-3.2744 \times 10^{+00}$	$-3.0301 \times 10^{+00}$	$-3.1891 \times 10^{+00}$	$-3.2173 \times 10^{+00}$	$-2.9486 \times 10^{+00}$	$-2.4973 \times 10^{+00}$	$-3.2644 \times 10^{+00}$	$-2.4371 \times 10^{+00}$
	S.D	7.6819×10^{-02}	5.9241×10^{-02}	3.3020×10^{-01}	1.1387×10^{-01}	1.0235×10^{-01}	2.9583×10^{-01}	6.3663×10^{-01}	7.4374×10^{-02}	4.9850×10^{-01}
	Best	$-3.3220 \times 10^{+00}$	$-3.3220 \times 10^{+00}$	$-3.3219 \times 10^{+00}$	$-3.3220 \times 10^{+00}$	$-3.3219 \times 10^{+00}$	$-3.1661 \times 10^{+00}$	$-3.0156 \times 10^{+00}$	$-3.3220 \times 10^{+00}$	$-3.0699 \times 10^{+00}$
	Rank	2	1	6	5	4	7	8	3	9
f_{14}	Mean	$-8.5353 \times 10^{+00}$	$-1.0153 \times 10^{+01}$	$-9.3357 \times 10^{+00}$	$-5.1037 \times 10^{+00}$	$-8.4517 \times 10^{+00}$	$-2.4910 \times 10^{+00}$	-7.3343×10^{-01}	$-9.4791 \times 10^{+00}$	$-2.8980 \times 10^{+00}$
	S.D	$2.7552 \times 10^{+00}$	6.4471×10^{-15}	$2.2166 \times 10^{+00}$	$3.0334 \times 10^{+00}$	$2.4427 \times 10^{+00}$	$2.0921 \times 10^{+00}$	1.5860×10^{-01}	$1.7469 \times 10^{+00}$	$2.1059 \times 10^{+00}$
	Best	$-1.0152 \times 10^{+01}$	$-1.0153 \times 10^{+01}$	$-1.0153 \times 10^{+01}$	$-1.0153 \times 10^{+01}$	$-1.0153 \times 10^{+01}$	$-6.2745 \times 10^{+00}$	$-1.1276 \times 10^{+00}$	$-1.0153 \times 10^{+01}$	$-5.0469 \times 10^{+00}$
	Rank	4	1	3	6	5	7	9	2	8
f_{15}	Mean	$-9.7286 \times 10^{+00}$	$-1.0403 \times 10^{+01}$	$-9.6957 \times 10^{+00}$	$-5.8579 \times 10^{+00}$	$-8.1441 \times 10^{+00}$	$-3.7514 \times 10^{+00}$	-8.9447×10^{-01}	$-1.0403 \times 10^{+01}$	$-3.7733 \times 10^{+00}$
	S.D	$2.0113 \times 10^{+00}$	6.5972×10^{-16}	$1.8260 \times 10^{+00}$	$3.6030 \times 10^{+00}$	$3.1109 \times 10^{+00}$	$2.2774 \times 10^{+00}$	3.3004×10^{-01}	2.3476×10^{-04}	$1.9370 \times 10^{+00}$
	Best	$-1.0402 \times 10^{+01}$	$-1.0403 \times 10^{+01}$	$-1.0403 \times 10^{+01}$	$-1.0403 \times 10^{+01}$	$-1.0403 \times 10^{+01}$	$-8.9582 \times 10^{+00}$	$-2.2132 \times 10^{+00}$	$-1.0403 \times 10^{+01}$	$-5.0688 \times 10^{+00}$
	Rank	2	1	3	5	4	7	8	1	6
f_{16}	Mean	$-8.1022 \times 10^{+00}$	$-9.5146 \times 10^{+00}$	$-1.0351 \times 10^{+01}$	$-6.8025 \times 10^{+00}$	$-9.1024 \times 10^{+00}$	$-4.4693 \times 10^{+00}$	-9.6138×10^{-01}	$-1.0266 \times 10^{+01}$	$-4.0959 \times 10^{+00}$
	S.D	$3.5173 \times 10^{+00}$	$2.6583 \times 10^{+00}$	9.7766×10^{-01}	$3.7992 \times 10^{+00}$	$2.6919 \times 10^{+00}$	$1.9074 \times 10^{+00}$	2.2087×10^{-01}	$1.4814 \times 10^{+00}$	$1.7673 \times 10^{+00}$
	Best	$-1.0533 \times 10^{+01}$	$-1.0536 \times 10^{+01}$	$-1.0536 \times 10^{+01}$	$-1.0536 \times 10^{+01}$	$-1.0536 \times 10^{+01}$	$-9.0906 \times 10^{+00}$	$-1.5282 \times 10^{+00}$	$-1.0536 \times 10^{+01}$	$-5.1066 \times 10^{+00}$
	Rank	5	3	1	6	4	7	9	2	8

Table 3. Cont.

Fun	Item	IPFA	PFA	PSO	DE	WOA	SCA	SOA	GWO	ChOA
F1	Mean	$3.2031 \times 10^{+05}$	$3.8477 \times 10^{+05}$	$4.4096 \times 10^{+09}$	$2.5608 \times 10^{+09}$	$1.5124 \times 10^{+09}$	$1.9143 \times 10^{+10}$	$2.2417 \times 10^{+10}$	$3.0315 \times 10^{+09}$	$2.8477 \times 10^{+10}$
	S.D	$1.7839 \times 10^{+05}$	$2.7042 \times 10^{+05}$	$4.5147 \times 10^{+09}$	$3.9989 \times 10^{+09}$	$8.0454 \times 10^{+08}$	$2.9889 \times 10^{+09}$	$3.6710 \times 10^{+09}$	$1.7244 \times 10^{+09}$	$5.7881 \times 10^{+09}$
	Best	$1.0000 \times 10^{+05}$	$1.2654 \times 10^{+05}$	$9.4917 \times 10^{+08}$	$1.1604 \times 10^{+05}$	$5.8993 \times 10^{+08}$	$1.4205 \times 10^{+10}$	$1.5594 \times 10^{+10}$	$4.3512 \times 10^{+08}$	$1.7267 \times 10^{+10}$
	Rank	1	2	6	4	3	7	8	5	9
F2	Mean	$3.2597 \times 10^{+16}$	$3.1168 \times 10^{+20}$	$7.7651 \times 10^{+40}$	$4.4781 \times 10^{+38}$	$4.0970 \times 10^{+34}$	$3.7841 \times 10^{+37}$	$5.5771 \times 10^{+34}$	$4.0208 \times 10^{+31}$	$6.2372 \times 10^{+34}$
	S.D	$1.4430 \times 10^{+17}$	$1.6332 \times 10^{+21}$	$2.3723 \times 10^{+41}$	$2.3261 \times 10^{+39}$	$2.2182 \times 10^{+35}$	$1.1458 \times 10^{+38}$	$5.9430 \times 10^{+34}$	$1.8976 \times 10^{+32}$	$1.6308 \times 10^{+35}$
	Best	$6.8586 \times 10^{+08}$	$1.1816 \times 10^{+15}$	$3.3609 \times 10^{+26}$	$1.9980 \times 10^{+24}$	$1.3264 \times 10^{+25}$	$1.0354 \times 10^{+32}$	$1.3585 \times 10^{+31}$	$9.4065 \times 10^{+17}$	$2.5089 \times 10^{+28}$
	Rank	1	2	9	8	4	7	5	3	6
F3	Mean	$3.5133 \times 10^{+04}$	$1.1406 \times 10^{+05}$	$1.4559 \times 10^{+05}$	$2.8678 \times 10^{+05}$	$2.6799 \times 10^{+05}$	$7.2540 \times 10^{+04}$	$1.2867 \times 10^{+05}$	$5.3046 \times 10^{+04}$	$9.0166 \times 10^{+04}$
	S.D	$7.7838 \times 10^{+03}$	$3.1471 \times 10^{+04}$	$6.7237 \times 10^{+04}$	$9.3639 \times 10^{+04}$	$6.5367 \times 10^{+04}$	$1.1034 \times 10^{+04}$	$2.5446 \times 10^{+04}$	$9.2375 \times 10^{+03}$	$1.1815 \times 10^{+04}$
	Best	$2.1456 \times 10^{+04}$	$5.1345 \times 10^{+04}$	$4.8397 \times 10^{+04}$	$1.3216 \times 10^{+05}$	$1.6389 \times 10^{+05}$	$4.9470 \times 10^{+04}$	$9.1851 \times 10^{+04}$	$3.5006 \times 10^{+04}$	$6.2633 \times 10^{+04}$
	Rank	1	5	6	9	8	3	7	2	4
F4	Mean	$4.8025 \times 10^{+02}$	$5.3420 \times 10^{+02}$	$1.2661 \times 10^{+03}$	$6.9072 \times 10^{+02}$	$8.3609 \times 10^{+02}$	$2.3814 \times 10^{+03}$	$3.0966 \times 10^{+03}$	$6.2332 \times 10^{+02}$	$6.6924 \times 10^{+03}$
	S.D	$2.4508 \times 10^{+01}$	$3.6579 \times 10^{+01}$	$8.2219 \times 10^{+02}$	$2.4902 \times 10^{+02}$	$1.8324 \times 10^{+02}$	$6.2395 \times 10^{+02}$	$1.6125 \times 10^{+03}$	$7.6240 \times 10^{+01}$	$3.8537 \times 10^{+03}$
	Best	$4.2153 \times 10^{+02}$	$4.7811 \times 10^{+02}$	$5.9131 \times 10^{+02}$	$4.7530 \times 10^{+02}$	$5.6490 \times 10^{+02}$	$1.5935 \times 10^{+03}$	$1.6294 \times 10^{+03}$	$5.1902 \times 10^{+02}$	$1.8798 \times 10^{+03}$
	Rank	1	3	5	4	5	7	8	2	9
F5	Mean	$6.2748 \times 10^{+02}$	$6.7506 \times 10^{+02}$	$7.7137 \times 10^{+02}$	$7.0768 \times 10^{+02}$	$8.2436 \times 10^{+02}$	$8.1259 \times 10^{+02}$	$8.4089 \times 10^{+02}$	$6.1896 \times 10^{+02}$	$8.3368 \times 10^{+02}$
	S.D	$2.8065 \times 10^{+01}$	$4.2444 \times 10^{+01}$	$4.0632 \times 10^{+01}$	$6.4519 \times 10^{+01}$	$5.0080 \times 10^{+01}$	$1.7890 \times 10^{+01}$	$2.3815 \times 10^{+01}$	$3.7013 \times 10^{+01}$	$3.4435 \times 10^{+01}$
	Best	$5.7272 \times 10^{+02}$	$5.9728 \times 10^{+02}$	$6.9902 \times 10^{+02}$	$5.5651 \times 10^{+02}$	$7.2355 \times 10^{+02}$	$7.7859 \times 10^{+02}$	$7.9316 \times 10^{+02}$	$5.7676 \times 10^{+02}$	$7.5516 \times 10^{+02}$
	Rank	2	3	5	4	7	6	9	1	8
F6	Mean	$6.0437 \times 10^{+02}$	$6.3732 \times 10^{+02}$	$6.4583 \times 10^{+02}$	$6.3622 \times 10^{+02}$	$6.7947 \times 10^{+02}$	$6.6093 \times 10^{+02}$	$6.7676 \times 10^{+02}$	$6.1108 \times 10^{+02}$	$6.7170 \times 10^{+02}$
	S.D	$3.3460 \times 10^{+00}$	$9.4945 \times 10^{+00}$	$1.3812 \times 10^{+01}$	$1.4159 \times 10^{+01}$	$1.1513 \times 10^{+01}$	$8.1954 \times 10^{+00}$	$4.0893 \times 10^{+00}$	$3.8978 \times 10^{+00}$	$7.0330 \times 10^{+00}$
	Best	$6.0044 \times 10^{+02}$	$6.1964 \times 10^{+02}$	$6.2539 \times 10^{+02}$	$6.1065 \times 10^{+02}$	$6.5740 \times 10^{+02}$	$6.3950 \times 10^{+02}$	$6.6739 \times 10^{+02}$	$6.0375 \times 10^{+02}$	$6.5509 \times 10^{+02}$
	Rank	1	4	5	3	9	6	8	2	7
F7	Mean	$9.1901 \times 10^{+02}$	$9.2109 \times 10^{+02}$	$1.0923 \times 10^{+03}$	$1.2915 \times 10^{+03}$	$1.3030 \times 10^{+03}$	$1.2140 \times 10^{+03}$	$1.2523 \times 10^{+03}$	$8.9029 \times 10^{+02}$	$1.2618 \times 10^{+03}$
	S.D	$5.5400 \times 10^{+01}$	$5.8953 \times 10^{+01}$	$5.5427 \times 10^{+01}$	$2.8304 \times 10^{+02}$	$1.0158 \times 10^{+02}$	$3.6266 \times 10^{+01}$	$4.0721 \times 10^{+01}$	$4.7196 \times 10^{+01}$	$4.4554 \times 10^{+01}$
	Best	$8.2121 \times 10^{+02}$	$8.2913 \times 10^{+02}$	$9.8539 \times 10^{+02}$	$9.6088 \times 10^{+02}$	$1.0884 \times 10^{+03}$	$1.1396 \times 10^{+03}$	$1.1869 \times 10^{+03}$	$8.3976 \times 10^{+02}$	$1.1679 \times 10^{+03}$
	Rank	2	3	4	8	9	5	6	1	7
F8	Mean	$9.1677 \times 10^{+02}$	$9.4171 \times 10^{+02}$	$1.0393 \times 10^{+03}$	$9.9407 \times 10^{+02}$	$1.0486 \times 10^{+03}$	$1.0787 \times 10^{+03}$	$1.0811 \times 10^{+03}$	$9.0647 \times 10^{+02}$	$1.0828 \times 10^{+03}$
	S.D	$3.3583 \times 10^{+01}$	$3.6845 \times 10^{+01}$	$4.5076 \times 10^{+01}$	$6.3144 \times 10^{+01}$	$5.1836 \times 10^{+01}$	$2.2188 \times 10^{+01}$	$2.1102 \times 10^{+01}$	$2.5680 \times 10^{+01}$	$2.3501 \times 10^{+01}$
	Best	$8.6992 \times 10^{+02}$	$8.7869 \times 10^{+02}$	$9.7723 \times 10^{+02}$	$8.8824 \times 10^{+02}$	$9.6004 \times 10^{+02}$	$1.0434 \times 10^{+03}$	$1.0307 \times 10^{+03}$	$8.6384 \times 10^{+02}$	$1.0258 \times 10^{+03}$
	Rank	2	4	5	3	6	7	8	1	9

Table 3. Cont.

Fun	Item	IPFA	PFA	PSO	DE	WOA	SCA	SOA	GWO	ChOA
F9	Mean	$3.8547 \times 10^{+03}$	$3.0937 \times 10^{+03}$	$5.8847 \times 10^{+03}$	$8.6777 \times 10^{+03}$	$1.0007 \times 10^{+04}$	$7.7026 \times 10^{+03}$	$9.1212 \times 10^{+03}$	$2.4220 \times 10^{+03}$	$8.4320 \times 10^{+03}$
	S.D	$1.2123 \times 10^{+03}$	$1.2522 \times 10^{+03}$	$2.6928 \times 10^{+03}$	$4.1457 \times 10^{+03}$	$2.9150 \times 10^{+03}$	$1.1743 \times 10^{+03}$	$1.1518 \times 10^{+03}$	$1.2459 \times 10^{+03}$	$1.5339 \times 10^{+03}$
	Best	$1.2662 \times 10^{+03}$	$1.1054 \times 10^{+03}$	$2.9701 \times 10^{+03}$	$3.0616 \times 10^{+03}$	$5.1769 \times 10^{+03}$	$5.5565 \times 10^{+03}$	$6.8441 \times 10^{+03}$	$1.0207 \times 10^{+03}$	$5.6489 \times 10^{+03}$
	Rank	3	2	4	8	9	5	7	1	6
F10	Mean	$4.3076 \times 10^{+03}$	$5.5265 \times 10^{+03}$	$7.2979 \times 10^{+03}$	$7.6795 \times 10^{+03}$	$7.0343 \times 10^{+03}$	$8.7115 \times 10^{+03}$	$7.5729 \times 10^{+03}$	$8.6346 \times 10^{+03}$	$8.6540 \times 10^{+03}$
	S.D	$6.5432 \times 10^{+02}$	$8.6930 \times 10^{+02}$	$7.6825 \times 10^{+02}$	$1.1535 \times 10^{+03}$	$7.4749 \times 10^{+02}$	$4.0612 \times 10^{+02}$	$4.7822 \times 10^{+02}$	$2.6047 \times 10^{+02}$	$2.2182 \times 10^{+02}$
	Best	$3.1236 \times 10^{+03}$	$3.9800 \times 10^{+03}$	$5.4293 \times 10^{+03}$	$5.0911 \times 10^{+03}$	$5.6275 \times 10^{+03}$	$7.6354 \times 10^{+03}$	$6.4401 \times 10^{+03}$	$7.5530 \times 10^{+03}$	$8.2531 \times 10^{+03}$
	Rank	1	2	5	7	4	9	6	3	8
F11	Mean	$1.2018 \times 10^{+03}$	$1.4173 \times 10^{+03}$	$4.6497 \times 10^{+03}$	$4.9019 \times 10^{+03}$	$7.4108 \times 10^{+03}$	$3.2053 \times 10^{+03}$	$4.4522 \times 10^{+03}$	$2.1514 \times 10^{+03}$	$4.5684 \times 10^{+03}$
	S.D	$2.7703 \times 10^{+01}$	$9.0032 \times 10^{+01}$	$7.4643 \times 10^{+03}$	$5.7398 \times 10^{+03}$	$2.5008 \times 10^{+03}$	$9.3077 \times 10^{+02}$	$8.4370 \times 10^{+02}$	$8.8683 \times 10^{+02}$	$1.1343 \times 10^{+03}$
	Best	$1.1536 \times 10^{+03}$	$1.2730 \times 10^{+03}$	$1.5958 \times 10^{+03}$	$1.2967 \times 10^{+03}$	$2.5724 \times 10^{+03}$	$2.1251 \times 10^{+03}$	$2.5780 \times 10^{+03}$	$1.3184 \times 10^{+03}$	$2.7426 \times 10^{+03}$
	Rank	1	2	7	8	9	4	5	3	6
F12	Mean	$8.2985 \times 10^{+06}$	$8.4177 \times 10^{+06}$	$6.1703 \times 10^{+08}$	$6.2965 \times 10^{+07}$	$2.3812 \times 10^{+08}$	$2.1908 \times 10^{+09}$	$2.2938 \times 10^{+09}$	$9.8263 \times 10^{+07}$	$6.8252 \times 10^{+09}$
	S.D	$1.1901 \times 10^{+07}$	$7.0883 \times 10^{+06}$	$9.0651 \times 10^{+08}$	$1.7005 \times 10^{+08}$	$1.5908 \times 10^{+08}$	$5.3778 \times 10^{+08}$	$1.0459 \times 10^{+09}$	$8.3703 \times 10^{+07}$	$3.2651 \times 10^{+09}$
	Best	$4.9157 \times 10^{+05}$	$5.7449 \times 10^{+05}$	$3.4099 \times 10^{+07}$	$4.1067 \times 10^{+05}$	$3.0842 \times 10^{+07}$	$1.0328 \times 10^{+09}$	$6.0040 \times 10^{+08}$	$5.0473 \times 10^{+06}$	$2.7086 \times 10^{+09}$
	Rank	1	2	9	3	5	6	7	4	8
F13	Mean	$4.2649 \times 10^{+04}$	$1.3986 \times 10^{+05}$	$6.7844 \times 10^{+08}$	$6.5946 \times 10^{+07}$	$2.1645 \times 10^{+06}$	$8.0275 \times 10^{+08}$	$4.0367 \times 10^{+08}$	$2.6393 \times 10^{+07}$	$5.7390 \times 10^{+09}$
	S.D	$2.3305 \times 10^{+04}$	$8.9848 \times 10^{+04}$	$1.4467 \times 10^{+09}$	$3.2724 \times 10^{+08}$	$1.6684 \times 10^{+06}$	$2.8392 \times 10^{+08}$	$7.6943 \times 10^{+08}$	$6.9637 \times 10^{+07}$	$4.7275 \times 10^{+09}$
	Best	$4.5744 \times 10^{+03}$	$2.4576 \times 10^{+04}$	$7.7534 \times 10^{+05}$	$7.8561 \times 10^{+03}$	$4.0791 \times 10^{+05}$	$3.8632 \times 10^{+08}$	$2.7978 \times 10^{+07}$	$2.6701 \times 10^{+04}$	$1.2994 \times 10^{+08}$
	Rank	1	2	7	5	3	8	6	4	9
F14	Mean	$8.2788 \times 10^{+04}$	$8.5509 \times 10^{+04}$	$9.7754 \times 10^{+05}$	$4.0456 \times 10^{+05}$	$1.9903 \times 10^{+06}$	$4.2828 \times 10^{+05}$	$4.4783 \times 10^{+05}$	$5.5038 \times 10^{+05}$	$1.2593 \times 10^{+06}$
	S.D	$1.4833 \times 10^{+05}$	$6.3780 \times 10^{+04}$	$3.4757 \times 10^{+06}$	$7.1797 \times 10^{+05}$	$2.4778 \times 10^{+06}$	$3.0649 \times 10^{+05}$	$4.9449 \times 10^{+05}$	$7.6649 \times 10^{+05}$	$1.7075 \times 10^{+06}$
	Best	$2.2287 \times 10^{+03}$	$3.3063 \times 10^{+03}$	$1.7826 \times 10^{+04}$	$4.7497 \times 10^{+03}$	$4.2143 \times 10^{+04}$	$3.9759 \times 10^{+04}$	$5.5357 \times 10^{+04}$	$3.0602 \times 10^{+04}$	$1.1296 \times 10^{+05}$
	Rank	1	2	7	3	9	4	5	6	8
F15	Mean	$2.2402 \times 10^{+04}$	$5.8515 \times 10^{+04}$	$8.1276 \times 10^{+05}$	$3.1002 \times 10^{+07}$	$2.3077 \times 10^{+06}$	$5.0355 \times 10^{+07}$	$7.8371 \times 10^{+06}$	$2.4859 \times 10^{+06}$	$9.5456 \times 10^{+07}$
	S.D	$2.3225 \times 10^{+04}$	$4.9457 \times 10^{+04}$	$7.6847 \times 10^{+05}$	$1.6475 \times 10^{+08}$	$3.9023 \times 10^{+06}$	$2.9236 \times 10^{+07}$	$1.3077 \times 10^{+07}$	$1.0367 \times 10^{+07}$	$4.0691 \times 10^{+08}$
	Best	$2.7296 \times 10^{+03}$	$8.6259 \times 10^{+03}$	$7.3810 \times 10^{+04}$	$1.8291 \times 10^{+03}$	$5.2271 \times 10^{+04}$	$7.4482 \times 10^{+06}$	$2.5170 \times 10^{+05}$	$1.6488 \times 10^{+04}$	$1.8609 \times 10^{+06}$
	Rank	1	2	3	7	4	8	6	5	9
F16	Mean	$2.7121 \times 10^{+03}$	$3.0447 \times 10^{+03}$	$3.6287 \times 10^{+03}$	$3.5558 \times 10^{+03}$	$4.1014 \times 10^{+03}$	$4.0332 \times 10^{+03}$	$3.8178 \times 10^{+03}$	$4.1852 \times 10^{+03}$	$4.1105 \times 10^{+03}$
	S.D	$2.7593 \times 10^{+02}$	$4.0471 \times 10^{+02}$	$5.0065 \times 10^{+02}$	$5.6068 \times 10^{+02}$	$5.6221 \times 10^{+02}$	$2.3612 \times 10^{+02}$	$3.0471 \times 10^{+02}$	$3.5916 \times 10^{+02}$	$3.7665 \times 10^{+02}$
	Best	$2.0122 \times 10^{+03}$	$2.3213 \times 10^{+03}$	$2.9486 \times 10^{+03}$	$1.9482 \times 10^{+03}$	$3.2065 \times 10^{+03}$	$3.5054 \times 10^{+03}$	$3.2284 \times 10^{+03}$	$3.6689 \times 10^{+03}$	$3.5176 \times 10^{+03}$
	Rank	1	2	4	3	7	6	5	9	8

Table 3. Cont.

Fun	Item	IPFA	PFA	PSO	DE	WOA	SCA	SOA	GWO	ChOA
F17	Mean	$2.1407 \times 10^{+03}$	$2.4179 \times 10^{+03}$	$2.5769 \times 10^{+03}$	$2.5874 \times 10^{+03}$	$2.6818 \times 10^{+03}$	$2.6567 \times 10^{+03}$	$2.6308 \times 10^{+03}$	$2.0917 \times 10^{+03}$	$3.2590 \times 10^{+03}$
	S.D	$1.9680 \times 10^{+02}$	$2.5815 \times 10^{+02}$	$3.4084 \times 10^{+02}$	$3.3583 \times 10^{+02}$	$2.6156 \times 10^{+02}$	$1.8352 \times 10^{+02}$	$1.6227 \times 10^{+02}$	$1.9005 \times 10^{+02}$	$1.3088 \times 10^{+03}$
	Best	$1.7951 \times 10^{+03}$	$2.0814 \times 10^{+03}$	$1.9739 \times 10^{+03}$	$1.8476 \times 10^{+03}$	$2.1446 \times 10^{+03}$	$2.3263 \times 10^{+03}$	$2.3473 \times 10^{+03}$	$1.8437 \times 10^{+03}$	$2.4275 \times 10^{+03}$
	Rank	2	3	4	5	8	7	6	1	9
F18	Mean	$1.0108 \times 10^{+06}$	$1.6471 \times 10^{+06}$	$2.1618 \times 10^{+07}$	$1.7515 \times 10^{+07}$	$1.0735 \times 10^{+07}$	$9.0368 \times 10^{+06}$	$1.5038 \times 10^{+06}$	$1.3408 \times 10^{+06}$	$3.7312 \times 10^{+06}$
	S.D	$9.0803 \times 10^{+05}$	$2.3211 \times 10^{+06}$	$7.9669 \times 10^{+07}$	$3.3616 \times 10^{+07}$	$9.2493 \times 10^{+06}$	$4.3702 \times 10^{+06}$	$8.1194 \times 10^{+05}$	$1.6433 \times 10^{+06}$	$1.9271 \times 10^{+06}$
	Best	$1.2574 \times 10^{+05}$	$7.6547 \times 10^{+04}$	$3.9099 \times 10^{+05}$	$1.9034 \times 10^{+05}$	$2.2865 \times 10^{+05}$	$3.8986 \times 10^{+06}$	$3.5949 \times 10^{+05}$	$1.4792 \times 10^{+05}$	$6.0237 \times 10^{+05}$
	Rank	1	4	9	8	7	6	3	2	5
F19	Mean	$1.0245 \times 10^{+04}$	$2.8419 \times 10^{+05}$	$1.1608 \times 10^{+07}$	$2.5108 \times 10^{+06}$	$1.4933 \times 10^{+07}$	$7.2644 \times 10^{+07}$	$3.1410 \times 10^{+07}$	$8.0087 \times 10^{+05}$	$6.2082 \times 10^{+08}$
	S.D	$9.6215 \times 10^{+03}$	$2.8579 \times 10^{+05}$	$3.9655 \times 10^{+07}$	$1.2544 \times 10^{+07}$	$1.6643 \times 10^{+07}$	$4.5453 \times 10^{+07}$	$4.4058 \times 10^{+07}$	$1.1964 \times 10^{+06}$	$8.2664 \times 10^{+08}$
	Best	$2.1822 \times 10^{+03}$	$3.2526 \times 10^{+04}$	$1.0743 \times 10^{+05}$	$2.5511 \times 10^{+03}$	$1.5216 \times 10^{+05}$	$2.8965 \times 10^{+06}$	$1.7347 \times 10^{+06}$	$1.3678 \times 10^{+04}$	$2.4740 \times 10^{+07}$
	Rank	1	2	5	4	6	9	7	3	8
F20	Mean	$2.5016 \times 10^{+03}$	$2.6103 \times 10^{+03}$	$2.7478 \times 10^{+03}$	$3.0946 \times 10^{+03}$	$2.9299 \times 10^{+03}$	$2.8502 \times 10^{+03}$	$2.7753 \times 10^{+03}$	$2.4158 \times 10^{+03}$	$3.1307 \times 10^{+03}$
	S.D	$1.8747 \times 10^{+02}$	$1.8793 \times 10^{+02}$	$2.2633 \times 10^{+02}$	$3.3156 \times 10^{+02}$	$2.5956 \times 10^{+02}$	$1.3199 \times 10^{+02}$	$1.6163 \times 10^{+02}$	$1.0027 \times 10^{+02}$	$1.5944 \times 10^{+02}$
	Best	$2.2101 \times 10^{+03}$	$2.2547 \times 10^{+03}$	$2.2403 \times 10^{+03}$	$2.5342 \times 10^{+03}$	$2.4882 \times 10^{+03}$	$2.4937 \times 10^{+03}$	$2.5064 \times 10^{+03}$	$2.2629 \times 10^{+03}$	$2.7606 \times 10^{+03}$
	Rank	2	3	4	8	7	6	5	1	9
F21	Mean	$2.4544 \times 10^{+03}$	$2.4682 \times 10^{+03}$	$2.5575 \times 10^{+03}$	$2.4834 \times 10^{+03}$	$2.6184 \times 10^{+03}$	$2.5878 \times 10^{+03}$	$2.6097 \times 10^{+03}$	$2.4077 \times 10^{+03}$	$2.5975 \times 10^{+03}$
	S.D	$4.2970 \times 10^{+01}$	$4.5204 \times 10^{+01}$	$4.3387 \times 10^{+01}$	$6.5060 \times 10^{+01}$	$7.6117 \times 10^{+01}$	$2.2646 \times 10^{+01}$	$2.4117 \times 10^{+01}$	$3.0143 \times 10^{+01}$	$3.4120 \times 10^{+01}$
	Best	$2.3794 \times 10^{+03}$	$2.3771 \times 10^{+03}$	$2.4707 \times 10^{+03}$	$2.3980 \times 10^{+03}$	$2.5022 \times 10^{+03}$	$2.5443 \times 10^{+03}$	$2.5494 \times 10^{+03}$	$2.3563 \times 10^{+03}$	$2.4993 \times 10^{+03}$
	Rank	2	3	5	4	9	6	8	1	7
F22	Mean	$5.4008 \times 10^{+03}$	$3.6862 \times 10^{+03}$	$6.5564 \times 10^{+03}$	$8.6375 \times 10^{+03}$	$7.5657 \times 10^{+03}$	$9.8280 \times 10^{+03}$	$6.0989 \times 10^{+03}$	$4.5680 \times 10^{+03}$	$9.9134 \times 10^{+03}$
	S.D	$1.6378 \times 10^{+03}$	$2.1666 \times 10^{+03}$	$2.8172 \times 10^{+03}$	$1.9543 \times 10^{+03}$	$1.9040 \times 10^{+03}$	$1.2932 \times 10^{+03}$	$1.6862 \times 10^{+03}$	$1.7447 \times 10^{+03}$	$2.2289 \times 10^{+02}$
	Best	$2.3042 \times 10^{+03}$	$2.3059 \times 10^{+03}$	$2.6614 \times 10^{+03}$	$2.8120 \times 10^{+03}$	$2.7455 \times 10^{+03}$	$4.4365 \times 10^{+03}$	$4.0314 \times 10^{+03}$	$2.3699 \times 10^{+03}$	$9.2662 \times 10^{+03}$
	Rank	3	2	5	7	6	8	4	1	9
F23	Mean	$2.8632 \times 10^{+03}$	$2.8515 \times 10^{+03}$	$3.2755 \times 10^{+03}$	$2.8440 \times 10^{+03}$	$3.0942 \times 10^{+03}$	$3.0547 \times 10^{+03}$	$3.0277 \times 10^{+03}$	$2.7709 \times 10^{+03}$	$3.1096 \times 10^{+03}$
	S.D	$6.4030 \times 10^{+01}$	$4.8214 \times 10^{+01}$	$2.1503 \times 10^{+02}$	$7.1286 \times 10^{+01}$	$9.4794 \times 10^{+01}$	$4.0942 \times 10^{+01}$	$2.4572 \times 10^{+01}$	$4.0442 \times 10^{+01}$	$4.9032 \times 10^{+01}$
	Best	$2.7421 \times 10^{+03}$	$2.7367 \times 10^{+03}$	$2.9140 \times 10^{+03}$	$2.7323 \times 10^{+03}$	$2.9213 \times 10^{+03}$	$2.9592 \times 10^{+03}$	$2.9833 \times 10^{+03}$	$2.7343 \times 10^{+03}$	$3.0236 \times 10^{+03}$
	Rank	4	3	9	2	7	6	5	1	8
F24	Mean	$3.1439 \times 10^{+03}$	$2.9964 \times 10^{+03}$	$3.6837 \times 10^{+03}$	$3.0314 \times 10^{+03}$	$3.2140 \times 10^{+03}$	$3.2203 \times 10^{+03}$	$3.2186 \times 10^{+03}$	$2.9305 \times 10^{+03}$	$3.3219 \times 10^{+03}$
	S.D	$8.6642 \times 10^{+01}$	$5.2673 \times 10^{+01}$	$2.2246 \times 10^{+02}$	$6.9814 \times 10^{+01}$	$1.0941 \times 10^{+02}$	$3.1077 \times 10^{+01}$	$2.5481 \times 10^{+01}$	$4.9094 \times 10^{+01}$	$4.6278 \times 10^{+01}$
	Best	$3.0059 \times 10^{+03}$	$2.9051 \times 10^{+03}$	$3.3540 \times 10^{+03}$	$2.9251 \times 10^{+03}$	$3.0350 \times 10^{+03}$	$3.1723 \times 10^{+03}$	$3.1726 \times 10^{+03}$	$2.8725 \times 10^{+03}$	$3.2491 \times 10^{+03}$
	Rank	4	2	9	3	5	7	6	1	8

Table 3. Cont.

Fun	Item	IPFA	PFA	PSO	DE	WOA	SCA	SOA	GWO	ChOA
F25	Mean	$2.8906 \times 10^{+03}$	$2.9307 \times 10^{+03}$	$3.1579 \times 10^{+03}$	$3.0033 \times 10^{+03}$	$3.1069 \times 10^{+03}$	$3.3901 \times 10^{+03}$	$3.6917 \times 10^{+03}$	$3.0091 \times 10^{+03}$	$4.4745 \times 10^{+03}$
	S.D	$1.1971 \times 10^{+01}$	$2.2101 \times 10^{+01}$	$9.4472 \times 10^{+01}$	$1.4912 \times 10^{+02}$	$5.4413 \times 10^{+01}$	$1.5695 \times 10^{+02}$	$3.0194 \times 10^{+02}$	$7.1573 \times 10^{+01}$	$5.4389 \times 10^{+02}$
	Best	$2.8813 \times 10^{+03}$	$2.8886 \times 10^{+03}$	$3.0194 \times 10^{+03}$	$2.8926 \times 10^{+03}$	$3.0310 \times 10^{+03}$	$3.1806 \times 10^{+03}$	$3.3264 \times 10^{+03}$	$2.9357 \times 10^{+03}$	$3.4101 \times 10^{+03}$
	Rank	1	2	6	3	5	7	8	4	9
F26	Mean	$5.5592 \times 10^{+03}$	$5.4684 \times 10^{+03}$	$7.2510 \times 10^{+03}$	$5.7929 \times 10^{+03}$	$8.4663 \times 10^{+03}$	$7.6331 \times 10^{+03}$	$6.8978 \times 10^{+03}$	$4.9280 \times 10^{+03}$	$7.2914 \times 10^{+03}$
	S.D	$8.9380 \times 10^{+02}$	$8.1585 \times 10^{+02}$	$1.5604 \times 10^{+03}$	$6.8537 \times 10^{+02}$	$8.3105 \times 10^{+02}$	$3.8632 \times 10^{+02}$	$2.3730 \times 10^{+02}$	$4.6748 \times 10^{+02}$	$4.9225 \times 10^{+02}$
	Best	$2.8133 \times 10^{+03}$	$3.1385 \times 10^{+03}$	$3.8296 \times 10^{+03}$	$4.5415 \times 10^{+03}$	$6.7174 \times 10^{+03}$	$6.9848 \times 10^{+03}$	$6.1900 \times 10^{+03}$	$4.1351 \times 10^{+03}$	$6.3202 \times 10^{+03}$
	Rank	3	2	6	4	9	8	5	1	7
F27	Mean	$3.2000 \times 10^{+03}$	$3.3003 \times 10^{+03}$	$3.6542 \times 10^{+03}$	$3.2632 \times 10^{+03}$	$3.4134 \times 10^{+03}$	$3.5118 \times 10^{+03}$	$3.5036 \times 10^{+03}$	$3.2740 \times 10^{+03}$	$3.7410 \times 10^{+03}$
	S.D	3.5746×10^{-04}	$8.4381 \times 10^{+01}$	$3.3626 \times 10^{+02}$	$3.0128 \times 10^{+01}$	$1.2073 \times 10^{+02}$	$7.6528 \times 10^{+01}$	$4.8087 \times 10^{+01}$	$3.5391 \times 10^{+01}$	$1.5108 \times 10^{+02}$
	Best	$3.2000 \times 10^{+03}$	$3.2330 \times 10^{+03}$	$3.2558 \times 10^{+03}$	$3.2174 \times 10^{+03}$	$3.2842 \times 10^{+03}$	$3.3841 \times 10^{+03}$	$3.4055 \times 10^{+03}$	$3.2164 \times 10^{+03}$	$3.4900 \times 10^{+03}$
	Rank	1	3	8	4	5	7	6	2	9
F28	Mean	$3.2934 \times 10^{+03}$	$3.2887 \times 10^{+03}$	$3.9852 \times 10^{+03}$	$3.9408 \times 10^{+03}$	$3.5815 \times 10^{+03}$	$4.2227 \times 10^{+03}$	$3.9834 \times 10^{+03}$	$3.5191 \times 10^{+03}$	$4.9530 \times 10^{+03}$
	S.D	$2.2046 \times 10^{+01}$	$3.6787 \times 10^{+01}$	$1.0041 \times 10^{+03}$	$6.7247 \times 10^{+02}$	$1.3088 \times 10^{+02}$	$2.3651 \times 10^{+02}$	$2.8420 \times 10^{+02}$	$1.8424 \times 10^{+02}$	$7.4931 \times 10^{+02}$
	Best	$3.2119 \times 10^{+03}$	$3.2251 \times 10^{+03}$	$3.3372 \times 10^{+03}$	$3.2399 \times 10^{+03}$	$3.3711 \times 10^{+03}$	$3.8666 \times 10^{+03}$	$3.6869 \times 10^{+03}$	$3.3142 \times 10^{+03}$	$3.7515 \times 10^{+03}$
	Rank	2	1	7	5	4	8	6	3	9
F29	Mean	$3.9505 \times 10^{+03}$	$4.3798 \times 10^{+03}$	$4.8424 \times 10^{+03}$	$4.4459 \times 10^{+03}$	$5.3745 \times 10^{+03}$	$5.0495 \times 10^{+03}$	$4.6276 \times 10^{+03}$	$3.8568 \times 10^{+03}$	$4.8108 \times 10^{+03}$
	S.D	$2.4865 \times 10^{+02}$	$2.6757 \times 10^{+02}$	$7.2174 \times 10^{+02}$	$4.0600 \times 10^{+02}$	$3.9015 \times 10^{+02}$	$3.1166 \times 10^{+02}$	$1.6918 \times 10^{+02}$	$2.1939 \times 10^{+02}$	$2.7074 \times 10^{+02}$
	Best	$3.4629 \times 10^{+03}$	$3.7985 \times 10^{+03}$	$3.9964 \times 10^{+03}$	$3.7717 \times 10^{+03}$	$4.6358 \times 10^{+03}$	$4.3992 \times 10^{+03}$	$4.1722 \times 10^{+03}$	$3.5280 \times 10^{+03}$	$4.3074 \times 10^{+03}$
	Rank	2	3	7	4	9	8	5	1	6
Total Rank		88	142	267	281	244	302	266	108	310
Final Rank		1	3	6	6	7	8	5	2	9

In the evaluation of the CEC2017 functions F1–F23, the IPFA consistently secures the 1st rank across all rotation and displacement functions F1, F2, F3, F4, and F6. This remarkable performance indicates its exceptional capability to rotate the problem and effectively adapt to the translation of the search space. Moreover, the algorithm demonstrates resilience in overcoming geometric changes within the search space, underscoring its robust adaptability. The IPFA algorithm is ranked 1 for all high-dimensional composite functions except for the function F18. This observation suggests that the incorporation of the escape energy from the Harris Hawks optimization in the algorithm achieves a harmonious balance between a global and local search. This allows for rapid convergence to the global optimal solution, preventing entrapment in local optima. Moreover, it underscores the algorithm's proficiency in handling high-dimensional functions. The IPFA achieves the 1st rank on mixed functions F25 and F27. These mixed functions exhibit highly complex shapes with pronounced nonlinearity, lack of smoothness, and irregular characteristics. The utilization of t-distribution-based multi-elite individuals for dimension-wise mutations in the algorithm proves beneficial for escaping the current position and converging toward the local minimum of the functions. This imparts significant adaptability to the algorithm, enabling it to overcome the challenges posed by the intricate nature of these functions. While the algorithm's performance is suboptimal on the remaining eight functions, it notably outperforms other algorithms with the smallest best values on functions F20, F22, F26, F28, and F29. This underscores the algorithm's superior optimization capability in comparison to the considered alternatives.

The IPFA secures the top position across all 45 function species in the search results, showcasing its exceptional global search and local exploration capabilities. The algorithm exhibits robustness and adaptability, further underscoring its prowess.

The convergence curves shown in Figures 4 and 5 indicate that IPFA outperforms the other eight algorithms in terms of global convergence and overall optimized search accuracy. In addition, the boxplots in Figures 6 and 7 illustrate the performance of these algorithms when dealing with test functions. Regarding unimodal functions f_1 – f_7 , the IPFA only falls into local optima for functions f_3 , f_4 , and f_5 , demonstrating a notable advantage in the remaining functions. It rapidly converges toward the global optimum. Regarding multimodal functions f_8 – f_{12} , the IPFA notably lacks the high convergence accuracy observed in the WOA for functions f_8 – f_{10} and experiences local optima. However, it is evident that the IPFA exhibits a faster early-stage convergence, indicating that elite opposition-based learning provides superior initial solutions. These solutions are closer to the global optimum than randomly initialized solutions. Despite all algorithms encountering local optima in functions f_{11} and f_{12} , the IPFA significantly outperforms others in terms of convergence accuracy and speed. For fixed-dimensional multi-peaked functions f_{13} – f_{16} , it can be seen that the PFA algorithm solves such functions more generally. This superiority arises from the characteristic nature of fixed-dimensional multi-peaked functions, which typically encompass numerous locally optimal solutions spaced relatively far apart. The dimension-by-dimension variation strategy employed by the IPFA concentrates solutions near certain local optima, hindering efficient escape and resulting in the algorithm being trapped in local convergence without the ability to explore global optimal solutions. In the realm of CEC2017 functions, the collaboration between the pathfinders and followers in the PFA has consistently exhibited superior optimization accuracy compared to other algorithms. Notably, for functions F7, F20, and F22, the PFA and GWO demonstrate continuous optimization efforts. However, the improved pathfinder algorithm (IPFA) outperforms all other algorithms on the remaining functions. This underscores the algorithm's exceptional global search capability, enabling it to swiftly and efficiently navigate the problem's search space and converge rapidly toward potential global optimal solutions.

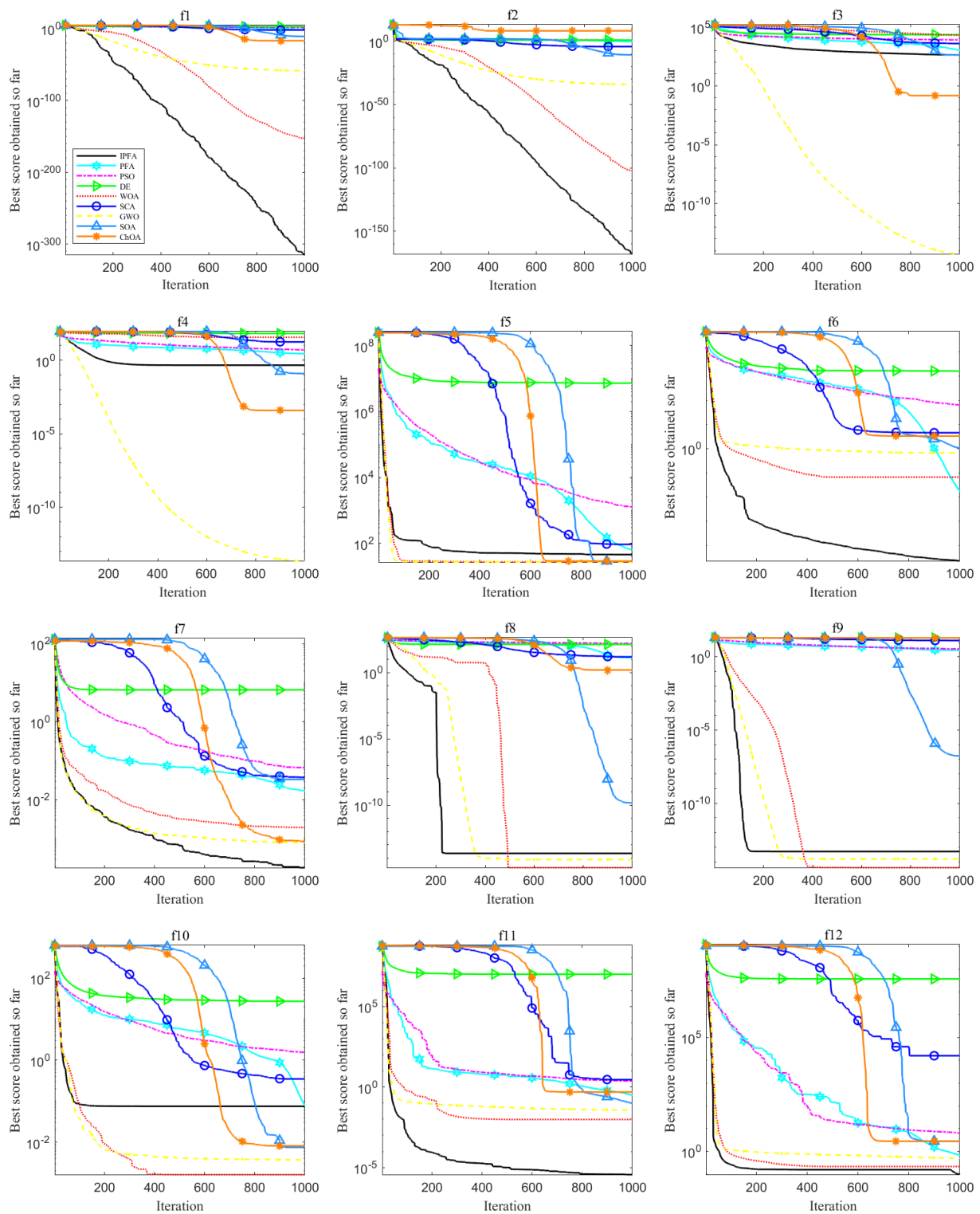


Figure 4. Cont.

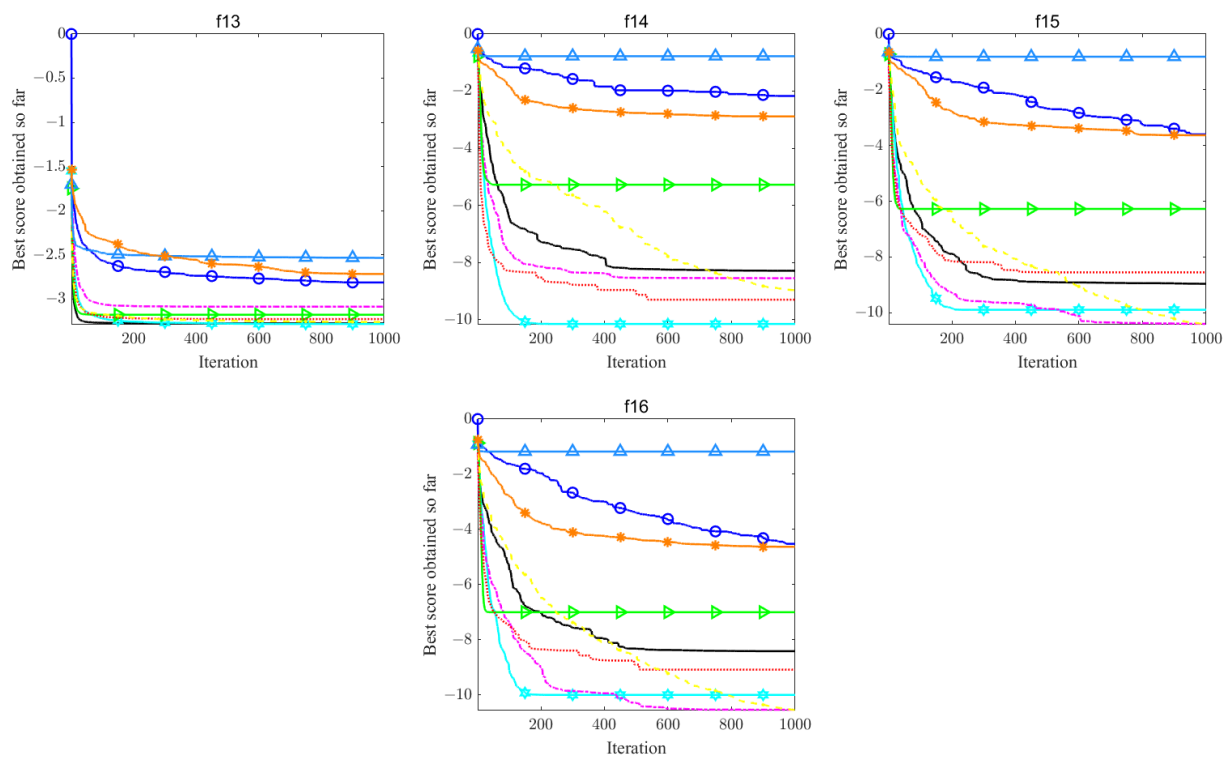


Figure 4. Convergence curves of functions f_1 – f_{16} .

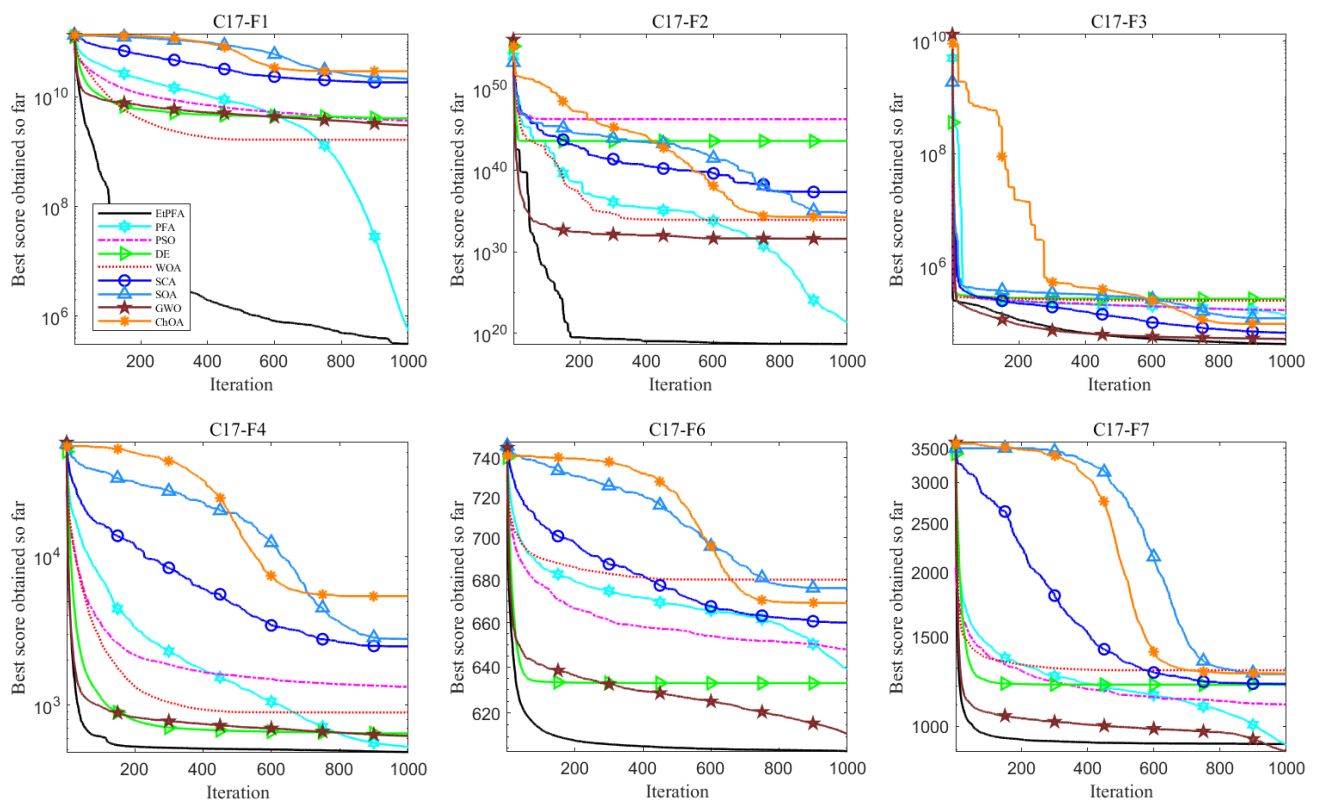


Figure 5. Cont.

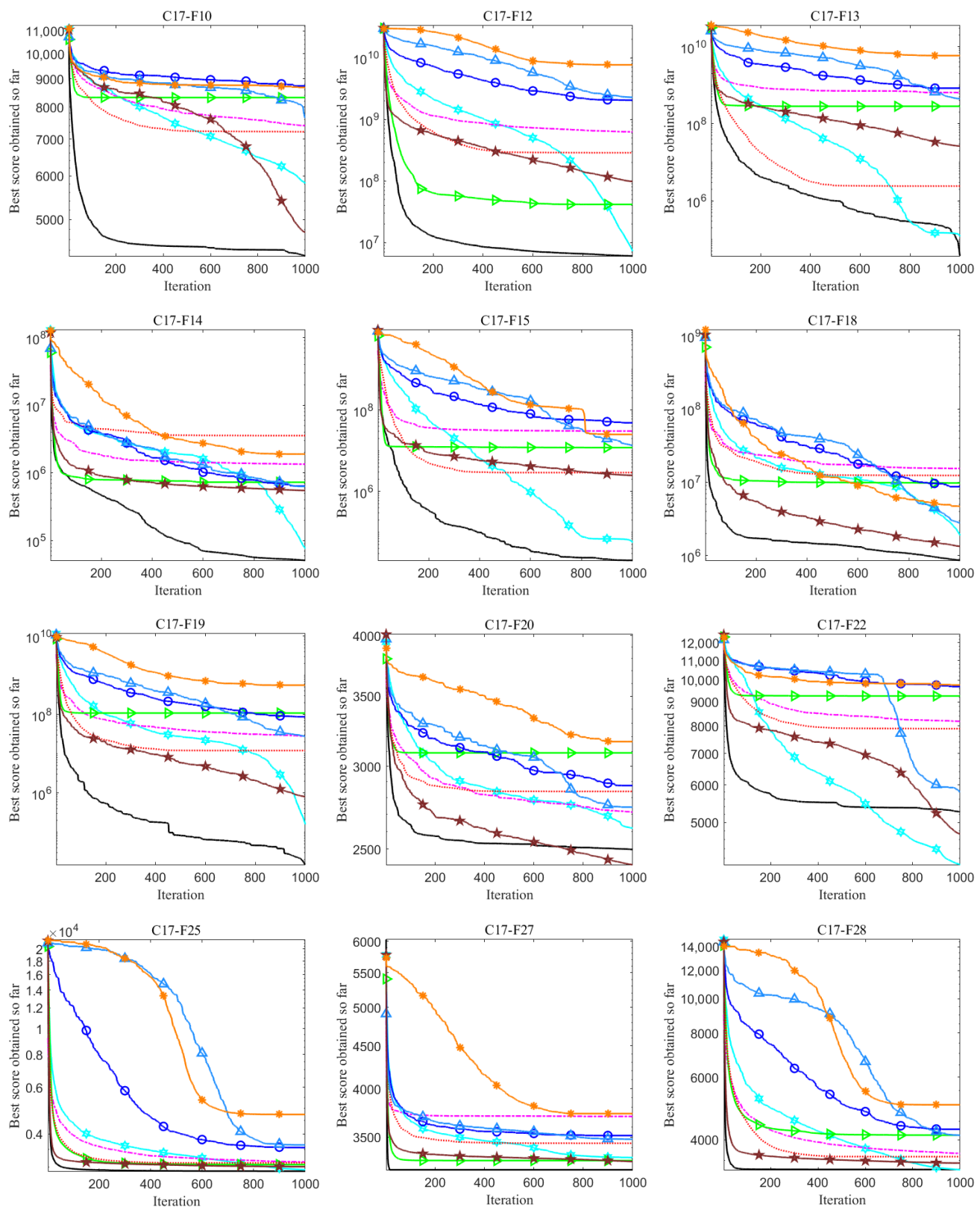


Figure 5. CEC2017 functions: convergence curves (parts).

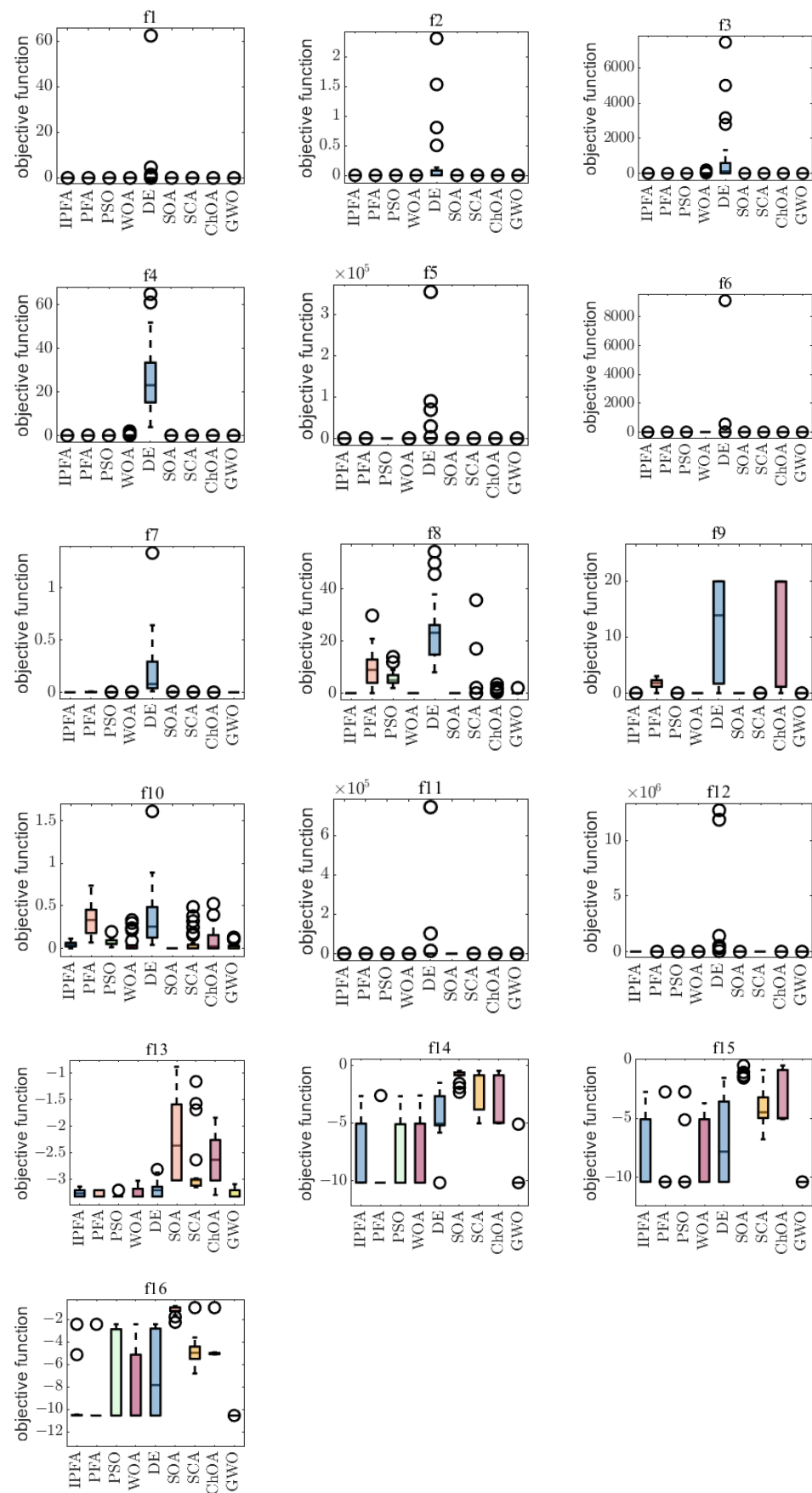


Figure 6. Boxplot of functions f_1 – f_{16} .

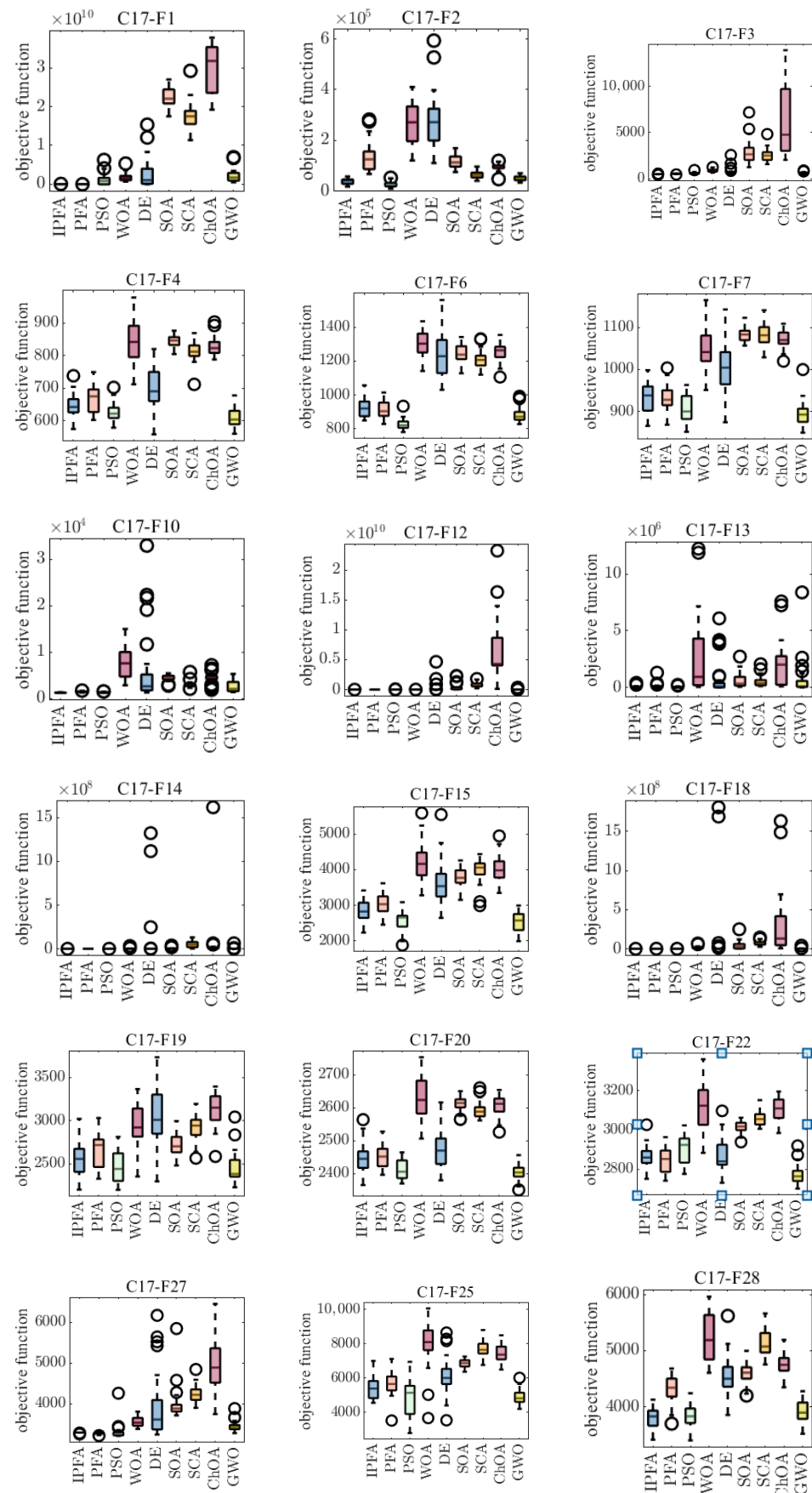


Figure 7. Boxplot of CEC2017 functions (parts).

4.3. Wilcoxon Rank Sum Test

One nonparametric hypothesis testing technique is the Wilcoxon signed-rank sum test (WRST) [38]. Because it refrains from making particular assumptions about the data distribution, it is very appropriate for a variety of intricate comparative data analyses. The Wilcoxon rank sum test (WRST) is a valuable tool for evaluating the convergence performance of multiple algorithms when comparing their performances. The outcomes of the nonparametric tests conducted through WRST are presented in Tables 4 and 5.

Eight distinct algorithms were employed in this analysis, each subjected to a comparative evaluation against the IPFA algorithm. In each experiment, the discrepancies in observations across the algorithms were quantified and ranked, based on their absolute magnitudes, and assigned corresponding ranks. Subsequently, the algorithms were classified, based on the positive and negative scenarios of the rank, with '+' indicating a positive rank sum and '-' indicating a negative rank sum. The p -value was then used to determine if there was a significant difference between the two algorithms. If the p -value exceeds 0.05, it indicates that there is no statistically significant difference between the two algorithms. Simultaneously, the concept of a "winner" was introduced to delineate the algorithm's performance relative to others in diverse situations.

Specifically, the magnitudes of positive and negative rank sums were compared when the p -value was less than 0.05. If the positive rank sum is less than the negative rank sum, it means that an algorithm shows an advantage in convergence performance compared to other algorithms at a statistical significance level of 5%, marking it as '+'; on the contrary, if the positive rank sum is greater than the negative rank sum, it means that the algorithm shows a weaker performance at the significance level, marking it as '-'; the p -value is marked with a '=' if it is larger than 0.05, which means that at the 5% significance level, there is no significant difference between the two algorithms.

In this way, the validity of the proposed IPFA algorithm is verified. Therefore, in this section, nonparametric tests are performed on the means in Table 3 using the Wilcoxon signed-rank sum test, the results of which are presented in Tables 4 and 5, to authoritatively assess the convergence performance of the compared algorithms.

Tables 4 and 5 demonstrate that in 27 of the 45 benchmark functions, the IPFA performs better than the PFA, there is no significant difference in 9 benchmark functions, and there are also 9 benchmark functions that are inferior to the PFA, and in general, the IPFA performs slightly better than the PFA, which indicates that the algorithm's improvement has an effect; compared with the PSO, the performance of the IPFA is inferior to that of PSO except for the functions f_{14} , f_{15} , and f_{16} , and the performance of the IPFA outperforms that of PSO for the rest of the 42 functions. Compared with the DE, except for 10 functions where the performance of the two algorithms is comparable, the DE is inferior to the IPFA in the remaining 35 functions; compared with the SCA, the IPFA outperforms the SCA in 45 functions. Compared to the GWO, IPFA has only slightly better performance; the IPFA has a significant advantage over the WOA, SOA, and ChOA in most of the functions. In conclusion, it is demonstrated that IPFA outperforms other intelligent optimization algorithms in terms of total performance.

Table 4. WRST comparison between IPFA and PFA, PSO, DE, and WOA.

Fun	IPFA vs. PFA				IPFA vs. PSO				IPFA vs. DE				IPFA vs. WOA			
	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner
<i>f</i> ₁	2.3657×10^{-12}	0	465	+	2.3657×10^{-12}	0	465	+	2.3657×10^{-12}	0	465	+	2.3657×10^{-12}	0	465	+
<i>f</i> ₂	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
<i>f</i> ₃	7.6171×10^{-03}	179	286	+	2.1544×10^{-10}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
<i>f</i> ₄	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
<i>f</i> ₅	4.5146×10^{-02}	193	272	+	8.1527×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	5.2014×10^{-01}	182	283	+
<i>f</i> ₆	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
<i>f</i> ₇	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.6351×10^{-05}	38	427	+
<i>f</i> ₈	1.4900×10^{-11}	0	465	+	1.4900×10^{-11}	0	465	+	1.4900×10^{-11}	0	465	+	1.3780×10^{-04}	213	0	+
<i>f</i> ₉	2.5856×10^{-11}	0	465	+	2.5856×10^{-11}	0	465	+	2.5692×10^{-11}	0	465	+	1.3946×10^{-11}	465	0	+
<i>f</i> ₁₀	9.5922×10^{-03}	187	278	+	2.5206×10^{-11}	0	465	+	9.2212×10^{-11}	0	465	+	8.8658×10^{-07}	298	0	+
<i>f</i> ₁₁	1.2057×10^{-10}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	5.5727×10^{-10}	27	438	+
<i>f</i> ₁₂	3.6459×10^{-08}	65	400	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	2.8389×10^{-04}	136	329	+
<i>f</i> ₁₃	8.8669×10^{-04}	335	130	+	1.0277×10^{-06}	84	381	+	1.8047×10^{-01}	239	226	=	2.2539×10^{-04}	141	324	+
<i>f</i> ₁₄	1.4488×10^{-11}	465	0	−	9.8461×10^{-06}	392	73	−	3.7412×10^{-02}	128	337	+	5.5699×10^{-03}	281	184	−
<i>f</i> ₁₅	4.0806×10^{-12}	465	0	−	1.6617×10^{-04}	347	118	−	9.6027×10^{-02}	152	313	=	3.0418×10^{-01}	236	229	=
<i>f</i> ₁₆	1.2551×10^{-07}	431	34	−	3.5149×10^{-08}	434	31	−	3.2402×10^{-01}	268	197	=	3.9881×10^{-04}	335	130	−
F1	6.1001×10^{-01}	153	312	−	3.0199×10^{-11}	0	465	+	7.3803×10^{-10}	20	445	+	3.0199×10^{-11}	0	465	+
F2	2.1544×10^{-10}	6	459	−	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
F3	3.0199×10^{-11}	0	465	−	4.0772×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
F4	7.1186×10^{-09}	15	450	+	3.0199×10^{-11}	0	465	+	1.1737×10^{-09}	44	421	+	3.0199×10^{-11}	0	465	+
F5	8.2919×10^{-06}	106	359	+	3.0199×10^{-11}	0	465	+	1.8731×10^{-07}	61	404	+	3.0199×10^{-11}	0	465	+
F6	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	4.5043×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
F7	9.9410×10^{-01}	239	226	=	9.9186×10^{-11}	0	465	+	8.1527×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
F8	4.5530×10^{-01}	200	265	=	7.3803×10^{-10}	3	462	+	2.3885×10^{-04}	140	325	+	1.1737×10^{-09}	0	465	+
F9	2.4157×10^{-02}	300	165	−	1.1143×10^{-03}	108	357	+	1.1567×10^{-07}	44	421	+	8.1527×10^{-11}	0	465	+
F10	5.5999×10^{-07}	24	441	+	4.0772×10^{-11}	0	465	+	5.4941×10^{-11}	0	465	+	4.5043×10^{-11}	0	465	+
F11	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
F12	1.2597×10^{-01}	193	272	=	4.9752×10^{-11}	0	465	+	1.6238×10^{-01}	203	262	=	4.5043×10^{-11}	0	465	+
F13	1.0666×10^{-07}	57	408	+	3.0199×10^{-11}	0	465	+	1.1199×10^{-01}	120	345	=	3.0199×10^{-11}	0	465	+
F14	8.5338×10^{-01}	259	206	=	3.6709×10^{-03}	158	307	+	3.1830×10^{-01}	137	328	=	6.0104×10^{-08}	57	408	+
F15	2.7726×10^{-05}	63	402	+	3.6897×10^{-11}	0	465	+	4.7335×10^{-01}	205	260	=	4.0772×10^{-11}	0	465	+
F16	1.2362×10^{-03}	97	368	+	1.3289×10^{-10}	0	465	+	5.9673×10^{-09}	12	453	+	3.0199×10^{-11}	0	465	+
F17	4.9426×10^{-05}	112	353	+	1.2860×10^{-06}	73	392	+	7.5991×10^{-07}	103	362	+	2.2273×10^{-09}	24	441	+

Table 4. Cont.

Fun	IPFA vs. PFA				IPFA vs. PSO				IPFA vs. DE				IPFA vs. WOA			
	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner
F18	4.3764×10^{-01}	235	230	=	6.2828×10^{-06}	65	400	+	1.0277×10^{-06}	46	419	+	1.2541×10^{-07}	94	371	+
F19	4.9752×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	9.5207×10^{-04}	127	338	+	3.0199×10^{-11}	0	465	+
F20	1.9883×10^{-02}	91	374	+	6.3560×10^{-05}	86	379	+	2.0338×10^{-09}	6	459	+	6.0104×10^{-08}	69	396	+
F21	2.5805×10^{-01}	212	253	=	8.8910×10^{-10}	0	465	+	1.5367×10^{-01}	207	258	=	1.3289×10^{-10}	23	442	+
F22	1.4945×10^{-01}	301	164	=	6.0971×10^{-03}	149	316	+	1.0105×10^{-08}	74	391	+	1.1077×10^{-06}	66	399	+
F23	8.4180×10^{-01}	128	337	=	1.0937×10^{-10}	0	465	+	1.7613×10^{-01}	330	135	=	1.7769×10^{-10}	0	465	+
F24	4.9980×10^{-09}	436	29	−	3.0199×10^{-11}	0	465	+	2.6784×10^{-06}	383	82	−	1.4412×10^{-02}	196	269	+
F25	1.8567×10^{-09}	6	459	+	3.0199×10^{-11}	0	465	+	5.5727×10^{-10}	5	460	+	3.0199×10^{-11}	0	465	+
F26	8.5338×10^{-01}	258	207	=	1.1937×10^{-06}	81	384	+	3.4029×10^{-01}	202	263	=	4.5043×10^{-11}	0	465	+
F27	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
F28	2.3243×10^{-02}	315	150	−	3.0199×10^{-11}	0	465	+	4.6159×10^{-10}	9	456	+	3.0199×10^{-11}	0	465	+
F29	2.0283×10^{-07}	21	444	+	1.5465×10^{-09}	25	440	+	8.1975×10^{-07}	45	420	+	3.0199×10^{-11}	0	465	+
+ / = / −				27/9/9	42/0/3				35/10/0				42/1/2			

Table 5. WRST comparison between IPFA and SCA, SOA, GWO, and ChOA.

Fun	IPFA vs. SCA				IPFA vs. SOA				IPFA vs. GWO				IPFA vs. ChOA			
	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner
f_1	2.3657×10^{-12}	0	465	+	1.1738×10^{-03}	0	465	+	3.0199×10^{-11}	0	465	+	2.3657×10^{-12}	0	465	+
f_2	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
f_3	1.4110×10^{-09}	26	439	+	7.7272×10^{-02}	323	142	−	3.0199×10^{-11}	465	0	−	3.0199×10^{-11}	465	0	−
f_4	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	465	0	−	3.0199×10^{-11}	0	465	+
f_5	1.7666×10^{-03}	160	305	+	3.0199×10^{-11}	171	294	=	4.6427×10^{-01}	253	212	=	7.7272×10^{-02}	171	294	=
f_6	3.0199×10^{-11}	0	465	+	4.8218×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
f_7	3.0199×10^{-11}	0	465	+	2.5856×10^{-11}	0	465	+	8.4848×10^{-09}	46	419	+	2.8389×10^{-04}	140	325	+
f_8	1.4900×10^{-11}	0	465	+	2.8186×10^{-01}	0	434	+	2.6733×10^{-01}	88	74	=	6.9360×10^{-10}	0	425	+
f_9	2.5856×10^{-11}	0	465	+	1.4643×10^{-10}	0	465	+	4.4427×10^{-12}	465	0	−	2.5856×10^{-11}	0	465	+
f_{10}	3.2146×10^{-03}	97	368	+	3.0199×10^{-11}	284	181	=	1.4809×10^{-07}	354	67	−	2.0266×10^{-02}	298	91	=
f_{11}	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+
f_{12}	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	8.8910×10^{-10}	0	465	+	3.0199×10^{-11}	0	465	+
f_{13}	8.1527×10^{-11}	25	440	+	3.0199×10^{-11}	0	465	+	1.9112×10^{-02}	314	151	−	3.0199×10^{-11}	0	465	+
f_{14}	2.4386×10^{-09}	27	438	+	3.0199×10^{-11}	0	465	+	4.1127×10^{-07}	407	58	−	2.2273×10^{-09}	28	437	+

Table 5. Cont.

Fun	IPFA vs. SCA				IPFA vs. SOA				IPFA vs. GWO				IPFA vs. ChOA				
	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	<i>p</i> -Value	R+	R−	Winner	
f_{15}	2.2273×10^{-09}	9	456	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	465	0	−	1.6947×10^{-09}	0	465	+	
f_{16}	4.2175×10^{-04}	69	396	+	3.0199×10^{-11}	0	465	+	7.3803×10^{-10}	438	27	−	8.6634×10^{-05}	57	408	+	
F1	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	
F2	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.6897×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	
F3	3.3384×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.3703×10^{-03}	87	378	+	3.0199×10^{-11}	0	465	+	
F4	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.3289×10^{-10}	6	459	+	3.0199×10^{-11}	0	465	+	
F5	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.0763×10^{-02}	341	124	−	3.0199×10^{-11}	0	465	+	
F6	3.0199×10^{-11}	0	465	+	3.3384×10^{-11}	0	465	+	3.1967×10^{-09}	27	438	+	3.0199×10^{-11}	0	465	+	
F7	3.0199×10^{-11}	0	465	+	3.3384×10^{-11}	0	465	+	5.0842×10^{-03}	337	128	−	3.0199×10^{-11}	0	465	+	
F8	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.3853×10^{-06}	397	68	−	3.3384×10^{-11}	0	465	+	
F9	7.3891×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.2362×10^{-03}	340	125	−	6.0658×10^{-11}	0	465	+	
F10	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	1.9579×10^{-01}	146	319	=	3.0199×10^{-11}	0	465	+	
F11	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	
F12	3.0199×10^{-11}	0	465	+	1.0907×10^{-05}	0	465	+	1.3289×10^{-10}	33	432	+	3.0199×10^{-11}	0	465	+	
F13	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.4971×10^{-09}	0	465	+	3.0199×10^{-11}	0	465	+	
F14	9.8329×10^{-08}	39	426	+	3.0199×10^{-11}	74	391	+	6.7650×10^{-05}	80	385	+	5.9673×10^{-09}	8	457	+	
F15	3.0199×10^{-11}	0	465	+	7.3803×10^{-10}	0	465	+	7.0881×10^{-08}	3	462	+	3.0199×10^{-11}	0	465	+	
F16	3.0199×10^{-11}	0	465	+	3.6709×10^{-03}	0	465	+	1.2477×10^{-04}	367	98	−	3.0199×10^{-11}	0	465	+	
F17	5.0723×10^{-10}	4	461	+	3.0199×10^{-11}	0	465	+	1.5014×10^{-02}	286	179	−	7.3891×10^{-11}	0	465	+	
F18	3.0199×10^{-11}	0	465	+	1.7290×10^{-06}	153	312	+	5.8945×10^{-01}	185	280	=	1.8500×10^{-08}	27	438	+	
F19	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.4742×10^{-10}	0	465	+	3.0199×10^{-11}	0	465	+	
F20	2.6015×10^{-08}	6	459	+	9.7052×10^{-01}	26	439	+	9.6263×10^{-02}	268	197	=	7.3891×10^{-11}	6	459	+	
F21	3.0199×10^{-11}	0	465	+	5.0723×10^{-10}	0	465	+	7.2951×10^{-04}	345	120	−	4.9752×10^{-11}	0	465	+	
F22	1.4110×10^{-09}	46	419	+	3.9881×10^{-04}	240	225	=	8.8830×10^{-01}	225	240	=	3.0199×10^{-11}	0	465	+	
F23	2.6099×10^{-10}	0	465	+	3.0199×10^{-11}	27	438	+	3.1967×10^{-09}	430	35	−	6.0658×10^{-11}	0	465	+	
F24	3.5638×10^{-04}	73	392	+	8.4848×10^{-09}	118	347	+	8.9934×10^{-11}	465	0	−	6.1210×10^{-10}	0	465	+	
F25	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.3384×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	
F26	6.0658×10^{-11}	0	465	+	3.0199×10^{-11}	36	429	+	1.0907×10^{-05}	391	74	−	5.0723×10^{-10}	11	454	+	
F27	3.0199×10^{-11}	0	465	+	7.3891×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	
F28	3.0199×10^{-11}	0	465	+	1.1738×10^{-03}	0	465	+	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	0	465	+	
F29	3.0199×10^{-11}	0	465	+	3.0199×10^{-11}	22	443	+	8.3026×10^{-01}	232	233	=	3.6897×10^{-11}	0	465	+	
+ / = / -		45/0/0				41/3/1				20/7/18				42/2/1			

4.4. Research on Engineering Design Problems of IPFA

4.4.1. Pressure Vessel Design Problem

Reducing the total cost is the aim of the pressure vessel design challenge [39]. Four decision variables make up this model: the thickness of the shell $T_s (0 \leq x_1 \leq 99)$, the thickness of the head $T_h (0 \leq x_2 \leq 99)$, the inner radius $R (0 \leq x_3 \leq 200)$, and the length of the cylindrical section without taking the head $L_s (10 \leq x_4 \leq 200)$, in addition to four constraints. The model's schematic structure is displayed in Figure 8:

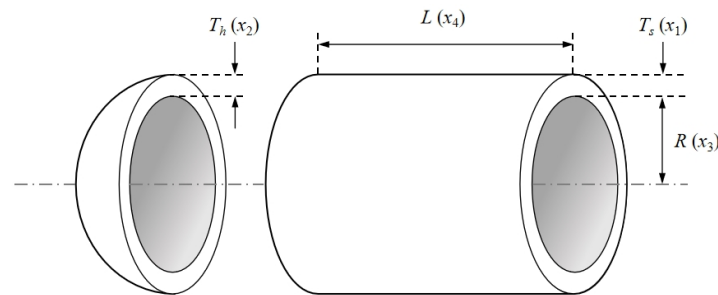


Figure 8. Pressure vessel design model.

The mathematical model of the pressure vessel design problem is as follows:

$$\begin{aligned}
 \min f(X) &= 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
 \text{s.t. } g_1(X) &= -x_1 + 0.0193x_2 \leq 0 \\
 g_2(X) &= -x_2 + 0.00954x_3 \leq 0 \\
 g_3(X) &= -\pi x_3^2x_4 - \frac{4\pi x_3^3}{3} + 1296000 \leq 0 \\
 g_4(X) &= x_4 - 240 \leq 0
 \end{aligned} \tag{27}$$

To optimize this problem, the IPFA is initialized with 30 individuals, executed for 30 independent runs, and iterated for 1000 cycles. The final optimization outcome is $f(X) = 5948.3597$ with $X = [0.9383 \ 0.0628 \ 54.6570 \ 184.6838]$. The results obtained by the proposed IPFA in solving this problem are compared with eight other optimization algorithms, including those reported in the literature. The experimental results are comprehensively presented in Tables 6 and 7. Upon analyzing the tables, it is evident that the IPFA in this study exhibits superior optimization performance compared to the other six meta-heuristic algorithms. It not only incurs lower design costs but also boasts the smallest S.D. among all the algorithms. Furthermore, the mean and worst rank second only to PFA, underscoring the efficacy of IPFA in optimizing the pressure vessel design problem.

Table 6. Optimal solutions for the pressure vessel design problem.

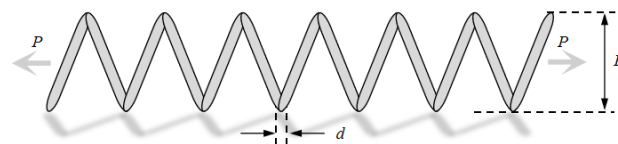
Algorithms	x_1	x_2	x_3	x_4	$f(X)$
IPFA	0.9383	0.0628	54.6570	184.6838	5948.3597
PFA	0.7784	0.3848	40.3303	199.8508	5984.8222
PSO	0.7779	0.3863	40.3241	200.0000	6054.2177
WOA	0.8604	0.4419	44.6154	147.7585	6103.924
DE	0.7833	0.3872	40.5855	200.0000	5973.8661
SOA	1.3609	0.0000	69.4207	1.5820	7951.2981
SCA	1.2594	0.0000	65.0338	11.8753	7675.005
ChOA	1.2628	0.0000	65.2543	10.0000	7608.0931
GWO	1.2417	0.3393	64.2907	14.0944	7421.4554
PHSSA [40]	0.8152	0.4265	42.0913	176.7423	6043.9861
HFA-GD [41]	0.8125	0.4375	42.0984	176.6366	6059.7143
BIANCA [42]	0.8125	0.4375	42.0968	176.6580	6059.9380

Table 7. Results of the pressure vessel design problem.

Algorithms	Best	Mean	Worst	S.D
IPFA	5948.3597	6603.1152	7325.0342	360.6949
PFA	5984.8222	6357.2572	7271.8175	390.2489
PSO	6054.2177	17550.2365	615848	16489.5647
WOA	6103.924	8791.4192	12457.216	2236.9964
DE	5973.8661	10892.4411	12522.3018	2314.5862
SOA	7951.2981	11620.3902	12416.8507	1505.3759
SCA	7675.005	11731.5107	12429.8212	1628.8661
ChOA	7608.0931	10932.7028	12418.6293	2197.2781
GWO	7421.45554	8831.1675	12415.5318	2091.9112

4.4.2. Tension Spring Design Problem

The tension spring design problem [43] has the objective of minimizing or maximizing the performance metrics of a tension spring under given constraints. This model has three decision variables: coil diameter d ($0.05 \leq x_1 \leq 2$), average coil diameter D ($0.25 \leq x_2 \leq 1.3$), and the number of active coils N ($2 \leq x_3 \leq 15$), as well as four constraints, and its structure is schematically shown in Figure 9:

**Figure 9.** Tension spring design model.

The mathematical model of the tension spring design problem is as follows:

$$\begin{aligned}
 \min f(X) &= (x_3 + 2)x_2x_1^2 \\
 \text{s.t. } g_1(X) &= 1 - \frac{x_2^3x_3}{71785} \leq 0 \\
 g_2(X) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^2 - x_4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\
 g_3(X) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
 g_4(X) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0
 \end{aligned} \tag{28}$$

The algorithm settings remain consistent with those employed in the pressure vessel design experiments. The results of each algorithm running independently for 30 iterations are tabulated in Tables 8 and 9. Based on the provided tables, the mean, worst, and S.D. of the IPFA are comparatively average in comparison to other optimization algorithms. Although there is room for improvement in its performance, the IPFA and GWO optimums are minimized at the same time, which is excellent compared to the other seven meta-heuristic algorithms, demonstrating the lowest design cost with $f(X) = 0.012699$, $X = [0.0504 \ 0.3978 \ 11.2764]$.

Table 8. Optimal solutions for the tension spring design problem.

Algorithms	x_1	x_2	x_3	$f(X)$
IPFA	0.0504	0.3978	11.2764	0.012699
PFA	0.0500	0.3174	14.0282	0.012719
PSO	0.0528	0.3845	9.8542	0.012719
WOA	0.0556	0.4575	7.1493	0.012926
DE	0.0531	0.3918	9.4925	0.012701
SOA	0.0517	0.3574	11.3142	0.012726
SCA	0.0500	0.3172	14.0598	0.012736
ChOA	0.0500	0.3170	14.0944	0.012756
GWO	0.0528	0.3832	9.8985	0.012699
SCADE [44]	0.0500	0.3145	15.0000	0.013365
GA [45]	0.0515	0.3517	11.6322	0.0127048
hHHO-SCA [46]	0.05469	0.054693	0.05469	0.054693

Table 9. Results of the tension spring design problem.

Algorithms	Best	Mean	Worst	S.D
IPFA	0.012699	0.015548	0.019405	0.0023197
PFA	0.012719	0.012751	0.013458	0.00013439
PSO	0.012719	0.061618	1.4434	0.26101
WOA	0.012926	0.013948	0.017773	0.0014544
DE	0.012701	0.013545	0.020274	0.0023698
SOA	0.012726	0.012939	0.013237	0.00018285
SCA	0.012736	0.013	0.013203	0.00013274
ChOA	0.012756	0.013096	0.014325	0.00031685
GWO	0.012699	0.012738	0.012878	0.0000486

5. Conclusions

In this study, we propose an improved pathfinding algorithm (IPFA) to address the problems of slow convergence and low optimization accuracy of standard pathfinding algorithms. Firstly, we introduce an elite opposition-based learning mechanism that incorporates learned exemplar individuals at the beginning of the algorithm. This improvement aims to increase the diversity and quality of the entire population, thus improving the global optimality performance and increasing the convergence accuracy. By selectively integrating individuals with exceptional performance into the initial population, the algorithm is better equipped to explore a diverse solution space and mitigate premature convergence to local optima.

Secondly, integrating the escape energy into the prey-hunting phase of the grey wolf optimizer, while substituting the follower phase in the pathfinder algorithm, serves to enhance the algorithm's diversity and elevate both the search efficiency and optimization performance. The incorporation of escape energy enables a subset of individuals to opt for escape during the search process, facilitating a more extensive exploration of the search space. This strategic approach helps prevent the algorithm from converging to a local optimal solution, thereby enhancing global search capabilities and fortifying the algorithm's overall robustness.

In conclusion, the dimension-by-dimension mutation method is employed to perturb the location of the optimal individual, facilitating a swift escape from local optimal solutions and promoting movement toward alternative regions. By implementing tiny random perturbations on every dimension of the ideal individual, this method successfully breaks apart the local structure of the existing solution. As a result, this directs the search procedure in a novel and maybe more advantageous route, improving the algorithm's overall search performance.

Incorporating these mechanisms into the pathfinder algorithm considers global optimization, diversity maintenance, escape mechanisms, and perturbation strategies. This

comprehensive enhancement aims to improve the robustness and performance of the algorithm. Such depth in algorithmic improvement is typically well-suited to address the distinct characteristics of various problems, consequently enhancing the algorithm's applicability.

Simultaneously, eight representative meta-heuristic algorithms were chosen for performance comparison, confirming the exceptional performance of the IPFA through function optimization experiments and an index evaluation system. The IPFA was successfully employed to address two engineering design problems, demonstrating stable and effective optimization outcomes that fully validate the algorithm's robust applicability. In future research endeavors, the aim will be to explore the integration of the IPFA with the field of deep learning, applying the optimization model to data prediction and analysis problems.

Author Contributions: Conceptualization, B.W. and X.M.; methodology, X.M.; software, B.W.; data curation, W.Y. and Y.C.; writing—original draft preparation, X.M.; visualization, B.W., W.Y. and Y.C.; funding acquisition, B.W. and X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the National Natural Science Foundation of China (grant no. 12271036), the Scientific Research Project of Mudanjiang Normal University (grant no. GP2020003), and the School-level Projects of Mudanjiang Normal University (grant no. kjcx2023-125mdjnu, kjcx2023-126mdjnu).

Data Availability Statement: Data are contained within the article.

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

References

- Huo, X.; Zhang, F.; Shao, K.; Tan, J.Q. Improved Meta-heuristic Optimization Algorithm and Its Application in Image Segmentation. *J. Softw.* **2021**, *32*, 3452–3467.
- Rajamoorthy, R.; Arunachalam, G.; Kasinathan, P.; Devendiran, R.; Ahmadi, P.; Pandiyan, S.; Muthusamy, S.; Panchal, H.; Kazem, H.A.; Sharma, P. A novel intelligent transport system charging scheduling for electric vehicles using Grey Wolf Optimizer and Sail Fish Optimization algorithms. *Energy Sources Part A Recover. Util. Environ. Eff.* **2022**, *44*, 3555–3575. [\[CrossRef\]](#)
- He, Y.; Venkatesh, B.; Guan, L. Optimal scheduling for charging and discharging of electric vehicles. *IEEE Trans. Smart Grid* **2012**, *3*, 1095–1105. [\[CrossRef\]](#)
- Cai, Y.; Du, P. Path planning of unmanned ground vehicle based on balanced whale optimization algorithm. *Control. Decis.* **2021**, *36*, 2647–2655.
- Guerrero-Ibáñez, J.; Zeadally, S.; Contreras-Castillo, J. Sensor technologies for intelligent transportation systems. *Sensors* **2018**, *18*, 1212. [\[CrossRef\]](#)
- Xia, H.; Chen, L.; Wang, D.; Lu, X. Improved Denclue outlier detection algorithm with differential privacy and attribute fuzzy priority relation ordering. *IEEE Access* **2023**, *11*, 90283–90297. [\[CrossRef\]](#)
- Xia, H.Z.; Chen, L.M.; Qi, F.; Mao, X.D.; Sun, L.Q.; Xue, F.Y. DP-Denclue: An outlier detection algorithm with differential privacy preservation. In Proceedings of the 2022 IEEE 24th IEEE International Conference on High Performance Computing and Communications (HPCC), Chengdu, China, 18–21 December 2022; pp. 2264–2269.
- Hsu, Y.P.; Modiano, E.; Duan, L. Age of information: Design and analysis of optimal scheduling algorithms. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 561–565.
- Zhang, H.Y.; Lin, W.m.; Chen, A.X. Path planning for the mobile robot: A review. *Symmetry* **2018**, *10*, 450. [\[CrossRef\]](#)
- Shi, Y. Particle swarm optimization. *IEEE Connect.* **2004**, *2*, 8–13.
- Draa, A.; Chettah, K.; Talbi, H. A compound sinusoidal differential evolution algorithm for continuous optimization. *Swarm Evol. Comput.* **2019**, *50*, 100450. [\[CrossRef\]](#)
- Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
- Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [\[CrossRef\]](#)
- Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [\[CrossRef\]](#)
- Yapici, H.; Cetinkaya, N. A new meta-heuristic optimizer: Pathfinder algorithm. *Appl. Soft Comput.* **2019**, *78*, 545–568. [\[CrossRef\]](#)
- Priyadarshani, S.; Subhashini, K.; Satapathy, J. Pathfinder algorithm optimized fractional order tilt-integral-derivative (FOTID) controller for automatic generation control of multi-source power system. *Microsyst. Technol.* **2021**, *27*, 23–35. [\[CrossRef\]](#)
- Janamala, V. A new meta-heuristic pathfinder algorithm for solving optimal allocation of solar photovoltaic system in multi-lateral distribution system for improving resilience. *SN Appl. Sci.* **2021**, *3*, 118. [\[CrossRef\]](#) [\[PubMed\]](#)
- Gouda, E.A.; Koth, M.F.; El-Fergany, A.A. Investigating dynamic performances of fuel cells using pathfinder algorithm. *Energy Convers. Manag.* **2021**, *237*, 114099. [\[CrossRef\]](#)

19. Yuan, Z.; Li, H.; Yousefi, N. Optimal hydrogen consumption of fuel cell-based locomotive using speed trajectory optimization by Improved Pathfinder algorithm. *J. Clean. Prod.* **2021**, *278*, 123430. [\[CrossRef\]](#)
20. Tang, C.; Zhou, Y.; Luo, Q.; Tang, Z. An enhanced pathfinder algorithm for engineering optimization problems. *Eng. Comput.* **2021**, *38*, 1481–1503. [\[CrossRef\]](#)
21. Tang, C.; Zhou, Y.; Tang, Z.; Luo, Q. Teaching-learning-based pathfinder algorithm for function and engineering optimization problems. *Appl. Intell.* **2021**, *51*, 5040–5066. [\[CrossRef\]](#)
22. Hu, R.; Dong, Y.; Qian, B. Pathfinder algorithm for green pipeline scheduling with limited buffers. *J. Syst. Simul.* **2021**, *33*, 1384.
23. Lu, M. Improvement and Application of Grey Wolf Optimization Algorithm. Master's Thesis, Guangxi Minzun University, Nanning, China, 2021.
24. Sun, Z.Z. Improved Pathfinder Algorithm and Its Application. Master's Thesis, University of Science and Technology Liaoning, Anshan, China, 2021.
25. Mahdavi, S.; Rahnamayan, S.; Deb, K. Opposition based learning: A literature review. *Swarm Evol. Comput.* **2018**, *39*, 1–23. [\[CrossRef\]](#)
26. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005; Volume 1, pp. 695–701.
27. Yildiz, B.S.; Pholdee, N.; Bureerat, S.; Yildiz, A.R.; Sait, S.M. Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems. *Eng. Comput.* **2022**, *38*, 4207–4219. [\[CrossRef\]](#)
28. Yuxin, G.; Sheng, L.; Wenxin, G.; Lei, Z. Elite Opposition-Based Learning Golden-Sine Harris Hawks Optimization. *J. Comput. Eng. Appl.* **2022**, *58*. [\[CrossRef\]](#)
29. Xie, C.W.; Xu, L.; Zhao, H.R.; Xia, X.W.; Wei, B. Multi-objective fireworks optimization algorithm using elite opposition-based learning. *Acta Electronica Sin.* **2016**, *44*, 1180.
30. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
31. Zhang, X.; Lin, Q.; Mao, W.; Liu, S.; Dou, Z.; Liu, G. Hybrid Particle Swarm and Grey Wolf Optimizer and its application to clustering optimization. *Appl. Soft Comput.* **2021**, *101*, 107061. [\[CrossRef\]](#)
32. Ou, Y.; Zhou, K.; Yin, P.; Liu, X. Improved grey wolf optimizer algorithm based on dual convergence factor strategy. *J. Comput. Appl.* **2023**, *43*, 2679.
33. Kamboj, V.K.; Nandi, A.; Bhadoria, A.; Sehgal, S. An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Appl. Soft Comput.* **2020**, *89*, 106018. [\[CrossRef\]](#)
34. Lange, K.L.; Little, R.J.; Taylor, J.M. Robust statistical modeling using the t distribution. *J. Am. Stat. Assoc.* **1989**, *84*, 881–896. [\[CrossRef\]](#)
35. Jones, M.C.; Faddy, M. A skew extension of the t-distribution, with applications. *J. R. Stat. Soc. Ser. Stat. Methodol.* **2003**, *65*, 159–174. [\[CrossRef\]](#)
36. Dhiman, G.; Kumar, V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl.-Based Syst.* **2019**, *165*, 169–196. [\[CrossRef\]](#)
37. Khishe, M.; Mosavi, M.R. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [\[CrossRef\]](#)
38. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 196–202.
39. Moss, D.R. *Pressure Vessel Design Manual*; Elsevier: Amsterdam, The Netherlands, 2004.
40. Abualigah, L.; Shehab, M.; Diabat, A.; Abraham, A. Selection scheme sensitivity for a hybrid Salp Swarm Algorithm: Analysis and applications. *Eng. Comput.* **2022**, *38*, 1149–1175. [\[CrossRef\]](#)
41. Cheng, Z.; Song, H.; Zheng, D.; Zhou, M.; Sun, K. Hybrid firefly algorithm with a new mechanism of gender distinguishing for global optimization. *Expert Syst. Appl.* **2023**, *224*, 120027. [\[CrossRef\]](#)
42. Montemurro, M.; Vincenti, A.; Vannucci, P. The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **2013**, *256*, 70–87. [\[CrossRef\]](#)
43. Arora, J.S. *Introduction to Optimum Design*; Elsevier: Amsterdam, The Netherlands, 2004.
44. Kaveh, A.; Bakhshpoori, T. Water evaporation optimization: A novel physically inspired optimization algorithm. *Comput. Struct.* **2016**, *167*, 69–85. [\[CrossRef\]](#)
45. Feng, L.; Li, S.; Feng, S. Preparation and characterization of silicone rubber with high modulus via tension spring-type crosslinking. *RSC Adv.* **2017**, *7*, 13130–13137. [\[CrossRef\]](#)
46. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [\[CrossRef\]](#)

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.