STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **9**, June 2021, pp 459–491.
Published online in International Academic Press (www.IAPress.org)

A Cooperation of the Multileader Fruit Fly and Probabilistic Random Walk Strategies with Adaptive Normalization for Solving the Unconstrained Optimization Problems

Wirote Apinantanakon [1], Khamron Sunat [2,*],Sirapat Chiewchanwattana[2]

[1]*Department of Computer Science, Faculty of Arts and Science, Rajabhat Chaiyaphum University, Chaiyaphum 36000, Thailand*
[2]*Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand*

Abstract A swarm-based nature-inspired optimization algorithm, namely, the fruit fly optimization algorithm (FOA), has a simple structure and is easy to implement. However, FOA has a low success rate and a slow convergence, because FOA generates new positions around the best location, using a fixed search radius. Several improved FOAs have been proposed. However, their exploration ability is questionable. To make the search process smooth, transitioning from the exploration phase to the exploitation phase, this paper proposes a new FOA, constructed from a cooperation of the multileader and the probabilistic random walk strategies (CPFOA). This involves two population types working together. CPFOAs performance is evaluated by 18 well-known standard benchmarks. The results showed that CPFOA outperforms both the original FOA and its variants, in terms of convergence speed and performance accuracy. The results show that CPFOA can achieve a very promising accuracy, when compared with the well-known competitive algorithms. CPFOA is applied to optimize two applications: classifying the real datasets with multilayer perceptron and extracting the parameters of a very compact T-S fuzzy system to model the Box and Jenkins gas furnace data set. CPFOA successfully find parameters with a very high quality, compared with the best known competitive algorithms.

Keywords Nature-inspired optimization algorithm; Fruit fly optimization algorithm; multileader strategy; random walk; cooperative algorithm.

DOI: 10.19139/soic-2310-5070-702

1. Introduction

Over the last few decades, researchers have started to adapt their knowledge of natural phenomena for the development of optimization techniques. These techniques were successfully applied for chemical process applications([1], [2], [3] and [4]). The main concepts of the aptly named, sources of nature-inspired algorithms, have been observed within the successful biological systems. Accordingly, most nature-inspired algorithms are biologically inspired, or bio-inspired, and mimic specific behavior in nature. Examples of such popular nature-inspired algorithms include the particle swarm optimization algorithm(PSO)([5]),which was inspired by the social behavior of flocking birds, or schooling fish; the ant colony optimization algorithm (ACO)([6]),which mimics an ant colonys behavior in their search for food; the artificial bee colony algorithm (ABC)([7]),motivated by the intelligent behavior of a honey bee swarm; the cuckoo search algorithm (CS)([8]),inspired by the parasitic bio-interactions of a cuckoo species, which lays their eggs in the nests of other host birds; and the bat-inspired algorithm (BA)([9]),which was inspired by the echolocation behavior of bats, to name but a few. However, each nature-inspired algorithm has different capabilities when it comes to finding solutions, depending on the personal ability

*Correspondence to: Khamron Sunat.Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

ISSN 2310-5070 (online) ISSN 2311-004X (print)

of the living things in nature. Developing a successful, modern nature-inspired algorithm is a challenging task, even today, as there is no one particular nature-inspired algorithm capable of solving every scientific problem. Hence, the continual development of a new algorithm is required. Details of the broadly classified nature-inspired algorithms, which brought about the development of algorithms in current modern literature, have been provided([10], [11]).

In 2011, a swarm intelligence method-based stochastic optimization technique, namely, the fruit fly optimization algorithm (FOA) was proposed by Pan([12]).It is based on the foraging behavior of fruit flies. FOA is very user-friendly because of its simplicity and shortness. FOA can be easily understood by most researchers in this field. This also means that it can be easily implemented into a program code, when compared with other well-known algorithms, such as differential evolution (DE) ([13]),the genetic algorithm (GA)([13]) and particle swarm optimization (PSO). The FOA has achieved success in several applications, including research into optimization problems ([15], [16], [17] and [18]), neural network parameter optimization ([19], [20], [21]), swarm techniques for mini autonomous surface vehicles (ASVs) ([22]), the identification of dynamic protein complexes based on the fruit fly optimization algorithm ([23]), support for vector regression using the fruit fly optimization algorithm for seasonal electricity consumption forecasting ([24]), efficient truss optimization using the contrast-based fruit fly optimization algorithm ([25]) and improving the fruit fly optimization algorithm to apply it to structural engineering design optimization problems ([26], [27]), a prediction model for Chinas agricultural output value, based on the optimization of the neural network ([28]), and a novel phase angle-encoded fruit fly optimization algorithm, with a mutation adaptation mechanism applied to UAV path planning ([29]).

Pans FOA has a good structure and mechanism for finding the solution of optimization problems. However, the algorithm is prone to trapping into a local extreme and premature convergence, since FOA generates a new fruit fly swarm around the current best solution by using random uniform distribution, with a fixed radius, especially when FOA deals with multi-dimensional and complex optimization problems.

The drawback caused by fixed radius updating has motivated many researches into various dynamic radius updating techniques to improve the FOA, e.g., an improved fruit fly optimization algorithm for solving optimization problems (LGMS-FOA) ([15]), an improved fruit fly optimization algorithm for continuous function optimization problems (IFFO) ([16]), a novel multi-swarm fruit fly optimization algorithm (MFOA) ([18]), a novel multi-scale cooperative mutation fruit fly optimization algorithm (MSFOA) ([30]), and a novel fruit fly optimization algorithm, with trend search and co-evolution (CEFOA) ([31]).

There is one remaining imperfection that is caused by the single leader usage, although the FOA variants showed that dynamic radius updating can improve the quality of the produced solutions. The single leader usage causes a lack of diversity in FOA when the search process has to deal with a multi-dimensional or a complex optimization problem. In this paper, we proposed a novel FOA that uses a multileader, instead a single leader, and probabilistic random walk radius updating instead of random uniform dynamic radius updating (CPFOA). The details of the proposed CPFOA are described in Section 3.

The remainder of the paper is organized as follows: Section 2 presents the FOA; Section 3 presents the strategies of CPFOA construction, the multileader, and the probabilistic random walk strategies; Section 4 explains the evaluation process of algorithms and settings; Section 5 shows the experimental results and discussion; Section 6 shows two applications of CPFOA; and lastly, Section 7 concludes the paper.

2. The Fruit Fly Optimization Algorithm

The drosophila optimized algorithm or fruit fly optimization algorithm (FOA) was developed in 2011 by Pan ([12]). FOA determines global optimization based on the foraging behavior of fruit flies. Compared to other species, the fruit fly possesses a keener sense of smell and sight in search of food. Their drosophila olfactory organ can detect a food source as far as 40 km away, which triggers a flight reaction toward the target location. For the intelligent sense of fruit flies, the novel FOA optimization algorithm is inspired and established through the simple behavior of fruit flies search for food.

The FOAs process is similar to that of other swarm optimization algorithms. The first phase of the fruit flies quest for food is initiated with a random uniform distribution ([10], [32], [33], [34], [35], [36]), with no specific

position or direction. In the second phase, the fruit fly with the best sense of smell or the best fitness within the group, from the first phase, is determined. The fruit fly with the best sense of smell, represented by X_axis^t is used as a center for generating new populations in the next generation. The computational steps of the FOA method are summarized in Algorithm 1.

Algorithm 1 The FOA algorithm

Step 1: Initialize FOA's parameters: random location (X_axis, Y_axis),size of population (popsize),and maximum iteration (Max_iter).

Step 2: Give the random position and fly direction of an individual fruit fly in search of food.

$$X_i = X\_axis + rand()$$

$$Y_i = Y\_axis + rand()$$

Step 3: Calculate the distance (Dist) to the food's origin, as the exact position of the food's location is not known at this stage.

$$Dist_i = \sqrt{x_i^2 + y_i^2}$$

Step 4: Calculate the smell concentration judgment value(S_i).

$$S_i = \frac{1}{Dist_i}$$

Step 5: Calculate the fitness: the smell concentration judgment of the individual fruit fly, obtained from Step 4, is calculated by substituting S_i into the smell concentration judgment function(also called the fitness function),in order to find the optimal smell.

$$Smell_i = objective\_function(S_i)$$

Step 6: Determine the fruit fly with the optimal smell concentration judgment among the fruit fly group.

$$[bestSmell, bestIndex] = find\_the\_best(Smell)$$

Step 7: Keep the best (x,y) position and the optimal concentration value, and use this position as the flight center towards the next location(in Step 2).

$$Smellbest = bestSmell$$

$$X\_axis = X(bestIndex)$$

$$Y\_axis = Y(bestIndex)$$

Step 8: Repeat Steps 2-7, and determine whether the smell concentration is better than the previous iterative smell concentration. If yes, go to Step 7. The process will stop if either the smell concentration no longer changes, or the iterative number reaches the maximum iteration number(Max_iter).The outputs are X_axis and Y_axis.

2.1. An Analysis of the Original FOA and the FOA Variants

2.1.1. Analysis of the Original FOA There are problems with FOA that make the algorithm unsuitable to dealing with multi-dimensional and complex optimization problems. The problems are investigated in

([15], [16], [18], [30], [31], [37], [38]) and are briefly described as follows.
1. Problems regarding the smell value S_i,according to Step 4, cannot appropriately evaluate the "objective function(S_i)" when there are negative numbers in the domain because $S_i = \frac{1}{Dist_i} > 0$,so that the function cannot determine S_i as a negative ([15], [18], [37], [38]).
2. The fixed radius, with random uniform distribution,$rand()$,within the initial process, limited the convergence of FOA in the processes of exploration and exploitation ([15], [16], [18], [30], [31]).

2.1.2. Analysis of the FOA Variants To overcome the disadvantages of the original FOA, researchers have continuously developed new strategies to improve the FOA for solving high-dimensional function optimization problems. The recently proposed FOAs can be grouped in the two categories. In the first category, each of the fruit flies is defined through the random initial base point of X_axis, Y_axis and the positions of X_i and Y_i,as in the original FOA, to update the new generation of populations. The other functions, such as the smell value$(S_i)$and the evaluation of the objective function(S_i),are modified, including the extra mechanisms. In the second category, these the problems are solved by (i) omitting Y_i;(ii) defining each of the fruit flies through $X \in R^{N\times D}$,where D is the number of decision variables $(ordimensions)$,and N is the population size, i.e.,$x_i = (x_i^1, x^1, x^2, \ldots, x_i^D) \in R^D$;and (iii) removing the distance $Dist_i$ and the smell concentration judgment function S_i.The fitness value is now calculated by substituting x_i into the smell concentration judgment function, with $Smell_i = objectivefunction(x_i)$.The position of the fruit fly with the minimal concentration value,X_axis,is the base point for flying towards the next location. A brief summary of the two categories of improved FOAs are as follows.
The first category is as follows.

- Babalık et al. ([39]) proposed an improvement of the fruit fly optimization algorithm using sign parameters (SFOA). The algorithm presents the improvement method by using two sign variables, r and q vectors, in order to determine a sign for each decision variable of fruit flies.

- CEFOA ([31]), proposed by Han et al., revealed that the simple structure of FOA limited the search space and easily trapped the fruit flies in a local minimum. To overcome this drawback, the CEFOA used two mechanisms: The trend search strategy and the co-evolution mechanism. The trend search enhances the local search capability of swarm. The co-evolution mechanism is employed to avoid premature convergence and to improve the global searching ability. However, the key of the CEFOA is the multi-scale equation for updating the fruit fly swarm. To set the variable capacity of each fruit fly, connected with its food quality, the CEFOA used the variance of the multi-level evolutionary operator. The search radius is dynamically adjusted through $\mathbf{X_i(t) = X_i(t-1) + LR \times N(0, \delta_i(t))}$ and $\mathbf{Y_i(t) = Y_i(t-1) + LR \times N(0, \delta_i(t))}$ where $\mathbf{\delta_i(t)}$ are the multi-scale factors of the i-th fruit fly.

The second category is as follows.

- LGMS-FOA, proposed by Shan et al.([15]),presented two parameters to tune up the search radius by adding the weight parameter w,when the radius is changed with respect to time. A new fruit fly location is generated as $\mathbf{x_i^d = X\_axis_i^d + w \times rand[0,1)}$, $\mathbf{w = w_0 \times \alpha^t}$,where $\mathbf{w_0 = 1}, \alpha = \mathbf{0.95}$,t=iteration index, and $\mathbf{X\_axis}$ is the best position obtained during iterations.
- IFFO, proposed by Pan et al.([16]),introduced a new control parameter to adjust the search radius adaptively. The search radius is dynamically changed during iterations through $\lambda = \lambda_{\mathbf{max}} \times \mathbf{exp}(\mathbf{log}(\frac{\lambda_{\mathbf{min}}}{\lambda_{\mathbf{max}}}) \times \frac{\mathbf{t}}{\mathbf{t_{max}}})$,where λ is the radius variants in each iteration,$\lambda_{\mathbf{max}} = \frac{\mathbf{UB-LB}}{\mathbf{2}}$,$\mathbf{UB}$ is the upper bound and $\mathbf{LB}$ is the lower bound of domain problems,$\lambda_{\mathbf{min}} = \mathbf{10^{-5}}$,$\mathbf{t}$ is the iteration index and $\mathbf{t_{max}}$ is the maximum iteration number $(\mathbf{Max\_iter})$.
- MFOA, proposed by Yuan et al.([18]), presented a multi-swarm fruit fly that employed sub-swarm to explore the solutions in the search space simultaneously. Moreover, MFOA shrinks the search radius through

$\mathbf{R(t)} = (\mathbf{UB} - \frac{\mathbf{LB}}{\mathbf{2}}) \times (\mathbf{G_{max}} - \frac{\mathbf{G}}{\mathbf{G_{max}}})^{\theta}$,where $\mathbf{t}$ is the iteration index,$\mathbf{UB}$ is the upper bound,$\mathbf{LB}$ is the lower bound,$\mathbf{G}$ is the number of sub-swarms and $\mathbf{G_{max}}$ is the maximum number of sub-swarms.

- MSFOA, proposed by Zhang et al.([30]), presented a strategy to analyze the convergence and showed that the convergence depends on the initial positions of the swarms. MSFOA used the Gaussian mutation operator, rather than the uniform random number (more details of which can be found in ([30])). For a flying fruit fly, MSFOA used a linear generation mechanism, through the equation,$\mathbf{x_{i,j}^t} = \mathbf{X_j^t} + \mathbf{w} \times \mathbf{rand}(\mathbf{R_{min}}, \mathbf{R_{max}})$,where $\mathbf{w} = \mathbf{w_0} \times \alpha^{\mathbf{t}}$,$\mathbf{w_0} = \mathbf{1}$,$\alpha = \mathbf{0.95}$,$\mathbf{t}$ is current iteration,$\mathbf{w}$ is the search coefficient,α is the initial weight and $\mathbf{R_{min}}, \mathbf{R_{max}}$ are obtained from the domain boundary of the problem.

To recap, the improved FOAs, such as SFOA, CEFOA, LGMS-FOA, IFFO, MFOA and MSFOA, were strategies proposed to enhance the search ability. The search radius was a main point to tackle in several proposed strategies. However, only the dynamic mechanism of the search radius itself might be insufficiently efficient to overcome the lack of diversity and premature convergence, because these FOA variants still use only one leader as the flying base point. The single leader strategy might affect the FOA by easily trapping a local optimum when optimizing a multi-dimensional optimization problem. In this paper, the proposed CPFOA is comprised of two strategies. The first strategy focuses on the enhancement of the search ability based on the multileader fruit fly. The latter is a probabilistic dynamic search radius, with adaptive normalization. These two mechanisms are different from the existing FOA variants. The details of the proposed CPFOA are presented in the next section.

3. The Proposed CPFOA

This section presents a cooperation of multileader fruit flies and a probability search of a random walk for FOA to solve the unconstrained optimization problems (CPFOA). The CPFOA consists of (i) the cooperation strategies, called the multileader strategies, and (ii) the probabilistic search by a random walk. The multileader strategy used a main leader and several other leaders as the flying bases. The probabilistic search by a random walk changes the search radius to control the search spaces of the main leader.

3.1. Multileader Strategy

Contrary to FOA, which uses a single leader fruit fly, the CPFOA uses multileader fruit flies. Suppose that there is a swarm of fruit flies,$X \in R^{N \times D}$,where D is the number of decision variables (or dimensions), and N is the number of fruit flies, i.e.,$x_i = (x_i^1, x^1, x^2, \ldots, x_i^D) \in R^D$.The generated multileader strategy has four computational steps, as follows.

Step 1. Sort X in ascending order, based on the individual fitness values, to be:

$$\dot{X} = \{\dot{X}_1, \dot{X}_2, \ldots, \dot{X}_N\}. \tag{1}$$

Step 2. Divide $\dot{X}$ into the M disjoint sub-swarms, as in equation (2):

$$\dot{X} = \{\dot{X}_1, \dot{X}_2, \ldots, \dot{X}_{\frac{N}{M}}\} \cup \{\dot{X}_{\frac{N}{M}+1}, \dot{X}_{\frac{N}{M}+2}, \ldots, \dot{X}_{\frac{2N}{M}}\} \cup \cdots \cup \{\dot{X}_{\frac{(M-1)N}{M}+1}, \dot{X}_{\frac{(M-1)N}{M}+2}, \ldots, \dot{X}_N\} \tag{2}$$

where M is the total number of sub-swarms (aka, the number of leaders).

Step 3. Compute $leader_1, \ldots, leader_M$ by equation (3):

$$leader_{j,d} = \frac{M}{N} \sum_{i=1}^{\frac{N}{M}} \dot{X}_{\frac{(j-1)N}{M}+l,d}, j = 1, 2, \ldots, \frac{N}{M}, d = 1, \ldots, D \tag{3}$$

Step 4. Generate M new fruit flies based on $leader_1, \ldots, leader_M$ by equation (4):

$$M\_C_j = \beta - X_{axis} \otimes leader_j \times rand[0, 1), j = 1, 3, \ldots, \frac{N}{M} \tag{4}$$

where $\otimes$ is the Hadamard product.
The multileader strategy generates M new fruit flies from M leaders, i.e., a new fruit fly is generated from each leader. These fruit flies are generated from the shared information and might result in the improvement of the exploration ability. Moreover, CPFOA controls the shrinking of the search radius (β) in Step 4 through the probability of Ps.There are two types of radius: The normal scope (N_{scope}) and high scope (H_{scope}).The generation of β is controlled by Ps,as follows:

$$\beta = \begin{cases} N_{scope}, & \text{if} \quad Ps > 0.1 \\ H_{scope}, & otherwise \end{cases} \tag{5}$$

$$Ps = \frac{t}{Max\_iter} \tag{6}$$

where Max_iter is the maximum number of iterations,t is the iteration index, and $0 \leq t \leq Max\_iter$.

$$N_{scope} = max(UB) + min(UB) \tag{7}$$

where $Ub = upper - bound$,and $Lb = lower - bound$ of the domain search problems.

$$H_{scope} = max(X_{axis}) + min(X_{axis}) \tag{8}$$

Based on Equation (6), the radius of the fruit fly search is probabilistically large,if Ps is a small value (or t is at the early phase of the optimization). Otherwise, it is probabilistically a small value, if the iteration is in the latter phase of the optimization.

3.2. Probabilistic Search Strategy Based on a Random Walk with Adaptive Normalization

As mentioned in Section 2.1.1, FOA uses the random uniform distribution, with a fixed radius of search. In this paper, CPFOA will employ a random walk generation, which is inspired by the ant lion optimizer ([40]), and a probabilistic control of the search radius. The random walk is a mathematical equation process, which can provide the series of consecutive random steps ([41], [42]).
The value generated from a random walk at time $n > 0$ is found by a recursive formula, as follows:

$$R^n = R^{n-1} + x^n \tag{9}$$

where x^n is a random value extracted from a random number generator, and $R^0 = 0$.
Equation (9) shows that the changing state of R^n is attached to the previous state of R^{n-1} and every step, obtained from current iteration to the next iteration. The details of the random walk in the CPFOA strategy can be described in this section.
1. The fruit fly with the best fitness (X_axis) is determined after the first generation of the evaluation.(X_axis) is used as a center for updating the N-M candidate solutions in the next generations.
2. As for the updating step, CPFOA uses the random walk to generate N-M individual fruit flies. The characteristics of the populations of the random walk movements are described as:

$$XR^t = XR^{t+1} + \rho x^t \tag{10}$$

where ρ is a function that controls the direction of the fruit fly at any changing step, and $XR^0 = 0$.In the proposed CPFOA,ρ is either-1 or 1 and is determined as:

$$\rho = 2r^t - 1 \tag{11}$$

$$r^t = \begin{cases} 1, & \text{if} \quad rand > 0.5 \\ 0, & otherwise \end{cases} \tag{12}$$

where t denotes the iteration that the random walk came to a halt,$r(t)$ is a stochastic function, and $rand$ is a random uniform point in [0, 1). In order to match the values generated from Equation (10) with the boundaries of the problem and, furthermore, to make the search process smooth, transitioning from the exploration phase to the exploitation phase, equation (10) is adaptively normalized as follows:

$$P\_XR_{i,d}^t = \frac{XR_{i,d}^t - a^t}{b^t - a^t} \times (d^t - c^t) + c^t, i = 1, \ldots, N, and \quad d = 1, \ldots, D \tag{13}$$

where a is the minimum of $\{XR^0, \ldots, XR^t\}$,b is the maximum of $\{XR^0, \ldots, XR^t\}$,b,c and d are the minimum and maximum radii at the t-th iteration to control the scope of the search space during the optimization steps, respectively. The value of c and d are determined through the changing values of L,as follows:

$$c^t = \begin{cases} X\_axis^t + LB^t, & \text{if} \quad rand < 0.5 \\ X\_axis^t - LB^t, & otherwise \end{cases} \tag{14}$$

$$d^t = \begin{cases} X\_axis^t + UB^t, & \text{if} \quad rand < 0.5 \\ X\_axis^t - UB^t, & otherwise \end{cases} \tag{15}$$

where

$$LB^t = \frac{LB^t}{L} \tag{16}$$

$$UB^t = \frac{UB^t}{L} \tag{17}$$

L is a special constant parameter determined from the probability of the Ps variable. The parameter of L and Ps can be calculated as follows:

$$Ps = \frac{t}{Max\_iter} \tag{18}$$

where t is the current iteration,Max_iter is the maximum number of iterations, and $L = Ps \times 10^2$ when $Ps > 0.25, L = Ps \times 10^3$ when $Ps > 0.5, L = Ps \times 10^4$ when $Ps > 0.75, L = Ps \times 10^5$,when $Ps > 0.8, and L = Ps \times 10^6$ when $Ps > 0.9$.In the proposed CPFOA,L is used to adjust the accuracy level of exploitation.

Equation (14) through Equation (18) perform the probabilistic control of the search radius for CPFOA, which is different from the mechanism in FOA variants. The simulation state of the cooperation of FOA's leader and CPFOA's co-leaders is shown in Figure 1. A graph of the search radius generated during CPFOA, optimized as "Exponential function" (f1 in Table 1), is shown in Figure 2 (a), and the example the graph of the search radius generated by IFFO is shown in Figure 2 (b).We have observed that the search radii in the two figures are very different. The behavior of the search radius in CPFOA is very similar to that of the chaotic gravitational constants in the gravitational search algorithm (GSA)([43]). This kind of behavior should help CPFOA in smoothly transitioning from the exploration stage to exploitation.

3.3. The Proposed CPFOA

The structure of the CPFOA is similar to that of the IFFO, in that the function $Dist_i$ and the smell concentration judgment value (S_i) are eliminated. The pseudo code of CPFOA is presented in Algorithm 2.

4. The Experiments and Evaluations

There are two groups of competitive algorithms. The first group, shown in Table 2, is the FOA variants, including FOA, LGMS, IFFO, MFOA, MSFOA, and CEFOA. The second group, shown in Table 3, comprises CEFOA and six meta-heuristic algorithms: PSO, DE, GSA, HS, BA, and FA.

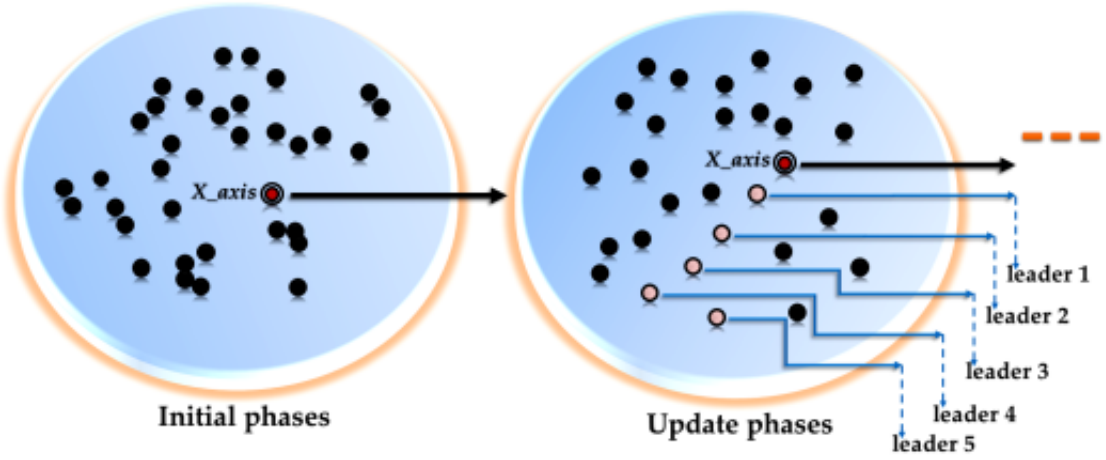


Figure 1. The simulation state of the cooperation of FOAs leader and CPFOAs co-leaders.

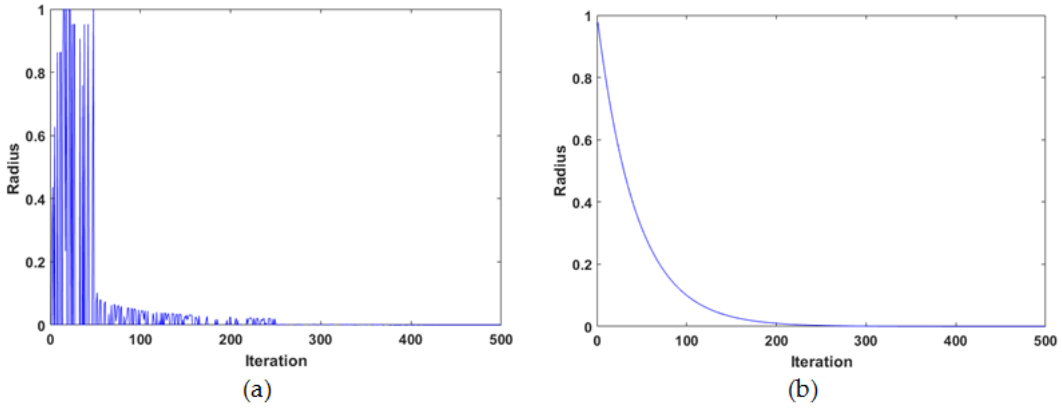


Figure 2. A graph of the search radius generated from the Exponential function, (a) is generated by CPFOA, and (b) is the example graph of the dynamic search radius generated by IFFO. The stochastic fluctuation of the radius in (a) make CPFOA smoothly transits from the global search to the local search

The performance of the proposed algorithm and that of the competitive algorithms are evaluated through 18 scalable functions taken from ([31]). The definition of the functions and their global optima are listed in Table 1: $f1 - f3$ are the uni-modal functions, and $f11 - f18$ are the multi-modal functions.
The experimental environment is MATLAB 9.2.0 (R2017a), which was run on a personal computer, with a 3.2 GHz CPU and 8 GB RAM. The operating system is Microsoft Windows 7 (64-bit).

Algorithm 2 CPFOA

Input: popsize,Max_iter,Lb,Ub,dim,benchmark function (Smell)
Output: $X\_axis(Max\_iter)$,bestSmell(Max_iter)

Initial random position of an individual fruit fly (X_i)
Compute $S = \{Smell(X_i^0)\}, i = 1, \ldots, N$
$bestIndex(0) = ArgMin(S, \{1, \ldots, N\})$
$bestSmell(0) = S(0)_{bestIndex(0)}$
$X\_axis(0) = X(0)_{bestIndex(0)}$
for $t = 1$ to Max_iter **do**
    Generate N-M fruit flies $(calledP\_XR^t)$ based on the X_axis using the random walk in equation (13)
    Sort X^t based on S
    Generate M leaders by Equation (3)
    Generate M cooperative fruit flies based on M leaders $(calledM\_candidate^t)$ by equation (4)
    $X^t = M\_candidate^t \cup P\_XR^t$
    Compute $S = \{Smell(X_i^t) | i = 1, \ldots, N\}$
    $bestIndex(t) = ArgMin(S, \{1, \ldots, N\})$
    $bestSmell(t) = S(t)_{bestIndex(t)}$
    $X\_axis(t) = X(t)_{bestIndex(t)}$
    If $bestSmell(t) > bestSmell(t-1)$
        $bestSmell(t) = bestSmell(t-1)$
        $X\_axis(t) = X\_axis(t-1)$
    End if
end for
Note: $ArgMin(f, X)$ gives a position X_{min} at which f is minimized.

4.1. Parameters and Settings

There are two experiments, each of which is as follows:
1. In the first experiment, the proposed CPFOA is compared with the competitive algorithms from the first group, i.e., CPFOA and the FOA variants are competing. The comparison is conducted on the 18 scalable functions taken from ([31]). The dimension of each problem is set to three values: 30, 50, and 1000.
2. In the second experiment, the proposed CPFOA is compared with the competitive algorithms from the second group, i.e., CPFOA and the original version of some state-of-the-art algorithms are competing. The comparison is based on the 18 scalable functions taken from ([31]). The dimension of each problem is set as those in the first experiment.

The following settings are set to comply with that of the existing CEFOA, the maximum iteration$(Max_iter)$of each algorithm is fixed to 500, the population size $(popsize)$ is 30, and the average (ave) and the standard deviations (std) of the final objective values are computed from 50 replications. The other parameters of each competitive algorithm are set in accordance with their original literature, which are listed in Tables 2 and 3.

4.2. Performance Criteria

The criteria for performance evaluation of the competing algorithms are the quality, the robustness, the success rate, and the statistical test, each of which is as follows:
1. The quality of the algorithms is determined by the average value (ave) and standard deviation (std) of the final objective values. A lower value is better. Moreover, the decision can be supported by the convergence

graph.
2. The success rate (SR) is determined by the number of successful runs over the total number of runs. A run is successful if the algorithm finds a feasible solution x, represented by $f(x) - f(x^*) \leq 1e-5, f(x) - f(x^*) \leq 1e-10$,within the maximum number of function calls (the terminated condition of each run (Max_{NFE}),where x is a feasible optimal solution of the function, and x^* is the best known solution of a specific problem f.A higher average SR_{ave} indicates a better performance. The average success rate (SR_{ave}) is written as:

$$SR_{ave} = \frac{number\ of\ successful\ runs}{total\ number\ of\ runs} \tag{19}$$

A statistical test, to investigate the significance of difference between CPFOA's outcome and the competitive algorithm's outcome, the Wilcoxon signed rank test, with the significance level of 0.05, is conducted to judge whether the 50 runs of CPFOA are statistically better than that of its competitors. The h values, signifying the results of the Rank-sum test, are indicated in Tables 6-13 by one of three symbols:"+","-" or "=", where the "+" symbol means that the solutions produced by the competitive algorithm are better than those of CPFOA, the "=" symbol means the outcomes of the competitive algorithm are comparable to or similar to those of CPFOA, and the "-" symbol means that the outcomes of the competitive algorithm are worse than those of CPFOA. To conclude the statistical test, the total h is represented as #1/#2/#3, where #1, #2, and #3 represent the number of wins, ties, and losses of the algorithm, respectively.

5. Results and Discussion

5.1. Convergence Behavior of CPFOA

This section will provide consistent information about the convergence behavior of CPFOA, when the problem has an unknown number of local optima. The contours of the Egg holder function ([55]) and Schaffer function ([56]) are plotted in Figures 3 and 4, respectively. In addition, the artificial fruit flies, appearing at several iterations, are also scattered in the different plots of the two functions. The optimal points of the two functions are located near the top right corner and at the origin. The starting points are far from the optimal points. We have observed, from Figures 3 and 4, that CPFOA has the capability of successfully optimizing the multimodal functions without being trapped in local optima. The experiment has been repeated for 50 runs, and every run reaches the optimal point. Now, we can conclude that CPFOA has a good balance between diversification $(exploration)$ and intensification $(exploitation)$, since it has not been trapped in local optima.

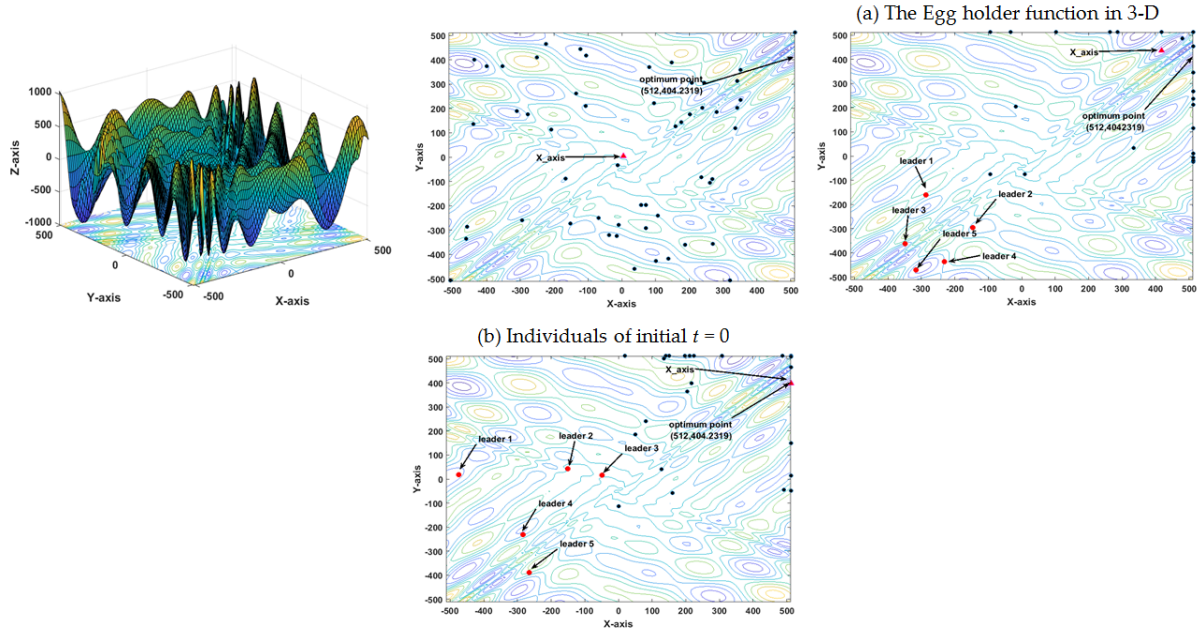


Figure 3. Convergence behavior of CPFOA on the Egg holder function.

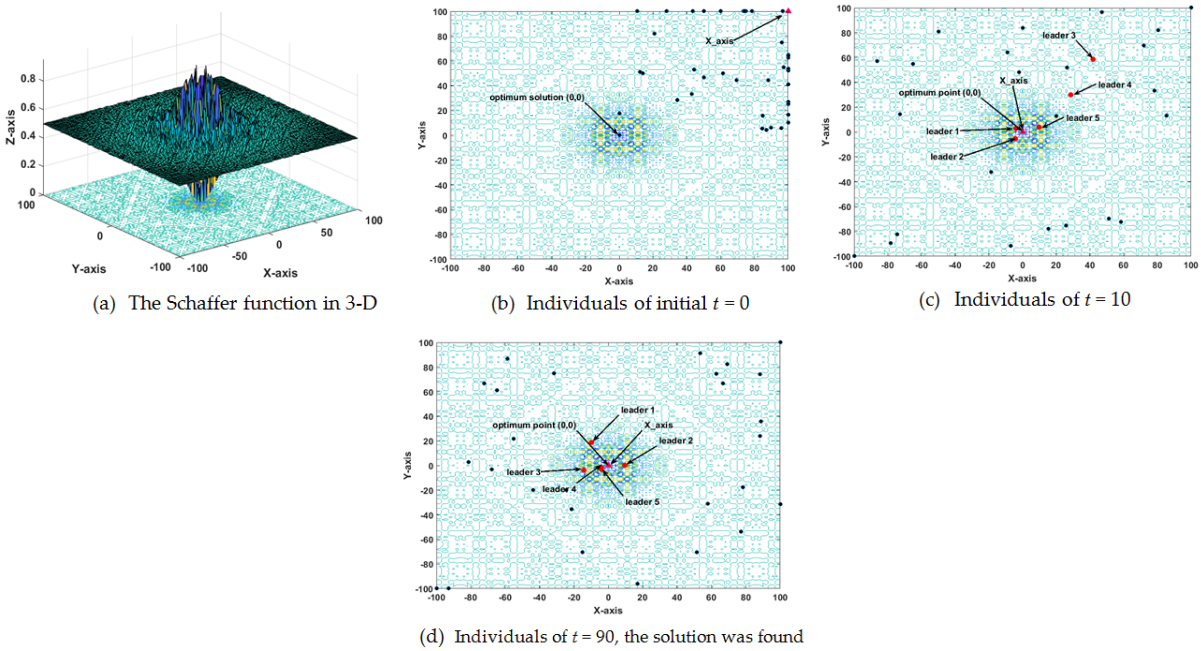


Figure 4. Convergence behavior of CPFOA on the Schaffer function.

5.2. Computational Time of CPFOA

The optimization problem should be solved in a short time. Therefore, a good meta-heuristic optimization should have a short computational time. The average computational time consumed by eights algorithms,

when they optimized 18 of the scalable functions taken from ([31]), is shown in Figure 5. From the figure, we found that CPFOA consumes quite a short computational time, compared to the other algorithms, especially when the dimension of the problem is 1000. Therefore, CPFOA can be used effectively in optimizing a large-scale problem.

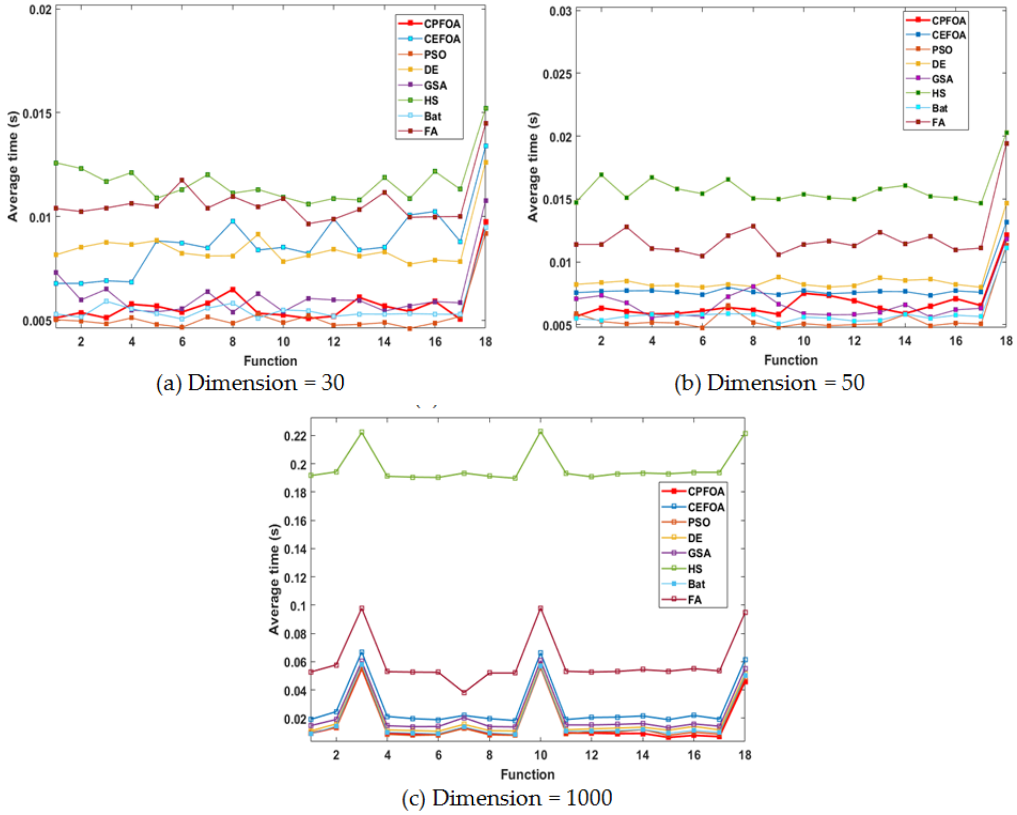


Figure 5. Comparison of the average computational time of 50 runs based on 18 benchmark functions (500 iterations).

5.3. The First Experiment: Comparison of CPFOA and Six FOA Variants

The average (ave) and standard deviation (std) of the final objective values, as well as the h and SR values, produced by seven algorithms performing 18 benchmark functions, with the dimensions of 30, 50 and 1000, are shown in Tables 5-7, respectively. To conclude the table, the totals of h and the average of the SR values are shown in the last two rows. The h values signify the results of the Rank-sum test that are used to compare the quality between FOA, LGMS, IFFO, MFOA, MSFOA, CEFOA and CPFOA. NA mean that CPFOA was not compared itself. The total h is represented as #1/#2/#3, where #1, #2, and #3 represent the number of wins, ties, and losses of the algorithm when it is compared with CPFOA,respectively. For example, the total h in the last row of Table 5, there is only the h value "+" of CEFOA = 1. It meant that from 18 benchmark functions, CEFOA win CPFOA an only 1 function. Moreover, the h value "-","=" of CEFOA = 11 and 6. It meant that from 18 benchmark functions, CEFOA losses CPFOA 11 and ties CPFOA 6, respectively. Hence, CPFOA outperforms CEFOA. The SR is the success rate. For each function, the lowest of ave and

std values, and the highest of SR value, are highlighted in boldface.

As can be seen from Tables 5 and 6, CPFOA has an excellent performance. It outperforms the competitive algorithms, FOA, LGMS, IFFO, MFOA, MSFOA and CEFOA, in every function except some values of CEFOA. For the comparison between CEFOA and CPFOA, in the uni-modal functions, $f1 - f10$, CEFOA has one win in function $f2$, and four ties in $f1$ and $f8 - f10$. For the multi-modal functions, $f11 - f18$, CEFOA has no wins, two ties in functions $f16$ and $f18$, and six losses in functions $f11 - f15$ and $f17$. Hence, CPFOA outperforms CEFOA. Moreover, the last rows of Tables 5 and 6 show that CPFOA has the highest average SR value (0.94).

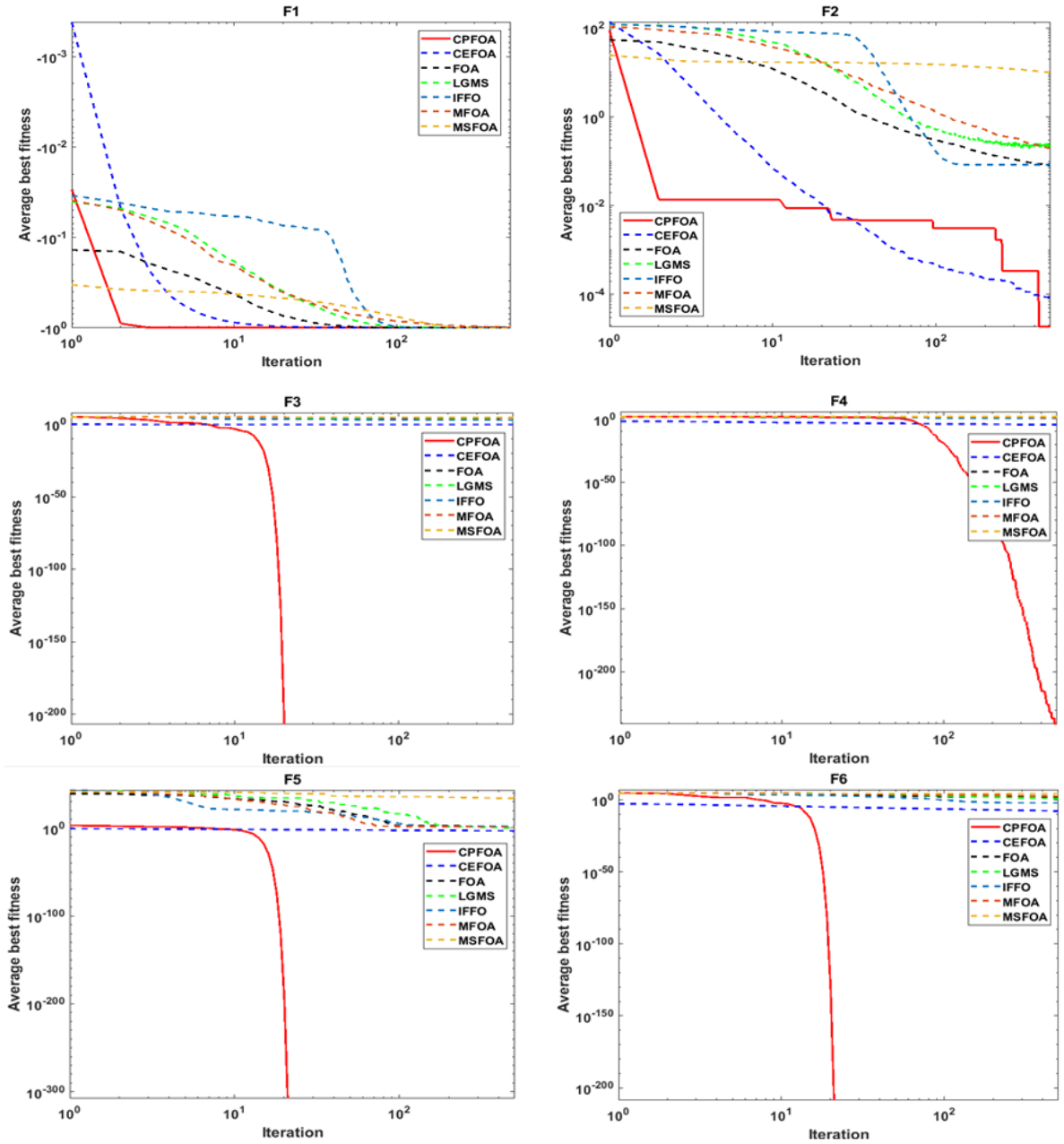
The results in relation to a large-scale problem are shown in Table 7. CPFOA still shows a very good performance, outperforming the competitive algorithms, FOA, LGMS, IFFO, MFOA, MSFOA and CEFOA, in every function except some values of CEFOA. For the comparison between CEFOA and CPFOA, for the uni-modal functions,$f1 - f10$,CEFOA has one win in function $f2$, six ties in $f1$, $f3$, $f5$, and $f8 - f10$, and three losses in functions $f4$, $f6$, and $f7$. For the multi-modal functions, $f11 - f18$, CEFOA has no wins, two ties in functions $f16$ and $f18$, and six losses in functions $f11 - f15$ and $f17$. In addition, from the last row of Table 7, it can be found that CPFOA has the highest average SR value (0.83).

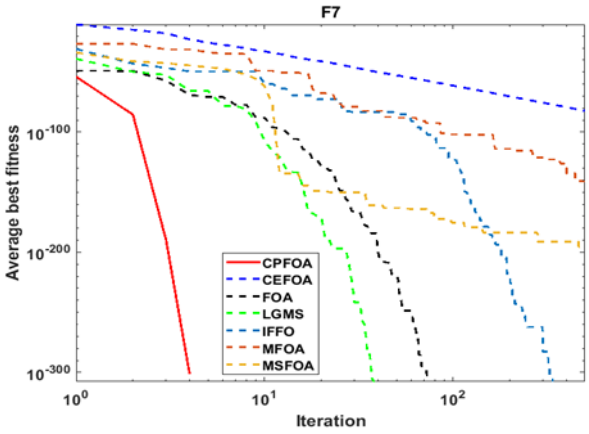
To confirm the efficiency of the CPFOA, the convergence graphs of all seven algorithms, when they are optimizing 18 benchmark test functions, with $dimension = 1000$, are shown in Figure 6, where the red lines represent the convergence graph of CPFOA. The graph is plotted in the $log - log$ scale, where the $x - axis$ is the number of iterations, and the $y - axis$ is the average fitness values, obtained at the corresponding iterations, of the algorithms. From the graphs, CPFOA can reach the best solution faster than the other six FOA variants.

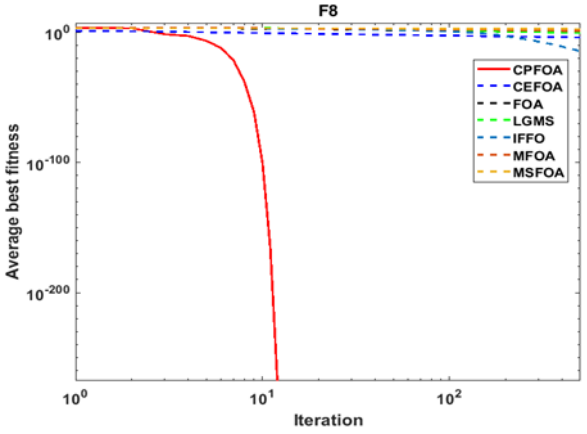
5.4. The Second Experiment: Comparison of the CPFOA and Meta-Heuristics Algorithms

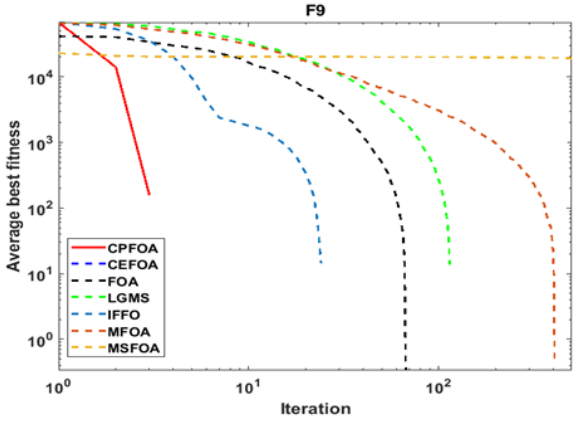
This section presents a comparison of the CPFOA and six meta-heuristics algorithms: PSO, DE, GSA, HS, BA, FA, and CEFOA. The average (ave) and standard deviation (std) of the final objective values, as well as the h and SR values, produced by the competitive algorithms performing 18 benchmark functions, with the dimension of 30, 50 and 1000, are shown in Tables 8-10, respectively. To conclude the table, the totals of h and the average of the SR values are shown in the last rows. The h values signify the results of the Rank-sum test which are used to compare the quality between PSO, DE, GSA, HS, BA, FA, CEFOA and CPFOA. NA mean that CPFOA was not compared itself. The total h is represented as #1/#2/#3, where #1, #2, and #3 represent the number of wins, ties, and losses of the algorithm, respectively. For example, the total h in the last row of Table 8, there is only the h value "+" of CEFOA = 1. It meant that from 18 benchmark functions, CEFOA win CPFOA an only 1 function. Moreover, the h value "-","=" of CEFOA = 11 and 6. It meant that from 18 benchmark functions, CEFOA losses CPFOA 11 and ties CPFOA 6 respectively. Hence, CPFOA outperforms CEFOA. The SR is the success rate. For each function, the lowest of ave and std values, and the highest of SR value, are highlighted in boldface.

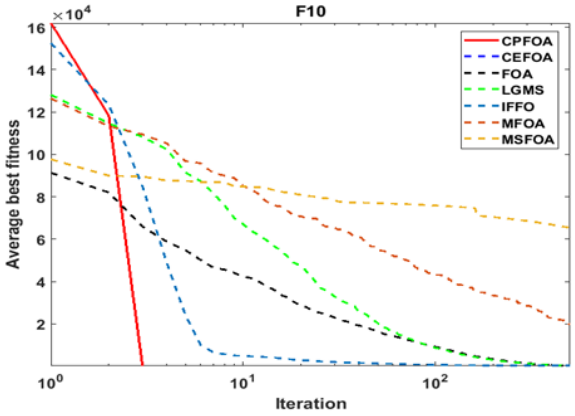
As can be seen from Tables 8 and 9, CPFOA has an excellent performance. It outperforms the competitive algorithms, PSO, DE, GSA, HS, BA, FA, and CEFOA in every function except some values of CEFOA. For the comparison between CEFOA and CPFOA, in the uni-modal functions, $f1 - f10$, CEFOA has one win in function $f2$, and four ties in $f1$ and $f8 - f10$. For the multi-modal functions, $f11 - f18$, CEFOA has no wins, two ties in functions $f16$ and $f18$, and six losses in functions $f11 - f15$ and $f17$. Hence, CPFOA outperforms CEFOA. Moreover, the last row of both Tables 8 and 9 show that CPFOA has the highest average SR value (0.94).

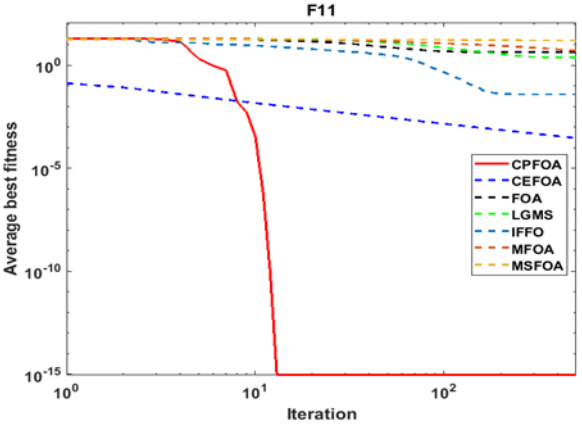

F1
F2
F3
F4
F5
F6
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA
Average best fitness
Iteration

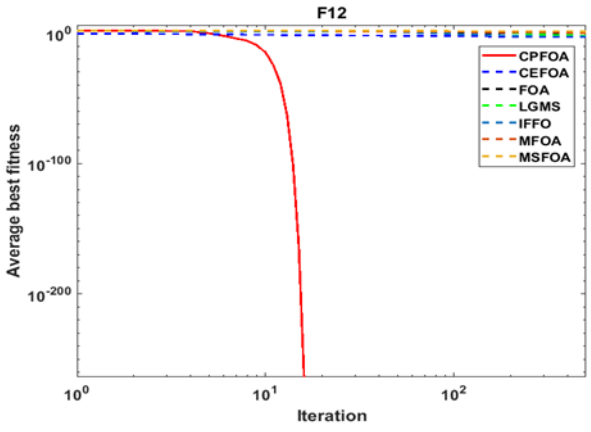

F7
Average best fitness
Iteration
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA


F8
Average best fitness
Iteration
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA


F9
Average best fitness
Iteration
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA


F10
Average best fitness
Iteration
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA


F11
Average best fitness
Iteration
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA


F12
Average best fitness
Iteration
CPFOA
CEFOA
FOA
LGMS
IFFO
MFOA
MSFOA

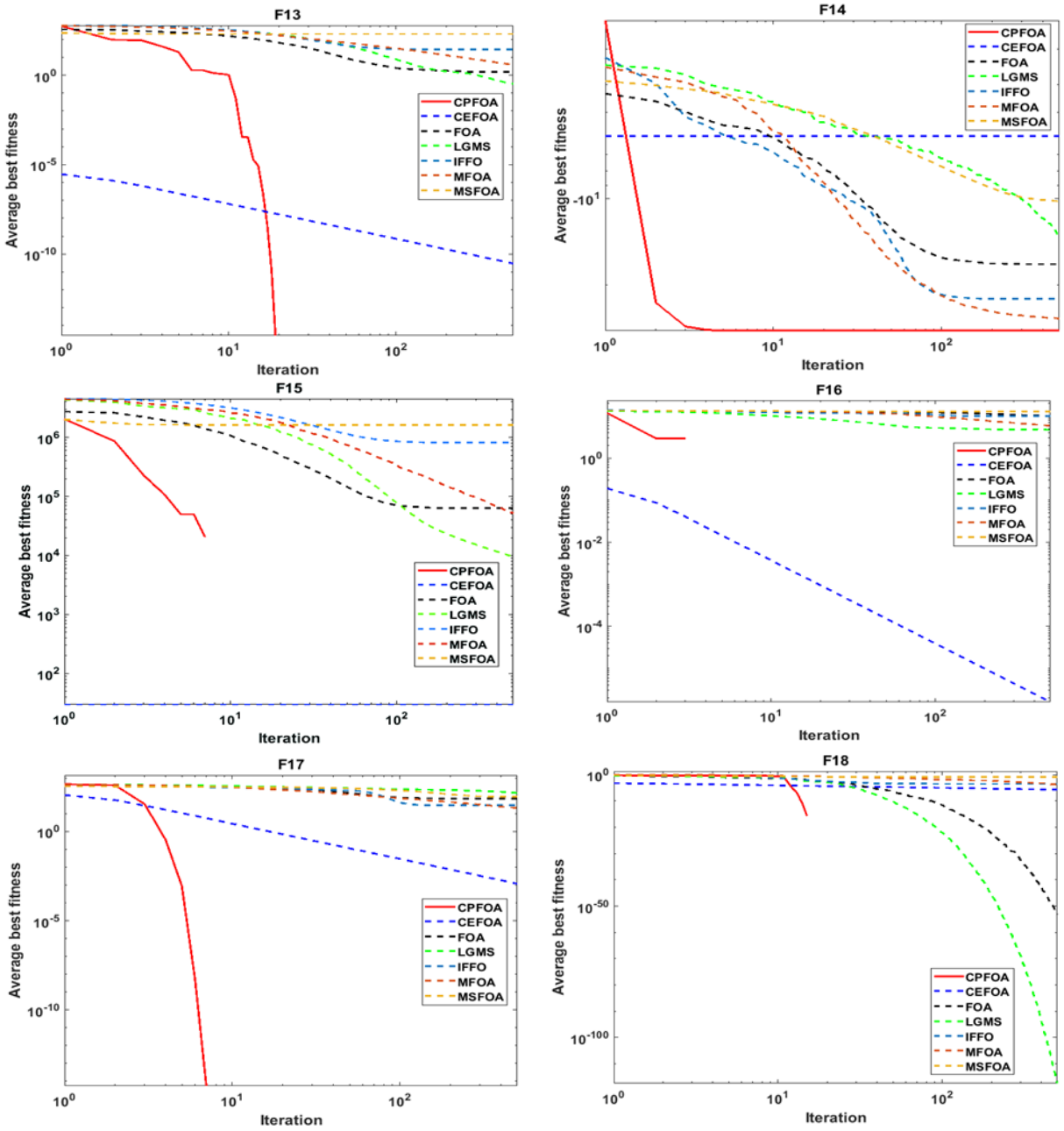


Figure 6. Convergence curves of 18 benchmark functions that compare CPFOA, CEFOA, FOA, LGMS, IFFO, MFOA and MSFOA. The dimension of the problem is 1000.

The results regarding a large-scale problem are shown in Table 10. CPFOA still shows a very good performance, outperforming the competitive algorithms, PSO, DE, GSA, HS, BA, FA, and CEFOA, in every function except some values of CEFOA. For the comparison between CEFOA and CPFOA, for the uni-modal functions, $f1 - f10$, CEFOA has one win in function $f2$, six ties in $f1, f3, f5$, and $f8 - f10$, and three losses in functions $f4, f6$, and $f7$. For the multi-modal functions, $f11 - f18$, CEFOA has no wins, two ties in functions $f16$ and $f18$, and six losses in functions $f11 - f15$ and $f17$. Hence, CPFOA

outperforms CEFOA. In addition, from the last row of Table 8, it can be found that CPFOA has the highest average SR value (0.83).

6. Applications

In this section, the proposed CPFOA is applied for (i) training the Multi-Layer Perceptron (MLP) to classify five datasets, and (ii) estimating the T-S fuzzy system parameters. MLPs and T-S fuzzy system method have been proposed as useful tools to model complex systems for process in chemical applications ([57], [58], [59]).

6.1. Bio-Medical Real-Life Classification Problems

The datasets are a synthesis dataset, 3-bits XOR, a small dataset, Iris, and three bio-medical datasets: Balloon, Breast cancer and Heart. The details of these datasets and parameter settings, including the MLP structure for solving these datasets, are taken from the literature ([60], [61]). Brief details of the dataset, for the implementation and performance comparison of the algorithms, are presented in Table 11. CPFOA is compared with six meta-heuristic algorithms in Table 4: JADE, BLPSO, CLPSO, GWO, MGWO and HAGWO ([61]). HAGWO was successfully tested using these five datasets. It is a hybrid nature-inspired optimization technique that has been constructed using a hybridization of the Mean Grey Wolf Optimizer (MGWO) and Whale Optimizer Algorithm (WOA). The parameters of GWO and HAGWO are set as in ([60], [61]). Each algorithm is coded and run in MATLAB environment. The convergent graphs, based on five datasets, are shown in Figure 7, where the red line represents the graph of CPFOA. From Figure 7, it can be found that CPFOA can produce the lowest MSE in every dataset. To confirm our claim, the statistical results, in which the minimum objective function and maximum objective function values are extracted, are shown in Table 12. A lower objective function value is better. In addition, the average and the standard deviation of the classification rate are shown in Table 12. A higher classification rate, with a lower standard deviation, is better.

The results from Table 12 are as follows:

(i) The XOR dataset contains 8 training and 8 testing samples. Each sample has 3 input attributes and 1 output. The outputs of the XOR dataset are the same as those of the input values. After encoding, the dimension of a fruit fly is 36. As can be seen in Table 12, the highest accuracy of 100% could be obtained by GWO, MGWO, HAGWO and CPFOA. However, CPFOA is the best, as it produces the lowest Best MSE Value (1.97×10^{-9}), Worst MSE Value (2.43×10^{-4}), ave (2.59×10^{-5}), and std (5.45×10^{-5}), compared with those of the other algorithms.

(ii) The Iris dataset contains 150 samples. Each sample has 4 attributes and 3 classes. The training and the testing sets are the same. This problem is harder than that of XOR, as it has three output classes. After encoding, the dimension of a fruit fly is 75. The results show that CPFOA can produce the highest classification rate (93.733%), which is highlighted in boldface. The second and third best competitors are two improved versions of GWO, HAGWO and MGWO, which could obtain an accuracy of 93.00% and 91.334%, respectively. The original GWO ranks forth, with an accuracy of (91.333%). Furthermore, CPFOA produces the lowest average MSE (5.39×10^{-2}), compared with those of the six competitive algorithms.

(iii) The Balloon dataset contains 18 training and 18 testing samples. Each sample has 4 input attributes and 2 output classes. After encoding, the dimension of a fruit fly is 55. As can obviously be seen from Table 12, this dataset is quite simple, because the several algorithms can produce the highest accuracy of 100%, except for CLPSO's accuracy, which is 89.75%. However, CPFOA produces the lowest Best MSE Value (1.87×10^{-20}), Worst MSE Value (2.82×10^{-8}), ave (1.51×10^{-9}) and std (6.29×10^{-9}) of MSE, compared with those of the six competitive algorithms.

(iv) The Breast cancer dataset contains 599 training and 100 testing samples, each of which has 9 input attributes and 2 output classes. After encoding, the dimension of a fruit fly is 209. Therefore, the computational time per iteration is quite long. However, the results confirm that CPFOA can produce the highest accuracy and the lowest values of Best MSE Value (1.21×10^{-3}), Worst MSE Value (1.70×10^{-3}), ave (1.67×10^{-3}) and std (1.19×10^{-3}) of the MSE.

(v) The Heart dataset, is the hardest problem among the four classification problems, containing 80 training and 187 testing samples, each of which has 22 input attributes and 2 classes. After encoding, the dimension of a fruit fly is 1081. It is a large-scale problem. As can be seen from Table 12, several algorithms could produce low accuracies. The accuracy of JADE, BLPSO, CLPSO, GWO, MGWO and HAGWO are 77.50%, 67.20%, 68.125%, 75.00%, 88.375%, respectively. HAGWO produces a very low accuracy of 47.625%. However, the CPFOA showed a very promising result, as it can reach the highest accuracy of 90.333%. Therefore, CPFOA outperforms the other algorithms.

To summarize, CPFOA has the ability and is suitable for training the Multi-Layer Perceptron (MLP) in order to solve real-life classification problems.

6.2. T-S Fuzzy System Parameter Extraction

This T-S fuzzy system parameter estimation has been conducted in several literary works, and seven of the existing works compare the other algorithms with CPFOA. The dataset is a Box and Jenkins gas furnace data set ([62]). It consists of 296 input and output measurements of a gas-furnace process. It is the collection of recorded data from a combustion process of a methane-air mixture. At each sampling time k, the input $x(k)$ is the gas flow rate, and the output $y(k)$ is the output CO_2 concentration.

To compare CPFOA with the other novel algorithms, we borrow a procedure from the literature, namely, the "parameter estimation of Takagi-Sugeno's fuzzy system using a heterogeneous cuckoo search algorithm" ([63]) by choosing $u(k), u(k-1), y(k-1)$ and $u(k-2)$, as the input variables of the model. To prepare the data, the first 148 input-output data were utilized as training data, and the last 148 were the testing data. The individual parameter settings of the simulation and the best competitor, HeCoS, are set as in ([63]), which is as follows: $P_a = 0.15, \alpha = 1.3, \beta = 1.3, \delta = 0.9$, the maximum iteration (Max_iter) = 2000, and the number of agents $(popsize) = 40$. CPFOA solve a very compact model as the number of rules in the model is 2–this is the smallest number of rules. After the encoding, the total number of parameters for a T-S fuzzy system model is 26.

The training results of CPFOA are visualized in Figure 8, and the testing results are shown in Figure 9. The blue line is the real data, and the red dashed line is the output of the model, trained by CPFOA. From those two figures, we have observed that the real data and the output from the model are not very different.

Based on the study in ([63]), the HeCoS optimizing iTaSuM claimed that it is the best model, with training and testing MSE values of 0.0271 and 0.138, respectively. We also re-run the HeCoS, and there is no better solution found. These two training and testing MSE values are not the lowest values, but the number of rules of the T-S fuzzy system is less than or equal to that of the other six approaches (model no. 1 through no. 6). Regarding CPFOA, from Table 13, it can be found that the training and testing MSEs of CPFOA do not obviously outperform that of HeCoS, but CPFOA can optimize a more compact model than HeCoS as the number of rules of the T-S fuzzy system trained by CPFOA is less than that of HeCoS. The parameters of the identified model are listed in Table 14.

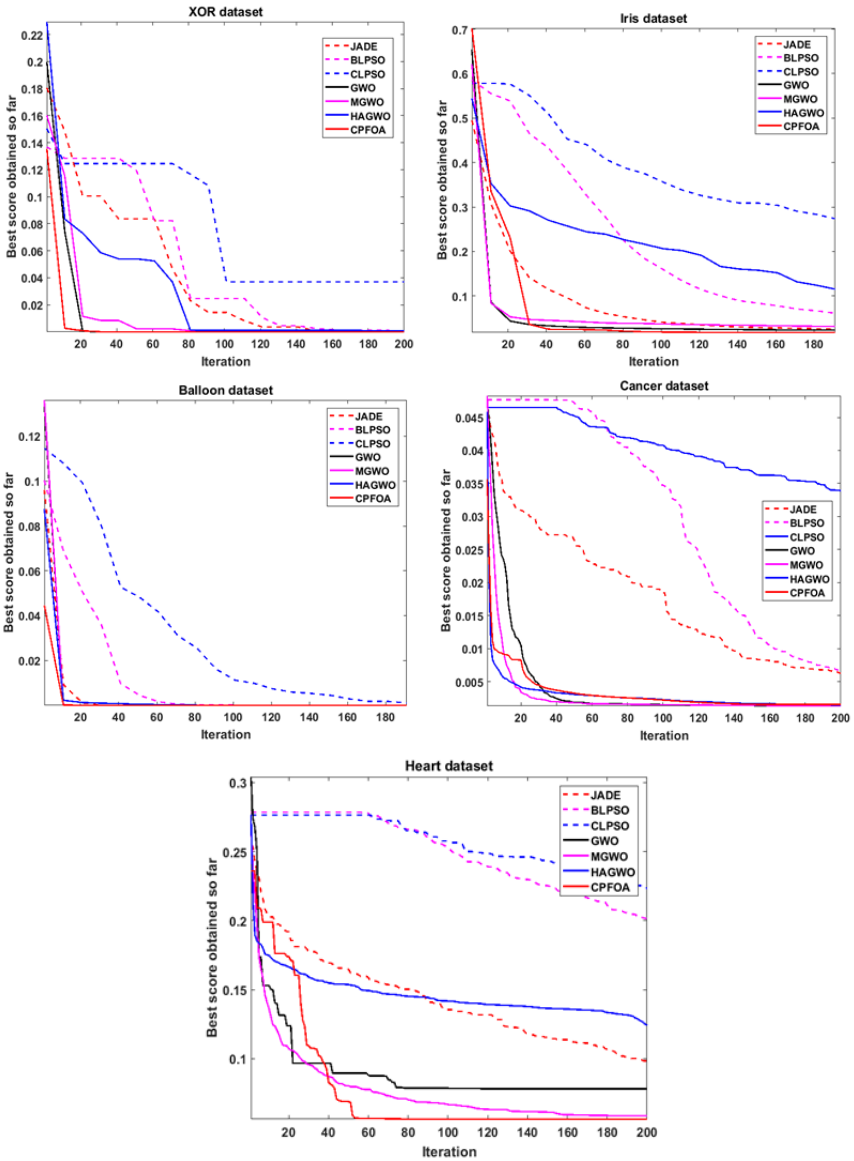


Figure 7. Convergence curves of XOR, Iris, Balloon, Cancer and Heart datasets that compare JADE, BLPSO, CLPSO, GWO, MGWO, HAGWO and CPFOA).

7. Conclusions

This paper addresses the problem of the low exploration ability of FOA and its variants. The cooperation of the multileader and the probabilistic random walk strategies forms the proposed CPFOA algorithm. CPFOA can smoothly transition from the exploration stage to the intensification stage. The experimental results show that the performance of CPFOA is improved, compared with the original FOA and the FOA variants, in finding the optimal solution.

The main characteristics of CPFOA in finding the optimal solution can be summarized as follows:

- The CPFOA uses the probabilistic random walk algorithm, with adaptive normalization, as the main procedure.
- The CPFOA uses the multileader strategy to further enhance the exploration ability.
- The population with two types of behavior can prevent the search from becoming trapped in a local optimum, whereas only one population behavior in the existing FOA variants can lead the algorithms to be easily trapped.
- The CPFOA demonstrated its promising performance in solving unconstrained function optimization problems, especially when the dimension of the problem is high.

We evaluated the CPFOA's performance in 18 well-known standard benchmark functions. The experimental results from the benchmark functions clearly illustrated that the CPFOA outperforms both the original FOA and the FOA variants, in terms of the convergence speed, the success rate, and the solution accuracy, in finding the optimal solution.

CPFOA is applied for training the MLPs in relation to classified real-life datasets, and the results demonstrate that it achieves a higher level of accuracy in the classification of the proposed CPFOA trainer than the competitive algorithms.

Moreover, CPFOA is applied for the T-S fuzzy system parameter extraction of a Box and Jenkins gas furnace data set. The results demonstrate that CPFOA can achieve a very promising accuracy of the T-S fuzzy system in modeling, when compared with the best known competitive algorithms.

Future work will involve applying CPFOA to optimize the multi-objective function problems, which are very challenging problems.

Author Contributions : The authors have contributed equally to this research and to the writing of this paper. W.A. designed the experiments, conceptualization, investigation, methodology, software and wrote the original draft of this paper. K.S. was responsible for supervision, performing the numerical experiments, conceptualization, investigation, methodology, software and the writing of the original draft of this paper. S.C. analyzed the numerical results and was responsible for conceptualization, investigation, methodology, software and the writing of the original draft of this paper. All authors have read and approved the final version of the manuscript.

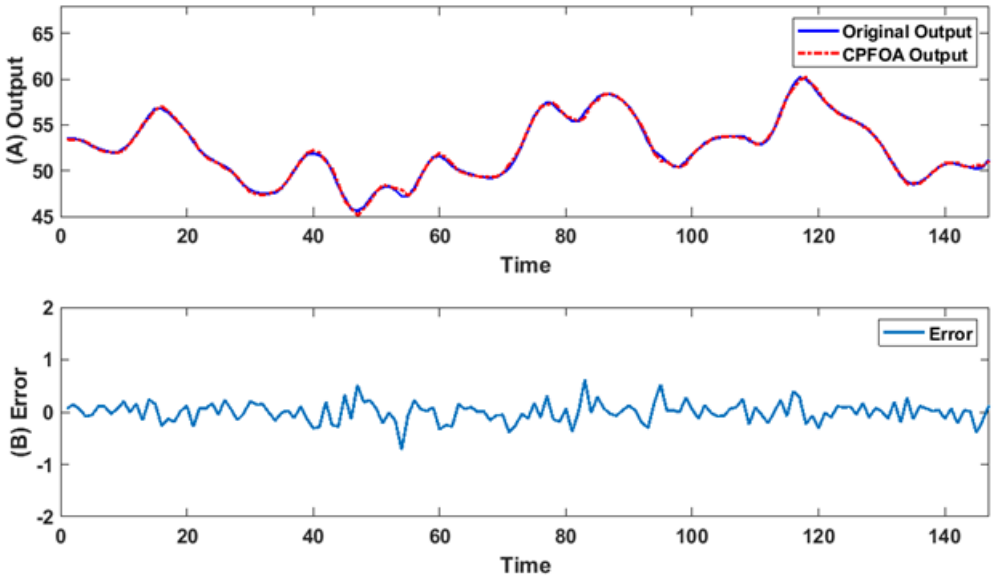


Figure 8. Comparison of iTaSuM optimized by CPFOA and the real system for Box and Jenkins gas furnace training data. (A) The real values and the CPFOA outputs for the data set. (B) The error between the real values and the CPFOA outputs).

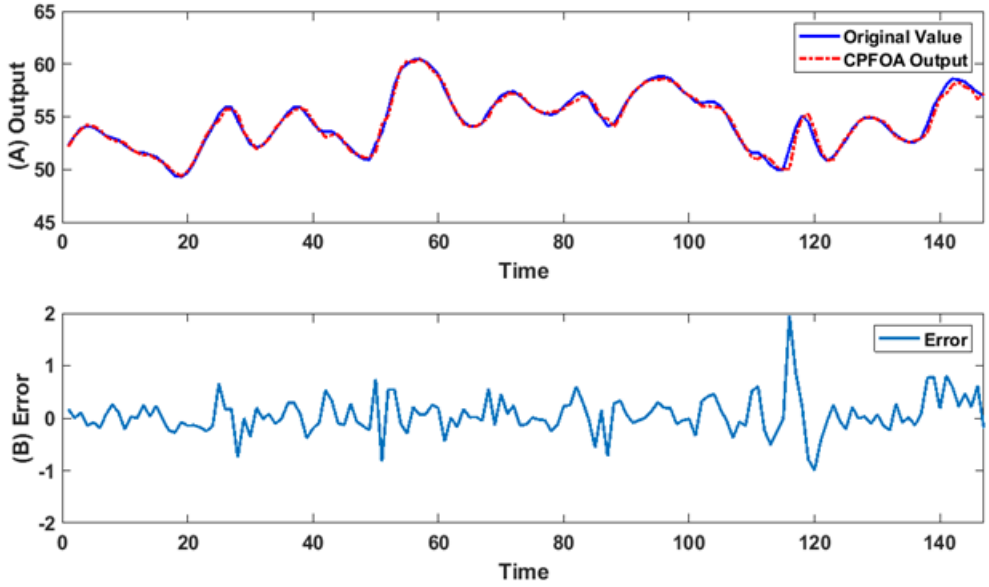


Figure 9. Comparison of iTaSuM optimized by CPFOA and the real system for Box and Jenkins gas furnace testing data. (A): The real values and the CPFOA outputs for the data set. (B) The error between the real values and the CPFOA outputs).

Table 1. The 18 standard benchmark functions ($f1-f10$ are the uni-modal, $f11-f18$ are the multi-modal functions).

No.	Function	x_i	$f(x)$
$f1$	$f(x)=-exp(-0.5\sum_{i=2}^{n}x_i^2)$	$(-1,1)$	-1
$f2$	$f(x)=\sum_{i=1}^{n}ix_i^4+rand(0,1)$	$(-1.28,1.28)$	0
$f3$	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_i)^2$	$(-100,100)$	0
$f4$	$f(x)=max(\mid x_i \mid)$	$(-100,100)$	0
$f5$	$f(x)=\sum_{i=1}^{n}\mid x_i \mid+\Pi_{i=1}^{n}\mid x_i \mid$	$(-10,10)$	0
$f6$	$f(x)=\sum_{i=1}^{n}x_i^2$	$(-100,100)$	0
$f7$	$f(x)=\sum_{i=1}^{n}\mid x_i \mid$	$(-1,1)$	0
$f8$	$f(x)=\sum_{i=1}^{n}ix_i^2$	$(-10,10)$	0
$f9$	$f(x)=\sum_{i=1}^{n}x_i^2-450$	$(-100,100)$	-450
$f10$	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_i)^2-450$	$(-100,100)$	-450
$f11$	$-exp(-0.2\sqrt{\frac{1}{n}\sum_{i=2}^{n}x_i^2})-20exp(\frac{1}{n}\sum_{i=2}^{n}\cos(2\pi x_i^2))$	$(-32,32)$	0
$f12$	$f(x)=\sum_{i=1}^{n}\mid x_i\sin(x_i)+0.1x_i \mid$	$(-10,10)$	0
$f13$	$f(x)=\frac{1}{400}\sum_{i=1}^{n}x_i^2-\Pi_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$(-600,600)$	0
$f14$	$f(x)=-\sum_{i=1}^{n}exp(-\frac{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}{8})\cos(4\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}})$	$(-5,5)$	$(1-n)$
$f15$	$f(x)=\sum_{i=1}^{n}x_i-1^2-\sum_{i=1}^{n}x_ix_{i-1}$	$(-n^2,n^2)$	$\frac{n(n+4)(1-n)}{6}$
$f16$	$f(x)=\sum_{i=1}^{n}(0.5+\frac{\sin^2\sqrt{100x_i^2+x_{i+1}^2}-0.5}{1+0.01(x_i^2-2x_ix_{i+1}+x_{i+1}^2)})$	$(-100,100)$	0
$f17$	$f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	$(-5.12,5.12)$	0
$f18$	$f(x)=1-\cos(2\pi(\sqrt{\sum_{i=1}^{n}x_{i+1}^2}))+0.1\sqrt{\sum_{i=1}^{n}x_{i+1}^2}$	$(-100,100)$	0

Table 2. The initial parameters of the FOA variants.

Algorithm	Strategy	Reference
FOA	A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example	Pan([45])
LGMS	LGMS-FOA: An Improved Fruit Fly Optimization Algorithm for Solving Optimization Problems	Shan([15])
IFFO	An improved fruit fly optimization algorithm for continuous function optimization problems	Pan([16])
MFOA	On a novel multi-swarm fruit fly optimization algorithm and its application	Yuan([18])
MSFOA	A novel multi-scale cooperative mutation fruit fly optimization algorithm	Zhang [30]
CEFOA	Novel fruit fly optimization algorithm with trend search and co-evolution	Han([31])

Table 3. The initial parameters of the six meta-heuristic algorithms.

Algorithm	Strategy	Reference
PSO	Particle swarm optimization	Kennedy([5])
DE	Differential Evolution-A Simple and Efficient Heuristic for global Optimization over Continuous Spaces	Storn([13])
GSA	GSA: A Gravitational Search Algorithm	Rashedi([46])
HS	A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice	Lee([47])
BA	A New Metaheuristic Bat-Inspired Algorithm	Yang([9])
FA	A comprehensive review of firefly algorithms	Fister([48])

Table 4. The initial parameters of the six state-of Cthe-art meta-heuristic algorithms.

Algorithm	Strategy	Reference
BLPSO	Biogeography-based learning particle swarm optimization	Chen([49])
CLPSO	Comprehensive learning particle swarm optimizer for global optimization of multimodal functions	Liang([50])
CoDE	Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters	Wang([51])
JADE	JADE: Adaptive Differential Evolution With Optional External Archive	Zhang([52])
MGWO	A Modified Mean Gray Wolf Optimization	Singh([53])
MFPA	A Modified Flower Pollination Algorithm for Global Optimization	Nabil([54])

Table 5: Solution quality comparisons of the CPFOA and variant FOAs performing 18 benchmark functions (Dimension = 30).

Function	Criteria	FOA	LGMS	FFO	MFOA	MSFOA	CEFOA	CPFOA
f_1	ave	-3.28×10^{-3}	4.50×10^{-2}	$\mathbf{-1.00 \times 10^{0}}$	-1.86×10^{1}	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$
	std	1.65×10^{-3}	1.39×10^{-2}	1.85×10^{-5}	1.62×10^{-9}	1.62×10^{-16}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	1	0	1	1	1
	h	-	-	-	-	-	=	NA
f_2	ave	3.00×10^{2}	3.39×10^{0}	3.14×10^{-3}	3.36×10^{-3}	1.08×10^{-1}	$\mathbf{1.02 \times 10^{-4}}$	1.00×10^{-3}
	std	8.92×10^{1}	6.99×10^{0}	1.84×10^{-3}	1.90×10^{-3}	3.25×10^{-2}	$\mathbf{1.98 \times 10^{-4}}$	9.21×10^{-4}
	SR	0	0	0	0	0	0	0
	h	-	-	-	-	-	+	NA
f_3	ave	2.42×10^{5}	1.43×10^{3}	2.32×10^{1}	3.15×10^{-2}	4.60×10^{3}	7.29×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	std	1.47×10^{5}	3.87×10^{2}	3.07×10^{1}	2.19×10^{-2}	3.48×10^{3}	1.98×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_4	ave	9.55×10^{0}	1.01×10^{0}	1.48×10^{-1}	2.14×10^{-3}	1.64×10^{1}	1.98×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	4.51×10^{0}	8.10×10^{-2}	1.14×10^{-1}	1.11×10^{-3}	6.42×10^{1}	3.46×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_5	ave	1.34×10^{3}	1.29×10^{0}	1.21×10^{0}	4.38×10^{-2}	1.53×10^{0}	2.67×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	1.91×10^{2}	1.53×10^{0}	6.26×10^{-1}	2.64×10^{-2}	2.49×10^{0}	1.06×10^{-6}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_6	ave	9.82×10^{4}	7.88×10^{0}	4.70×10^{-1}	1.09×10^{-4}	1.13×10^{-17}	1.95×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	1.56×10^{4}	1.27×10^{0}	5.35×10^{-1}	1.22×10^{-4}	2.20×10^{-18}	2.08×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	1	1	1
	h	-	-	-	-	-	-	NA
f_7	ave	2.54×10^{0}	3.00×10^{0}	3.59×10^{-45}	8.64×10^{-40}	$\mathbf{0.00 \times 10^{0}}$	1.45×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	5.68×10^{0}	1.64×10^{-26}	1.97×10^{-44}	1.93×10^{-46}	$\mathbf{0.00 \times 10^{0}}$	1.94×10^{-11}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	1	0	1	1	1
	h	-	-	-	-	-	-	NA
f_8	ave	2.98×10^{3}	9.09×10^{1}	1.08×10^{-2}	2.52×10^{-6}	3.45×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	6.20×10^{2}	8.95×10^{1}	1.20×10^{-2}	2.38×10^{-6}	5.75×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_9	ave	9.03×10^{4}	-4.42×10^{2}	-4.49×10^{2}	-4.40×10^{2}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	2.33×10^{4}	1.31×10^{1}	3.78×10^{-1}	1.78×10^{-4}	8.18×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{10}	ave	4.56×10^{5}	8.79×10^{2}	-4.48×10^{2}	-4.45×10^{2}	4.93×10^{3}	-4.50×10^{2}	-4.50×10^{2}
	std	3.96×10^{5}	4.90×10^{2}	4.37×10^{1}	2.89×10^{-2}	3.44×10^{3}	0.00×10^{0}	0.00×10^{0}
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{11}	ave	2.05×10^{1}	3.46×10^{1}	1.80×10^{-1}	1.43×10^{0}	2.76×10^{1}	1.60×10^{-10}	$\mathbf{8.88 \times 10^{-16}}$
	std	1.84×10^{-1}	1.33×10^{-1}	1.34×10^{-1}	1.96×10^{0}	4.60×10^{1}	4.01×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{12}	ave	6.96×10^{1}	6.78×10^{1}	1.56×10^{-2}	1.21×10^{0}	7.64×10^{1}	9.15×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	std	1.20×10^{1}	6.26×10^{-1}	1.15×10^{-2}	2.70×10^{0}	2.92×10^{1}	1.55×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{13}	ave	8.51×10^{2}	1.01×10^{1}	2.27×10^{-1}	1.52×10^{-4}	1.09×10^{-2}	8.89×10^{-16}	$\mathbf{0.00 \times 10^{0}}$
	std	2.76×10^{2}	1.98×10^{-2}	2.57×10^{-1}	7.46×10^{-5}	1.29×10^{-2}	4.70×10^{-15}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{14}	ave	-3.17×10^{0}	1.80×10^{1}	2.13×10^{1}	-2.14×10^{1}	-8.54×10^{1}	-2.73×10^{1}	$\mathbf{-2.70 \times 10^{1}}$
	std	1.08×10^{0}	1.30×10^{1}	8.91×10^{-1}	2.82×10^{-4}	1.77×10^{1}	1.05×10^{0}	$\mathbf{2.80 \times 10^{-2}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{15}	ave	6.04×10^{6}	-5.18×10^{2}	5.30×10^{1}	3.06×10^{1}	3.19×10^{4}	-1.89×10^{4}	$\mathbf{-3.32 \times 10^{3}}$
	std	7.89×10^{5}	8.13×10^{1}	2.67×10^{1}	3.67×10^{-1}	2.82×10^{4}	3.60×10^{3}	$\mathbf{6.38 \times 10^{1}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_{16}	ave	1.33×10^{1}	8.40×10^{1}	9.12×10^{-1}	9.51×10^{-7}	1.23×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	4.26×10^{-1}	7.17×10^{-1}	2.71×10^{-1}	1.89×10^{-6}	3.84×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{17}	ave	4.19×10^{2}	4.19×10^{2}	1.17×10^{-1}	1.05×10^{2}	1.33×10^{2}	1.46×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	std	5.27×10^{1}	2.06×10^{1}	1.58×10^{-1}	1.31×10^{2}	3.90×10^{1}	4.42×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_{18}	ave	3.72×10^{0}	6.60×10^{-2}	1.81×10^{-4}	9.99×10^{-2}	6.48×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	3.90×10^{0}	4.97×10^{-2}	2.55×10^{-4}	3.45×10^{-7}	7.76×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	1	1	1
	h	-	-	-	-	-	=	NA
$total$	$+$	0	0	0	0	0	1	NA
	$-$	18	18	18	18	18	11	NA
h	=	0	0	0	0	0	6	NA
	$AverageSR$	0.00	0.00	0.11	0.00	0.22	0.72	**0.94**

Table 6: Solution quality comparisons of the CPFOA and variant FOAs performing 18 benchmark functions (Dimension = 50).

Function	Criteria	FOA	LGMS	FFO	MFOA	MSFOA	CEFOA	CPFOA
f_1	ave	-5.05×10^{-4}	4.17×10^{-3}	$\mathbf{-1.00 \times 10^{0}}$	-9.92×10^{-1}	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$
	std	1.21×10^{-4}	2.44×10^{-3}	3.03×10^{-5}	3.18×10^{-9}	2.96×10^{-8}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	1	0	1	1	1
	h	-	-	-	-	-	=	NA
f_2	ave	7.69×10^{2}	1.29×10^{2}	4.99×10^{-3}	3.80×10^{-3}	5.38×10^{-1}	$\mathbf{9.52 \times 10^{-5}}$	1.03×10^{-3}
	std	1.09×10^{2}	1.76×10	3.39×10^{-3}	1.56×10^{-3}	1.88×10^{-1}	$\mathbf{2.08 \times 10^{-4}}$	8.76×10^{-4}
	SR	0	0	0	0	0	0	0
	h	-	-	-	-	-	+	NA
f_3	ave	1.22×10^{6}	6.49×10^{3}	4.12×10^{1}	7.52×10^{-2}	5.06×10^{4}	1.02×10^{-13}	$\mathbf{0.00 \times 10^{0}}$
	std	1.03×10^{6}	1.68×10^{3}	4.45×10^{1}	3.17×10^{-2}	1.89×10^{4}	2.10×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_4	ave	9.69×10^{1}	1.07×10^{1}	1.95×10^{-1}	1.26×10^{-3}	4.70×10^{1}	1.99×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	2.37×10^{1}	7.91×10^{-2}	1.49×10^{-1}	2.08×10^{-4}	7.62×10^{1}	3.50×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_5	ave	2.42×10^{3}	2.23×10^{1}	1.71×10^{1}	8.01×10^{-2}	1.64×10^{2}	1.97×10^{-29}	$\mathbf{0.00 \times 10^{0}}$
	std	1.80×10^{2}	1.63×10^{1}	1.01×10^{1}	5.92×10^{-2}	6.57×10^{1}	1.01×10^{-6}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_6	ave	1.59×10^{5}	1.53×10^{1}	2.08×10^{-1}	1.32×10^{-4}	8.76×10^{-5}	1.02×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	1.73×10^{4}	1.90×10^{1}	2.58×10^{-1}	1.14×10^{-4}	4.37×10^{-4}	1.88×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	1	1	1
	h	-	-	-	-	-	-	NA
f_7	ave	2.57×10^{-1}	5.19×10^{-53}	2.91×10^{-22}	4.98×10^{-44}	$\mathbf{0.00 \times 10^{0}}$	1.95×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	4.33×10^{-1}	2.84×10^{-52}	2.00×10^{-21}	3.50×10^{-43}	$\mathbf{0.00 \times 10^{0}}$	1.48×10^{-11}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	1	1	0	1	1	1
	h	-	-	-	-	-	-	NA
f_8	ave	9.81×10^{3}	2.82×10^{2}	2.57×10^{-2}	1.38×10^{-5}	6.24×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	3.15×10^{3}	2.27×10^{1}	4.39×10^{-2}	1.09×10^{-5}	4.80×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_9	ave	1.65×10^{5}	-4.46×10^{2}	-4.48×10^{2}	-4.45×10^{2}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	1.90×10^{4}	5.82×10^{-1}	2.23×10^{0}	9.61×10^{-5}	1.88×10^{-4}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	1	1	1
	h	-	-	-	-	-	=	NA
f_{10}	ave	3.51×10^{6}	-2.53×10^{2}	-2.60×10^{1}	-4.35×10^{2}	4.80×10^{4}	$\mathbf{-4.50E2}$	$\mathbf{-4.50E2}$
	std	2.99×10^{6}	3.50×10^{2}	4.23×10^{2}	1.45×10^{-1}	1.64×10^{4}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{11}	ave	2.06×10^{1}	3.53×10^{1}	1.19×10^{-1}	2.50×10^{-3}	1.15×10^{1}	1.69×10^{-10}	$\mathbf{8.88E-16}$
	std	1.68×10^{-1}	1.12×10^{-1}	1.17×10^{-1}	6.19×10^{-4}	7.12×10^{1}	2.13×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{12}	ave	1.29×10^{2}	1.21×10^{1}	1.93×10^{-2}	1.57×10^{-3}	2.03×10^{1}	1.05×10^{-15}	$\mathbf{0.00 \times 10^{0}}$
	std	1.71×10^{1}	9.39×10^{-1}	1.83×10^{-2}	7.03×10^{-4}	5.40×10^{1}	1.95×10^{-13}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{13}	ave	1.37×10^{3}	1.04×10^{1}	2.01×10^{-1}	2.47×10^{-4}	3.23×10^{-2}	1.09×10^{-17}	$\mathbf{0.00 \times 10^{0}}$
	std	2.77×10^{2}	8.73×10^{-3}	2.05×10^{-1}	1.17×10^{-4}	1.55×10^{-2}	8.70×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{14}	ave	-6.00×10^{0}	2.83×10^{1}	3.38×10^{1}	-3.62×10^{1}	-1.27×10^{1}	-2.58×10^{1}	$\mathbf{-4.45 \times 10^{1}}$
	std	1.80×10^{0}	1.41×10^{1}	1.15×10^{1}	3.77×10^{-4}	2.04×10^{1}	1.52×10^{0}	$\mathbf{1.56 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_{15}	ave	8.73×10^{7}	1.03×10^{3}	3.90×10^{8}	5.47×10^{1}	9.22×10^{5}	-1.69×10^{4}	$\mathbf{-8.60 \times 10^{3}}$
	std	2.00×10^{7}	5.99×10^{2}	5.16×10^{8}	2.60×10^{0}	4.47×10^{5}	2.64×10^{3}	$\mathbf{2.02 \times 10^{2}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_{16}	ave	2.30×10^{1}	1.65×10^{1}	9.40×10^{-1}	1.12×10^{-6}	2.18×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	4.12×10^{-1}	9.15×10^{-1}	3.19×10^{-1}	1.38×10^{-6}	5.77×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{17}	ave	7.63×10^{2}	9.15×10^{2}	1.92×10^{-1}	4.01×10^{-5}	1.47×10^{4}	1.06×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	std	6.98×10^{1}	3.24×10^{1}	1.95×10^{-1}	3.32×10^{-5}	2.21×10^{2}	2.27×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	NA	2.14100
f_{18}	ave	2.14×10^{0}	3.53×10^{-2}	1.60×10^{-4}	9.99×10^{-2}	6.47×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	3.72×10^{0}	4.68×10^{-2}	2.51×10^{-4}	2.09×10^{-6}	6.24×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	1	1	1
	h	-	-	-	-	-	=	NA
Average SR		0.00	0.05	0.11	0.00	0.27	0.72	**0.94**

Table 7: Solution quality comparisons of the CPFOA and variant FOAs performing 18 benchmark functions (Dimension = 1000).

Function	Criteria	FOA	LGMS	FFO	MFOA	MSFOA	CEFOA	CPFOA
f_1	ave	-1.87×10^{-35}	7.20×10^{-62}	$\mathbf{-1.00 \times 10^{0}}$	-9.98×10^{-6}	-3.14×10^{-54}	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$
	std	4.74×10^{-35}	8.73×10^{-62}	1.62×10^{-4}	2.07×10^{-4}	8.37×10^{-54}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_2	ave	4.01×10^{5}	7.36×10^{4}	2.21×10^{-2}	1.24×10^{-2}	1.60×10^{5}	$\mathbf{0.00 \times 10^{0}}$	2.53×10^{-4}
	std	1.85×10^{4}	2.29×10^{3}	1.28×10^{-2}	4.92×10^{-3}	5.16×10^{3}	$\mathbf{0.00 \times 10^{0}}$	2.55×10^{-4}
	SR	0	0	0	0	0	1	0
	h	-	-	-	-	-	+	NA
f_3	ave	1.24×10^{9}	7.01×10^{7}	4.96×10^{2}	5.33×10^{3}	3.26×10^{7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	1.16×10^{9}	3.02×10^{6}	7.53×10^{2}	4.70×10^{3}	9.74×10^{6}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_4	ave	9.99×10^{1}	1.36×10^{1}	1.49×10^{-1}	4.71×10^{-3}	9.71×10^{1}	1.80×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	1.90×10^{-1}	1.47×10^{-2}	1.10×10^{-1}	1.91×10^{-3}	5.75×10^{-1}	1.06×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_5	ave	4.98×10^{4}	5.02×10^{2}	1.12×10^{1}	4.45×10^{0}	4.13×10^{3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	8.52×10^{2}	1.01×10^{1}	8.06×10^{1}	3.26×10^{0}	6.01×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_6	ave	3.35×10^{6}	3.67×10^{2}	2.34×10^{1}	1.98×10^{-2}	2.53×10^{6}	1.74×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	7.07×10^{4}	8.96×10^{1}	3.61×10^{1}	2.50×10^{-2}	4.64×10^{4}	3.28×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_7	ave	1.92×10^{0}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	7.57×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	1.15×10^{-13}	$\mathbf{0.00 \times 10^{0}}$
	std	1.11×10^{0}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	1.88×10^{-5}	$\mathbf{0.00 \times 10^{0}}$	1.72×10^{-11}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	1	1	0	1	1	1
	h	-	=	=	-	=	-	NA
f_8	ave	4.46×10^{6}	1.41×10^{5}	2.97×10^{1}	3.66×10^{-2}	3.06×10^{6}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	8.13×10^{4}	2.37×10^{3}	3.97×10^{1}	3.60×10^{-2}	7.01×10^{4}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_9	ave	3.35×10^{6}	$?3.66 \times 10^{2}$	$?4.48 \times 10^{2}$	$?4.40 \times 10^{2}$	2.52×10^{6}	$\mathbf{?4.50 \times 10^{2}}$	$\mathbf{?4.50 \times 10^{2}}$
	std	7.25×10^{4}	3.21×10^{1}	2.23×10^{1}	2.05×10^{-2}	5.85×10^{4}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{10}	ave	1.57×10^{9}	6.34×10^{5}	4.23×10^{2}	4.66×10^{3}	7.77×10^{6}	$\mathbf{?4.50 \times 10^{2}}$	$\mathbf{?4.50 \times 10^{2}}$
	std	2.72×10^{9}	1.14×10^{6}	1.22×10^{2}	4.36×10^{3}	3.64×10^{4}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{11}	ave	2.11×10^{1}	3.81×10^{1}	5.96×10^{-2}	2.86×10^{0}	2.07×10^{1}	7.64×10^{-11}	$\mathbf{8.88E-16}$
	std	2.71×10^{-2}	1.60×10^{-2}	5.02×10^{-2}	1.59×10^{0}	2.66×10^{-2}	2.00×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{12}	ave	2.93×10^{3}	3.06×10^{2}	2.00×10^{-1}	3.01×10^{-2}	2.18×10^{3}	1.15×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	std	4.12×10^{1}	3.75×10^{1}	1.58×10^{-1}	1.41×10^{-2}	5.35×10^{1}	1.51×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{13}	ave	2.96×10^{4}	1.81×10^{1}	1.76×10^{-1}	3.16×10^{-3}	2.26×10^{4}	2.09×10^{-16}	$\mathbf{0.00 \times 10^{0}}$
	std	8.47×10^{2}	1.98×10^{-2}	2.08×10^{-1}	1.40×10^{-3}	4.17×10^{2}	3.79×10^{-15}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	NA
f_{14}	ave	$?2.31 \times 10^{1}$	4.52×10^{2}	$?5.29 \times 10^{2}$	-7.37×10^{2}	-1.11×10^{2}	-2.32×10^{1}	$\mathbf{-9.23E2}$
	std	3.91×10^{1}	8.21×10^{1}	6.83×10^{0}	8.41×10^{-3}	1.29×10^{1}	1.05×10^{0}	$\mathbf{6.45}$
	SR	0	0	0	0	0	0	0
	h	-	-	-	-	-	-	NA
f_{15}	ave	3.37×10^{14}	8.36×10^{9}	3.90×10^{8}	1.00×10^{5}	2.23×10^{14}	-1.29×10^{4}	$\mathbf{-1.02E6}$
	std	$1.29E \times 10^{13}$	3.83×10^{8}	5.16×10^{8}	3.39×10^{4}	6.18×10^{12}	2.54×10^{3}	$\mathbf{1.64E3}$
	SR	0	0	0	0	0	0	0
	h	-	-	-	-	-	-	NA
f_{16}	ave	4.89×10^{2}	4.63×10^{2}	3.25×10^{1}	3.86×10^{-4}	4.78×10^{2}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	1.14×10^{1}	3.87×10^{1}	4.74×10^{1}	7.92×10^{-4}	4.52×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
f_{17}	ave	1.79×10^{4}	1.02×10^{4}	1.39×10^{1}	7.96×10^{2}	1.47×10^{4}	1.14×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	std	3.77×10^{2}	1.15×10^{2}	1.25×10^{1}	4.45×10^{2}	1.61×10^{2}	3.27×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	NA
f_{18}	ave	3.93×10^{1}	4.05×10^{-2}	1.02×10^{-4}	9.99×10^{-2}	9.33×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	3.99×10^{1}	4.73×10^{-2}	9.30×10^{-5}	1.88×10^{-5}	9.48×10^{-14}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	NA
$total$	$+$	0	0	0	0	0	1	NA
	$-$	18	18	18	18	18	11	NA
h	=	0	0	0	0	0	6	NA
Average SR		0.00	0.05	0.05	0.00	0.05	0.77	**0.83**

Table 8: Solution quality comparisons of the CPFOA and the meta-heuristic algorithms performing 18 benchmark functions (Dimension = 30).

Function	Criteria	PSO	DE	GSA	HS	BA	FA	CPFOA	CPFOA
f_1	ave	-0.99×10^{0}	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$	$-9.83E-1$	$\mathbf{-1.00 \times 10^{0}}$	$-9.99E-1$	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$
	std	3.75×10^{3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$4.11E-3$	$1.51E-6$	$6.34E-4$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0.03	0.73	1	0	1	0	1	1
	h	-	=	=	-	=	-	=	NA
f_2	ave	7.94×10^{-2}	2.12×10^{-1}	8.34×10^{-2}	1.99×10^{-1}	9.95×10^{1}	3.59×10^{-2}	$\mathbf{1.02 \times 10^{-4}}$	$1.00E-3$
	std	4.07×10^{-2}	8.95×10^{-2}	3.02×10^{-2}	5.29×10^{-2}	5.92×10^{1}	1.86×10^{-2}	$\mathbf{1.98 \times 10^{-4}}$	$9.21E-4$
	SR	0	0	0	0	0	0	0	0
	h	-	-	-	-	-	-	+	NA
f_3	ave	8.28×10^{2}	6.89×10^{2}	9.49×10^{2}	2.11×10^{4}	5.29×10^{4}	1.16×10^{3}	7.29×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	std	1.05×10^{3}	4.97×10^{2}	3.11×10^{2}	5.78×10^{3}	2.27×10^{4}	4.47×10^{2}	1.98×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_4	ave	1.59×10^{1}	2.65×10^{1}	7.15×10^{1}	2.25×10^{1}	5.18×10^{1}	8.37×10^{-2}	1.98×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	7.79×10^{1}	6.47×10^{1}	2.41×10^{1}	2.34×10^{0}	7.41×10^{1}	1.84×10^{-2}	3.46×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_5	ave	1.87×10^{2}	1.18×10^{1}	1.95×10^{2}	1.81×10^{1}	5.41×10^{33}	1.51×10^{1}	2.67×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	1.26×10^{2}	3.90×10^{1}	4.77×10^{1}	4.22×10^{0}	2.27×10^{34}	2.13×10^{1}	1.06×10^{-6}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_6	ave	1.65×10^{2}	7.90×10^{1}	7.94×10^{-3}	3.21×10^{2}	2.12×10^{4}	5.71×10^{-3}	1.95×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	2.30×10^{2}	1.98×10^{1}	4.35×10^{-2}	8.92×10^{1}	5.47×10^{3}	2.43×10^{-3}	2.08×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_7	ave	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$1.45E-12$	$\mathbf{0.00 \times 10^{0}}$
	std	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$1.94E-11$	$\mathbf{0.00 \times 10^{0}}$
	SR	1	1	1	1	1	1	1	1
	h	=	=	=	=	=	=	-	NA
f_8	ave	3.20×10^{1}	5.23×10^{-1}	3.10×10^{-15}	8.23×10^{1}	2.51×10^{2}	3.89×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	5.52×10^{1}	1.79×10^{1}	1.90×10^{-15}	1.97×10^{1}	1.38×10^{2}	4.01×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0.03	1	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_9	ave	-3.37×10^{2}	-4.37×10^{2}	-4.50×10^{2}	-1.40×10^{2}	1.93×10^{4}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	1.14×10^{2}	3.08×10^{1}	3.17×10^{-14}	5.78×10^{1}	7.05×10^{3}	$\mathbf{2.30 \times 10^{-3}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0.3	0.96	0	0	0	0	1	1
	h	-	-	-	-	-	=	=	NA
f_{10}	ave	3.81×10^{2}	2.46×10^{2}	5.14×10^{2}	2.00×10^{4}	6.53×10^{4}	6.84×10^{2}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	7.24×10^{2}	4.14×10^{2}	3.39×10^{2}	6.05×10^{3}	3.58×10^{4}	4.32×10^{2}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{11}	ave	4.41×10^{1}	2.47×10^{0}	3.85×10^{-2}	5.15×10^{1}	1.62×10^{1}	4.11×10^{-2}	1.60×10^{-10}	$\mathbf{8.88E-16}$
	std	1.53×10^{1}	2.24×10^{0}	2.11×10^{-1}	5.67×10^{-1}	9.90×10^{-1}	1.46×10^{-2}	4.01×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{12}	ave	8.51×10^{-1}	7.52×10^{-3}	3.25×10^{-3}	9.89×10^{-1}	2.17×10^{1}	3.16×10^{-2}	9.15×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	std	7.59×10^{-1}	1.43×10^{-2}	2.88×10^{-3}	2.42×10^{-1}	5.31×10^{1}	2.14×10^{-2}	1.55×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{13}	ave	1.53×10^{1}	3.34×10^{-1}	2.75×10^{1}	3.81×10^{1}	2.01×10^{2}	3.59×10^{-3}	8.89×10^{-16}	$\mathbf{0.00 \times 10^{0}}$
	std	1.07×10^{1}	5.58×10^{-1}	6.78×10^{1}	6.52×10^{1}	6.38×10^{1}	9.05×10^{-4}	4.70×10^{-15}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{14}	ave	-1.70×10^{1}	-1.35×10^{1}	-2.25×10^{1}	-2.63×10^{1}	-1.02×10^{1}	-2.32×10^{1}	-2.73×10^{1}	$\mathbf{-2.75 \times 10^{1}}$
	std	2.05×10^{1}	3.40×10^{0}	1.86×10^{1}	5.13×10^{-1}	3.20×10^{1}	1.47×10^{0}	1.05×10^{0}	$\mathbf{2.80 \times 10^{-2}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{15}	ave	6.35×10^{4}	9.45×10^{13}	8.22×10^{5}	5.21×10^{4}	1.63×10^{6}	-1.02×10^{3}	-1.89×10^{4}	$\mathbf{-3.12 \times 10^{3}}$
	std	5.76×10^{4}	9.75×10^{13}	1.19×10^{5}	1.92×10^{4}	5.80×10^{5}	2.25×10^{3}	3.60×10^{3}	$\mathbf{5.79 \times 10^{1}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_{16}	ave	1.02×10^{1}	4.74×10^{1}	1.00×10^{1}	5.90×10^{1}	1.28×10^{1}	1.10×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	7.62×10^{-1}	1.35×10^{1}	1.04×10^{1}	4.98×10^{-1}	4.30×10^{-1}	6.30×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{17}	ave	7.13×10^{1}	1.44×10^{12}	2.95×10^{1}	2.11×10^{1}	9.26×10^{1}	2.78×10^{1}	1.46×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	std	1.99×10^{1}	3.41×10^{11}	8.19×10^{1}	3.99×10^{1}	3.40×10^{1}	8.13×10^{1}	4.42×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_{18}	ave	6.26×10^{-54}	$\mathbf{0.00 \times 10^{0}}$	3.72×10^{-4}	2.05×10^{-4}	1.73×10^{-1}	3.46×10^{-7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	1.24×10^{-53}	$\mathbf{0.00 \times 10^{0}}$	8.54×10^{-4}	2.45×10^{-4}	1.34×10^{-1}	3.68×10^{-7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	1	1	0	0	0	1	1	1
	h	-	=	-	-	-	-	=	NA
$total$ h	+	0	0	0	0	0	0	1	NA
	−	17	16	17	17	17	16	11	NA
	=	1	2	1	1	1	2	6	NA
Average SR		0.11	0.11	0.05	0.05	0.05	0.11	0.72	**0.94**

Table 9: Solution quality comparisons of the CPFOA and the meta-heuristic algorithms performing 18 benchmark functions (Dimension = 50).

Function	Criteria	PSO	DE	GSA	HS	BA	FA	CPFOA	CPFOA
f_1	ave	-8.91×10^{-1}	-9.88×10^{1}	$\mathbf{-1.00 \times 10^{0}}$	-8.13×10^{-1}	$\mathbf{-1.00 \times 10^{0}}$	-9.88×10^{1}	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$
	std	7.14×10^{-2}	1.46×10^{-2}	2.96×10^{-6}	2.74×10^{-2}	4.27×10^{-6}	3.79×10^{-3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_2	ave	4.59×10^{-1}	4.13×10^{-1}	3.31×10^{-1}	1.82×10^{1}	2.21×10^{1}	6.48×10^{-2}	$\mathbf{9.52 \times 10^{-5}}$	1.03×10^{-3}
	std	2.99×10^{-1}	1.78×10^{-1}	1.10×10^{-1}	4.16×10^{-1}	9.78×10^{1}	3.35×10^{-2}	$\mathbf{2.08 \times 10^{-4}}$	8.76×10^{-4}
	SR	0	0	0	0	0	0	0	0
	h	-	-	-	-	-	-	+	NA
f_3	ave	9.62×10^{3}	6.75×10^{3}	3.24×10^{3}	9.61×10^{4}	1.48×10^{5}	5.41×10^{3}	1.02×10^{-13}	$\mathbf{0.00 \times 10^{0}}$
	std	3.58×10^{3}	2.23×10^{3}	9.46×10^{2}	1.24×10^{4}	5.98×10^{4}	1.46×10^{3}	2.10×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_4	ave	3.67×10^{1}	5.99×10^{1}	1.23×10^{1}	4.36×10^{1}	5.79×10^{1}	1.13×10^{1}	1.99×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	7.64×10^{0}	2.74×10^{1}	2.31×10^{1}	2.49×10^{1}	9.36×10^{0}	6.65×10^{-1}	3.50×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_5	ave	7.84×10^{2}	5.14×10^{1}	3.90×10^{2}	1.03×10^{2}	3.82×10^{56}	8.28×10^{1}	1.97×10^{-29}	$\mathbf{0.00 \times 10^{0}}$
	std	2.12×10^{2}	6.07×10^{1}	3.43×10^{1}	1.63×10^{1}	1.63×10^{57}	5.11×10^{1}	1.01×10^{-6}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_6	ave	1.71×10^{3}	1.23×10^{2}	2.14×10^{2}	3.93×10^{3}	3.32×10^{4}	4.27×10^{-2}	1.02×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	1.08×10^{3}	1.14×10^{2}	1.44×10^{2}	5.43×10^{2}	7.92×10^{3}	1.11×10^{-2}	1.88×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_7	ave	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	1.95×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	1.48×10^{-11}	$\mathbf{0.00 \times 10^{0}}$
	SR	1	1	1	1	1	1	1	1
	h	=	=	=	=	=	=	-	NA
f_8	ave	1.47×10^{2}	1.57×10^{1}	1.29×10^{1}	2.11×10^{2}	1.13×10^{3}	2.99×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	8.35×10^{1}	1.89×10^{1}	1.26×10^{1}	3.94×10^{1}	6.72×10^{2}	1.21×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_9	ave	1.55×10^{3}	-3.05×10^{2}	-2.37×10^{2}	3.59×10^{3}	3.59×10^{4}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	1.09×10^{3}	1.37×10^{2}	1.77×10^{2}	7.50×10^{2}	1.41×10^{4}	9.56×10^{-3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	=	=	NA
f_{10}	ave	8.40×10^{3}	5.87×10^{3}	2.76×10^{3}	9.69×10^{4}	1.57×10^{5}	5.78×10^{3}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	2.70×10^{3}	1.64×10^{3}	9.91×10^{2}	1.32×10^{4}	8.08×10^{4}	1.45×10^{3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{11}	ave	9.26×10^{1}	5.52×10^{1}	1.63×10^{1}	1.05×10^{1}	1.67×10^{1}	1.30×10^{-1}	1.69×10^{-10}	$\mathbf{8.88 \times 10^{-16}}$
	std	2.20×10^{1}	2.97×10^{1}	6.29×10^{-1}	5.63×10^{-1}	8.83×10^{-1}	2.52×10^{-2}	2.13×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{12}	ave	7.70×10^{1}	3.29×10^{-1}	2.09×10^{-1}	7.14×10^{0}	4.20×10^{1}	2.05×10^{-1}	1.05×10^{-15}	$\mathbf{0.00 \times 10^{0}}$
	std	3.72×10^{1}	2.79×10^{-1}	3.37×10^{-1}	9.85×10^{-1}	7.84×10^{1}	1.18×10^{-1}	1.95×10^{-13}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{13}	ave	1.87×10^{1}	3.40×10^{1}	1.28×10^{2}	3.75×10^{1}	3.48×10^{2}	1.28×10^{-2}	1.09×10^{-17}	$\mathbf{0.00 \times 10^{0}}$
	std	8.63×10^{1}	3.85×10^{1}	1.43×10^{1}	5.20×10^{1}	1.13×10^{2}	3.12×10^{-3}	8.70×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{14}	ave	2.40×10^{1}	-1.43×10^{1}	-3.63×10^{1}	-4.09×10^{1}	-1.67×10^{1}	-3.79×10^{1}	-2.58×10^{1}	$\mathbf{-4.45 \times 10^{1}}$
	std	2.72×10^{1}	2.42×10^{1}	2.48×10^{1}	1.24×10^{1}	4.74×10^{1}	2.33×10^{1}	1.52×10^{0}	$\mathbf{1.56 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_{15}	ave	1.95×10^{6}	2.99×10^{5}	4.62×10^{7}	2.89×10^{6}	2.01×10^{7}	1.54×10^{4}	-1.69×10^{4}	$\mathbf{-8.60 \times 10^{3}}$
	std	8.42×10^{5}	1.64×10^{5}	5.81×10^{6}	5.24×10^{5}	6.01×10^{6}	1.72×10^{4}	2.64×10^{3}	$\mathbf{2.02 \times 10^{2}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_{16}	ave	1.98×10^{1}	9.93×10^{1}	1.71×10^{1}	1.38×10^{1}	2.23×10^{1}	1.97×10^{1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	9.25×10^{-1}	1.07×10^{1}	1.46×10^{1}	6.27×10^{-1}	5.77×10^{-1}	6.80×10^{-1}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{17}	ave	1.72×10^{2}	3.07×10^{2}	6.12×10^{1}	8.20×10^{1}	1.36×10^{2}	5.70×10^{1}	1.06×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	std	3.34×10^{1}	7.58×10^{1}	1.15×10^{1}	8.57×10^{1}	4.96×10^{1}	9.44×10^{1}	2.27×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_{18}	ave	7.22×10^{-54}	$\mathbf{0.00 \times 10^{0}}$	1.19×10^{-3}	2.86×10^{-4}	1.73×10^{-1}	3.47×10^{-7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	2.23×10^{-53}	$\mathbf{0.00 \times 10^{0}}$	4.25×10^{-3}	4.91×10^{-4}	1.44×10^{-1}	3.65×10^{-7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	1	1	0	0	0	1	1	1
	h	-	=	-	-	-	-	=	NA
$total$ h	+	0	0	0	0	0	0	1	NA
	−	17	16	17	17	17	16	11	NA
	=	1	2	1	1	1	2	6	NA
Average SR		0.11	0.11	0.05	0.05	0.05	0.11	0.72	**0.94**

Table 10: Solution quality comparisons of the CPFOA and the meta-heuristic algorithms performing 18 benchmark functions (Dimension = 1000).

Function	Criteria	PSO	DE	GSA	HS	BA	FA	CPFOA	CPFOA
f_1	ave	-1.36×10^{-28}	-1.01×10^{-14}	-4.42×10^{-13}	-7.28×10^{-38}	-3.33×10^{-7}	-6.34×10^{-5}	$\mathbf{-1.00 \times 10^{0}}$	$\mathbf{-1.00 \times 10^{0}}$
	std	7.05×10^{-28}	5.51×10^{-14}	1.09×10^{-12}	6.62×10^{-38}	1.44×10^{-6}	4.02×10^{-5}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_2	ave	5.96×10^{4}	2.43×10^{5}	5.38×10^{3}	9.42×10^{4}	6.24×10^{3}	1.35×10^{3}	$\mathbf{0.00 \times 10^{0}}$	2.53×10^{-4}
	std	6.61×10^{3}	6.28×10^{3}	5.64×10^{2}	2.48×10^{3}	3.44×10^{3}	1.63×10^{2}	$\mathbf{0.00 \times 10^{0}}$	2.55×10^{-4}
	SR	0	0	0	0	0	0	1	0
	h	-	-	-	-	-	-	+	NA
f_3	ave	8.10×10^{6}	1.44×10^{7}	6.35×10^{6}	4.53×10^{7}	7.27×10^{7}	3.43×10^{6}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	1.40×10^{6}	2.70×10^{6}	3.01×10^{6}	6.72×10^{6}	3.54×10^{7}	5.22×10^{5}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_4	ave	8.79×10^{1}	9.96×10^{1}	3.28×10^{1}	9.66×10^{1}	7.78×10^{1}	4.85×10^{1}	1.80×10^{-9}	$\mathbf{0.00 \times 10^{0}}$
	std	1.15×10^{0}	1.52×10^{-1}	1.53×10^{0}	1.70×10^{-1}	7.27×10^{0}	1.69×10^{0}	1.06×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_5	ave	2.86×10^{4}	4.08×10^{4}	8.95×10^{3}	3.21×10^{4}	2.31×10^{4}	9.60×10^{3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	7.51×10^{2}	5.12×10^{2}	2.28×10^{2}	3.69×10^{2}	3.05×10^{3}	2.82×10^{2}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_6	ave	1.39×10^{6}	2.90×10^{6}	1.25×10^{5}	1.73×10^{6}	7.91×10^{5}	1.50×10^{5}	1.74×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	std	5.28×10^{4}	7.53×10^{5}	4.92×10^{3}	2.62×10^{4}	1.86×10^{5}	1.21×10^{4}	3.28×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_7	ave	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	1.15×10^{-13}	$\mathbf{0.00 \times 10^{0}}$
	std	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$	1.72×10^{-11}	$\mathbf{0.00 \times 10^{0}}$
	SR	1	1	1	1	1	1	1	1
	h	=	=	=	=	=	=	-	NA
f_8	ave	1.70×10^{6}	1.72×10^{6}	1.52×10^{5}	2.07×10^{6}	8.87×10^{5}	1.81×10^{5}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	7.82×10^{4}	1.47×10^{6}	1.12×10^{4}	3.91×10^{4}	2.61×10^{5}	1.31×10^{4}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_9	ave	1.36×10^{6}	2.81×10^{6}	1.23×10^{5}	1.71×10^{6}	8.18×10^{5}	1.51×10^{5}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	5.93×10^{4}	8.27×10^{5}	5.06×10^{3}	3.18×10^{4}	2.17×10^{5}	9.18×10^{3}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{10}	ave	7.95×10^{6}	1.46×10^{7}	7.66×10^{6}	4.71×10^{7}	5.64×10^{7}	3.22×10^{6}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
	std	1.19×10^{6}	3.66×10^{6}	3.73×10^{6}	5.54×10^{6}	3.04×10^{7}	5.05×10^{5}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{11}	ave	1.99×10^{1}	1.86×10^{1}	1.09×10^{1}	2.03×10^{1}	1.78×10^{1}	1.28×10^{1}	7.64×10^{-11}	$\mathbf{8.88 \times 10^{-16}}$
	std	9.78×10^{-2}	7.49×10^{-1}	1.69×10^{-1}	4.16×10^{-2}	7.21×10^{-1}	2.59×10^{-1}	2.00×10^{-10}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{12}	ave	1.59×10^{3}	9.62×10^{2}	4.57×10^{2}	1.74×10^{3}	1.34×10^{3}	5.44×10^{2}	1.15×10^{-14}	$\mathbf{0.00 \times 10^{0}}$
	std	7.39×10^{1}	4.90×10^{1}	2.46×10^{1}	2.56×10^{1}	1.50×10^{2}	2.32×10^{1}	1.51×10^{-12}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{13}	ave	1.26×10^{4}	2.61×10^{4}	2.05×10^{4}	1.55×10^{4}	7.71×10^{3}	1.35×10^{3}	2.09×10^{-16}	$\mathbf{0.00 \times 10^{0}}$
	std	8.42×10^{2}	6.83×10^{3}	2.83×10^{2}	2.82×10^{2}	2.22×10^{3}	7.89×10^{1}	3.79×10^{-15}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	-	NA
f_{14}	ave	-1.82×10^{2}	-8.67×10^{1}	-5.10×10^{2}	-1.11×10^{2}	-1.87×10^{2}	-4.84×10^{2}	-2.32×10^{1}	$\mathbf{-9.23 \times 10^{2}}$
	std	1.51×10^{1}	3.13×10^{1}	1.58×10^{1}	5.52×10^{0}	4.06×10^{1}	1.35×10^{1}	1.05×10^{0}	$\mathbf{6.45 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	0
	h	-	-	-	-	-	-	-	NA
f_{15}	ave	1.11×10^{14}	5.34×10^{13}	3.04×10^{14}	1.87×10^{14}	8.44×10^{13}	1.12×10^{13}	-1.29×10^{4}	$\mathbf{-1.02 \times 10^{6}}$
	std	7.06×10^{12}	5.40×10^{12}	6.30×10^{12}	3.98×10^{12}	2.31×10^{13}	8.49×10^{11}	2.54×10^{3}	$\mathbf{1.64 \times 10^{3}}$
	SR	0	0	0	0	0	0	0	0
	h	-	-	-	-	-	-	-	NA
f_{16}	ave	4.79×10^{2}	2.79×10^{2}	4.82×10^{2}	4.84×10^{2}	4.70×10^{2}	4.48×10^{2}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	5.99×10^{1}	1.30×10^{1}	1.43×10^{0}	1.49×10^{0}	7.36×10^{0}	3.35×10^{0}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	1	1
	h	-	-	-	-	-	-	=	NA
f_{17}	ave	1.14×10^{4}	8.31×10^{3}	6.63×10^{3}	1.24×10^{4}	8.06×10^{3}	6.10×10^{3}	1.14×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	std	3.50×10^{2}	3.09×10^{2}	1.62×10^{2}	1.27×10^{2}	5.30×10^{2}	1.78×10^{2}	3.27×10^{-8}	$\mathbf{0.00 \times 10^{0}}$
	SR	0	0	0	0	0	0	0	1
	h	-	-	-	-	-	-	-	NA
f_{18}	ave	5.09×10^{-54}	$\mathbf{0.00 \times 10^{0}}$	4.99×10^{-3}	1.97×10^{-4}	2.27×10^{-1}	2.88×10^{-7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	std	2.44×10^{-53}	$\mathbf{0.00 \times 10^{0}}$	1.84×10^{-2}	2.35×10^{-4}	2.03×10^{-1}	3.19×10^{-7}	$\mathbf{0.00 \times 10^{0}}$	$\mathbf{0.00 \times 10^{0}}$
	SR	1	1	0	0	0	0	1	1
	h	-	=	-	-	-	-	=	NA
$total$	$+$	0	0	0	0	0	0	1	NA
	$-$	17	16	17	17	17	17	9	NA
h	=	1	2	1	1	1	1	8	NA
Average SR		0.11	0.11	0.05	0.05	0.05	0.05	0.77	**0.83**

Table 11. The classification datasets [60,61].

Classification Datasets	Numbe of Attributes	Number of Training Samples	Number of Test Samples	Number of classes
3-bits XOR	3	8	8 as training samples	2
Ballon	4	16	6 as training Samples	2
Iris	4	150	150 as training Samples	3
Breast cancer	9	599	100	2
Heart	22	89	187	2

Table 12. Experimental results of real-life Bio-Medical problems. In the column with the ∇symbol, the lower values are better, whereas in the column with the △symbol, the higher values are better. The best value for each measure is highlighted in boldface.

(i)XOR Dataset Problem					
Algorithm	Best MSE $Value^{\nabla}$	Worst MSE $Value^{\nabla}$	*ave* MSE $Value^{\nabla}$	*std* MSE $Value^{\nabla}$	Classification $Rate^{\triangle}$
JADE	2.03×10^{-6}	1.61×10^{-3}	5.04×10^{-4}	5.43×1^{-4}	76.875%
BLPSO	8.81×10^{-4}	1.42×10^{-2}	4.89×10^{-3}	3.79×10^{-3}	58.125%
CLPSO	3.17×10^{-9}	8.71×10^{-2}	3.36×10^{-2}	2.17×10^{-2}	60.833%
GWO	8.27×10^{-6}	$1.33 \times 10{-}1$	1.56×10^{-2}	3.75×10^{-2}	100%
MGWO	5.06×10^{-5}	2.16×10^{-1}	3.48×10^{-2}	6.34×10^{-2}	**100**%
HAGWO4.27×10^{-2}	2.30×10^{-1}	2.90×10^{-3}	4.69×10^{-2}	**100**%	
CPFOA	$\mathbf{1.97 \times 10^{-9}}$	$\mathbf{2.43 \times 10^{-4}}$	$\mathbf{2.59 \times 10^{-5}}$	$\mathbf{5.45 \times 10^{-5}}$	**100**%
(ii)Iris Dataset Problem					
Algorithm	Best MSE $Value^{\nabla}$	Worst MSE $Value^{\nabla}$	*ave* MSE $Value^{\nabla}$	*std* MSE $Value^{\nabla}$	Classification $Rate^{\triangle}$
JADE	2.02×10^{-2}	3.70×10^{-2}	2.52×10^{-1}	2.89×1^{-2}	75.266%
BLPSO	3.51×10^{-2}	8.17×10^{-2}	5.68×10^{-2}	1.45×10^{-2}	43%
CLPSO	1.36×10^{-1}	3.67×10^{-1}	2.59×10^{-1}	6.35×10^{-2}	47.333%
GWO	6.67×10^{-1}	8.76×10^{-1}	6.71×10^{-1}	2.49×10^{-2}	91.333%
MGWO	6.67×10^{-1}	8.13×10^{-1}	6.79×10^{-1}	1.54×10^{-2}	91.34%
HAGWO6.67×10^{-1}	8.81×10^{-1}	6.73×10^{-1}	2.81×10^{-2}	93.00%	
CPFOA	$\mathbf{1.81 \times 10^{-2}}$	$\mathbf{3.42 \times 10^{-2}}$	$\mathbf{5.39 \times 10^{-2}}$	$\mathbf{1.01 \times 10^{-2}}$	**93.733**%
(iii)Balloon Dateset Problem					
Algorithm	Best MSE $Value^{\nabla}$	Worst MSE $Value^{\nabla}$	*ave* MSE $Value^{\nabla}$	*std* MSE $Value^{\nabla}$	Classification $Rate^{\triangle}$
JADE	4.44×10^{-18}	3.37×10^{-8}	2.24×10^{-9}	7.62×1^{-9}	**100**%
BLPSO	1.42×10^{-11}	9.61×10^{-8}	1.37×10^{-8}	2.42×10^{-8}	**100**%
CLPSO	7.45×10^{-6}	3.18×10^{-3}	9.71×10^{-4}	8.77×10^{-4}	89.75%
GWO	3.51×10^{-17}	1.17×10^{-1}	6.40×10^{-3}	2.61×10^{-2}	**100**%
MGWO	2.25×10^{-15}	5.56×10^{-2}	7.10×10^{-3}	4.38×10^{-2}	**100**%
HAGWO1.64×10^{-5}	1.80×10^{-1}	1.43×10^{-2}	4.38×10^{-2}	**100**%	
CPFOA	$\mathbf{1.87 \times 10^{-20}}$	$\mathbf{2.82 \times 10^{-8}}$	$\mathbf{1.51 \times 10^{-9}}$	$\mathbf{6.29 \times 10^{-9}}$	**100**%
(iv)Breast Cancer Dataset Problem					
Algorithm	Best MSE $Value^{\nabla}$	Worst MSE $Value^{\nabla}$	*ave* MSE $Value^{\nabla}$	*std* MSE $Value^{\nabla}$	Classification $Rate^{\triangle}$
JADE	3.79×10^{-3}	7.98×10^{-3}	6.28×10^{-3}	1.30×1^{-3}	67.90%
BLPSO	4.56×10^{-3}	8.34×10^{-3}	6.63×10^{-3}	1.07×10^{-3}	92.40%
CLPSO	2.26×10^{-2}	3.90×10^{-2}	3.39×10^{-2}	3.91×10^{-3}	60%
GWO	1.40×10^{-3}	4.64×10^{-2}	6.50×10^{-3}	9.30×10^{-3}	99.00%
MGWO	1.70×10^{-3}	3.87×10^{-2}	6.60×10^{-3}	9.60×10^{-3}	99.11%
HAGWO1.30×10^{-3}	4.64×10^{-2}	2.60×10^{-3}	4.20×10^{-3}	**100**%	
CPFOA	$\mathbf{1.21 \times 10^{-3}}$	$\mathbf{1.70 \times 10^{-3}}$	$\mathbf{1.67 \times 10^{-3}}$	$\mathbf{1.19 \times 10^{-4}}$	**100**%
(v)Heart Dataset Problem					
Algorithm	Best MSE $Value^{\nabla}$	Worst MSE $Value^{\nabla}$	*ave* MSE $Value^{\nabla}$	*std* MSE $Value^{\nabla}$	Classification $Rate^{\triangle}$
JADE	8.49×10^{-2}	1.13×10^{-1}	9.86×10^{-2}	8.35×1^{-2}	77.5%
BLPSO	1.65×10^{-1}	2.33×10^{-1}	2.01×10^{-1}	2.30×10^{-2}	67.20%
CLPSO	1.98×10^{-1}	2.48×10^{-1}	2.23×10^{-1}	1.80×10^{-2}	68.125%
GWO	4.69×10^{-2}	8.13×10^{-1}	6.44×10^{-2}	1.07×10^{-2}	75%
MGWO	4.69×10^{-2}	7.81×10^{-1}	5.88×10^{-2}	8.69×10^{-3}	88.375%
HAGWO	4.69×10^{-2}	1.42×10^{-1}	1.12×10^{-1}	1.78×10^{-2}	47.625%
CPFOA	$\mathbf{5.63 \times 10^{-2}}$	$\mathbf{1.06 \times 10^{-1}}$	$\mathbf{4.17 \times 10^{-2}}$	$\mathbf{1.14 \times 10^{-2}}$	**90.333**%

Table 13. Comparison of different models for the Box-Jenkins system.

NO.	Models	No. of Rules	No. of Parameters	Training MSE	Testing MSE
1	Lin et al.([64])	4	44	0.0710	0.261
2	Kim et al.([65])	2	38	0.0340	0.244
3	Tsekouras([66])	7	91	0.220	0.236
4	Li et al.([67])	3	57	0.0159	0.126
5	Li et al.([68])	3	57	0.0150	0.147
6	Cheung et al.([69])	3	21	0.590	0.152
7	iTaSuM([63])	3	39	0.0271	0.138
8	CPFOA	2	26	0.0364	0.131

Table 14: The parameters in each rule of CPFOA in the Box-Jenkins system.

Rule	Parameter Vectors				
1	C_1	45.2402 47.6302 0.8332 1.5923	σ_1	41.3468 44.0452 41.0143 39.0507	W_1, b_1 3.9853 2.9633 -4.1245 -5.9999 -5.4895
2	C_2	47.7534 46.5581 -2.5197 2.8912	σ_2	42.1889 41.7317 36.6326 36.8915	W_2, b_2 0.0791 -4.8267 3.1219 5.5687 -4.5793

REFERENCES

1. Mirlekar, G.; Gebreslassie, B.; Diwekar, U.; Lima,F.V. *Biomimetic model-based advanced control strategy integrated with multi-agent optimization for nonlinear chemical processes*, Chemical Engineering Research and Design 2018, 140, 229-240, doi:https://doi.org/10.1016/j.cherd.2018.10.005.
2. Tian, W.; Zhang, G.; Liang, H. *Alarm clustering analysis and ACO based multi-variable alarms thresholds optimization in chemical processes*, Process Safety and Environmental Protection 2018, 113, 132-140, doi:https://doi.org/10.1016/j.psep.2017.09.020.
3. Chen, X.; Mei, C.; Xu, B.; Yu, K.; Huang, X. *Quadratic interpolation based teaching-learning-based optimization for chemical dynamic system optimization*, Knowledge-Based Systems 2018, 145, 250-263, doi:https://doi.org/10.1016/j.knosys.2018.01.021.
4. Ma, H.; Fei, M.; Yang, Z. *Biogeography-based optimization for identifying promising compounds in chemical process*, Neurocomputing 2016, 174, 494-499, doi:https://doi.org/10.1016/j.neucom.2015.05.125.
5. Eberhart, R.; Kennedy, J. *A new optimizer using particle swarm theory*, In Proceedings of Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, 4-6 Oct 1995; pp. 39-43.
6. Dorigo, M.; Maniezzo, V.; Colorni, A. *Ant system: optimization by a colony of cooperating agents.*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 1996, 26, 29-41, doi:10.1109/3477.484436.
7. Karaboga, D.; Basturk, B., *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*, Journal of Global Optimization 2007, 39, 459-471, doi:10.1007/s10898-007-9149-x.
8. Yang, X.S.; Suash, D. *Cuckoo Search via Lévy flights*, In Proceedings of 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 9-11 Dec. 2009; pp. 210-214.
9. Yang, X.-S. *A New Metaheuristic Bat-Inspired Algorithm*, In Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Gonzá lez, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds. Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; 10.1007/978-3-642-12538-6_6pp. 65-74.
10. Jain, M.; Singh, V.; Rani, A. *A novel nature-inspired algorithm for optimization: Squirrel search algorithm*, Swarm and Evolutionary Computation 2018, https://doi.org/10.1016/j.swevo.2018.02.013, doi:https://doi.org/10.1016/j.swevo.2018.02.013.
11. Ma, H.; Shen, S.; Yu, M.; Yang, Z.; Fei, M.; Zhou, H. *Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey*, Swarm and Evolutionary Computation 2018, https://doi.org/10.1016/j.swevo.2018.04.011, doi:https://doi.org/10.1016/j.swevo.2018.04.011.
12. Pan, W.-T. *A new evolutionary computation approach: Fruit Fly Optimization Algorithm*, 2011 Conference of Digital Technology and Innovation Management, Taipei 2011.
13. Storn, R.; Price, K. *Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*, Journal of Global Optimization 1997, 11, 341-359, doi:10.1023/A:1008202821328.

14. Goldberg, D.E.; Holland, J.H. *Genetic Algorithms and Machine Learning* Machine Learning 1988, 3, 95-99, doi:10.1023/A:1022602019183.
15. Shan, D.; Cao, G.; Dong, H. *LGMS-FOA: An Improved Fruit Fly Optimization Algorithm for Solving Optimization Problems*, Mathematical Problems in Engineering 2013, 2013, 9, doi:10.1155/2013/108768.
16. Pan, Q.-K.; Sang, H.-Y.; Duan, J.-H.; Gao, L. *An improved fruit fly optimization algorithm for continuous function optimization problems*, Knowledge-Based Systems 2014, 62, 69-83, doi:https://doi.org/10.1016/j.knosys.2014.02.021.
17. Pan, W.-T. *Using modified fruit fly optimisation algorithm to perform the function test and case studies*, Connect. Sci 2013, 25, 151-160, doi:10.1080/09540091.2013.854735.
18. Yuan, X.; Dai, X.; Zhao, J.; He, Q. *On a novel multi-swarm fruit fly optimization algorithm and its application*, Applied Mathematics and Computation 2014, 233, 260-271, doi:https://doi.org/10.1016/j.amc.2014.02.005.
19. Li, H.-z.; Guo, S.; Li, C.-j.; Sun, J.-q. *A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm*, Knowledge-Based Systems 2013, 37, 378-387, doi:https://doi.org/10.1016/j.knosys.2012.08.015.
20. Hu, R.; Wen, S.; Zeng, Z.; Huang, T. *A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm*, Neurocomputing 2017, 221, 24-31, doi:https://doi.org/10.1016/j.neucom.2016.09.027.
21. Niu, D.; Wang, H.; Chen, H.; Liang, Y. *The General Regression Neural Network Based on the Fruit Fly Optimization Algorithm and the Data Inconsistency Rate for Transmission Line Icing Prediction*, Energies 2017, 10, doi:10.3390/en10122066.
22. Abidin, Z.Z.; Arshad, M.R.; Ngah, U.K. *A simulation based fly optimization algorithm for swarms of mini autonomous surface vehicles application*, Indian Journal of Marine Sciences 2011, 40, 250-266.
23. Lei, X.; Ding, Y.; Fujita, H.; Zhang, A. *Identification of dynamic protein complexes based on fruit fly optimization algorithm*, Knowledge-Based Systems 2016, 105, 270-277, doi:https://doi.org/10.1016/j.knosys.2016.05.019.
24. Cao, G.; Wu, L. *Support vector regression with fruit fly optimization algorithm for seasonal electricity consumption forecasting*, Energy 2016, 115, 734-745, doi:https://doi.org/10.1016/j.energy.2016.09.065.
25. Kanarachos, S.; Griffin, J.; Fitzpatrick, M.E. *Efficient truss optimization using the contrast-based fruit fly optimization algorithm*, Computers & Structures 2017, 182, 137-148, doi:https://doi.org/10.1016/j.compstruc.2016.11.005.
26. Du, T.-S.; Ke, X.-T.; Liao, J.-G.; Shen, Y.-J. *DSLC-FOA : Improved fruit fly optimization algorithm for application to structural engineering design optimization problems*, Applied Mathematical Modelling 2018, 55, 314-339, doi:https://doi.org/10.1016/j.apm.2017.08.013.
27. Wu, L.; Liu, Q.; Tian, X.; Zhang, J.; Xiao, W. *A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems*, Knowledge-Based Systems 2018, 144, 153-173, doi:https://doi.org/10.1016/j.knosys.2017.12.031.
28. Han, S.-Z.; Pan, W.-T.; Zhou, Y.-Y.; Liu, Z.-L. *Construct the prediction model for China agricultural output value based on the optimization neural network of fruit fly optimization algorithm*, Future Generation Computer Systems 2018, 86, 663-669, doi:https://doi.org/10.1016/j.future.2018.04.058.
29. Zhang, X.; Lu, X.; Jia, S.; Li, X. *A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning*, Applied Soft Computing 2018, 70, 371-388, doi:https://doi.org/10.1016/j.asoc.2018.05.030.
30. Zhang, Y.; Cui, G.; Wu, J.; Pan, W.-T.; He, Q. *A novel multi-scale cooperative mutation Fruit Fly Optimization Algorithm*, Knowledge-Based Systems 2016, 114, 24-35, doi:https://doi.org/10.1016/j.knosys.2016.09.027.
31. Han, X.; Liu, Q.; Wang, H.; Wang, L. *Novel fruit fly optimization algorithm with trend search and co-evolution*, Knowledge-Based Systems 2018, 141, 1-17, doi:https://doi.org/10.1016/j.knosys.2017.11.001.
32. Krishnanand, K.N.; Ghose, D. *Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications*, Multiagent Grid Syst. 2006, 2, 209-222.
33. Rajabioun, R. *Cuckoo Optimization Algorithm*, Applied Soft Computing 2011, 11, 5508-5518, doi:https://doi.org/10.1016/j.asoc.2011.05.008.
34. Chen, Z.; Tang, H. *Cockroach Swarm Optimization*, 10.1109/ICCET.2010.5485993.
35. Saremi, S.; Mirjalili, S.; Lewis, A. *Grasshopper Optimisation Algorithm: Theory and application*, Advances in Engineering Software 2017, 105, 30-47, doi:https://doi.org/10.1016/j.advengsoft.2017.01.004.
36. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. *Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems*, Advances in Engineering Software 2017, 114, 163-191, doi:https://doi.org/10.1016/j.advengsoft.2017.07.002.
37. Dai, H.; Zhao, G.; Lu, J.; Dai, S. *Comment and improvement on "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example"*, Knowledge-Based Systems 2014, 59, 159-160, doi:https://doi.org/10.1016/j.knosys.2014.01.010.
38. Niu, J.; Zhong, W.; Liang, Y.; Luo, N.; Qian, F. *Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization*, Knowledge-Based Systems 2015, 88, 253-263, doi:https://doi.org/10.1016/j.knosys.2015.07.027.
39. Babalık, A.; İ ş can, H.; Babaoğlu,İ.; Gündüz, M. *An improvement in fruit fly optimization algorithm by using sign parameters*, Soft Computing 2017, 10.1007/s00500-017-2733-1, doi:10.1007/s00500-017-2733-1.
40. Mirjalili, S. *The Ant Lion Optimizer*, Advances in Engineering Software 2015, 83, 80-98, doi:https://doi.org/10.1016/j.advengsoft.2015.01.010.
41. Yang, X.-S. *Nature-inspired Metaheuristic Algorithms*, Luniver Press, 2010.
42. Gupta, S.; Deep, K. *A novel Random Walk Grey Wolf Optimizer*, Swarm and Evolutionary Computation 2018, https://doi.org/10.1016/j.swevo.2018.01.001, doi:https://doi.org/10.1016/j.swevo.2018.01.001.

43. Mirjalili, S.; Gandomi, A.H. *Chaotic gravitational constants for the gravitational search algorithm*, Applied Soft Computing 2017, 53, 407-419, doi:https://doi.org/10.1016/j.asoc.2017.01.008.
44. Ab. Aziz, N.A.; Ibrahim, Z.; Mubin, M.; Nawawi, S.W.; Mohamad, M.S. *Improving particle swarm optimization via adaptive switching asynchronous C synchronous update*, Applied Soft Computing 2018, 72, 298-311, doi:https://doi.org/10.1016/j.asoc.2018.07.047.
45. Pan, W.-T. *A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example*, Knowledge-Based Systems 2012, 26, 69-74, doi:https://doi.org/10.1016/j.knosys.2011.07.001.
46. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. *GSA: A Gravitational Search Algorithm*, Information Sciences 2009, 179, 2232-2248, doi:https://doi.org/10.1016/j.ins.2009.03.004.
47. Lee, K.S.; Geem, Z.W. *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice*, Computer Methods in Applied Mechanics and Engineering 2005, 194, 3902-3933, doi:https://doi.org/10.1016/j.cma.2004.09.007.
48. Fister, I.; Fister, I.; Yang, X.-S.; Brest, J. *A comprehensive review of firefly algorithms*, Swarm and Evolutionary Computation 2013, 13, 34-46, doi:https://doi.org/10.1016/j.swevo.2013.06.001.
49. Chen, X.; Tianfield, H.; Mei, C.; Du, W.; Liu, G. *Biogeography-based learning particle swarm optimization*; 2017; Vol. 21, pp. 7519C7541.
50. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. *Comprehensive learning particle swarm optimizer for global optimization of multimodal functions*, Trans. Evol. Comp 2006, 10, 281-295, doi:10.1109/tevc.2005.857610.
51. Wang, Y.; Cai, Z.; Zhang, Q. *Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters* IEEE Transactions on Evolutionary Computation 2011, 15, 55-66, doi:10.1109/TEVC.2010.2087271.
52. Zhang, J.; Sanderson, A.C. *JADE: Adaptive Differential Evolution With Optional External Archive*, IEEE Transactions on Evolutionary Computation 2009, 13, 945-958, doi:10.1109/TEVC.2009.2014613.
53. Singh, N.; Singh, S. *A Modified Mean Gray Wolf Optimization Approach for Benchmark and Biomedical Problems*, Evolutionary Bioinformatics 2017, 13, 1176934317729413, doi:10.1177/1176934317729413.
54. Nabil, E. *A Modified Flower Pollination Algorithm for Global Optimization*, Expert Systems with Applications 2016, 57, 192-203, doi:https://doi.org/10.1016/j.eswa.2016.03.047.
55. Liu, Z.; Liu, X.; Cai, X. *A new hybrid aerodynamic optimization framework based on differential evolution and invasive weed optimization*, Chinese Journal of Aeronautics 2018, 31, 1437-1448, doi:https://doi.org/10.1016/j.cja.2018.05.002.
56. Zhao, X.; Yao, Y.; Yan, L. *Learning algorithm for multimodal optimization*, Computers & Mathematics with Applications 2009, 57, 2016-2021, doi:https://doi.org/10.1016/j.camwa.2008.10.008.
57. Messikh, N.; Bousba, S.; Bougdah, N. *The use of a multilayer perceptron (MLP) for modelling the phenol removal by emulsion liquid membrane*, Journal of Environmental Chemical Engineering 2017, 5, 3483-3489, doi:https://doi.org/10.1016/j.jece.2017.06.053.
58. Díaz-Rodrłguez, P.; Cancilla, J.C.; Matute, G.; Chicharro, D.; Torrecilla, J.S. *Inputting molecular weights into a multilayer perceptron to estimate refractive indices of dialkylimidazolium-based ionic liquidsłA purity evaluation*, Applied Soft Computing 2015, 28, 394-399, doi:https://doi.org/10.1016/j.asoc.2014.12.004.
59. Qiao, J.; Li, W.; Han, H. *Soft Computing of Biochemical Oxygen Demand Using an Improved TCS Fuzzy Neural Network*, Chinese Journal of Chemical Engineering 2014, 22, 1254-1259, doi:https://doi.org/10.1016/j.cjche.2014.09.023.
60. Mirjalili, S. *How effective is the Grey Wolf optimizer in training multi-layer perceptrons*, Applied Intelligence 2015, 43, 150-161, doi:10.1007/s10489-014-0645-7.
61. Singh, N.; Hachimi, H. *A New Hybrid Whale Optimizer Algorithm with Mean Strategy of Grey Wolf Optimizer for Global Optimization*, Mathematical and Computational Applications 2018, 23, 14.
62. Box, G.E.P.; Jenkins, G. *Time Series Analysis, Forecasting and Control*; Holden-Day, Inc.: 1990; pp. 500.
63. Ding, X.; Xu, Z.; Cheung, N.J.; Liu, X. *Parameter estimation of TakagiCSugeno fuzzy system using heterogeneous cuckoo search algorithm*, Neurocomputing 2015, 151, 1332-1342, doi:https://doi.org/10.1016/j.neucom.2014.10.063.
64. Yinghua, L.; Cunningham, G.A. *A new approach to fuzzy-neural system modeling*, IEEE Transactions on Fuzzy Systems 1995, 3, 190-198, doi:10.1109/91.388173.
65. Euntai, K.; Minkee, P.; Seungwoo, K.; Mignon, P. *A transformed input-domain approach to fuzzy modeling*, IEEE Transactions on Fuzzy Systems 1998, 6, 596-604, doi:10.1109/91.728458.
66. Tsekouras, G.E. *On the use of the weighted fuzzy c-means in fuzzy modeling*, Advances in Engineering Software 2005, 36, 287-300, doi:https://doi.org/10.1016/j.advengsoft.2004.12.001.
67. Li, C.; Zhou, J.; Fu, B.; Kou, P.; Xiao, J. *T-S Fuzzy Model Identification With a Gravitational Search-Based Hyperplane Clustering Algorithm*, IEEE Transactions on Fuzzy Systems 2012, 20, 305-317, doi:10.1109/TFUZZ.2011.2173693.
68. Li, C.; Zhou, J.; Xiao, J.; Xiao, H. *Hydraulic turbine governing system identification using T-S fuzzy model optimized by chaotic gravitational search algorithm*, Engineering Applications of Artificial Intelligence 2013, 26, 2073-2082, doi:https://doi.org/10.1016/j.engappai.2013.04.002.
69. Cheung, N.J.; Ding, X.; Shen, H. *OptiFel: A Convergent Heterogeneous Particle Swarm Optimization Algorithm for Takagi-Sugeno Fuzzy Modeling*, IEEE Transactions on Fuzzy Systems 2014, 22, 919-933, doi:10.1109/TFUZZ.2013.2278972.