Knowledge-Based Systems 176 (2019) 77–96

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

New fruit fly optimization algorithm with joint search strategies for function optimization problems ☆

Lin Wang [a], Yingnan Xiong [a], Shuwen Li [b], Yu-Rong Zeng [c,*]

[a] School of Management, Huazhong University of Science and Technology, Wuhan 430074, China
[b] School of Economics and Management, China University of Geosciences (Wuhan), Wuhan 430074, China
[c] Hubei University of Economics, Wuhan 430205, China

HIGHLIGHTS

- A new parameter and new move direction guided by biological memory are used for search in proposed JS-FOA.
- Using a formula similar to the gradient descent in mathematical to escape from the local extreme.
- Comparative experiment containing 29 benchmark functions is tested to verify the performance of JS-FOA.
- Perform sensitivity analysis experiments on the two parameters of JS-FOA to verify the stability of the algorithm.

ARTICLE INFO

Article history:
Received 21 July 2018
Received in revised form 16 February 2019
Accepted 24 March 2019
Available online 9 April 2019

Keywords:
Fruit fly optimization algorithm
Group collaboration
Memory move direction
Gradient descent search

ABSTRACT

An enhanced fruit fly optimization algorithm (FOA) with joint search strategies named JS-FOA is proposed to optimize continuous function problems. First, a collaborative group search, which includes a new parameter, is conducted to obtain the critical value. Second, a new search strategy similar to biological memory, namely, memory move direction, is proposed to improve solution accuracy. Third, a gradient descent search is used in the collaborative group search to ensure that it does not fall into a local optimum. Finally, a new function, which is similar to the excitation function in a neural network, is proposed to combine the three search strategies. To test the robustness and convergence of the proposed JS-FOA, we used 29 complex continuous benchmark functions. Results show that the proposed JS-FOA outperforms other heuristic algorithms for most functions. The performance of JS-FOA is also evaluated for different parameter values and the results show that parameter values affect convergence speed within a certain range, but do not change the convergence accuracy for the continuous benchmark functions. The proposed JS-FOA may potentially solve high-dimensional optimization problems.

1. Introduction

In the past 20 years, many swarm intelligence algorithms, including the genetic algorithm (GA) [1], ant colony optimization [2], particle swarm optimization (PSO) [3], simulated annealing (SA) [4], and other intelligent algorithms have been proposed to solve complex problems such as financial distress situations [5,6], energy load performance [7], test scheduling problems [8,9], web auction logistic servicing [10], engineering optimization problems [11], nonlinear bi-level programming problems [12], neural network parameter optimization [13–15], and others [16–18]. These swarm intelligence algorithms have become increasingly popular and can solve the abovementioned problems effectively and efficiently.

However, the structures of these algorithms are complex because their parameters have to be adjusted. Reference [5] recently proposed a new swarm intelligence algorithm with a simple structure, named the fruit fly optimization algorithm (FOA), to solve optimization problems while [6] formally tested the FOA's performance. The basic FOA simulates a swarm of fruit flies in search of food. Fruit flies can easily find food because of their cooperative behavior and good eyesight. However, the basic FOA usually produces local optimal solutions when dealing with high-dimensional functions and problems. Thus, various studies have been conducted to improve the FOA. Some enhanced FOAs can improve the solution accuracy, while others can increase the convergence speed or focus on escaping from a local optimum, which may help to improve both the solution accuracy and convergence speed.

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.03.028.

* Corresponding author.
E-mail addresses: wanglin@hust.edu.cn (L. Wang), hust.xyn@gmail.com (Y. Xiong), 294072711@qq.com (S. Li), zyrhbue@gmail.com (Y.-R. Zeng).

https://doi.org/10.1016/j.knosys.2019.03.028
0950-7051/

Firstly, some researchers focused on improving the solution accuracy. Reference [12] proposed an enhanced bi-level FOA (BIFOA) for nonlinear bi-level programming problems. In BIFOA, two enhanced FOAs are combined to test 10 problems including low- and high-dimensional problems. The results showed that BIFOA outperforms other algorithms used for comparison and is significantly better than the regular FOA. Reference [19] introduced an improved FOA named IFFO with a new control parameter. The control parameter can change the flies' search range, thus improving the algorithm's accuracy. Reference [20] developed a multi-swarm FOA (MFOA) that employs multi-swarm behavior to significantly improve its performance. When splitting into several sub-swarms, the MFOA performs better than the original FOA. Reference [21] proposed a new method to maintain population diversity and improve exploration ability. Experimental results showed that the enhanced FOA has better overall performance than the original FOA and other algorithms.

Secondly, new approaches for improving convergence speed are also currently of interest. Reference [22] introduced another enhanced FOA with nonlinear and linear generation mechanisms for candidate solutions, which can greatly improve the search efficiency and quality. An ensemble FOA (EFFO) for solving range image registration was proposed by [23]. The EFFO mainly provides a parameter pool using the smell-based search of a fruit fly swarm and sorts the individuals based on their ability to identify the leaders in a vision-based search. The experiment showed that the proposed algorithm is effective in range image registration. Reference [24] proposed an enhanced FOA that uses Cauchy mutation to improve convergence and other capabilities. Reference [25] proposed an enhanced FOA with balanced exploitation and exploration that contributes to the cooperation within a swarm in order to solve multi-dimensional knapsack problems. Reference [11] proposed an improved FOA with four extra mechanisms (IAFOA) to solve engineering optimization problems. In IAFOA, four adaptive functions, namely the search direction, iteration step value, crossover and mutation mechanism, and the multi-sub-swarm mechanism are combined, and 29 benchmark functions are tested. The results showed that the proposed IAFOA has quicker convergence speed and higher accuracy than comparative algorithms. In addition, IAFOA has an advantage in solving engineering optimization problems.

Finally, some scholars focused on preventing FOAs from falling into local optima. Reference [26] proposed a bimodal adaptive FOA using normal cloud learning; the enhanced FOA divides a population into two groups according to their duties of either searching or capturing. The searching group is mainly concerned with finding a possible global optimum, whereas the capturing group monitors the neighborhood of the currently available best food found by the searching group. The enhanced FOA has better convergence performance and accuracy than other algorithms, such as PSO, differential evolution algorithm (DE), and GA. Reference [27] proposed a novel multi-scale cooperative mutation FOA (MSFOA) that employs a multi-scale cooperative mutation and the Gaussian mutation operator. Using the Gaussian mutation operator can address the limitation of a local optimum. The experimental results showed that MSFOA has better benchmark functions than other enhanced FOAs at the time.

Although many studies on FOAs have been conducted, room for improvement remains. Firstly, some FOAs improve the solution accuracy but need more CPU time. Secondly, the structures of some FOAs are too complex to be easily implemented. Thirdly, some FOAs may fall into local optima during the iterative process. Finally, a suitable initial value enables FOAs to find a global optimum, but few researchers have studied this problem. Therefore, new FOAs with improved effectiveness and efficiency should be further studied.

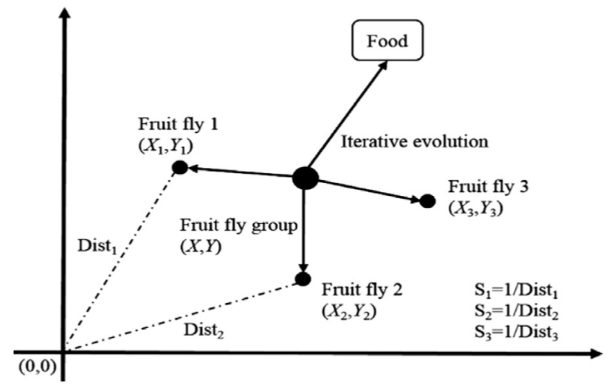


Fig. 1. Process of a fruit fly swarm searching for food.

In order to overcome the disadvantages discussed above, this study proposes an enhanced FOA with joint search strategies named JS-FOA to quickly and accurately solve high-dimensional problems. The main contributions of this study are as follows:

(a) The overall improvements of JS-FOA lie in three aspects: Firstly, a parameter is added and the normal distribution is used to control the convergence rate; secondly, the gradient descent is calculated using a mathematical formula to escape from local optima; finally, the course of the fruit flies is guided by a memory move direction to find food quickly and accurately.

(b) We develop a new nonlinear joint algorithm to combine the three aspects in (a) in a particular order.

(c) Compared with existing FOAs, the JS-FOA has a simpler structure, which guarantees easier implementation in software.

To the best of our knowledge, no study has ever utilized the proposed policies and obtained an enhanced FOA with satisfactory performance. In fact, the proposed JS-FOA is composed of three sub-algorithms which have synergetic effects on the improvement of the existing algorithms. The three sub-algorithms have different purposes; two of them are used to search for a global optimal solution and the third helps to avoid local optima. Through the coordination of the three sub-algorithms, the new FOA has a faster convergence speed, obtains a more accurate solution, and escapes more easily from local optima.

In fact, experiments based on 29 benchmark functions showed that JS-FOA has a greater probability to find the global optimum and a quicker convergence rate than some of the current algorithms when optimizing high-dimensional functions. Moreover, the JS-FOA parameter can affect the convergence rate and probability of finding a global optimum. Experimental results also showed that the parameter can be changed according to practical necessity.

The rest of this paper is organized as follows. Section 2 introduces the basic fruit fly optimization algorithm. In Section 3, the proposed enhanced FOA is discussed. The experiments and sensitivity analysis are described and the results are listed in Section 4. Finally, conclusions and future research are discussed in Section 5.

2. Basic fruit fly optimization algorithm (FOA)

Fruit flies are well known for their cooperative hunting with a superior sense of smell and vision; they can find food even 40 km away. When a swarm hunts for food, each fruit fly keeps track of its location and the concentration of food. Smell concentrations are compared, and the largest concentration of food is selected. The location with the largest concentration is the best current location. The fruit flies then move to the current location from

Algorithm 1: Basic FOA

Parameters: population size (*popsize*), maximum iteration (*maxgen*), upper bound (*Ub*), and lower bound (*Lb*)

Output: *S** and *BestSmell*

1. **Set** *popsize* **and** *maxgen* //*parameter initialization*
2. $X_{ini} = Lb + (Ub - Lb) * rand()$ // *swarm location initialization*
3. $Y_{ini} = Lb + (Ub - Lb) * rand()$
4. $iter = 0$ **and** $SmellBest = Fitness(S_{ini})$
5. **Repeat** // *searching for food by smell and vision*

   For $i = 1,2,\ldots,population$

   $X_i = X_{ini} + rand()$ ***and*** $Y_i = Y_{ini} + rand()$

   $$S_{ij} = \frac{1}{\sqrt{X_{ij}^2 + Y_{ij}^2}}$$

   If $S_{ij} > Ub$ **then** $S_{ij} = Ub$

   If $S_{ij} < Lb$ **then** $S_{ij} = Lb$

   Endfor
6. *[BestSmell, BestIndex] = Fitness(Si)* // *calculate the adaptive value*
7. **If** *BestSmell < SmellBest* **then** $X_{ini} = X(BestIndex)$ $Y_{ini} = Y(BestIndex)$

   and then *BestSmell = SmellBest*
8. $t = t + 1$

   Until $t = maxgen$ **or** *find food*

Fig. 2. Basic fruit fly optimization algorithm.

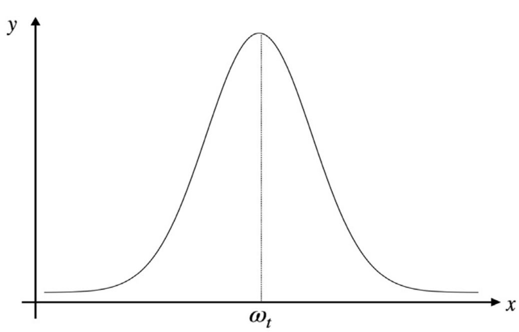


Fig. 3. Probability value of ω_t.

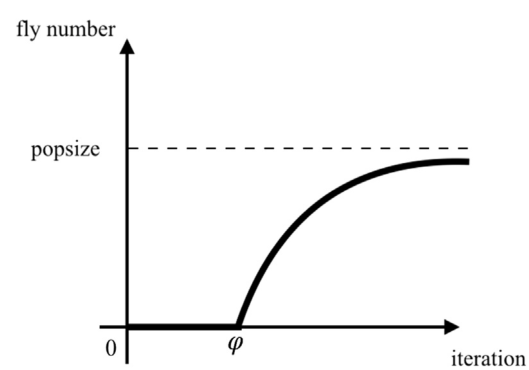


Fig. 4. Number of flies with memories.

where they find the next best location based on smell and vision. The process of a fruit fly swarm finding food is shown in Fig. 1.

This process can also be described by a mathematical model as follows:

$$\begin{aligned} &\min Fitness\,(X) \\ &s.t.x_j \in \left[Lb_j, Ub_j\right] j = 1, 2, \ldots, n \end{aligned} \tag{1}$$

where $Fitness(X)$ is the objective function, which is the reciprocal of the smell concentration in the fruit fly optimization algorithm. $X = (x_1, x_2, \ldots, x_n)$ is the vector of fruit flies, Lb_j is the lower bound, and Ub_j is the upper bound of the search range. The algorithm can be described in four steps.

Step 1: Initialize algorithm parameters, which are the population size of the fruit fly swarm (*popsize*), maximum number of iterations (*maxgen*), upper bound (*Ub*), lower bound (*Lb*), and index of an individual fly (*N*). The initial location of the swarm is generated as:

$$\begin{cases} X_{ini} = Lb + (Ub - Lb) * rand() \\ Y_{ini} = Lb + (Ub - Lb) * rand() \end{cases} \tag{2}$$

where $rand()$ is a function that generates a random number between 0 and 1.

Step 2: A fruit fly searches for food by smell and shares the location with others.

Step 2.1: Generate the new location of an individual fruit fly by using Eq. (3) and calculate its distance from the initial location ($Dist_i$) by using Eq. (4) as follows:

$$\begin{cases} X_i = X_{ini} + rand() \\ Y_i = Y_{ini} + rand() \end{cases} \tag{3}$$

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \tag{4}$$

Algorithm 2: JS-FOA

Parameters: population size (*popsize*), maximum iteration (*maxgen*), upper bound (*Ub*), lower bound (*Lb*), and excitation iteration (φ)

Outputs: *Location** **and** *BestSmell*

Set *popsize*, *maxgen*, **and** φ // *initialization*
Eq. (7) // *swarm location initialization*
Iteration = 0 **and** *BestSmell = Fitness*(S_{ini})
Repeat
While *iteration* < φ
    sub-algorithm 2.1 // *group collaborative search*
    sub-algorithm 2.2 // gradient descent
End
While *iteration* > φ
    sub-algorithm 2.3 // *move memory direction*
    sub-algorithm 2.2 // *gradient descent*
end
Until *iteration = maxgen* **or** *find best solution*

Fig. 5. Structure of the JS-FOA.

Sub-algorithm 2.1: Group collaborative search

Outputs: X_t, Y_t **and** *BestSmell*

$step_x^t(i) = X_t(i) - X_{t-1}(i)\ (i = 1,2,\ldots,N)$ // *calculate the length of step*
$\omega_t(i) = \left(Ls_t(i) - Ss_t(i)\right) * \left[e^{\frac{step_{t-1}(i) - Ss_t(i)}{Ls_t(i) - Ss_t(i)}} - 1\right] + Ss_t(i)$
$X_t(i) = X_{ini}(i) + \omega_x(i) * rand()$ // *group search*
$Y_t(i) = Y_{ini}(i) + \omega_y(i) * rand()$
$Location_t = \begin{cases} \sqrt{X_t^2 + Y_t^2} & if\ X_i + Y_i > 0 \\ -\sqrt{X_t^2 + Y_t^2} & if\ X_i + Y_i < 0 \end{cases}$ // *calculate location*
[SmellBest,BestIndex] = Fitness($Location_{iter}$)
If *SmellBest* < *BestSmell* **then** *BestSmell = SmellBest*
    and then $X_{ini} = X(BestIndex)\ Y_{ini} = Y(BestIndex)$
Endif

Fig. 5(a). Collaborative group search.

Step 2.2: Calculate the fruit fly's smell concentration judgment value (S_i) as the reciprocal of its distance, and obtain its smell concentration ($Smell_i$) by substituting S_i into the fitness as shown in Eq. (5):

$$\begin{cases} S_i = 1/Dist_i \\ Smell_i = fitness(S_i) \end{cases} \quad (5)$$

Step 3: Select the minimum smell concentration of all the fruit flies and move to the corresponding location by using Eq. (6):

$$\begin{cases} [BestSmell, BestIndex] = min(Smell) \\ \begin{cases} X_{ini} = X(BestIndex) \\ Y_{ini} = Y(BestIndex) \end{cases} \end{cases} \quad (6)$$

Step 4: Repeat Steps 2 and 3 until food has been found or the maximum number of iterations has been reached. The complete FOA algorithm is outlined in Fig. 2.

3. Proposed JS-FOA

In this section, an enhanced FOA named JS-FOA is discussed. JS-FOA is consistent with fruit flies' search for food and enables an easy escape from local optima to reach the global optimum. The details are described in the following sections.

3.1. Collaborative group search—mainly for initial position

Fruit flies are wise swarms. When fruit flies begin to search for food, they are initially far from the food source. Then, as they search in various directions, their locations are randomly

Sub-algorithm 2.2: Gradient descent search
Outputs: X^* Y^*

Input X_t **and** Y_t
$Gd_t(i) = \frac{N * \| X_t(i) - X_{t-1}(i) \|}{\sum_i \| X_t(i) \|}$ // Gd_t *is the logoram of* Gd_t^x *and* Gd_t^y
$P(i) = \frac{Gd(i)}{\sum_i Gd(i)}$
If $rand() < P(i)$ **then** $X(i) = rand(Domain)$ // *the same as* $Y(i)$
Endif

Fig. 5(b). Gradient descent search.

Sub-algorithm 2.3: Memory move direction
Outputs: *Location** **and** *BestSmell*

Input X_t **and** Y_t
Eq. (7) // *the number of fly with memory*
$MStep_t = X_t - X_{ini}$ // *Y_t is as same as X_t*
$X_t(i) = X_{ini}(i) + \lambda \cdot MStep_t \cdot rand() \quad i = 1,2,\ldots,f(x)$
$X_t(i) = X_{ini}(i) + \omega_x(i) * rand() \quad i = f(x)+1,\ldots,N$
$Location_t = \begin{cases} \sqrt{X_t^2 + Y_t^2} & if\ X_i + Y_i > 0 \\ -\sqrt{X_t^2 + Y_t^2} & if\ X_i + Y_i < 0 \end{cases}$ // *calculate location*
[SmellBest,BestIndex] = Fitness($Location_t$*)*
If *SmellBest < BestSmell* **then** *BestSmell = SmellBest*
and then $X_{ini} = X(BestIndex)$ $Y_{ini} = Y(BestIndex)$
Endif

Fig. 5(c). Memory move direction.

scattered over a plane. Eq. (7) describes the situation as follows:

$$\begin{cases} \begin{cases} X_{ini} = rand\,(Domain) \\ Y_{ini} = rand(Domain) \end{cases} \\ Location_{ini} = \begin{cases} \sqrt{X_{ini}^2 + Y_{ini}^2} & if\ \ X_i + Y_i > 0 \\ -\sqrt{X_{ini}^2 + Y_{ini}^2} & if\ \ X_i + Y_i < 0 \end{cases} \\ BestSmell_{ini} = Fitness\,(Location_{ini}) \end{cases} \tag{7}$$

where $i = 1, 2, \ldots, N$ is the index of the fruit fly, X_{ini} and Y_{ini} are locations in a coordinate system, and $Location_{ini}$ is the location of a fruit fly after dimensional reduction. These initial values are randomly generated and can form an N-dimensional column vector. Dimensionality reduction can streamline calculations for the smell concentration judgment value and simplify the structure.

When a fly starts searching for food, the food concentration is initially extremely small. Hence, the fly searches randomly by flying in all directions. Eq. (8) describes the situation. Soon, the fly is likely to arrive at a location with a small amount of food. This phenomenon is called a local optimum.

$$\begin{cases} \begin{cases} X_t(i) = X_{ini}(i) + \omega_x(i) \cdot rand() \\ Y_t(i) = Y_{ini}(i) + \omega_y(i) \cdot rand() \end{cases} \\ Location_t = \begin{cases} \sqrt{X_t^2 + Y_t^2} & if\ \ X_i + Y_i > 0 \\ -\sqrt{X_t^2 + Y_t^2} & if\ \ X_i + Y_i < 0 \end{cases} \end{cases} \tag{8}$$

where $i = 1, 2, \ldots, N$ is the index of the fruit fly, $\omega(i)$ $(i = 1, 2, \ldots, N)$ (ω_x and ω_y are unified as ω) is the search range, and $\omega(i)(i = 1, 2, \ldots, N)$ changes according to each change of the i_{th} index. These changes in ω are discussed next together with the relevant equations.

Suppose the steps of the t_{th} iteration are $step_x^t$ and $step_y^t$. The steps can then be calculated as follows:

$$step_x^t\,(i) = X_t\,(i) - X_{t-1}\,(i)\,(i = 1, 2, \ldots, N) \tag{9}$$

where $step_y^t$ is calculated in the same way as $step_x^t$, $step_x^t$ is N-dimensional, and the largest (Ls) and the smallest steps (Ss) can be recorded. For convenience, $step_x$ and $step_y$ are unified as $step$. Thus, $\omega(i)$ can be described as follows:

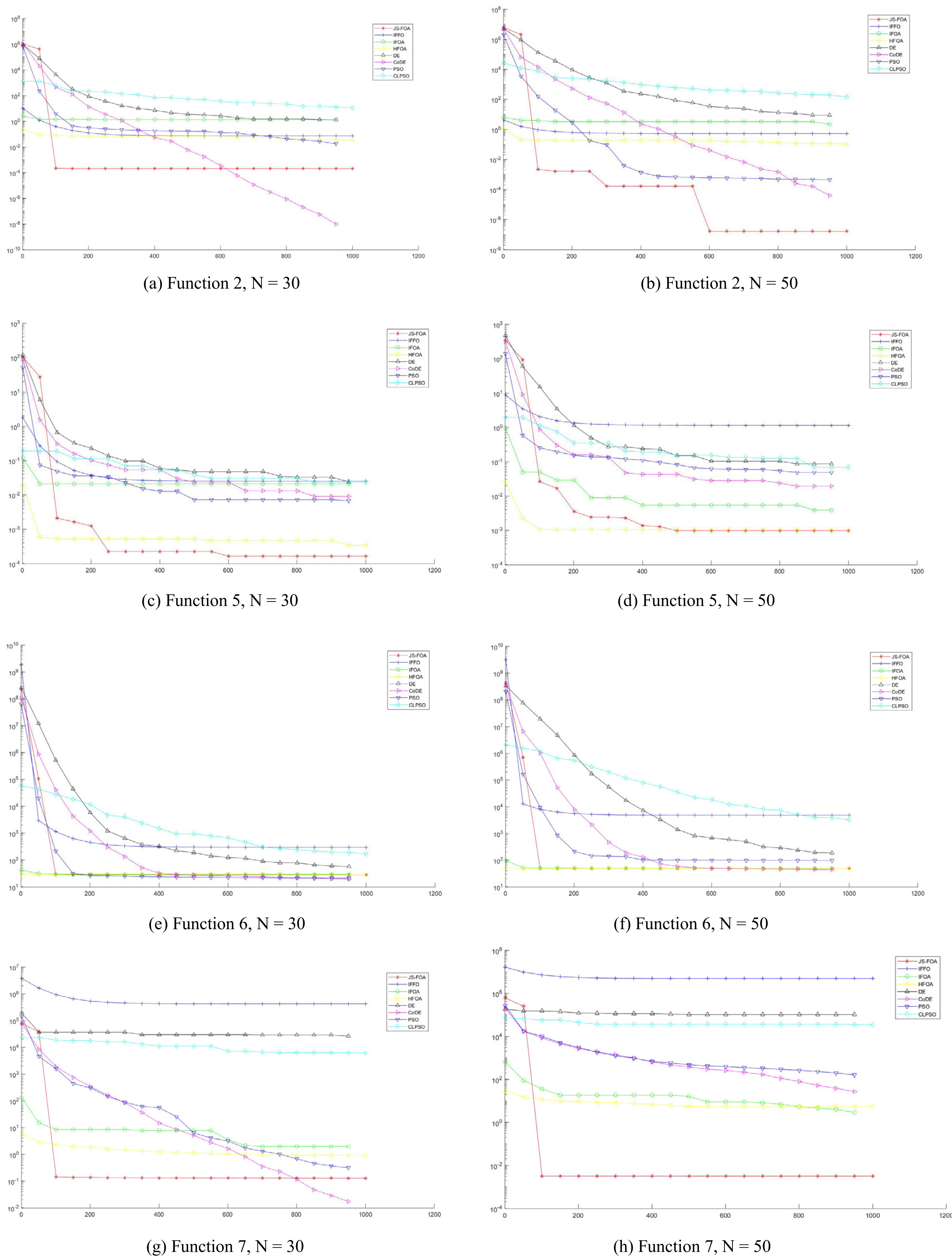
$$\omega_t\,(i) = (Ls_t\,(i) - Ss_t\,(i)) * \left[e^{\frac{step_{t-1}(i) - Ss_t(i)}{Ls_t(i) - Ss_t(i)}} - 1 \right] + Ss_t\,(i) \tag{10}$$

where subscript t is the t_{th} iteration and i is the i_{th} individual fly. From Eq. (10), $\omega_t\,(i) \in [Ss_t, Ls_t]$ and $\omega_t\,(i)$ is changed according to the previous iteration's step.

However, $\omega_t(i)$ is linear. Hence, a normal distribution is proposed to improve the spread of ω_t, as described by Eq. (11):

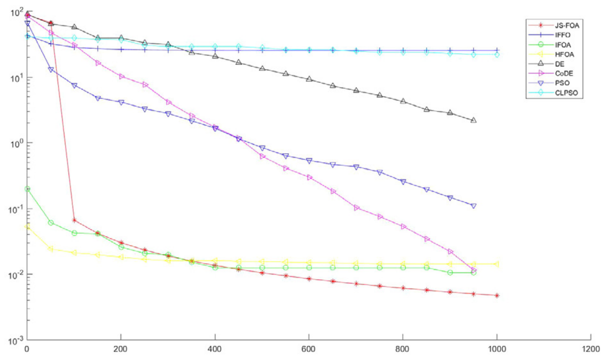
$$f\,(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\,(x - \omega_t)^2\right) \tag{11}$$

where $f\,(x)$ has a normal distribution with mean ω_t and variance, and $x \in [Ss_t, Ls_t]$ has a uniform distribution. Consequently, $f(x)$ can be described as shown by Fig. 3.

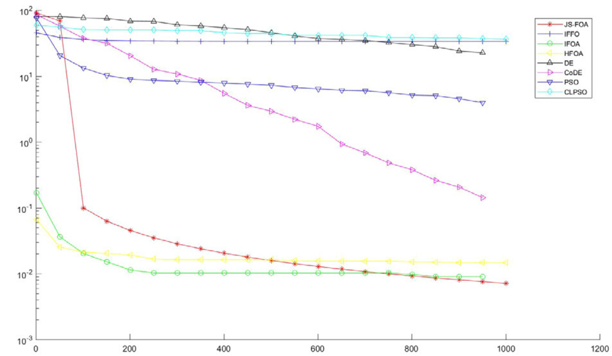


(a) Function 2, N = 30
(b) Function 2, N = 50
(c) Function 5, N = 30
(d) Function 5, N = 50
(e) Function 6, N = 30
(f) Function 6, N = 50
(g) Function 7, N = 30
(h) Function 7, N = 50

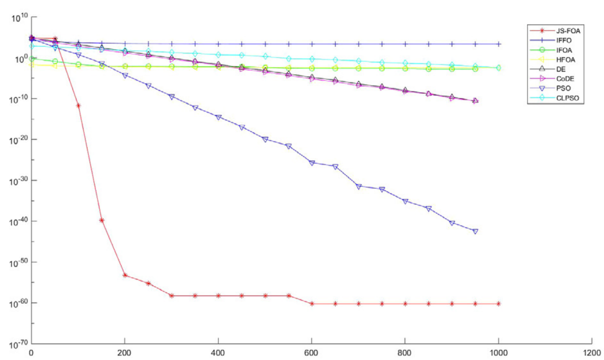
Fig. 6. Convergence graphs for several benchmark functions.



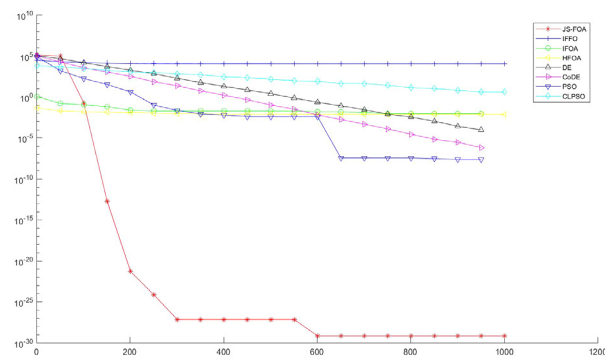
(i) Function 8, N = 30



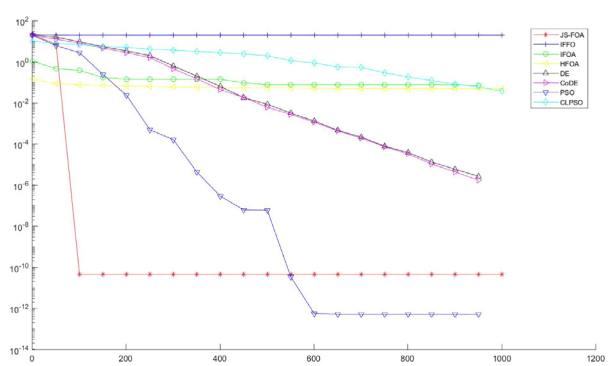
(j) Function 8, N = 50



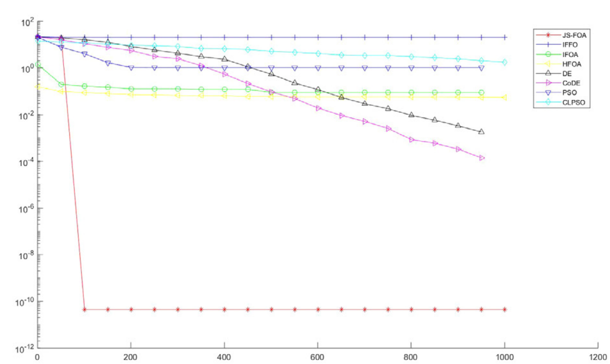
(k) Function 10, N = 30



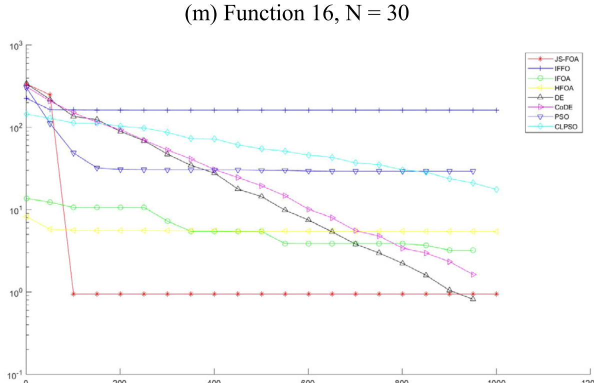
(l) Function 10, N = 50



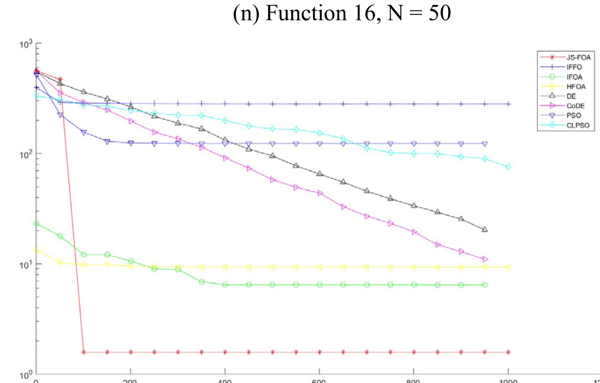
(m) Function 16, N = 30



(n) Function 16, N = 50



(o) Function 18, N = 30



(p) Function 18, N = 50

Fig. 6. (*continued*).

Table 1
Details of benchmark functions.

No.	Functions	x_i	x^*	$f(x^*)$
F1	$f(x)=\sum_{i=2}^{n} ix^2$	$(-5.12,\ 5.12)$	0	0
F2	$\sum_{i=2}^{n} i\left(2x_i^2-x_{i-1}^2\right)^2+(x_1-1)^2$	$(-10,\ 10)$	$2^{(2-2^i)/2^i}$	0
F3	$f(x)=-\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right)$	$(-1,\ 1)$	0	-1
F4	$f(x)=\sum_{i=1}^{n}\left(10^6\right)^{\frac{i-1}{n-1}} x_i^2$	$(-100,\ 100)$	0	0
F5	$f(x)=\sum_{i=1}^{n} ix_i^4+rand(0,\ 1)$	$(-1.28,\ 1.28)$	0	0
F6	$f(x)=\sum_{i+1}^{n-1}[1000\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2]$	$(-30,\ 30)$	0	1
F7	$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	$(-100,\ 100)$	0	0
F8	$f(x)=\max\left(\lvert x_i\rvert\right)$	$(-100,\ 100)$	0	0
F9	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$(-10,\ 10)$	0	0
F10	$f(x)=\sum_{i=1}^{n} x_i^2$	$(-100,\ 100)$	0	0
F11	$f(x)=\sum_{i=1}^{n}\left(\lfloor x_i+0.5\rfloor\right)^2$	$(-100,\ 100)$	0	0
F12	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	$(-1,\ 1)$	0	0
F13	$f(x)=\sum_{i=1}^{n} ix_i^2$	$(-10,\ 10)$	0	0
F14	$f(x)=\sum_{i=1}^{n} x_i^2-450$	$(-100,\ 100)$	0	-450
F15	$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2-450$	$(-100,\ 100)$	0	-450
F16	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	$(-32,\ 32)$	0	0
F17	$f(x)=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	$(-10,\ 10)$	0	0
F18	$f(x)=f_1(x_1,x_2)+f_1(x_2,x_3)+\cdots+f_1(x_n,x_1)$ $f_1(x,y)=\left(x^2+y^2\right)^{0.25}[\sin^2\left(50\left(x^2+y^2\right)^{0.1}\right)+1]$	$(-100,\ 100)$	0	0
F19	$f(x)=f_1(x_1,x_2)+f_1(x_2,x_3)+\cdots+f_1(x_n,x_1)$ $f_1(x,y)=0.5+(\sin^2\left(\sqrt{x^2+y^2}\right)-0.5)/\left(1+0.001\left(x^2+y^2\right)\right)^2$	$(-100,\ 100)$	0	0
F20	$f(x)=\frac{\pi}{n}\left\{10\sin^2(\pi y_i)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]\right.$ $\left.+(y_n-1)^2\right\}+\sum_{i=1}^{n}\mu(x_i,\ 10,\ 100,\ 4)$ $y_i=1+0.25(x_i+1)$ $\mu(x_i,\alpha,k,m)=\begin{cases}k(x_i-\alpha)^m, & x_i>\alpha\\ 0, & -\alpha<x_i<\alpha\\ k(-x_i-\alpha)^m, & x_i<-\alpha\end{cases}$	$(-50,\ 50)$	-1	0
F21	$f(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$(-600,\ 600)$	0	0
F22	$f(x)=$ $-\sum_{i=1}^{n-1}\left(\exp\left(-\frac{x_i^2+x_{x+1}^2+0.5x_ix_{i+1}}{8}\right)\cos\left(4\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}\right)\right)$	$(-5,\ 5)$	0	$1-n$
F23	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n} x_ix_{i-1}$	$(-n^2,n^2)$	$i(n+1-i)$	$\frac{n(n+4)(1-n)}{6}$
F24	$f(x)=\sum_{i=1}^{n-1}\left(0.5+\sin^2\frac{\sqrt{100x_i^2+x_{i+1}^2}-0.5}{1+0.001\left(x_i^2-2x_ix_{i-1}+x_{i+1}^2\right)^2}\right)^2$	$(-100,\ 100)$	0	0
F25	$f(x)=\sum_{i=1}^{n-1}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	$(-5.12,\ 5.12)$	0	0
F26	$f(y)=\sum_{i=1}^{n}(y_i^2-10\cos(2\pi y_i)+10)$ $y_i=\begin{cases}x_i, & \lvert x_i\rvert<1/2\\ \frac{round(2x_i)}{2}, & \lvert x_i\rvert\geq 1/2\end{cases}$	$(-5.12,\ 5.12)$	0	0
F27	$f(x)=1-\cos\left(2\pi\left(\sqrt{\sum_{i=1}^{2} x_i^2}\right)+0.1\sqrt{\sum_{i=1}^{n} x_i^2}\right)$	$(-100,\ 100)$	0	0
F28	$f(x)=$ $\sum_{i=1}^{n}\left\{\sum_{k=0}^{k_{max}}\left[\alpha^k\cos\left(2\pi b^k(x_i+0.5)\right)\right]\right\}-n\sum_{k=0}^{k_{max}}\left[\alpha^k\cos\left(2\pi b^k\cdot 0.5\right)\right]$ $\alpha=0.5,b=0.3,k_{max}=30$	$(-0.5,\ 0.5)$	0	0
F29	$f(x)=\sum_{k=1}^{n}\sum_{j=1}^{n}(\frac{y_{jk}^2}{4000}-\cos\left(y_{jk}\right)+1)$ $y_{jk}=100\left(x_k-x_j^2\right)^2+\left(1-x_j^2\right)^2$	$(-100,\ 100)$	1	0

Finally, instead of using $Location_t$ ($Location_{ini}$) directly, the X_t (X_{ini}) and Y_t (Y_{ini}) are used to enable the algorithm to search in negative fields. If the optimal solution to the problem is negative, $X_t+Y_t<0$ ($X_{ini}+Y_{ini}<0$) will be preserved in this iteration and $X_t+Y_t>0$ ($X_{ini}+Y_{ini}>0$) will be discarded.

3.2. Gradient descent search and memory move direction

In this section, two new methods are used to help the FOA escape from local optima and search for food efficiently and quickly. After a series of collaborative group searches, a swarm

Table 2
Results of the five algorithms (N = 30, *maxgen* = 500).

		JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO	$f(x^*)$
1	Median	**0.00E+00**	9.46E−03	9.39E−02	6.91E−02	2.08E−05	6.55E−06	1.81E−21	1.91E−02	0
	Std.	0.00E+00	4.65E+00	3.23E−18	3.29E−03	3.19E−06	1.59E−06	7.81E−22	9.70E−04	
2	Median	3.99E−02	2.42E+05	5.81E−03	3.47E−02	6.46E+00	**3.76E−03**	1.77E−01	4.14E+01	0
	Std.	3.83E−03	9.81E+03	7.77E−05	7.46E−04	7.07E−01	4.21E−03	9.40E−02	9.78E+00	
3	Median	**−1.00E+00**	0.00E+00	−9.12E−01	−9.97E−01	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	−1
	Std.	1.94E−04	−9.53E−01	−9.97E−01	3.84E−06	1.94E−19	8.51E−11	5.91E−01	3.39E−06	
4	Median	**0.00E+00**	**0.00E+00**	2.54E−05	1.89E+02	3.20E−09	5.59E−01	1.85E−15	1.50E+03	0
	Std.	1.32E+01	5.20E+03	4.11E+00	3.35E−01	8.39E−02	3.77E−02	2.57E−14	9.12E+01	
5	Median	**3.98E−04**	4.73E+04	2.29E−03	**3.58E−04**	4.93E−02	2.34E−02	2.51E−02	4.72E−02	0
	Std.	2.57E−05	3.23E+03	2.83E−04	9.29E−05	4.15E−03	3.21E−03	4.01E−03	7.41E−03	
6	Median	**3.15E+01**	3.17E+05	**3.27E+01**	**2.80E+01**	1.68E+02	2.70E+01	5.78E+01	1.13E+03	1
	Std.	7.89E−02	1.95E+04	2.96E−02	1.14E−01	1.97E+01	2.00E−01	1.75E+01	7.49E+01	
7	Median	**1.95E−02**	1.40E+03	3.01E+00	1.50E+00	3.85E+04	5.69E+00	5.63E+00	1.06E+04	0
	Std.	9.06E−03	3.84E+01	2.10E−02	1.47E−01	1.00E+03	2.94E+00	1.87E+00	2.62E+03	
8	Median	**2.87E−08**	3.59E−02	3.20E−02	1.67E−02	1.73E+01	8.81E−01	9.83E−01	2.89E+01	0
	Std.	1.88E−08	5.48E−04	3.29E−04	1.42E−04	6.91E−01	1.29E−01	9.92E−02	1.26E+00	
9	Median	**2.37E−19**	5.13E+09	4.63E−01	3.81E−01	2.64E−03	7.24E−03	7.53E−04	1.81E−01	0
	Std.	6.86E−20	3.79E+09	5.81E−17	9.01E−03	1.94E−04	3.00E−03	3.20E−04	5.33E−04	
10	Median	2.47E−04	7.76E+00	9.79E−03	5.11E−03	6.51E−04	2.28E−04	**2.47E−19**	7.65E−01	0
	Std.	8.00E−05	2.36E−01	1.75E−04	2.89E−04	1.31E−04	1.23E−04	1.94E−19	3.82E−02	
11	Median	**0.00E+00**	1.74E+00	6.00E−04	3.74E−04	5.91E−01	3.12E−01	8.38E−01	8.32E−01	0
	Std.	0.00E+00	1.99E+04	−1.95E+02	−3.03E+02	8.22E+06	−2.90E+02	2.07E+04	−2.90E+02	
12	Median	5.08E−15	1.04E+29	2.67E−04	2.51E−05	6.76E−14	**3.50E−25**	1.34E−05	2.98E−14	0
	Std.	5.50E−15	9.52E+28	2.80E−20	1.16E−06	2.31E−24	4.20E−26	1.94E−54	4.07E−14	
13	Median	**0.00E+00**	9.41E+01	9.81E−02	7.29E−02	7.67E−05	2.72E−05	2.68E−17	1.23E−01	0
	Std.	0.00E+00	3.66E+00	5.28E−18	3.78E−03	9.93E−06	1.02E−05	2.61E−17	6.03E−03	
14	Median	**−4.50E+02**	−4.37E+02	**−4.50E+02**	**−4.50E+02**	−4.29E+02	**−4.50E+02**	0.00E+00	**−4.50E+02**	−450
	Std.	7.84E−05	2.19E+02	8.56E−05	5.93E−05	1.67E−13	1.34E−04	5.14E−01	9.07E−02	
15	Median	**−4.50E+02**	−5.43E+01	−4.50E+02	−4.49E+02	−4.06E+02	−4.43E+02	−4.41E+02	1.21E+04	−450
	Std.	6.52E−11	5.37E+03	4.15E−14	1.26E−02	3.28E+04	2.77E+00	6.92E−03	1.40E+03	
16	Median	**3.41E−15**	2.71E−02	7.46E−02	6.09E−02	7.14E−03	4.84E−03	8.69E−01	1.46E+00	0
	Std.	4.10E−16	2.04E−05	0.00E+00	9.96E−04	4.99E−04	7.10E−04	3.25E−01	8.06E−02	
17	Median	1.00E+01	7.87E−01	**5.66E−02**	−3.97E−01	0.00E+00	−1.47E+02	0.00E+00	−1.66E+02	0
	Std.	4.63E+00	1.23E−02	4.01E−18	4.04E−01	0.00E+00	1.71E+00	3.15E+00	4.09E−02	
18	Median	**6.16E−01**	**2.30E−01**	6.42E+00	5.43E+00	2.22E+01	1.49E+01	4.50E+01	4.01E+01	0
	Std.	6.26E−02	2.46E−03	4.56E−02	4.30E−03	5.82E−01	3.30E+00	5.83E+00	5.14E−01	
19	Median	**4.52E−04**	1.96E−02	2.29E−02	1.03E−02	9.61E+00	8.70E+00	8.68E+00	6.27E+00	0
	Std.	2.03E−04	2.28E−04	2.88E−04	1.11E−04	2.11E−01	7.20E−02	4.86E−01	1.08E+00	
20	Median	7.81E−01	2.17E+04	2.57E+00	1.73E+00	7.23E−05	**8.49E−06**	2.29E−02	2.47E−02	0
	Std.	2.54E−02	3.19E+03	2.07E−16	1.86E−03	8.82E−06	3.61E−06	2.08E−02	8.39E−03	
21	Median	**6.82E−05**	2.88E−03	6.84E−04	2.81E−04	1.34E−02	1.51E−02	3.62E−03	6.00E−01	0
	Std.	2.11E−05	6.16E−05	6.61E−06	4.38E−06	3.63E−03	2.06E−02	2.77E−03	3.58E−01	
22	Median	**−2.90E+01**	−2.42E−03	−2.89E+01	−2.89E+01	−6.79E+00	−1.73E+01	−1.49E+01	−2.09E+01	1 − n
	Std.	6.86E−15	1.59E−01	2.59E−15	9.55E−03	3.97E−01	2.65E−01	5.72E−14	2.92E−01	
23	Median	**−5.02E−02**	−1.43E−02	−1.96E+02	−3.65E+02	1.23E+05	−2.90E+01	0.00E+00	−2.79E+01	$\frac{n(n+4)(1-n)}{6}$
	Std.	4.58E+00	3.70E−04	1.77E+01	9.52E+01	2.25E+04	1.27E−05	9.66E+02	3.88E−03	
24	Median	**1.85E−04**	**3.98E−04**	3.85E−02	1.02E−02	2.51E+00	2.61E+00	3.88E+00	2.18E+00	0
	Std.	7.03E−04	3.80E−05	7.67E−04	1.30E−03	1.94E−01	1.76E−01	2.02E−01	4.82E−01	
25	Median	**2.75E−14**	8.94E+00	2.05E+00	1.00E+00	8.91E+01	5.20E+01	4.90E+01	1.90E+01	0
	Std.	2.24E−14	3.22E−01	0.00E+00	1.11E−01	2.63E+00	6.86E+00	4.48E+00	3.61E−01	
26	Median	**0.00E+00**	8.66E+00	2.10E+00	9.69E−01	5.95E+01	4.12E+01	5.38E+01	2.04E+01	0
	Std.	0.00E+00	2.62E−01	0.00E+00	4.82E−02	3.10E+00	1.54E+00	2.58E+00	1.43E+00	
27	Median	**6.17E−14**	0.00E+00	3.44E−02	1.98E−02	3.20E−08	2.69E−09	8.02E−09	3.14E−09	0
	Std.	8.02E−09	0.00E+00	2.33E−03	2.81E−02	2.01E−08	3.46E−09	5.37E−14	4.33E−09	
28	Median	−6.20E+00	**−3.83E−02**	−5.31E+00	−5.18E+00	−8.55E+00	−1.44E+01	−1.35E+01	−1.42E+01	0
	Std.	2.52E−01	1.71E−04	8.54E−02	3.53E−01	8.35E−16	2.61E−06	3.33E−15	1.00E−02	
29	Median	**3.32E+02**	2.21E+13	4.34E+02	4.00E+02	8.35E+02	6.54E+02	5.84E+02	7.04E+09	0
	Std.	2.00E+00	3.56E+12	2.30E+00	4.72E−01	1.21E+01	3.29E+01	5.81E+01	3.04E+09	

can sense a food concentration clearly. However, some locations have small food concentrations called local optima that influence fruit fly judgment, causing them to fly to these locations. A gradient descent search can be used to ameliorate this issue.

3.2.1. Gradient descent search—escape from local optimum

A gradient descent modeled on a continuous function gradient change is proposed to calculate changes in fly movements. Supposing that some fruit flies arrive at the local optimum, their

Table 3
Average CPU time for each algorithm to run 30 times (seconds).

maxgen	N	JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO
500	30	32.94	28.01	26.61	41.01	33.81	44.88	33.42	45.15
500	50	52.72	47.03	43.68	62.05	53.33	71.22	55.89	68.37
1000	30	64.26	57.31	55.86	85.88	66.79	81.37	68.36	76.7
1000	50	103.33	93.62	90.13	130.84	103.54	129.12	109.6	133.25
5000	30	326.24	283.42	277.94	423.68	333.27	406.85	340.52	442.54
5000	50	583.48	521.97	495.31	650.21	590.42	645.66	547.24	662.12

Table 4
Wilcoxon rank sum test for 29 functions.

h	N	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO
1	30	27	28	25	27	24	29	26
1	50	28	28	26	29	26	28	28
0	30	2	1	2	2	2	0	2
0	50	1	1	3	0	2	1	0
−1	30	0	0	2	0	3	0	1
−1	50	0	0	0	0	1	0	1

Note: [a]When JS-FOA is significantly better than the contrastive algorithm, $h = 1$;
[b]When JS-FOA is significantly worse than the contrastive algorithm, $h = -1$;
[c]When JS-FOA is not statistically different from the contrastive algorithm, $h = 0$.

gradient descent as described by Eq. (12) reduces gradually.

$$Gd_t(i) = \frac{N * \|X_t(i) - X_{t-1}(i)\|}{\sum_i \|X_t(i)\|} \tag{12}$$

where $\|\cdot\|$ is a 2-norm, N is the number of fruit flies, $Gd(i)$ is the i_{th} fruit fly's gradient descent, and $X_t(i)$ is the i_{th} fruit fly's position during the t_{th} iteration. The sub-swarm is identified by a large Gd as described in Eq. (13):

$$P(i) = \frac{Gd(i)}{\sum_i Gd(i)} \tag{13}$$

where P is the probability of fruit flies escaping from the current position. If a fruit fly escapes from its current position, it obtains a random position from Eq. (7).

3.2.2. Memory move direction—guide for global solution

When a certain number (φ) of iterations has been reached, a fruit fly will have a memory and will not fly randomly. In this case, a new sub-algorithm named memory move direction is proposed to describe the flight path.

Suppose that fruit flies reached X_t (the best location) during the tth iteration. Some of them will not fly randomly, but will fly according to a previous step. As the number of iterations increases, many flies will follow previous paths while other flies still randomly find food. The sub-algorithm is described as follows:

$$MStep_t = X_t - X_{t-1} \tag{14}$$

where the upper limit of Eq. (14) is similar to Eq. (9), and $MStep_t$ is the memory direction. Eqs. (15) and (16) describe the search process as follows:

$$\begin{cases} X_{t+1} = X_t + \lambda \cdot MStep_t \cdot rand() \\ Y_{t+1} = Y_t + \lambda \cdot MStep_t \cdot rand(), \end{cases} \tag{15}$$

$$\lambda = \lambda_{max} \cdot \exp(\log\left(\frac{\lambda_{min}}{\lambda_{max}}\right) \cdot \frac{t}{t_{max}}) \tag{16}$$

where λ is the sensitivity factor determined by an iterative process similar to IFFO.

3.3. Combined search strategy

In this section, a curve modeled on the excitation function of a neural network is proposed to combine the three sub-algorithms. Fig. 4 shows the details of the curve.

The thick line represents the number of flies with memories. Before the φ iteration, a fruit fly participates in collaborative group search, and its memory move direction is activated when the number of iterations reaches φ. The number of fruit flies with memory can be described by Eq. (17) as follows:

$$f(x) = \begin{cases} 0 & x < \varphi \\ \left\lceil M \cdot 2 \cdot \left(\frac{1}{1 + e^{-\frac{x-\varphi}{\varphi}}} - \frac{1}{2} \right) \right\rceil & x \geq \varphi \end{cases} \tag{17}$$

where x is the iteration number and $\lceil\cdot\rceil$ is the ceiling function.

In short, the collaborative group search is responsible for finding the initial solution. During the iterative process, it is still used with fewer flies for a breadth search. At the same time, more files with memory movement help to improve the search accuracy. Thus, the satisfactory performance of JS-FOA is achieved by balancing its exploration and exploitation abilities. The complete enhanced algorithm is outlined in Fig. 5 (Details can be found in Figs 5(a)–5(c)).

3.4. Positive and negative effects of joint search and individual search policies

By using the proposed joint search strategy, the sub-algorithms can support each other. Firstly, the collaborative group search can quickly find an initial position. A good initial position will have a significant impact on finding the global optimum value, while a bad initial position may lead to falling into a local optimum. At this point, the gradient descent search can support the collaborative group search by helping the fruit fly to escape from the local optimum. Thus, collaborative group search and gradient descent search are used together to improve the convergence speed.

Secondly, although the collaborative group search can quickly find the initial position, the search precision may be low. Here, the memory move direction will be very helpful. By using a combined search strategy, collaborative group search and memory move direction can gradually transition from position search to precision search. Thus, collaborative group search and memory move direction are used together to improve the solution accuracy.

In summary, the three sub-algorithms have different effects. If each sub-algorithm is applied individually, the collaborative group search may search quickly but its accuracy is not high, and the memory move direction may have a high precision result but searches slowly. As they may both fall into a local optimum, the gradient descent search is important to help escape from the local optimum. Therefore, the joint search strategy is utilized in this study.

4. Performance tests and analysis of results

The algorithms discussed in this section were coded in Matlab 2014b and computations were conducted on a personal computer with Intel®Core i5 processor, 2.7 GHz CPU, 8 GB RAM, and a Mac OS 10.13.1 operating system.

4.1. Benchmark functions

The 29 benchmark functions listed in Table 1, are all well-known benchmark functions [19,20,22,28]. The first 15 functions (F1 to F15) are unimodal, whereas the rest (F16 to F29) are multimodal. Unimodal functions find global solutions more easily than multimodal functions. Furthermore, some multimodal function structures are complex, with peaks, valleys, channels, and flat hyperplanes of different heights. Hence, these functions are suitable for verifying the proposed JS-FOA.

4.2. Algorithms used for comparison

The following algorithms with good structures, appropriate convergence speeds, and robustness are used for comparison for the following reasons:

(a) The IFFO proposed by Reference [19] has a better structure and convergence speed than the basic FOA, MFOA, and few harmony search algorithms.

(b) The IFOA proposed by Reference [21] also has good structure and robustness. Compared with the basic FOA and few other heuristic algorithms, the IFOA has a good convergence speed.

(c) The hybrid FOA (HFOA) proposed by Reference [29] is the latest enhanced FOA known at the date of writing this paper. Compared to other harmony search algorithms, HFOA usually performs well on the test functions. Also, HFOA has a simple structure.

(d) DE is a meta-heuristic algorithm with an efficient and effective structure. This algorithm is similar to GA but has a simpler structure, is easier to understand, has a faster convergence speed, and is more robust than GA.

(e) CoDE is an enhanced algorithm based on DE, which was proposed in 2011 by Reference [30]. So far, CoDE has been cited more than one hundred times.

(f) PSO is a classic algorithm with a good structure. Furthermore, PSO is a parallel algorithm that has a fast convergence speed.

(g) CLPSO was first proposed in 2006 but has been cited more than a thousand times. In most cases, CLPSO performs better than PSO [31].

4.3. Parameter setting

The termination condition of the experiment is the number of iterations (*maxgen*). In IFFO, IFOA and HFOA *maxgen* is 5000, 1000 and 1000, respectively, and in DE (CoDE) and PSO (CLPSO), the reasonable parameter selection range is 500 to 1000. Hence, the termination conditions of this experiment are set to 500, 1000, and 5000, respectively.

4.3.1. Parameter setting of the proposed JS-FOA

For the JS-FOA, the transformation parameter φ is set to a value in the neighborhood of $maxgen/10$. For example, when $maxgen = 500$, φ is varied from 30 to 70. When φ is small, the convergence speed is fast. When φ is large, the enhanced algorithm will have a good probability of escaping from a local optimum. Thus, $\varphi = maxgen/10$ is suggested for a comparison with other algorithms. For example, when $maxgen = 500$, φ is set to 50 for comparison with other algorithms and varied from 30 to 70 for comparison with itself. The value of M (the number of fruit flies with memory) is set to $N/2$ for comparison with other algorithms and set to $N, N/2, N/3, N/4$ for comparison with itself. The population size (*popsize*) of JS-FOA is set to 50, the same as the other algorithms, while λ_{max} is set to $(Ub-Lb)/2$ and λ_{min} is set to 10^{-5}, which is similar to the values used in IFFO.

4.3.2. Parameter setting of the algorithms used for comparison

According to the suggestion of reference [19] for IFFO, k_{max} is set to $(Ub-Lb)/2$ and k_{min} is set to 10^{-5}. For IFOA, the parameters are similar to those of IFFO and JS-FAO. Based on the experience of reference [29] for HFOA, $ct = 0.6$, $mt = 0.2$, $\alpha = 0.6$, $FP_{min} = 0.05$, and $FP_{max} = 0.9$. The other parameters are similar to those for IFFO and JS-FAO.

For DE, CoDE, PSO, and CLPSO, classic parameter selection is used. The parameters of DE and CoDE are selected as follows: crossover probability $Cr = 0.9$ and mutation probability $Mu = 0.1$. For PSO and CLPSO, the learning factor $c_1 = c_2 = 2$, inertia weight $\omega_{max} = 0.9$, $\omega_{min} = 0.4$, and $V_{max} = 0.2 \cdot (Ub - Lb)$.

4.4. Results and analysis

4.4.1. Results

Following the related experimental design experiences of references [32–36], experiments of six combinations with different dimensionality (N) and maximum number of iterations (*maxgen*) are designed to further verify the overall performance of JS-FOA. For all the experiments, the entire problem is solved 30 times with $maxgen = 500$, 1000, and 5000. Each benchmark function is tested 30 times for each algorithm, and every solution is recorded. The median and standard deviations (Std) are listed in Tables. 2 and A.1 to A.5, respectively. The result is shown as 0 if Std. is less than 10^{-100}. Average convergence graphs for several 30-dimensional and 50-dimensional problems are shown in Fig. 6. In order to show the efficiency of every algorithm, the average CPU time for the algorithm to run 30 times for all functions is listed in Table 3.

4.4.2. Analysis of results

The performance of intelligent algorithms is always evaluated by their convergence accuracy, computational stability, and convergence speed. In related studies [25,26], convergence accuracy and computation reliability are often represented by the mean and standard deviation, respectively, while convergence speed is often displayed in graphical form. Therefore, the results are shown in the previous section and Appendix, and analyzed below:

(a) The results in Table 2 list that JS-FOA obtains better solutions than the other algorithms for F1, F4, F5, F7 to F9, F11, F13, F15, F16, F19, F21, F23, F25 to F27, and F29 when $N = 30$ and $maxgen = 500$. For F3, F6, F14, F18, and F24, the quality of the solutions are highly similar. For F2, F10, F12, F17, F20, F22, and F28, the solutions of JS-FOA are poor when compared to the other algorithms. Therefore, JS-FOA obtains better solutions than the other algorithms for most benchmark functions. Even when the dimensionality is increased to 50 and *maxgen* is increased to 1000 and 5000, the results are similar to when the dimensionality is 30. For DE, CoDE, PSO and CLPSO, the algorithms do not easily fall into local optima when the number of iterations is large enough. Thus, the results in the previous section and Appendix show that the good structure of JS-FOA prevents it from falling into local optima.

(b) Computational stability is evaluated by the standard deviation (Std). As seen in Table 2, the Stds of JS-FOA are lower than those of the other three algorithms () for most of the 29 functions. Therefore, JS-FOA is more robust than the other algorithms.

(c) JS-FOA is highly efficient. Fig. 6 shows that JS-FOA has efficient global search capabilities, which are reflected in the fast convergence speed of its iterative process and its high accuracy.

(d) CPU time is an important basis for measuring the structure of an algorithm. The shorter the CPU time, the simpler the algorithm structure. As summarized in Table 3, the enhanced FOAs have simpler structures than other algorithms. For the proposed JS-FOA, the structure is simpler than that of the latest enhanced

Table A.1
Results of eight algorithms ($N = 50$, $maxgen = 500$).

		JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO	$f(x^*)$
1	Median	**0.00E+00**	3.38E−01	9.31E−02	2.39E−01	2.52E−05	6.08E−03	2.32E−21	4.99E+00	0
	Std.	0.00E+00	1.64E−02	3.28E−18	1.51E−02	2.61E−06	1.22E−03	7.14E−22	3.61E−01	
2	Median	4.39E−02	9.05E+00	5.64E−03	1.19E−01	7.31E+00	**1.37E+00**	2.55E−01	1.07E+03	0
	Std.	3.55E−03	1.25E+00	7.96E−05	2.98E−03	6.57E−01	5.04E−01	8.79E−02	5.44E+02	
3	Median	**−1.00E+00**	**−1.00E+00**	−9.36E−01	−9.95E−01	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	−1
	Std.	0.00E+00	2.54E−04	2.86E−05	2.24E−04	3.37E−09	1.45E−06	1.87E−15	7.33E−04	
4	Median	3.19E+01	1.32E+03	6.02E+02	4.16E+02	9.83E−01	3.00E+02	**4.82E−14**	3.41E+05	0
	Std.	1.40E+01	4.05E+01	4.40E+00	1.84E+01	8.66E−02	1.76E+01	2.65E−14	1.31E+05	
5	Median	**4.08E−04**	1.24E−01	2.78E−03	1.07E−03	4.34E−02	5.01E−02	2.63E−02	1.59E−01	0
	Std.	3.30E−05	1.99E−02	2.85E−04	6.18E−05	3.38E−03	1.28E−02	3.42E−03	2.85E−02	
6	Median	**3.23E+01**	3.95E+02	**2.87E+01**	**4.83E+01**	1.93E+02	7.59E+01	4.98E+01	2.75E+04	1
	Std.	7.32E−02	4.21E+01	2.39E−02	5.51E−02	1.59E+01	5.87E+00	2.02E+01	9.25E+03	
7	Median	**1.91E−02**	3.65E+04	3.13E+00	7.89E+00	3.61E+04	3.76E+02	5.66E+00	4.45E+04	0
	Std.	9.05E−03	1.09E+03	2.96E−02	3.92E−01	9.96E+02	8.61E+01	2.43E+00	8.24E+02	
8	Median	**2.71E−08**	1.82E+01	2.77E−02	1.79E−02	1.43E+01	2.82E+00	9.75E−01	4.13E+01	0
	Std.	1.99E−08	1.16E+00	3.39E−04	3.70E−04	6.47E−01	4.07E−01	1.04E−01	2.95E+00	
9	Median	**1.62E−19**	1.37E+00	4.80E−01	6.80E−01	3.53E−03	2.01E−01	8.06E−04	3.32E+00	0
	Std.	6.93E−20	7.37E−03	5.93E−17	5.46E−03	1.44E−04	1.43E−03	3.34E−04	2.93E−01	
10	Median	2.42E−04	1.05E−01	1.01E−02	9.31E−03	7.02E−04	8.50E−02	**3.01E−19**	8.98E+01	0
	Std.	8.32E−05	2.86E−02	2.03E−04	1.03E−03	1.85E−04	2.72E−02	2.84E−19	1.11E+01	
11	Median	**0.00E+00**	6.44E+00	1.03E−02	1.58E−02	1.03E−03	1.14E−01	6.30E+00	8.20E+01	0
	Std.	0.00E+00	3.78E+00	2.06E−02	3.16E−02	2.06E−03	2.27E−01	3.50E+00	2.40E+01	
12	Median	5.80E−15	2.91E−04	2.50E−04	2.60E−05	7.33E−14	**2.14E−20**	1.51E−05	4.37E−12	0
	Std.	5.07E−15	6.17E−07	2.64E−20	6.17E−07	2.18E−24	3.02E−20	1.30E−54	5.02E−12	
13	Median	**0.00E+00**	3.71E−01	1.02E−01	2.33E−01	7.74E−05	3.57E−02	3.17E−17	1.58E+01	0
	Std.	0.00E+00	2.82E−02	5.47E−18	7.73E−03	9.77E−06	2.05E−02	2.31E−17	2.33E+00	
14	Median	**−4.50E+02**	−2.18E+03	**−3.84E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−3.52E+02	−450
	Std.	8.05E−05	4.53E−02	9.20E−05	9.52E−05	6.30E−05	4.51E−02	1.67E−13	1.61E+01	
15	Median	**−4.50E+02**	3.83E+04	−4.07E+02	−4.47E+02	3.97E+04	−1.48E+02	−4.41E+02	4.22E+04	−450
	Std.	7.22E−11	9.98E+02	5.13E−14	5.57E−04	9.86E+02	9.05E+00	2.86E+00	7.58E+02	
16	Median	**2.91E−15**	1.13E+00	7.20E−02	6.51E−02	7.21E−03	7.49E−02	9.14E−01	4.64E+00	0
	Std.	4.52E−16	3.43E−01	0.00E+00	3.62E−04	4.19E−04	7.80E−03	3.34E−01	1.71E−01	
17	Median	1.01E+01	5.26E+02	**5.12E−02**	7.93E+00	1.93E+02	1.99E+02	1.26E+02	2.49E+02	0
	Std.	4.71E+00	1.69E+01	4.04E−18	1.13E+01	0.00E+00	2.90E+00	2.64E+00	1.40E+00	
18	Median	**6.18E+00**	1.47E+02	6.12E+00	9.31E+00	2.10E+01	5.83E+01	5.26E+01	1.43E+02	0
	Std.	6.26E−02	1.26E+01	5.04E−02	1.14E−01	5.67E−01	5.99E+00	5.83E+00	5.38E+00	
19	Median	**4.51E−04**	3.59E+01	2.05E−02	1.89E−02	9.67E+00	1.75E+01	8.70E+00	1.50E+01	0
	Std.	1.63E−04	1.02E+00	2.16E−04	2.09E−03	1.96E−01	2.61E−01	5.58E−01	1.37E−01	
20	Median	7.40E−01	4.16E+00	2.59E+00	1.53E+00	6.82E−05	**3.51E−03**	2.99E−02	4.36E+00	0
	Std.	2.59E−02	2.73E−02	2.61E−16	6.34E−04	8.93E−06	2.51E−03	2.42E−02	1.22E+00	
21	Median	**6.67E−05**	1.41E−01	6.73E−04	3.83E−04	1.22E−02	1.24E−01	3.52E−03	1.74E+00	0
	Std.	1.56E−05	7.25E−02	6.67E−06	1.78E−05	3.72E−03	6.59E−02	2.90E−03	6.23E−02	
22	Median	**−4.90E+01**	−1.16E+02	−2.30E+01	−4.88E+01	−1.36E+01	−2.09E+01	−9.91E+00	−2.96E+01	1 − n
	Std.	7.28E−15	2.01E+00	3.22E−15	2.91E−02	4.42E−01	2.02E−01	1.33E+00	1.21E+00	
23	Median	−6.80E+01	1.21E+05	−2.04E+02	−1.29E+03	1.25E+05	−4.90E+01	−2.39E+03	**−4.09E+01**	$\frac{n(n+4)(1-n)}{6}$
	Std.	4.14E+00	2.65E+04	2.29E+01	1.14E+03	2.44E+04	1.28E−03	9.60E+02	2.86E+00	
24	Median	1.98E−03	**1.28E+01**	3.31E−02	2.33E−02	2.48E+00	6.21E+00	4.02E+00	5.02E+00	0
	Std.	7.33E−04	4.66E−01	7.41E−04	4.36E−03	1.86E−01	2.52E−02	2.50E−01	2.73E−02	
25	Median	**2.57E−14**	3.93E+02	1.37E+00	2.08E+00	9.05E+01	2.43E+02	5.61E+01	9.13E+01	0
	Std.	2.40E−14	1.28E+01	0.00E+00	1.41E−01	2.89E+00	5.02E+00	4.74E+00	2.61E+00	
26	Median	**0.00E+00**	2.85E+02	2.15E+00	1.99E+00	6.15E+01	1.73E+02	4.65E+01	7.05E+01	0
	Std.	0.00E+00	7.88E+00	0.00E+00	8.46E−02	4.01E+00	1.04E+00	2.75E+00	6.66E+00	
27	Median	7.58E−09	2.97E−02	2.97E−02	5.12E−13	2.94E−08	3.01E−09	**5.86E−14**	1.45E−08	0
	Std.	8.02E−09	2.53E−03	2.53E−03	4.26E−13	1.61E−08	5.60E−10	5.69E−14	1.36E−08	
28	Median	−1.00E+00	**−6.39E+01**	−6.14E+00	−5.87E+00	−1.40E+01	−2.39E+01	−1.39E+01	−2.24E+00	0
	Std.	2.51E−01	1.13E−01	9.23E−02	2.04E−02	8.65E−16	9.69E−05	3.86E−15	1.52E−01	
29	Median	**5.05E+02**	5.29E+03	4.97E+02	1.11E+03	7.62E+02	2.37E+03	5.47E+02	4.18E+12	0
	Std.	2.15E+00	1.18E+02	1.78E+00	1.06E+01	2.05E+01	2.79E+01	5.68E+01	1.46E+11	

FOA (HFOA) and almost the same as the other two enhanced FOAs (IFFO, IFOA). Therefore, JS-FOA has good potential for solving real-world problems.

(e) JS-FOA performs excellently because the two search strategies that jointly optimize and the strategy that helps it to escape from local optima are designed to increase diversity in the

Table A.2
Results of eight algorithms ($N = 30$, $maxgen = 1000$).

		JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO	$f(x^*)$
1	Median	**0.00E+00**	5.06E−01	2.83E−01	4.74E−02	1.76E−01	6.74E−14	3.39E−06	1.05E−04	0
	Std.	0.00E+00	8.08E−03	2.84E−04	1.78E−03	6.02E−03	2.91E−14	2.45E−06	3.37E−05	
2	Median	2.14E−01	9.05E+01	**1.07E−02**	3.18E−02	9.02E+01	3.27E−08	2.32E−01	7.17E+00	0
	Std.	7.11E−03	6.54E+00	2.12E−04	2.43E−04	6.33E+00	2.55E−08	2.10E−01	2.69E+00	
3	Median	**−1.00E+00**	−9.10E−01	−9.41E−01	−9.98E−01	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	−1
	Std.	6.03E−17	4.77E−05	3.91E−05	5.59E−06	3.04E−06	1.57E−16	1.69E−08	3.90E−08	
4	Median	6.06E+01	4.05E+03	1.00E+03	9.49E+01	2.95E+03	**3.91E−09**	8.00E+00	1.24E+01	0
	Std.	1.97E+01	2.49E+02	1.83E+01	2.21E+00	2.24E+02	1.76E−09	4.56E+00	2.99E+00	
5	Median	**7.58E−04**	2.94E−01	7.86E−03	3.15E−04	2.05E−01	1.14E−02	6.97E−02	2.93E−02	0
	Std.	7.50E−05	3.20E−02	9.07E−04	3.69E−05	1.70E−02	7.67E−03	6.39E−03	1.25E−02	
6	Median	5.30E+01	2.87E+03	**4.99E+01**	2.78E+01	2.69E+03	1.78E+01	8.31E+01	1.91E+02	1
	Std.	9.84E−02	1.10E+02	2.35E−02	1.13E−01	9.29E+01	7.96E−01	1.62E+01	7.41E−02	
7	Median	2.50E−01	1.98E+05	2.19E+01	9.47E−01	1.97E+05	**7.88E−03**	8.36E+02	6.66E+03	0
	Std.	6.09E−02	4.37E+03	2.02E−01	3.41E−02	4.13E+03	4.01E−03	2.32E+02	6.88E+02	
8	Median	**1.00E−06**	5.97E+01	3.62E−02	1.40E−02	5.25E+01	9.05E−03	7.11E+00	2.04E+01	0
	Std.	7.72E−07	2.47E+00	5.65E−04	1.21E−04	2.05E+00	3.43E−03	4.11E−01	1.02E+00	
9	Median	**6.27E−20**	1.93E+00	7.46E−01	3.22E−01	4.98E−01	5.52E−07	3.65E−01	8.99E−03	0
	Std.	2.87E−20	1.55E−01	8.81E−17	1.07E−02	2.05E−02	3.59E−07	1.23E−01	1.45E−03	
10	Median	4.26E−04	2.49E+00	3.16E−02	3.38E−03	2.45E+00	**1.31E−12**	3.66E−03	2.49E−03	0
	Std.	2.34E−04	1.80E−01	2.37E−04	3.67E−04	1.76E−01	3.12E−13	3.54E−03	3.56E−04	
11	Median	**0.00E+00**	8.54E+01	8.02E−04	1.12E−02	3.05E+00	3.43E−03	8.23E+01	1.03E+00	0
	Std.	0.00E+00	2.93E+01	3.26E−02	1.50E−02	6.11E−01	3.43E−03	2.86E+01	1.04E+00	
12	Median	3.09E−14	2.89E−04	2.80E−04	9.49E−06	5.78E−14	4.29E−19	**5.93E−37**	**3.79E−37**	0
	Std.	5.40E−13	3.43E−08	2.54E−20	3.43E−08	1.43E−14	6.08E−50	4.40E−37	3.89E−22	
13	Median	**0.00E+00**	7.48E−01	3.00E−01	4.81E−02	4.00E−01	2.81E−13	5.53E−05	2.99E−04	0
	Std.	0.00E+00	2.95E−02	1.29E−17	4.53E−03	2.49E−02	1.83E−13	5.19E−05	3.73E−05	
14	Median	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−4.48E+02	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−450
	Std.	2.08E−04	1.92E−01	2.78E−04	1.26E−05	1.92E−01	2.17E−12	1.76E−05	6.19E−05	
15	Median	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	2.04E+05	**−4.50E+02**	4.23E+02	4.31E+03	−450
	Std.	0.00E+00	4.57E+03	2.80E−03	6.78E−03	4.45E+03	2.55E−03	1.15E+02	1.63E+03	
16	Median	**2.24E−15**	3.25E+00	7.34E−02	4.84E−02	6.91E−01	3.88E−07	2.43E+00	3.40E−02	0
	Std.	3.09E−16	2.57E−01	2.84E−04	3.86E−03	7.02E−02	6.72E−10	1.83E−01	1.42E−02	
17	Median	4.73E+00	−7.29E+02	**4.71E−02**	−4.24E+01	−3.22E+02	−1.66E+02	−1.98E+02	−1.75E+02	0
	Std.	2.16E+00	1.78E+01	2.22E−01	9.83E+00	0.00E+00	3.85E+00	3.87E+00	1.35E+00	
18	Median	1.82E+01	2.93E+02	1.07E+01	**5.43E+00**	1.03E+02	1.07E+00	1.73E+02	1.79E+01	0
	Std.	1.20E−01	2.06E+01	5.96E−02	4.77E−03	9.38E−01	5.71E−03	1.96E+01	1.01E−01	
19	Median	**1.92E−03**	5.03E+01	4.50E−02	**7.24E−03**	2.35E+01	7.15E+00	1.96E+01	3.93E+00	0
	Std.	3.31E−04	9.51E−01	6.46E−04	8.17E−06	3.54E−01	1.11E−01	4.85E−01	5.00E−01	
20	Median	8.52E−01	6.08E+00	1.89E+00	1.71E+00	2.27E+00	4.36E−04	2.07E−01	**1.02E−15**	0
	Std.	2.61E−02	3.53E−01	7.79E−05	5.74E−04	2.79E−01	1.40E−14	7.37E−02	2.46E−05	
21	Median	5.86E−05	1.07E+00	8.90E−04	1.90E−04	1.04E+00	**6.99E−11**	2.85E−02	1.54E−02	0
	Std.	2.06E−05	3.65E−02	1.28E−05	5.06E−06	2.04E−02	2.11E−11	1.61E−02	5.88E−03	
22	Median	**−2.90E+01**	**−2.90E+01**	−2.89E+01	−2.89E+01	−9.24E+00	−2.01E+01	**−2.90E+01**	−2.28E+01	1 − n
	Std.	5.53E−16	1.74E+00	6.00E−04	3.74E−04	5.91E−01	3.12E−01	8.38E−01	8.32E−01	
23	Median	**−1.93E+02**	1.99E+04	−1.95E+02	−3.03E+02	8.22E+06	−2.90E+02	2.07E+04	−2.90E+02	$\frac{n(n+4)(1-n)}{6}$
	Std.	4.26E+00	6.36E+03	8.07E+00	2.36E+02	7.61E+05	5.71E−11	6.11E+03	9.70E−03	
24	Median	**1.85E−03**	1.53E+01	9.80E−02	**3.99E−03**	7.24E+00	2.04E+00	5.96E+00	1.91E+00	0
	Std.	7.01E−04	3.14E−01	3.13E−03	5.73E−04	1.57E−01	6.09E−02	9.29E−02	1.65E−02	
25	Median	**0.00E+00**	3.79E+02	2.59E+00	6.61E−01	2.59E+02	2.24E+01	9.45E+01	3.58E+00	0
	Std.	0.00E+00	1.52E+01	5.56E−03	1.07E−01	7.01E+00	7.81E−01	7.29E+00	4.83E−03	
26	Median	**6.12E−15**	3.99E+02	2.47E+00	6.84E−01	2.76E+02	2.41E+01	9.59E+01	9.83E+00	0
	Std.	4.76E−15	1.30E+01	4.32E−03	4.85E−02	8.52E+00	1.27E+00	3.19E+00	2.16E+00	
27	Median	1.46E−11	5.26E−02	5.26E−02	**1.69E−13**	2.28E−08	1.96E−10	5.00E−11	4.45E−10	0
	Std.	1.54E−11	4.11E−03	4.11E−03	5.16E−14	8.10E−09	2.42E−10	3.20E−11	5.96E−10	
28	Median	−6.30E+00	−6.69E+01	−6.55E+00	**−5.86E+00**	−2.14E+01	−1.44E+01	−1.87E+01	−1.43E+01	0
	Std.	1.80E−01	3.13E−01	9.81E−02	2.14E−01	4.77E−09	1.83E−12	2.16E−14	1.30E−03	
29	Median	1.49E+03	5.42E+03	1.99E+03	**3.84E+02**	6.98E+08	**4.93E+02**	2.56E+03	1.98E+08	0
	Std.	3.19E+00	7.53E+01	4.03E+00	4.45E−01	1.62E+08	1.55E+01	5.53E+01	1.43E+08	

population search. Compared with DE, CoDE, PSO, CLPSO, and the three enhanced FOAs, JS-FOA has better global search ability. The collaborative group search can easily find the initial position, and the memory move direction can guide the search to a global optimum. These factors increase the robustness of JS-FOA. The gradient descent search can significantly reduce the risk of falling into a local optimum. When dealing with multimodal problems, the gradient descent search can balance iterations and prevent early maturation.

Table A.3
Results of eight algorithms ($N = 50$, $maxgen = 1000$).

		JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO	$f(x^*)$
1	Median	**0.00E+00**	3.48E−01	9.80E−02	1.54E−01	2.28E−05	3.20E−09	2.50E−21	1.68E−01	0
	Std.	0.00E+00	1.94E−01	2.66E−18	7.92E−03	3.46E−06	8.84E−10	7.39E−22	1.83E−02	
2	Median	5.00E−02	2.23E+02	5.72E−03	1.05E−01	7.16E+00	**1.08E−05**	1.97E−01	2.16E+02	0
	Std.	2.78E−03	2.96E+01	2.06E−05	1.28E−02	6.79E−01	4.54E−06	9.60E−02	2.88E+01	
3	Median	**−1.00E+00**	**−1.00E+00**	−8.98E−01	−9.97E−01	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	−1
	Std.	0.00E+00	1.73E−04	1.92E−16	1.49E−04	3.22E−09	8.90E−13	2.49E−15	2.40E−05	
4	Median	7.55E−02	8.97E+03	6.16E+02	2.30E+02	1.01E+00	2.02E−04	**4.75E−14**	8.12E+03	0
	Std.	7.57E−02	1.82E+03	8.20E−01	7.29E+00	8.25E−02	7.26E−05	2.35E−14	1.82E+03	
5	Median	**3.09E−04**	1.58E−01	2.89E−03	**8.98E−04**	4.23E−02	1.66E−02	2.74E−02	6.79E−02	0
	Std.	2.42E−05	2.69E−02	3.78E−04	3.51E−04	4.07E−03	3.43E−03	3.81E−03	1.49E−02	
6	Median	3.19E+01	**2.55E+01**	3.43E+01	4.82E+01	1.84E+02	4.33E+01	4.92E+01	1.80E+03	1
	Std.	8.27E−02	2.47E+02	2.72E−02	1.06E−01	1.95E+01	9.16E−01	2.01E+01	2.06E+02	
7	Median	**4.18E−06**	6.84E+04	2.46E+00	5.14E+00	3.69E+04	2.40E+01	5.88E+00	3.14E+04	0
	Std.	2.80E−06	1.32E+03	1.74E−02	2.98E−01	1.02E+03	1.62E+01	1.94E+00	2.79E+02	
8	Median	**3.01E−37**	5.40E+01	2.86E−02	1.54E−02	1.82E+01	1.12E−01	1.02E+00	3.46E+01	0
	Std.	3.31E−37	2.57E+00	3.34E−04	6.59E−04	7.16E−01	5.24E−03	9.88E−02	1.75E+00	
9	Median	**2.55E−19**	1.55E+00	4.43E−01	5.72E−01	3.49E−03	2.65E−04	7.97E−04	5.28E−01	0
	Std.	1.05E−19	6.27E−02	5.35E−17	2.59E−03	1.50E−04	3.80E−05	3.46E−04	5.95E−02	
10	Median	7.03E−09	2.48E+00	8.37E−03	7.04E−03	6.26E−04	1.35E−07	**2.97E−19**	2.47E+00	0
	Std.	7.62E−09	3.74E−01	5.21E−05	5.46E−04	1.22E−04	1.35E−07	2.86E−19	3.73E−01	
11	Median	**0.00E+00**	6.36E+01	4.80E−01	5.95E−01	1.82E+01	1.12E−01	6.60E+00	3.76E+01	0
	Std.	0.00E+00	6.51E+01	9.32E−01	1.17E+00	1.82E+01	1.12E−01	4.09E+00	4.06E+01	
12	Median	**0.00E+00**	2.68E−04	2.58E−04	9.86E−06	6.71E−24	9.59E−41	2.10E−54	1.70E−18	0
	Std.	0.00E+00	3.08E−07	2.19E−20	3.08E−07	1.85E−24	1.18E−40	1.29E−54	1.82E−18	
13	Median	0.00E+00	7.17E−01	9.87E−02	1.51E−01	7.89E−05	1.82E−08	3.45E−17	4.67E−01	0
	Std.	0.00E+00	6.80E−02	5.75E−18	1.25E−02	1.01E−05	1.47E−08	2.32E−17	5.55E−02	
14	Median	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−4.47E+02	−450
	Std.	3.64E−15	5.61E−01	7.30E−05	8.15E−04	6.87E−05	9.27E−09	2.60E−13	5.60E−01	
15	Median	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−4.48E+02	4.22E+04	−4.38E+02	−4.41E+02	3.66E+04	−450
	Std.	1.94E−15	7.94E+03	4.66E−14	3.07E−02	9.85E+02	3.05E+00	2.36E+00	6.95E+03	
16	Median	**3.90E−15**	2.78E+00	7.32E−02	5.56E−02	7.62E−03	7.22E−05	9.28E−01	1.71E+00	0
	Std.	4.48E−16	3.88E−01	0.00E+00	7.98E−04	4.15E−04	6.83E−06	3.76E−01	1.01E−02	
17	Median	**−4.78E+00**	−9.03E+02	5.46E−02	−8.21E+01	−1.93E+02	−2.35E+02	−1.26E+02	−2.67E+02	0
	Std.	3.00E+00	6.76E+01	4.11E−18	5.69E+01	0.00E+00	5.06E+00	3.10E+00	2.52E+00	
18	Median	6.45E+00	1.64E+02	**5.98E+00**	9.31E+00	2.11E+01	8.32E+00	4.50E+01	7.47E+01	0
	Std.	5.89E−02	1.68E+01	4.80E−02	6.76E−03	5.99E−01	7.01E−01	5.12E+00	1.04E+01	
19	Median	**2.81E−16**	4.58E+01	2.05E−02	1.44E−02	9.67E+00	1.65E+01	8.11E+00	1.15E+01	0
	Std.	1.29E−16	1.65E+00	1.05E−04	1.33E−03	1.64E−01	1.36E−01	5.44E−01	8.07E−01	
20	Median	7.51E−01	4.21E+00	2.50E+00	1.52E+00	6.89E−05	**2.53E−09**	2.52E−02	1.67E−01	0
	Std.	9.55E−03	1.62E−01	2.53E−16	2.65E−04	8.19E−06	5.67E−10	2.29E−02	1.38E−01	
21	Median	**2.64E−07**	9.85E−01	5.41E−04	2.50E−04	1.36E−02	**1.60E−07**	4.15E−03	9.66E−01	0
	Std.	1.66E−07	5.36E−02	4.33E−06	1.04E−05	3.97E−03	1.15E−07	2.73E−03	4.69E−02	
22	Median	**−4.90E+01**	−2.75E+01	−1.91E+01	−4.89E+01	−1.11E+01	−2.33E+01	−1.51E+01	−3.19E+01	1 − n
	Std.	2.94E−16	4.14E+00	3.50E−15	1.19E−03	4.35E−01	1.93E+00	1.20E+00	5.68E−01	
23	Median	−1.31E+02	1.15E+05	−6.75E+02	−3.79E+02	1.18E+05	**−4.90E+01**	−2.39E+03	−4.70E+01	$\frac{n(n+4)(1-n)}{6}$
	Std.	5.65E+00	2.76E+04	5.17E+01	1.65E+02	2.64E+04	2.34E−06	9.35E+02	9.43E−02	
24	Median	**6.66E−04**	1.73E+01	2.11E−02	1.07E−02	2.83E+00	5.66E+00	4.31E+00	4.45E+00	0
	Std.	5.49E−04	8.65E−01	4.04E−04	1.32E−03	2.14E−01	8.60E−02	2.04E−01	3.59E−01	
25	Median	**6.00E−15**	3.38E+02	2.11E+00	1.32E+00	8.52E+01	1.42E+02	5.65E+01	5.02E+01	0
	Std.	3.79E−15	1.70E+01	0.00E+00	4.21E−02	2.78E+00	5.85E+00	5.23E+00	3.06E+00	
26	Median	**4.26E−15**	2.71E+02	1.36E+00	1.26E+00	6.32E+01	1.07E+02	5.10E+01	4.68E+01	0
	Std.	3.38E−15	1.17E+01	0.00E+00	2.22E−02	3.73E+00	4.71E+00	2.43E+00	8.54E−01	
27	Median	1.81E−09	3.17E−02	3.17E−02	4.20E−13	3.46E−08	1.96E−09	**5.78E−14**	2.85E−09	0
	Std.	2.27E−09	1.70E−08	0.00E+00	1.34E−13	1.28E−08	2.69E−09	6.00E−14	1.43E−09	
28	Median	**−5.66E+00**	−8.57E+01	−8.57E+00	−7.12E+00	−1.20E+01	−2.39E+01	−1.05E+01	−2.35E+01	0
	Std.	2.60E−01	7.97E−01	6.90E−02	6.79E−01	8.30E−16	6.97E−07	3.57E−15	4.90E−02	
29	Median	**5.03E+02**	4.57E+03	4.14E+02	1.07E+03	7.84E+02	1.75E+03	5.45E+02	6.15E+10	0
	Std.	1.63E+00	2.43E+02	1.83E+00	3.17E+01	2.02E+01	1.27E+02	6.27E+01	3.83E+10	

4.4.3. Statistical analysis

To further test the performance of JS-FOA, we apply a multiple -problem statistical analysis to compare its performance with those of the other four algorithms. Following reference [37], the Wilcoxon test, a nonparametric test that performs pairwise comparisons, is used at a 5% significance level. The results are listed in Table 4.

The conclusions drawn from Table 4 are stated below:

(a) JS-FOA's results are superior to those of the other four algorithms. The h-values of the Wilcoxon test confirm that the differences are significant.

Table A.4
Results of eight algorithms ($N = 30$, $maxgen = 5000$).

		JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO	$f(x^*)$
1	Median	**0.00E+00**	2.48E+05	6.19E−02	1.28E−02	1.57E−01	8.34E−77	3.53E−06	1.01E−22	0
	Std.	0.00E+00	5.86E+03	5.09E−03	1.67E−03	5.70E−03	1.18E−76	2.81E−06	3.73E−24	
2	Median	2.33E−01	4.27E+08	9.93E+01	1.70E−02	9.15E+01	**6.32E−56**	3.01E−01	1.51E−01	0
	Std.	7.68E−03	2.96E+07	2.78E+01	3.11E−03	6.08E+00	4.61E−57	1.91E−01	1.36E−05	
3	Median	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	−1
	Std.	2.93E−17	0.00E+00	6.28E−06	2.18E−05	3.01E−06	0.00E+00	1.90E−08	0.00E+00	
4	Median	2.22E−05	4.69E+07	7.94E+02	1.48E+01	2.47E+03	**3.55E−19**	8.00E+00	6.03E−18	0
	Std.	2.02E−05	3.57E+06	8.68E+01	1.58E−01	2.34E+02	4.00E−74	4.91E+00	5.49E−19	
5	Median	**3.94E−04**	9.51E+07	2.03E−01	**3.29E−04**	1.98E−01	3.14E−03	7.30E−02	6.62E−03	0
	Std.	2.56E−05	5.17E+06	3.84E−03	1.19E−05	2.37E−02	5.26E−04	6.16E−03	1.14E−03	
6	Median	**4.72E+01**	**4.27E+01**	6.21E+02	6.71E+01	2.52E+03	1.52E+03	8.88E+01	4.90E+00	1
	Std.	9.40E−02	2.67E+07	8.28E+01	6.73E−02	9.04E+01	7.34E+01	1.84E+01	2.71E+00	
7	Median	1.15E−05	5.48E+06	1.88E+04	2.28E−01	1.96E+05	**1.71E−27**	8.79E+02	7.86E+02	0
	Std.	1.74E−05	1.50E+05	8.42E+02	3.13E−03	4.59E+03	1.03E−27	1.90E+02	1.38E+02	
8	Median	**3.82E−40**	3.74E+01	1.96E+01	7.72E−03	4.91E+01	8.74E−18	6.68E+00	3.08E+00	0
	Std.	2.57E−40	3.54E−01	7.68E−01	3.05E−04	1.34E+00	6.24E−18	3.40E−01	2.45E−01	
9	Median	**1.82E−19**	4.91E+17	1.11E+00	1.58E−01	4.32E−01	9.34E−41	3.73E−01	2.91E−13	0
	Std.	6.64E−20	2.26E+17	9.20E−02	6.57E−04	1.86E−02	7.68E−41	2.10E−01	1.25E−14	
10	Median	**3.84E−38**	1.46E+04	2.00E+00	1.03E−03	2.88E+00	3.68E−37	3.44E−03	2.08E−21	0
	Std.	3.06E−38	3.16E+02	2.58E−01	1.27E−05	2.27E−01	2.04E−77	3.66E−03	6.42E−22	
11	Median	**0.00E+00**	1.48E+04	3.81E+00	**0.00E+00**	3.33E+00	**0.00E+00**	8.01E+01	**0.00E+00**	0
	Std.	0.00E+00	4.44E+02	6.62E−01	0.00E+00	6.10E−01	0.00E+00	2.43E+01	0.00E+00	
12	Median	**0.00E+00**	4.67E+47	3.97E−13	6.44E−07	5.34E−14	3.21E−231	6.35E−37	6.25E−73	0
	Std.	0.00E+00	2.51E+47	2.69E−13	9.66E−09	1.93E−14	0.00E+00	4.72E−37	8.28E−73	
13	Median	**0.00E+00**	2.51E+05	3.10E−01	1.29E−02	3.90E−01	4.52E−78	5.56E−05	2.89E−22	0
	Std.	0.00E+00	6.35E+03	3.16E−02	1.68E−04	2.60E−02	1.77E−78	5.50E−05	1.53E−22	
14	Median	**−4.50E+02**	**−4.50E+02**	**−4.49E+02**	**−4.50E+02**	−4.48E+02	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−450
	Std.	3.84E−15	3.84E+02	1.92E−01	2.26E−05	1.36E−01	0.00E+00	2.00E−05	0.00E+00	
15	Median	**−4.50E+02**	**−4.50E+02**	**2.13E+04**	**−4.50E+02**	1.74E+05	**−4.50E+02**	3.74E+02	3.24E+02	−450
	Std.	0.00E+00	7.46E+05	1.92E+03	1.03E−02	4.56E+03	0.00E+00	1.92E+02	8.76E+01	
16	Median	**4.32E−15**	2.29E+01	5.12E−01	2.54E−02	7.56E−01	**4.44E−15**	2.19E+00	2.03E−11	0
	Std.	5.27E−16	1.45E−02	5.80E−02	3.13E−04	6.71E−02	0.00E+00	1.58E−01	2.56E−13	
17	Median	**−2.37E+01**	−1.16E+03	−8.15E+01	−4.59E+02	−3.22E+02	−1.88E+02	−1.98E+02	−1.89E+02	0
	Std.	1.86E+01	1.78E+01	1.77E+00	3.36E+02	0.00E+00	2.40E+00	3.34E+00	1.60E+00	
18	Median	1.98E+01	3.02E+02	7.09E+01	5.39E+00	9.63E+01	**3.88E−15**	2.13E+02	3.75E−03	0
	Std.	2.13E−01	3.00E+00	2.31E+00	1.55E−03	1.01E+00	2.84E−15	1.65E+01	7.29E−05	
19	Median	**2.51E−06**	2.25E+01	2.52E+01	2.04E−03	2.46E+01	1.49E+00	2.63E+01	7.71E−01	0
	Std.	2.08E−06	2.44E−01	4.72E−02	2.25E−06	3.18E−01	7.68E−01	5.45E−01	4.64E−02	
20	Median	8.56E−01	3.80E+07	1.65E+00	1.69E+00	2.22E+00	**1.57E−32**	1.51E−01	8.18E−23	0
	Std.	2.41E−02	2.82E+06	2.15E−01	7.03E−04	2.01E−01	0.00E+00	7.54E−02	1.90E−23	
21	Median	**4.17E−09**	3.92E+00	8.03E−01	4.78E−05	1.05E+00	0.00E+00	3.43E−02	5.11E−06	0
	Std.	3.45E−09	6.88E−02	3.93E−02	1.39E−06	1.61E−02	0.00E+00	1.32E−02	5.97E−15	
22	Median	**−2.90E+01**	−3.89E+00	−9.38E+00	**−2.90E+01**	−1.29E+01	−2.85E+01	**−2.90E+01**	−2.68E+01	1 − n
	Std.	2.16E−15	3.36E−01	1.56E−01	1.44E−03	5.17E−01	7.41E−01	8.06E−01	1.24E−01	
23	Median	−1.92E+02	**−2.78E+01**	6.79E+04	−2.76E+02	7.55E+06	**−2.90E+01**	1.98E+04	**−2.90E+01**	$\frac{n(n+4)(1-n)}{6}$
	Std.	5.65E+00	3.09E−01	4.91E+03	1.25E+02	7.90E+05	0.00E+00	6.29E+03	7.82E−10	
24	Median	1.68E−03	4.90E−01	9.20E+00	**4.14E−04**	6.93E+00	9.44E−01	5.60E+00	9.44E−01	0
	Std.	6.50E−04	3.72E−02	6.42E−02	9.05E−06	1.89E−01	2.68E−01	1.00E−01	1.04E−01	
25	Median	**3.25E−15**	1.37E+04	4.72E+02	1.85E−01	2.76E+02	2.35E+02	9.36E+01	7.84E−11	0
	Std.	3.27E−15	4.06E+02	5.09E+00	7.51E−03	7.06E+00	7.35E+00	6.82E+00	3.13E−11	
26	Median	**4.27E−02**	2.20E+04	4.47E+02	1.91E−01	2.57E+02	4.64E+02	1.04E+02	3.69E−09	0
	Std.	4.03E−02	3.61E+02	3.74E+00	5.85E−03	8.18E+00	9.36E+00	3.41E+00	2.81E−09	
27	Median	1.92E−10	0.00E+00	4.06E−09	**2.84E−14**	1.66E−08	1.66E−08	4.82E−11	1.61E−09	0
	Std.	7.67E−11	0.00E+00	1.82E−09	4.40E−15	8.31E−09	8.31E−09	3.56E−11	2.02E−09	
28	Median	**−5.94E+00**	−7.35E+01	−1.48E+01	−7.93E+00	−1.78E+01	−1.44E+01	−2.20E+01	−1.44E+01	0
	Std.	2.32E−01	2.02E−01	2.18E−01	4.24E−01	4.93E−09	0.00E+00	2.13E−14	3.61E−11	
29	Median	**1.31E+02**	4.89E+16	2.98E+10	3.43E+02	6.87E+08	**1.72E+02**	2.32E+03	5.04E+02	0
	Std.	3.21E+00	4.40E+15	9.46E+09	2.85E+01	1.41E+08	1.86E+01	5.71E+01	6.02E+01	

(b) JS-FOA performs better when the dimensionality (N) increases. Therefore, JS-FOA can solve high-dimensional complex problems and has good behavior.

4.5. Effect of φ and M on the performance of JS-FOA

Parameter φ affects the convergence speed and probability of finding a global optimum. Different values of φ were evaluated.

Table A.5
Results of eight algorithms ($N = 50$, $maxgen = 5000$).

		JS-FOA	IFFO	IFOA	HFOA	DE	CoDE	PSO	CLPSO	$f(x^*)$
1	Median	**0.00E+00**	2.61E+01	9.65E−02	5.82E−02	8.51E−78	6.44E−56	2.23E−52	1.44E−12	0
	Std.	0.00E+00	0.00E+00	2.61E−18	1.56E−03	2.52E−78	1.30E−56	0.00E+00	5.43E−13	
2	Median	4.66E−02	1.56E−01	6.15E−03	5.16E−02	4.63E−01	**7.40E−44**	4.94E−02	2.50E+00	0
	Std.	3.17E−03	2.14E−02	2.46E−07	4.87E−03	9.76E−02	8.43E−44	4.34E−02	2.31E+00	
3	Median	**−1.00E+00**	**−1.00E+00**	−9.83E−01	−9.99E−01	**−1.00E+00**	−1.00E+00	**−1.00E+00**	**−1.00E+00**	−1
	Std.	4.37E−18	0.00E+00	2.15E−16	2.42E−04	5.66E−17	0.00E+00	3.53E−15	3.14E−16	
4	Median	**0.00E+00**	2.64E+02	5.88E+02	4.67E+01	2.86E−73	9.59E−52	1.86E−46	4.47E−08	0
	Std.	0.00E+00	2.90E−14	4.57E−14	7.75E−01	2.03E−73	3.25E−52	0.00E+00	4.17E−09	
5	Median	**8.45E−05**	2.18E+02	1.43E−03	7.33E−04	5.21E−03	2.86E−03	3.02E−03	1.23E−02	0
	Std.	5.48E−06	1.56E−06	1.01E−04	5.51E−05	4.45E−04	2.84E−04	5.85E−04	4.14E−03	
6	Median	3.44E+01	3.11E+01	2.61E+01	4.79E+01	3.35E+01	**1.00E−01**	3.93E+01	3.60E+01	1
	Std.	6.91E−02	3.85E+00	3.28E−02	2.73E−03	5.47E−01	6.49E−02	1.76E+01	3.92E+00	
7	Median	**5.86E−35**	9.66E−01	2.64E+00	1.83E+00	6.10E+03	9.08E−09	2.95E−16	1.17E+04	0
	Std.	4.41E−35	8.15E−17	1.65E−16	7.44E−02	6.28E+02	1.02E−08	2.27E−16	1.16E+03	
8	Median	**0.00E+00**	3.91E+00	1.70E−02	8.76E−03	7.59E−07	2.04E−12	1.23E−08	1.23E+01	0
	Std.	0.00E+00	4.35E−01	3.09E−04	5.27E−04	8.50E−08	1.65E−13	4.90E−09	4.29E−01	
9	Median	2.89E−19	3.00E+00	4.62E−01	3.26E−01	**1.84E−45**	3.08E−28	7.97E−04	1.72E−07	0
	Std.	1.84E−19	2.08E−16	5.59E−17	1.74E−02	2.87E−46	2.91E−28	4.34E−04	2.07E−08	
10	Median	**0.00E+00**	3.06E−03	6.58E−03	2.50E−03	1.76E−76	4.30E−54	5.21E−48	1.71E−11	0
	Std.	0.00E+00	0.00E+00	0.00E+00	2.99E−05	4.32E−77	1.17E−54	4.04E−48	6.53E−12	
11	Median	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.00E+00	0.00E+00	0
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.66E+00	0.00E+00	
12	Median	**0.00E+00**	3.98E+01	2.78E−04	6.36E−07	4.56E−231	3.90E−199	0.00E+00	7.39E−58	0
	Std.	0.00E+00	0.00E+00	2.74E−20	1.11E−08	0.00E+00	0.00E+00	0.00E+00	1.02E−57	
13	Median	**0.00E+00**	4.91E+00	9.46E−02	5.37E−02	2.72E−77	8.90E−55	4.74E−51	4.36E−12	0
	Std.	0.00E+00	3.45E−16	5.62E−18	7.76E−03	6.28E−78	1.03E−54	4.43E−51	6.26E−13	
14	Median	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−450
	Std.	5.00E−15	4.35E−14	3.12E−14	1.02E−04	2.25E−14	0.00E+00	1.86E−12	3.64E−13	
15	Median	**−4.50E+02**	−4.49E+02	**−4.50E+02**	−4.49E+02	5.65E+03	**−4.50E+02**	**−4.50E+02**	1.30E+04	−450
	Std.	0.00E+00	2.66E−14	4.58E−14	9.19E−02	6.38E+02	6.85E−09	2.54E−13	8.46E+02	
16	Median	**2.71E−15**	1.44E+00	7.60E−02	3.08E−02	8.54E−15	**4.44E−15**	2.06E+00	2.07E−06	0
	Std.	5.36E−16	8.60E−01	0.00E+00	2.78E−03	0.00E+00	0.00E+00	3.51E−01	2.79E−07	
17	Median	−8.24E+00	−7.30E+02	4.86E−02	−1.40E+02	−1.93E+02	−3.06E+02	−1.25E+02	**−3.04E+02**	0
	Std.	7.80E+00	2.06E+01	3.97E−18	1.59E+02	0.00E+00	2.40E+00	6.64E+00	1.60E+00	
18	Median	6.73E+00	2.09E+02	6.52E+00	9.26E+00	**2.32E−18**	1.42E−09	4.20E+01	1.02E+00	0
	Std.	5.99E−02	3.94E+00	4.94E−02	2.08E−02	3.46E−19	4.66E−11	5.28E+00	1.78E−01	
19	Median	**6.51E−17**	1.01E+01	1.47E−02	3.82E−03	4.52E+00	7.40E+00	8.87E+00	3.27E+00	0
	Std.	5.87E−17	2.18E−01	0.00E+00	4.08E−04	9.63E−02	3.91E+00	3.24E−01	4.71E−02	
20	Median	7.85E−01	9.02E+00	1.91E+00	1.50E+00	2.44E−32	**9.42E−33**	4.93E−02	6.03E−13	0
	Std.	3.13E−02	5.11E−01	1.66E−16	1.03E−03	0.00E+00	0.00E+00	4.41E−02	1.23E−13	
21	Median	**0.00E+00**	6.37E−03	4.09E−04	7.86E−05	0.00E+00	0.00E+00	2.39E−02	8.33E−10	0
	Std.	0.00E+00	2.02E−03	0.00E+00	1.19E−05	0.00E+00	0.00E+00	1.91E−02	2.10E−10	
22	Median	**−4.90E+01**	−3.49E+00	−2.32E+01	**−4.90E+01**	−1.30E+01	−3.98E+01	−1.73E+01	−4.02E+01	1 − n
	Std.	3.76E−16	4.33E−01	2.92E−15	7.36E−03	1.97E−01	2.96E−02	9.26E−01	1.94E+00	
23	Median	−1.23E+02	−2.90E+01	**−4.29E+03**	−6.00E+02	1.82E+03	−4.90E+01	**−4.19E+03**	−4.90E+01	$\frac{n(n+4)(1-n)}{6}$
	Std.	4.19E+00	0.00E+00	4.22E+00	3.25E+02	1.59E+03	0.00E+00	2.29E+02	1.82E−05	
24	Median	**2.73E−04**	2.56E−03	1.77E−02	1.69E−03	5.47E−01	3.88E+00	3.22E+00	2.94E+00	0
	Std.	2.25E−04	3.32E−04	5.35E−05	3.67E−04	7.11E−02	7.47E−02	3.31E−01	7.87E−02	
25	Median	**0.00E+00**	3.58E+01	2.18E+00	4.02E−01	0.00E+00	2.13E−14	6.06E+01	1.13E−03	0
	Std.	0.00E+00	2.34E−13	0.00E+00	1.15E−02	0.00E+00	3.01E−14	5.38E+00	8.51E−04	
26	Median	**2.49E−15**	3.90E+01	2.03E+00	4.76E−01	2.09E+01	1.96E+01	5.37E+01	1.26E−02	0
	Std.	1.75E−15	0.00E+00	0.00E+00	3.99E−02	7.43E−01	2.07E+00	5.09E+00	1.46E−03	
27	Median	6.74E−12	**0.00E+00**	3.09E−02	3.22E−15	5.38E−11	0.00E+00	1.00E−13	5.42E−10	0
	Std.	6.44E−12	0.00E+00	0.00E+00	7.85E−16	3.11E−11	0.00E+00	7.12E−14	7.49E−10	
28	Median	**−5.28E+00**	−3.71E+01	−9.21E+00	−7.95E+00	−1.17E+01	−2.39E+01	−7.20E+00	−2.39E+01	0
	Std.	1.65E−01	9.20E−02	7.47E−16	1.83E−01	8.42E−16	0.00E+00	2.71E−15	1.16E−06	
29	Median	4.19E+02	**1.82E+02**	4.40E+02	9.99E+02	9.01E+02	4.73E+02	5.79E+02	2.31E+04	0
	Std.	2.65E+00	6.73E+00	2.33E+00	1.69E+01	3.58E+01	3.26E−08	6.99E+01	7.31E+03	

Testing every *maxgen* value is not necessary because the algorithm is efficient and has fast convergence speed. Hence, setting φ to 30, 40, 50, 60, and 70 when $maxgen = 500$ provides a good overview. The dimensionality of each benchmark function is set to 30, and each complex problem is solved independently 30 times. The other parameters of JS-FOA are the same as those in Section 4.3. The results are listed in Tables A.6 and A.7 in the Appendix.

Table A.6
Results of comparing with itself ($N = 30$).

Function		φ				
		30	40	50	60	70
1	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	Median	4.07E−02	4.10E−02	4.69E−02	4.41E−02	4.58E−02
	Std.	3.62E−03	3.24E−03	2.45E−03	2.07E−03	1.78E−03
3	Median	−1.00E+00	−1.00E+00	−1.00E+00	−1.00E+00	−1.00E+00
	Std.	5.23E−18	0.00E+00	9.81E−17	0.00E+00	4.65E−16
4	Median	1.86E+01	1.31E+02	2.92E+02	2.97E+02	4.32E+02
	Std.	1.08E+01	2.49E+01	2.47E+01	2.46E+01	1.60E+01
5	Median	3.93E−04	4.34E−04	4.48E−04	4.10E−04	4.55E−04
	Std.	2.87E−05	2.03E−05	2.75E−05	3.34E−05	3.17E−05
6	Median	2.74E+01	2.73E+01	2.73E+01	2.72E+01	2.70E+01
	Std.	6.82E−02	6.47E−02	7.39E−02	6.50E−02	6.49E−02
7	Median	1.81E−02	1.17E−01	3.94E−01	6.81E−01	1.48E+00
	Std.	7.36E−03	3.07E−02	7.22E−02	1.17E−01	1.03E−01
8	Median	5.48E−07	1.65E−04	1.29E−03	7.31E−03	1.17E−02
	Std.	3.74E−07	1.36E−04	3.87E−04	6.78E−04	3.06E−04
9	Median	2.78E−20	2.61E−19	2.29E−20	8.09E−20	1.46E−19
	Std.	1.02E−20	2.00E−19	1.02E−20	3.66E−20	1.09E−19
10	Median	1.10E−04	5.57E−04	1.44E−03	3.30E−03	5.04E−03
	Std.	4.43E−05	1.10E−04	1.91E−04	3.46E−04	2.83E−04
11	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	Median	2.70E−148	1.38E−147	2.09E−122	1.66E−125	5.63E−109
	Std.	2.66E−148	1.36E−147	2.05E−122	1.63E−125	5.54E−109
13	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	1.03E−04	1.51E−04	1.86E−04	3.34E−04	3.12E−04
15	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	0.00E+00	0.00E+00	5.66E−10	1.89E−15	0.00E+00
16	Median	2.90E−15	2.05E−14	2.31E−15	4.91E−15	3.02E−15
	Std.	3.99E−16	1.70E−14	3.18E−16	8.82E−16	3.18E−16
17	Median	−5.28E+00	−9.06E+00	−1.49E+00	−1.51E+00	−3.62E+00
	Std.	2.31E+00	2.73E+00	1.40E+00	8.02E−01	2.05E+00
18	Median	6.10E+00	5.97E+00	6.03E+00	5.96E+00	5.97E+00
	Std.	7.80E−02	5.84E−02	5.75E−02	4.62E−02	4.48E−02
19	Median	4.23E−04	1.24E−03	3.11E−03	6.10E−03	1.09E−02
	Std.	1.92E−04	3.08E−04	4.08E−04	6.02E−04	5.18E−04
20	Median	6.68E−01	6.81E−01	6.86E−01	6.38E−01	6.23E−01
	Std.	8.56E−03	2.35E−02	2.66E−02	1.46E−02	6.79E−03
21	Median	6.06E−05	1.42E−04	3.29E−04	4.36E−04	4.79E−04
	Std.	1.37E−05	2.19E−05	2.37E−05	1.68E−05	5.94E−06
22	Median	−2.90E+01	−2.90E+01	−2.90E+01	−2.90E+01	−2.90E+01
	Std.	2.05E−16	2.05E−16	2.37E−16	3.35E−16	2.44E−15
23	Median	−1.47E+02	−1.52E+02	−1.58E+02	−1.58E+02	−1.68E+02
	Std.	6.02E+00	5.72E+00	6.12E+00	8.33E+00	5.67E+00
24	Median	1.62E−03	2.65E−03	5.99E−03	1.15E−02	1.24E−02
	Std.	6.98E−04	8.51E−04	1.56E−03	1.79E−03	1.03E−03
25	Median	4.44E−14	5.33E−15	3.55E−15	5.33E−15	7.40E−15
	Std.	3.84E−14	2.92E−15	2.43E−15	3.85E−15	3.30E−15
26	Median	1.78E−15	3.55E−15	3.55E−15	7.11E−15	1.78E−15
	Std.	1.75E−15	2.43E−15	2.43E−15	4.15E−15	1.75E−15
27	Median	6.55E−11	0.00E+00	8.77E−12	5.65E−14	1.57E−10
	Std.	6.42E−11	0.00E+00	8.62E−12	5.56E−14	1.54E−10
28	Median	−6.54E+00	−7.80E+00	−9.20E+00	−8.83E+00	−1.02E+01
	Std.	2.05E−01	1.76E−01	3.18E−01	3.33E−01	3.73E−01
29	Median	4.09E+02	4.08E+02	4.08E+02	4.10E+02	4.10E+02
	Std.	1.69E+00	1.76E+00	1.97E+00	1.60E+00	1.59E+00

Table A.7
Results of comparing with itself ($N = 50$).

Function		φ				
		30	40	50	60	70
1	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	Median	1.30E−01	1.38E−01	1.42E−01	1.41E−01	1.37E−01
	Std.	9.38E−03	6.47E−03	5.30E−03	5.06E−03	6.87E−03
3	Median	−1.00E+00	−1.00E+00	−1.00E+00	−1.00E+00	−1.00E+00
	Std.	5.23E−18	7.40E−18	9.06E−18	1.85E−17	2.96E−06
4	Median	5.59E+01	2.14E+02	3.78E+02	5.42E+02	6.75E+02
	Std.	2.42E+01	2.82E+01	3.68E+01	3.70E+01	1.83E+01
5	Median	6.59E−04	6.88E−04	7.59E−04	7.53E−04	8.76E−04
	Std.	5.28E−05	3.37E−05	3.53E−05	5.02E−05	6.53E−05
6	Median	4.78E+01	4.76E+01	4.73E+01	4.73E+01	4.72E+01
	Std.	7.75E−02	8.83E−02	6.94E−02	6.83E−02	7.82E−02
7	Median	1.25E−01	4.45E−01	1.14E+00	3.43E+00	8.11E+00
	Std.	4.29E−02	1.65E−01	2.61E−01	5.30E−01	4.39E−01
8	Median	4.42E−09	1.86E−04	2.32E−03	8.76E−03	1.31E−02
	Std.	2.33E−09	1.04E−04	6.69E−04	8.82E−04	3.62E−04
9	Median	1.84E−19	9.29E−20	6.03E−20	2.21E−19	1.85E−19
	Std.	1.36E−19	2.97E−20	3.24E−20	9.44E−20	9.91E−20
10	Median	4.67E−04	1.14E−03	2.61E−03	5.86E−03	1.07E−02
	Std.	1.54E−04	2.33E−04	4.70E−04	5.97E−04	4.53E−04
11	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	Median	1.08E−131	3.72E−146	1.27E−49	3.79E−47	1.26E−36
	Std.	1.07E−131	3.66E−146	1.25E−49	3.73E−47	1.23E−36
13	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	6.65E−05	2.55E−04	4.61E−04	6.79E−04	4.03E−04
15	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	0.00E+00	0.00E+00	1.89E−15	0.00E+00	0.00E+00
16	Median	2.78E−15	4.09E−15	3.49E−15	8.70E−15	3.61E−15
	Std.	3.64E−16	4.84E−16	9.46E−16	5.65E−15	3.62E−16
17	Median	−3.90E+01	−6.30E+00	−6.86E+00	−3.01E+00	−1.27E+01
	Std.	2.23E+01	3.15E+00	2.42E+00	2.04E+00	9.58E+00
18	Median	1.07E+01	1.05E+01	1.04E+01	1.03E+01	1.02E+01
	Std.	9.21E−02	7.67E−02	9.08E−02	8.91E−02	6.91E−02
19	Median	6.26E−04	2.15E−03	5.08E−03	1.17E−02	2.08E−02
	Std.	2.74E−04	4.47E−04	7.82E−04	1.21E−03	1.04E−03
20	Median	8.25E−01	7.85E−01	7.91E−01	7.53E−01	7.54E−01
	Std.	1.97E−02	1.40E−02	1.63E−02	6.57E−03	5.40E−03
21	Median	7.67E−05	1.75E−04	4.25E−04	5.69E−04	6.08E−04
	Std.	1.49E−05	2.61E−05	3.26E−05	1.27E−05	1.04E−05
22	Median	−4.90E+01	−4.90E+01	−4.90E+01	−4.90E+01	−4.90E+01
	Std.	7.11E−16	1.14E−15	7.11E−16	7.11E−16	8.20E−16
23	Median	−2.06E+02	−2.13E+02	−2.13E+02	−2.39E+02	−2.33E+02
	Std.	6.15E+00	6.19E+00	9.08E+00	7.60E+00	6.32E+00
24	Median	3.74E−03	7.19E−03	1.69E−02	2.54E−02	3.73E−02
	Std.	1.23E−03	2.35E−03	5.48E−03	3.50E−03	3.00E−03
25	Median	8.82E−15	1.63E−13	1.17E−14	4.83E−13	5.92E−15
	Std.	4.83E−15	1.54E−13	6.83E−15	4.65E−13	4.04E−15
26	Median	5.80E−15	1.18E−14	3.34E−02	2.10E−13	3.62E−02
	Std.	3.96E−15	5.51E−15	3.29E−02	1.80E−13	3.56E−02
27	Median	5.98E−10	5.94E−12	4.07E−10	1.49E−10	1.34E−10
	Std.	5.67E−10	5.76E−12	4.01E−10	1.47E−10	9.11E−11
28	Median	−7.30E+00	−8.24E+00	−1.05E+01	−1.25E+01	−1.27E+01
	Std.	1.82E−01	2.10E−01	3.25E−01	4.13E−01	5.52E−01
29	Median	1.14E+03	1.14E+03	1.14E+03	1.14E+03	1.14E+03
	Std.	2.67E+00	3.06E+00	2.82E+00	2.83E+00	2.49E+00

Tables A.6 and A.7 summarizes that the value of φ had a small impact on the results. In other words, φ only slightly affects the balancing of the global search (collaborative group search) and local search (memory move direction). Therefore, changing φ in JS-FOA can be used to solve different types of complex problems according to different needs in the future.

Table A.8
Results of comparing with itself ($N = 30$).

Function		M			
		N	N/2	N/3	N/4
1	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	Median	3.85E−02	4.21E−02	4.15E−02	4.32E−02
	Std.	3.48E−03	2.72E−03	2.43E−03	2.24E−03
3	Median	−1.00E+00	−1.00E+00	−1.00E+00	−1.00E+00
	Std.	7.40E−18	9.06E−18	1.43E−17	7.20E−15
4	Median	2.86E+02	3.52E+02	4.33E+02	4.28E+02
	Std.	2.28E+01	2.04E+01	1.28E+01	1.54E+01
5	Median	4.94E−04	5.89E−04	6.59E−04	6.66E−04
	Std.	3.35E−05	3.62E−05	3.32E−05	4.71E−05
6	Median	2.72E+01	2.69E+01	2.70E+01	2.71E+01
	Std.	7.05E−02	8.11E−02	7.55E−02	7.80E−02
7	Median	2.35E−01	8.13E−01	1.24E+00	1.56E+00
	Std.	4.05E−02	1.05E−01	9.87E−02	7.48E−02
8	Median	1.74E−03	6.73E−03	1.11E−02	1.18E−02
	Std.	4.64E−04	7.57E−04	4.20E−04	4.97E−04
9	Median	5.01E−20	3.27E−19	1.01E−19	2.21E−19
	Std.	2.40E−20	2.30E−19	6.13E−20	2.14E−19
10	Median	1.65E−03	3.27E−03	4.89E−03	5.44E−03
	Std.	2.69E−04	2.82E−04	2.40E−04	1.67E−04
11	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	Median	3.80E−149	1.68E−62	1.34E−31	5.17E−13
	Std.	3.74E−149	1.65E−62	1.32E−31	5.05E−13
13	Median	0.00E+00	0.00E+00	2.30E−45	0.00E+00
	Std.	0.00E+00	0.00E+00	2.26E−45	0.00E+00
14	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	1.89E−04	2.62E−04	2.24E−04	1.64E−04
15	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	0.00E+00	0.00E+00	1.51E−13	1.75E−11
16	Median	1.66E−14	3.49E−15	5.03E−12	4.19E−10
	Std.	1.33E−14	2.87E−16	4.94E−12	4.12E−10
17	Median	−3.67E+00	−7.36E−02	−1.03E+00	−2.50E−01
	Std.	1.67E+00	7.10E−05	7.84E−01	1.74E−01
18	Median	5.88E+00	5.88E+00	5.94E+00	5.85E+00
	Std.	4.14E−02	3.64E−02	5.13E−02	5.05E−02
19	Median	3.64E−03	7.64E−03	9.05E−03	1.11E−02
	Std.	4.77E−04	5.67E−04	5.06E−04	2.57E−04
20	Median	6.68E−01	5.96E−01	5.71E−01	5.75E−01
	Std.	1.57E−02	9.00E−03	9.29E−03	6.37E−03
21	Median	3.15E−04	4.43E−04	4.41E−04	4.51E−04
	Std.	2.53E−05	6.49E−06	5.92E−06	4.26E−06
22	Median	−2.90E+01	−2.90E+01	−2.90E+01	−2.90E+01
	Std.	3.25E−15	1.05E−13	2.42E−13	6.96E−15
23	Median	−1.72E+02	−2.30E+02	−2.10E+02	−2.44E+02
	Std.	7.42E+00	1.49E+01	1.06E+01	1.24E+01
24	Median	5.94E−03	1.01E−02	1.40E−02	1.20E−02
	Std.	1.89E−03	1.14E−03	2.57E−03	9.93E−04
25	Median	0.00E+00	7.11E−15	3.55E−15	1.31E−11
	Std.	0.00E+00	4.15E−15	2.43E−15	1.29E−11
26	Median	4.90E−13	7.11E−15	1.58E−11	1.95E−14
	Std.	4.78E−13	4.15E−15	1.56E−11	1.41E−14
27	Median	9.03E−12	3.96E−11	9.68E−13	4.61E−13
	Std.	6.67E−12	3.88E−11	9.52E−13	4.53E−13
28	Median	−8.51E+00	−9.07E+00	−7.88E+00	−8.49E+00
	Std.	3.31E−01	3.13E−01	3.34E−01	3.04E−01
29	Median	4.11E+02	4.11E+02	4.10E+02	4.08E+02
	Std.	1.36E+00	1.36E+00	1.59E+00	1.71E+00

Table A.9
Results of comparing with itself ($N = 50$).

Function		M			
		N	N/2	N/3	N/4
1	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	Median	1.34E−01	1.30E−01	1.37E−01	1.30E−01
	Std.	8.37E−03	6.52E−03	2.02E−03	3.17E−03
3	Median	−1.00E+00	−1.00E+00	−1.00E+00	−1.00E+00
	Std.	5.23E−18	1.23E−16	2.72E−17	1.32E−11
4	Median	3.76E+02	5.48E+02	6.40E+02	7.09E+02
	Std.	3.82E+01	3.63E+01	2.13E+01	1.10E+01
5	Median	7.23E−04	9.90E−04	1.23E−03	1.25E−03
	Std.	6.68E−05	6.19E−05	7.86E−05	7.03E−05
6	Median	4.72E+01	4.70E+01	4.70E+01	4.71E+01
	Std.	7.18E−02	8.01E−02	1.04E−01	9.76E−02
7	Median	1.04E+00	3.74E+00	5.66E+00	7.46E+00
	Std.	2.04E−01	5.67E−01	4.97E−01	3.49E−01
8	Median	7.25E−04	7.34E−03	1.15E−02	1.27E−02
	Std.	2.10E−04	7.66E−04	5.59E−04	3.04E−04
9	Median	6.28E−20	1.69E−19	9.45E−22	3.80E−20
	Std.	5.22E−20	1.08E−19	9.29E−22	3.74E−20
10	Median	3.14E−03	6.79E−03	9.83E−03	1.02E−02
	Std.	4.64E−04	5.96E−04	3.66E−04	3.27E−04
11	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	Median	6.02E−146	6.82E−74	1.06E−30	4.75E−18
	Std.	5.91E−146	6.63E−74	1.05E−30	4.67E−18
13	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
14	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	3.66E−04	6.29E−04	4.29E−04	3.40E−04
15	Median	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
	Std.	1.89E−15	0.00E+00	4.31E−11	5.68E−15
16	Median	3.25E−14	6.45E−15	1.24E−12	6.27E−13
	Std.	2.02E−14	2.24E−15	1.22E−12	4.92E−13
17	Median	−5.71E+00	−6.76E+00	−3.95E+00	−8.81E−01
	Std.	2.62E+00	3.17E+00	1.76E+00	4.36E−01
18	Median	1.04E+01	1.01E+01	1.01E+01	1.00E+01
	Std.	8.44E−02	5.60E−02	7.08E−02	8.00E−02
19	Median	7.31E−03	1.28E−02	1.89E−02	2.06E−02
	Std.	9.98E−04	1.27E−03	9.55E−04	6.15E−04
20	Median	7.75E−01	7.48E−01	7.38E−01	7.30E−01
	Std.	1.20E−02	1.52E−02	1.50E−02	9.76E−03
21	Median	4.71E−04	5.33E−04	5.49E−04	5.51E−04
	Std.	2.44E−05	6.44E−06	5.35E−06	6.96E−06
22	Median	−4.90E+01	−4.90E+01	−4.90E+01	−4.90E+01
	Std.	4.10E−16	4.90E−13	9.17E−16	8.20E−16
23	Median	−2.19E+02	−2.70E+02	−3.01E+02	−2.89E+02
	Std.	5.99E+00	9.03E+00	1.19E+01	1.17E+01
24	Median	1.37E−02	3.34E−02	3.08E−02	3.22E−02
	Std.	6.18E−03	4.62E−03	3.48E−03	2.24E−03
25	Median	5.03E−13	6.22E−14	5.67E−11	7.11E−14
	Std.	4.86E−13	3.84E−14	5.58E−11	5.32E−14
26	Median	2.96E−15	3.40E−02	5.92E−15	3.75E−12
	Std.	2.91E−15	3.34E−02	4.04E−15	3.60E−12
27	Median	3.66E−10	4.49E−13	1.48E−07	2.97E−12
	Std.	2.38E−10	3.18E−13	1.45E−07	2.30E−12
28	Median	−1.02E+01	−1.21E+01	−1.10E+01	−1.17E+01
	Std.	3.67E−01	3.80E−01	3.43E−01	3.53E−01
29	Median	1.14E+03	1.14E+03	1.14E+03	1.15E+03
	Std.	2.29E+00	2.34E+00	2.67E+00	2.26E+00

The maximum number of fruit flies with memory (M) can affect convergence speed but will also influence the local search. Similar to the evaluation of φ, $maxgen = 500$, $N = 30$, and M is set to N, $N/2$, $N/3$, and $N/4$. The other parameters remain unchanged. The results are listed in Tables A.8 and A.9 in the Appendix.

Tables A.8 and A.9 summarizes that the results for different values of M vary only slightly. However, as the value of M increases, the result approaches a global optimum. This trend shows that larger the value of M, the stronger the local search ability. Nevertheless, the value of M cannot be increased to N to avoid the premature termination of the algorithm. In future studies, the value of M should approach $N/2$ to ensure both convergence and variability.

5. Conclusions and future research

In this study, an enhanced FOA named JS-FOA is proposed to solve high dimensional continuous optimization problems. The JS-FOA concludes three search strategies and one combined strategy. Twenty-nine benchmark functions are utilized to compare JS-FOA's performance with other meta-heuristic algorithms. The results show that JS-FOA has an advantage in handling high-dimensional continuous problems; it can solve the problems with fewer iterations and faster convergence rate than the other algorithms. JS-FOA also outperforms the other algorithms in terms of the median and standard deviation. Therefore, the proposed JS-FOA is highly robust. When *maxgen* is 500 and the dimensionality is 30 or 50, most results confirm the superiority of JS-FOA. Even when *maxgen* is increased to 1000 and 5000, JS-FOA solutions are still better than those of the other algorithms. The twenty-nine benchmark functions are also used to compare JS-FOA's performance with itself for different parameter values. The results show that the parameter values only slightly affect specific problems, while for other cases parameter values can improve calculation speed or accuracy.

Other intelligent algorithms, such as harmony search [38], salp swarm [39], population algorithm [40], differential evolution [41] and locust swarm algorithms, perform well in solving different optimization problems [42,43]. We intend to design more hybrid FOAs by taking advantage of the aforementioned algorithms in handling optimization problems. In future, JS-FOA can be used to solve actual problems such as the optimizations of new deep learning approaches [44,45] and more realistic and complex operations management problems [46–48].

Acknowledgment

This research is partially supported by National Natural Science Foundation of China (Nos: 71771095; 71531009).

Appendix

See Tables A.1–A.9.

References

[1] S.P. Hong, Y.H. Kim, A genetic algorithm for joint replenishment based on the exact inventory cost, Comput. Oper. Res. 36 (1) (2009) 167–175.
[2] S. Gajjar, M. Sarkar, K. Dasgupta, FAMACROW: Fuzzy And ant colony optimization based combined mac, routing, and unequal clustering cross-layer protocol for wireless sensor networks, Appl. Soft Comput. 43 (2016) 235–247.
[3] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (11) (2016) 292–305.
[4] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[5] W.T. Pan, A new evolutionary computation approach: fruit fly optimization algorithm, in: 2011 Conference of Digital Technology and Innovation Management, Taipei, pp.382–391.
[6] W.T. Pan, A new fruit fly optimization algorithm: taking the financial distress model, Knowl. Based Syst. 26 (2012) 69–74.
[7] H. Li, S. Guo, H. Zhao, C. Su, B. Wang, Annual electric load forecasting by a least squares support vector machine with a fruit fly optimization algorithm, Energies 5 (2012) 4430–4445.
[8] X. Zheng, L. Wang, S. Wang, A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem, Knowl. Based Syst. 57 (2014) 95–103.
[9] L. Yin, X. Li, L. Gao, C. Lu, A new improved fruit fly optimization algorithm for traveling salesman problem, in: 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, 2016, pp. 21–28.
[10] S.M. Lin, Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network, Neur. Comput. Appl. 7 (2013) 459–465.
[11] L. Wu, Q. Liu, X. Tian, J.X. Zhang, W.S. Xiao, A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems, Knowl. Based Syst. 144 (2018) 153–173.
[12] G.M. Wang, L.M. Ma, J.W. Chen, A bilevel improved fruit fly optimization algorithm for the nonlinear bilevel programming problem, Knowl. Based Syst. 138 (2017) 113–123.
[13] L. Wang, H.L. Hu, X.Y. Ai, H. Liu, Effective electricity energy consumption forecasting using echo state network improved by differential evolution algorithm, Energy 153 (2018) 801–815.
[14] Y.R. Zeng, Y. Zeng, B.J. Choi, L. Wang, Multifactor-influenced energy consumption forecasting using enhanced back-propagation neural network, Energy 127 (2017) 381–396.
[15] H. Li, S. Guo, C. Li, J. Sun, A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm, Knowl. Based Syst. 37 (2013) 378–387.
[16] L. Wang, S.X. Lv, Y.R. Zeng, Effective sparse adaboost method with esn and foa for industrial electricity consumption forecasting in china, Energy 155 (2018) 1013–1031.
[17] L. Wang, Z.G. Wang, H. Qu, S. Liu, Optimal forecast combination based on neural networks for time series forecasting, Appl. Soft Comput. 66 (2018) 1–17.
[18] L. Yin, X. Li, L. Gao, et al., A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem, Sustainable Comput. Inf. Syst. 13 (3) (2017) 15–30.
[19] Q.K. Pan, H.Y. Sang, J.H. Duan, L. Gao, An improved fruit fly optimization algorithm for continuous function optimization problems, Konwl. Based Syst. 62 (2014) 69–83.
[20] X. Yuan, X. Dai, J. Zhao, Q. He, On a novel multi-swarm fruit fly optimization algorithm and its application, Appl. Math. Comput. 233 (2014) 260–271.
[21] L. Wang, Y.L. Shi, S. Liu, An improved fruit fly optimization algorithm and its application to joint replenishment problems, Expert Syst. Appl. 42 (2015) 4310–4323.
[22] D. Shan, G. Cao, H. Dong, LGMS-FOA: an improved fruit fly optimization algorithm for solving optimization, Math. Probl. Eng. (2013) 108768, http://dx.doi.org/10.1155/2013/108768, 1–9.
[23] T.F. Li, L. Gao, P.G. Li, Q.K. Pan, An ensemble fruit fly optimization algorithm for solving range image registration to improve quality inspection of free-form surface parts, Inform. Sci. 367 (2016) 953–974.
[24] D.X. Niu, T.N. Ma, B.Y. Liu, Power load forecasting by wavelet least squares support vector machine with improved fruit fly optimization algorithm, J. Comb. Optim. 33 (3) (2017) 1122–1143.
[25] T. Meng, Q.K. Pan, An improved fruit fly optimization algorithm for solving the multidimensional knapsack problem, Appl. Soft Comput. 50 (2017) 79–93.
[26] L.H. Wu, C.L. Zuo, H.Q. Zhang, Z.H. Liu, Bimodal fruit fly optimization algorithm based on cloud model learning, Soft Comput. 21 (7) (2017) 1877–1893.
[27] Y. Zhang, G. Cui, J. Wu, W.T. Pan, Q. He, A novel multi-scale cooperative muta- tion fruit fly optimization algorithm, Knowl. Based Syst. 114 (2016) 24–35.
[28] P.N. Suganthan, N. Hansen, J.J. Liang, et al., Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, 2005 and KanGAL Report #2005005, IIT Kanpur, India.
[29] S.X. Lv, Y.R. Zeng, L. Wang, An effective fruit fly optimization algorithm with hybrid information exchange and its applications, Int. J. Mach. Learn. Cyb. 9 (10) (2018) 1623–1648.
[30] Y. Wang, Z.X. Cai, Q.F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, Trans. Evol. Comput. 15 (1) (2011) 55–66.
[31] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[32] L. Shen, H.L. Chen, Z. Yu, W.C. Kang, B.Y. Zhang, H.Z. Li, B. Yang, D.Y. Liu, Evolving support vector machines using fruit fly optimization for medical data classification, Knowl. Based Syst. 96 (2016) 61–75.
[33] L. Peng, S. Liu, R. Liu, L. Wang, Effective long short-term memory with differential evolution algorithm for electricity price prediction, Energy 162 (2018) 1301–1314.

[34] J.Q. Li, Q.K. Pan, K. Mao, A hybrid fruit fly optimization algorithm for the realistic hybrid flowshop rescheduling problem in steel making systems, IEEE Trans. Autom. Sci. Eng. 13 (2) (2016) 932–949.
[35] L. Wang, H.L. Hu, R. Liu, X.J, An improved differential harmony search algorithm for function optimization problems, Soft Comput. (2018) http://dx.doi.org/10.1007/s00500-018-3139-4.
[36] R.M. Rizk-Allah, R.A. EI-Sehiemy, S. Deb, G.G. Wang, A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor, J. Supercomput. 73 (3) (2017) 1235–1256.
[37] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric test for analyzing the evolutionary algorithms' behaviour: a case study on the CEC' 2005 special session on real parameter optimization, J. Heuristics 15 (6) (2005) 617–644.
[38] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.
[39] H. Faris, M.M. Mafarja, A.A. Heidari, I. Aljarah, A. Al-Zoubi, S. Mirjalili, H. Fujita, An efficient binary salp swarm algorithm with crossover scheme for feature selection problems, Knowl. Based Syst. 154 (2018) 43–67.
[40] Y.L. Lu, U. Benlic, Q.H. Wu, A population algorithm based on randomized tabu thresholding for the multi-commodity pickup-and-delivery traveling salesman problem, Comput. Oper. Res. 101 (2019) 285–297.
[41] Y.R. Zeng, L. Peng, J.L. Zhang, L. L. Wang, An effective hybrid differential evolution algorithm incorporating simulated annealing for joint replenishment and delivery problem with trade credit, Int. J. Comput. Intell. Syst. 9 (6) (2016) 1001–1015.
[42] H. Zhou, H. Wang, W. Zeng, Smart construction site in mega construction projects: A case study on island tunneling project of Hong Kong-Zhuhai-Macao bridge, Front. Eng. Manage. 5 (1) (2018) 78–87.
[43] R. Liu, Y.-R. Zeng, H. Qu, L. Wang, Optimizing the new coordinated replenishment and delivery model considering quantity discount and resource constraints, Comput. Ind. Eng. 116 (2018) 82–96.
[44] U.R. Acharya, H. Fujita, O.S. Lih, Y. Hagiwara, J.H. Tan, M. Adam, Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network, Inform. Sci. 405 (2017) 81–90.
[45] U. Raghavendra, H. Fujita, S.V. Bhandary, A. Gudigar, J.H. Tan, Acharya, U.R. Acharya, Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images, Inform. Sci. 441 (2018) 41–49.
[46] W. Zeng, J. Zhang, H. Wang, H. Zhou, Supplier development and its incentives in infrastructure mega-projects: a case study on Hong Kong-zhuhai-macao bridge project, Front. Eng. Manage 5 (1) (2018) 88–97.
[47] L. Wang, R. Liu, S. Liu, An effective and efficient fruit fly optimization algorithm with level probability policy and its applications, Knowl. Based Syst. 97 (2016) 158–174.
[48] Q. Zhou, U. Benlic, Q.H. Wu, J.K. Hao, Heuristic search to the capacitated clustering problem, Eur. J. Oper. Res. 273 (2) (2019) 464–487.