Knowledge-Based Systems 62 (2014) 69–83

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

An improved fruit fly optimization algorithm for continuous function optimization problems

Quan-Ke Pan [a,b], Hong-Yan Sang [b], Jun-Hua Duan [b], Liang Gao [c,*]

[a] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, PR China
[b] College of Computer Science, Liaocheng University, Liaocheng 252059, PR China
[c] State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science & Technology, Wuhan 430074, PR China

ARTICLE INFO

Article history:
Received 4 September 2013
Received in revised form 6 February 2014
Accepted 26 February 2014
Available online 12 March 2014

Keywords:
Fruit fly optimization
Evolutionary algorithms
Meta-heuristics
Continuous optimization
Harmony search

ABSTRACT

This paper presents an improved fruit fly optimization (IFFO) algorithm for solving continuous function optimization problems. In the proposed IFFO, a new control parameter is introduced to tune the search scope around its swarm location adaptively. A new solution generating method is developed to enhance accuracy and convergence rate of the algorithm. Extensive computational experiments and comparisons are carried out based on a set of 29 benchmark functions from the literature. The computational results show that the proposed IFFO not only significantly improves the basic fruit fly optimization algorithm but also performs much better than five state-of-the-art harmony search algorithms.

1. Introduction

Fruit fly optimization (FFO) is one of the latest meta-heuristic methods presented in the literature. FFO simulates the intelligent foraging behavior of fruit flies or vinegar flies in finding food. It was proposed by Pan [1,2] for global optimization. Fruit flies live in the temperate and tropical climate zones. They have very sensitive osphresis and vision organs which are superior to other species. They feed chiefly on rotten fruits. In the process of finding food, their osphresis organs smell all kinds of scents in the air. They then fly towards the corresponding locations. When they get close to the food locations, they find foods using their visions and then fly towards that direction.

The FFO algorithm has many advantages such as a simple structure, immediately accessible for practical applications, ease of implementation and speed to acquire solutions. Therefore, it has captured much attention and has been successfully applied to solve a wide range of practical optimization problems including the financial distress [2], power load forecasting [3], web auction logistics service [4], PID controller tuning [5,6], and multidimensional knapsack problem [7]. However, to the best of our knowledge, the FFO algorithm has not yet been used to solve high-dimensional continuous function optimization problems which are used extensively in science, engineering, and finance. Most of these problems are no longer linear, quadratic, nor unimodal. Their objective functions are often multimodal with peaks, valleys, channels, and flat hyper-planes of different heights. Solving these types of problems to optimality undoubtedly becomes a true challenge [8].

In this paper, we adapt the FFO algorithm to high-dimensional functions and present an improved variant, namely improved fruit fly optimization (IFFO). Some advanced techniques are introduced to improve the effectiveness of the FFO algorithm including a new control parameter and an effective solution generating method. The IFFO is applied to various standard optimization problems and compared with the basic FFO, a variant of the FFO, and five existing harmony search (HS) algorithms. Numerical results reveal that the proposed IFFO is a new powerful search algorithm for various optimization problems.

The rest of the paper is organized as follows. In Section 2, the basic FFO algorithm is introduced, followed by the presented IFFO in Section 3. Section 4 lists a total of 29 benchmark functions. Experimental design and comparisons are presented in Section 5. Finally, Section 6 gives the concluding remarks.

2. The basic fruit fly optimization algorithm

The basic FFO consists of four consecutive phases. These are initialization, osphresis foraging, population evaluation, and vision

* Corresponding author. Tel.: +86 27 87544522.
E-mail address: gaoliang@mail.hust.edu.cn (L. Gao).

http://dx.doi.org/10.1016/j.knosys.2014.02.021
0950-7051/

foraging. Firstly, FFO sets its control parameters, i.e., population size and a termination criterion, and initializes its fruit fly swarm location. Then FFO is repeated with the search process of osphresis foraging and vision foraging until the termination criterion is satisfied. At the osphresis foraging phase, a population of fruit flies randomly search food sources around the fruit fly swarm location. After that, the smell concentration value or fitness value is evaluated for each of the food sources. In the vision foraging phase, the best food source with the maximum smell concentration value is found and then the fruit fly group flies towards it. The basic FFO was presented for the financial distress [2]. We adapt it to solve high-dimensional function optimization problems as follows.

2.1. Initialize the problem and algorithm parameters

In general, the global optimization problem can be summarized as follows.

$$\min\ f(X)$$

$$\text{st}:\ x_j \in [LB_j, UB_j], \quad j = 1, 2, \ldots, n \tag{1}$$

where $f(X)$ is the objective function, $X = (x_1, x_2, \ldots, x_n)$ is the set of decision variables, n is the number of decision variables, and LB_j and UB_j are the lower and upper bounds for the decision variable x_j, respectively.

The parameters of the FFO algorithm are the population size (PS) and the maximum number of iterations ($Iter_{\max}$). It is obvious that a good set of parameters can enhance the algorithm's ability to search for the global optimum or near optimum region with a high convergence rate.

2.2. Initialize the fruit fly swarm location

The fruit fly swarm location, $\Delta = (\delta_1, \delta_2, \ldots, \delta_n)$, is randomly initialized in the search space as follows.

$$\delta_j = LB_j + (UB_j - LB_j) \times rand(), \quad j = 1, 2, \ldots, n \tag{2}$$

where $rand()$ is a function which returns a value from the uniform distribution on the interval $[0, 1]$.

2.3. Osphresis foraging phase

In the osphresis foraging phase, a population of PS food sources are generated randomly around the current fruit fly swarm location Δ. Let $\{X_1, X_2, \ldots, X_{PS}\}$ represent the generated food sources, where $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$, $i = 1, 2, \ldots, PS$, is yielded as follows.

$$x_{i,j} = \delta_j \pm rand(), \quad j = 1, 2, \ldots, n \tag{3}$$

2.4. Vision foraging phase

In the vision foraging phase, FFO carries out a greedy selection procedure. The best food source with the lowest fitness, X_{best}, is first found, i.e., $X_{best} = \arg(\min_{i=1,2,\ldots,PS} f(X_i))$. If X_{best} is better than the current fruit fly swarm location Δ, it will replace the swarm location and become a new one in the next iteration, i.e., $\Delta = X_{best}$ if $f(X_{best}) < f(\Delta)$.

2.5. Check stopping criterion

If the stopping criterion (maximum number of iterations) is satisfied, computation is terminated. Otherwise, the osphresis foraging and vision foraging phases are repeated.

2.6. Procedure of the basic FFO

The complete computational procedure of the presented FFO algorithm is outlined in Fig. 1. Note in the algorithm, $\Delta = (\delta_1, \delta_2, \ldots, \delta_n)$ is the fruit fly swam location.

Algorithm 1. The FFO algorithm
Parameters: Population size (PS) and the maximum iterations ($Iter_{\max}$)
Output: Solution X^*

//Initialization
Set PS and $Iter_{\max}$
//Initialize swarm location $\Delta = (\delta_1, \delta_2, .., \delta_n)$
$\delta_j = LB_j + (UB_j - LB_j) \times rand(\ )$, $j = 1, 2, \ldots, n$
$Iter = 0$
$X^* = \Delta$
Repeat
//Osphresis foraging phase
For $i = 1, 2, \ldots, PS$
//Generate food source $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$
$x_{i,j} = \delta_j \pm rand(\ )$, $j = 1, 2, \ldots, n$
If $x_{i,j} > UB_j$ **then** $x_{i,j} = UB_j$
If $x_{i,j} < LB_j$ **then** $x_{i,j} = LB_j$
Endfor

//Vision foraging phase
$X_{best} = \arg\left(\min_{i=1,2,\ldots,PS} f(X_i)\right)$
If $f(X_{best}) < f(\Delta)$ then $\Delta = X_{best}$
If $f(\Delta) < f(X^*)$ then $X^* = \Delta$
$Iter = Iter + 1$
Until $Iter == Iter_{\max}$

Fig. 1. The basic FFO algorithm.

3. The improved fruit fly optimization algorithm

This section presents an improved FFO (IFFO) by introducing a new parameter and solution generating method. We detail the presented IFFO as follows.

3.1. A new parameter and solution generating method

In the osphresis foraging phase, a new solution is produced by adding each decision variable of the swarm location and a random value in the range of $[-1,1]$. In other words, the basic FFO generates food sources around its swarm location within a radius equal to one. This radius is fixed and cannot be changed during iterations. The main drawback of this method appears in the number of iterations that the algorithm needs to find an optimal solution. In the early iterations, the fruit fly swarm location is often far from an optimum solution, this search radius may be too small and considerable increase in iterations needed to find a promising region. In the final generations, the swarm location is close to an optimum or a near optimum solution. A very small scope is needed for the fine-tuning of solution vectors. This search radius is too large. To improve the performance of the FFO and eliminate the drawbacks lies with fixed values of search radius, inspired by [9], we change the search radius dynamically with iteration number as shown in Fig. 2. We also express it in the following formulation.

$$\lambda = \lambda_{\max} \cdot \exp\left(\log\left(\frac{\lambda_{\min}}{\lambda_{\max}}\right) \cdot \frac{Iter}{Iter_{\max}}\right) \tag{4}$$

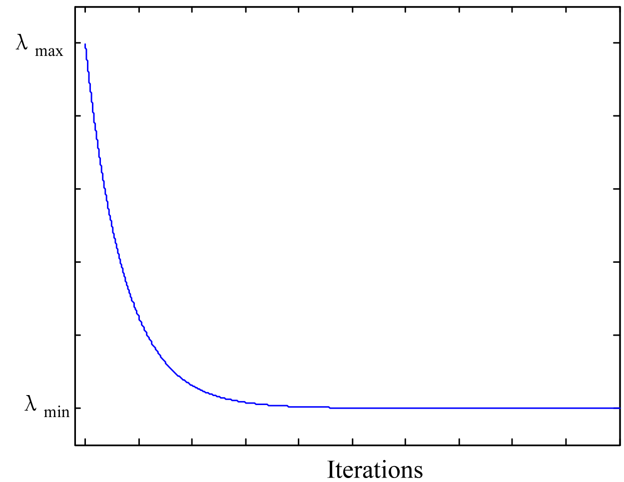

Fig. 2. Variation of λ versus iterations.

where λ is the search radius in each iteration, $\lambda_{\max}$ is the maximum radius, $\lambda_{\min}$ is the minimum radius, $Iter$ is the iteration number and $Iter_{\max}$ is the maximum iteration number.

To further enhance the intensive search, we do not change all the decision variants of the swarm location for producing a new solution. Instead, we randomly choose one decision variant with uniform distribution. Let $d \in \{1, 2, \ldots, n\}$ is a randomly chosen index. A new solution $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ is generated as follows.

Algorithm 2. The IFFO algorithm
Parameters: PS, $\lambda_{\max}$, $\lambda_{\min}$, and $Iter_{\max}$
Output: Solution X^*

//Initialization
Set PS, $\lambda_{\max}$, $\lambda_{\min}$, and $Iter_{\max}$
for $i = 1,2,\ldots,PS$ *// Generate PS food sources* $X_1, X_2, \ldots, X_{PS}$
  $x_{i,j} = LB_j + (UB_j - LB_j) \times rand(\ )$, $j = 1,2,\ldots,n$
end
$\Delta \leftarrow \arg\left(\min_{i=1,2,\ldots,PS} f(X_i)\right)$ *//Set swarm location* Δ
$Iter = 0$
$X^* = \Delta$
Repeat
  $\lambda = \lambda_{\max} \cdot \exp\left(\log\left(\frac{\lambda_{\min}}{\lambda_{\max}}\right) \cdot \frac{Iter}{Iter_{\max}}\right)$

  //Osphresis foraging phase
  For $i = 1,2,\ldots,PS$
    d = a random integer in the range of $[1, n]$
    // Generate food source $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$
    $x_{i,j} = \begin{cases} \delta_j \pm \lambda \cdot rand(\ ) & \text{if } j = d \\ \delta_j & \textit{otherwise} \end{cases}, \quad j = 1,2,\ldots,n$
    If $x_{i,j} > UB_j$ **then** $x_{i,j} = UB_j$
    If $x_{i,j} < LB_j$ **then** $x_{i,j} = LB_j$
  Endfor

  //Vision foraging phase
  $X_{best} = \arg\left(\min_{i=1,2,\ldots,PS} f(X_i)\right)$
  If $f(X_{best}) < f(\Delta)$ then $\Delta = X_{best}$
  If $f(\Delta) < f(X^*)$ then $X^* = \Delta$
  $Iter \leftarrow Iter + 1$
Until $Iter == Iter_{\max}$

Fig. 3. The proposed IFFO algorithm.

$$x_{i,j} = \begin{cases} \delta_j \pm \lambda \cdot rand() & \text{if } j = d \\ \delta_j & \text{otherwise} \end{cases}, \quad j = 1, 2, \ldots, n \tag{5}$$

3.2. Initialization

A good initial swarm location may result in a faster convergence to good solutions. To obtain a good swarm location, we randomly generate a population of *PS* solutions and the best one among them is selected as the initial fruit fly swarm location.

3.3. Procedure of IFFO

The complete computational procedure of the presented IFFO algorithm is outlined in Fig. 3.

4. Test functions

To test the performance of the proposed IFFO algorithm, we consider a total of 30 benchmark problems commonly used in the literature [10–14,8]. They include 15 unimodal and 14 multi-modal functions. We list them as follows.

4.1. Unimodal functions

1. Axis Parallel Hyperellipsoid function, defined as

$$f(x) = \sum_{i=2}^{n} i x_i^2$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-5.12 \leqslant x_i \leqslant 5.12$.

Table 1
Comparison of FFO, FFO_LGMS and IFFO on unimodal functions (Minimum medians are in bold. The function number is in bold if the IFFO is better than all the compared algorithms).

Function	Optimum		$n = 30$			$n = 50$		
			FFO	FFO_LGMS	IFFO	FFO	FFO_LGMS	IFFO
1	0	Median	7.54×10	2.29×10^3	$\mathbf{6.17 \times 10^{-12}}$	3.60×10^2	7.74×10^3	$\mathbf{4.85 \times 10^{-11}}$
		Std	1.29×10	3.36×10^2	3.51×10^{-12}	4.00×10	$9.26e \times 10^2$	1.96×10^{-11}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
2	0	Median	2.67×10^2	1.59×10^6	$\mathbf{6.67 \times 10^{-1}}$	1.65×10^3	6.17×10^6	$\mathbf{9.48 \times 10^{-1}}$
		Std	8.84×10	3.18×10^5	5.36×10^{-1}	3.72×10^2	1.42×10^6	1.98
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
3	−1	Median	-9.54×10^{-2}	-3.64×10^{-2}	**−1.00**	-4.77×10^{-3}	-1.56×10^{-3}	**−1.00**
		Std	2.16×10^{-2}	1.56×10^{-2}	0.00	1.97×10^{-3}	9.57×10^{-4}	0.00
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
4	0	Median	1.86×10^7	6.79×10^9	$\mathbf{3.72 \times 10^{-7}}$	5.87×10^7	2.38×10^{10}	$\mathbf{1.61 \times 10^{-6}}$
		Std	4.93×10^6	4.11×10^9	5.74×10^{-7}	1.09×10^7	7.98×10^9	1.27×10^{-6}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
5	0	Median	2.45×10	1.05×10^2	$\mathbf{3.66 \times 10^{-2}}$	1.79×10^2	4.55×10^2	$\mathbf{1.00 \times 10^{-1}}$
		Std	5.06	2.65×10^1	1.50×10^{-2}	2.93×10	8.04×10^1	2.13e−002
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
6	0	Median	7.67×10^2	2.54×10^8	$\mathbf{7.34 \times 10}$	3.06×10^3	6.02×10^8	$\mathbf{9.00 \times 10}$
		Std	8.81×10^2	3.70×10^7	2.72×10^2	1.50×10^3	1.07×10^8	2.46×10^2
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
7	0	Median	$\mathbf{1.50 \times 10}$	7.80×10^4	3.68×10^2	$\mathbf{1.66 \times 10^2}$	1.84×10^5	7.15×10^3
		Std	3.02	1.66×10^4	2.38×10^2	3.22×10	5.40×10^4	2.30×10^3
			$h = -1$	$h = 1$		$h = -1$	$h = 1$	
8	0	Median	2.59	9.33×10^1	$\mathbf{2.87 \times 10^{-6}}$	3.31×10	9.54×10	$\mathbf{2.21 \times 10}$
		Std	3.37	5.33	3.73×10^{-7}	6.54	1.94	1.25×10
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
9	0	Median	1.36×10^2	2.31×10^8	$\mathbf{2.33 \times 10^{-6}}$	1.15×10^5	6.37×10^{19}	$\mathbf{7.31 \times 10^{-6}}$
		Std	4.25×10^2	1.22×10^{12}	4.25×10^{-7}	9.02×10^{11}	9.85×10^{22}	1.09×10^{-6}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
10	0	Median	5.22	6.63×10^4	$\mathbf{4.96 \times 10^{-13}}$	1.46×10	1.29×10^5	$\mathbf{2.95 \times 10^{-12}}$
		Std	5.99×10^{-1}	7.56×10^3	3.18×10^{-13}	1.13	1.16×10^4	6.62×10^{-13}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
11	0	Median	**0.00**	6.44×10^4	**0.00**	5.00	1.27×10^5	**0.00**
		Std	4.07×10^{-1}	7.81×10^3	0.00	2.59	1.13×10^4	0.00
			$h = 0$	$h = 1$		$h = 1$	$h = 1$	
12	0	Median	1.04	3.17×10^{19}	$\mathbf{4.90 \times 10^{-15}}$	9.95×10^4	7.25×10^{36}	$\mathbf{1.56 \times 10^{-14}}$
		Std	1.99	4.66×10^{21}	2.15×10^{-14}	1.35×10^6	5.49×10^{39}	7.07×10^{-14}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
13	0	Median	8.34×10	8.72×10^3	$\mathbf{6.44 \times 10^{-12}}$	3.82×10^2	2.95×10^4	$\mathbf{4.77 \times 10^{-11}}$
		Std	1.42×10	1.28×10^3	2.50×10^{-12}	4.67×10	3.53×10^3	2.66×10^{-11}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
14	−450	Median	-4.45×10^2	5.25×10^4	$\mathbf{-4.50 \times 10^2}$	-4.35×10^2	1.63×10^5	$\mathbf{-4.50 \times 10^2}$
		Std	5.45×10^{-1}	1.00×10^4	3.00×10^{-13}	1.34	2.84×10^4	8.53×10^{-13}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
15	−450	Median	$\mathbf{-4.34 \times 10^2}$	1.04×10^5	4.75×10^2	$\mathbf{-1.72 \times 10^2}$	2.51×10^5	1.51×10^4
		Std	3.78	2.05×10^4	4.70×10^2	9.69×10	8.06×10^4	5.01×10^3
			$h = -1$	$h = 1$		$h = -1$	$h = 1$	

2. Dixon-Price function, defined as

$$f(x) = \sum_{i=2}^{n} i\left(2x_i^2 - x_{i-1}\right)^2 + (x_1 - 1)^2$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-10 \leqslant x_i \leqslant 10$.

3. Exponential problem, defined as

$$f(x) = -\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right)$$

Subject to $-1 \leqslant x_i \leqslant 1$, $i = 1, 2, \ldots, n$. The optimal value $f(x^*) = -1$ is located at the origin.

4. High Conditioned Elliptic Function, defined as

$$f(x) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.

5. Quartic function, defined as

$$f(x) = \sum_{i=1}^{n} i x_i^4 + rand()$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-1.28 \leqslant x_i \leqslant 1.28$.

6. Rosenbrock function, defined as

$$f(x) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right),$$

where global optimum $x^* = (1, 1, \ldots, 1)$ and $f(x^*) = 0$ for $-30 \leqslant x_i \leqslant 30$.

7. Schwefel's Problem 12, defined as

$$f(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.

Table 2
Comparison of FFO, FFO_LGMS and IFFO on Multi-modal functions (Minimum medians are in bold. The function number is in bold if the IFFO is better than all the compared algorithms).

Function	Optimum		$n = 30$			$n = 50$		
			FFO	FFO_LGMS	IFFO	FFO	FFO_LGMS	IFFO
1	0	Median	2.05×10	2.04×10	$\mathbf{5.13 \times 10^{-7}}$	2.07×10	2.05×10	$\mathbf{8.95 \times 10^{-7}}$
		Std	1.45×10^{-1}	6.39×10^{-2}	9.35×10^{-8}	1.15×10^{-1}	6.00×10^{-2}	1.41×10^{-7}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
2	0	Median	4.05×10	5.21×10	$\mathbf{2.40 \times 10^{-6}}$	8.31×10	1.03×10^2	$\mathbf{9.04 \times 10^{-6}}$
		Std	5.82	6.50	1.26×10^{-6}	7.57	8.25	3.15×10^{-6}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
3	0	Median	2.79×10^2	3.30×10^2	$\mathbf{1.82 \times 10^{-2}}$	4.71×10^2	5.72×10^2	$\mathbf{3.52 \times 10^{-2}}$
		Std	1.08×10	2.22×10	2.18×10^{-1}	1.49×10	2.70×10	6.00
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
4	0	Median	1.42×10	1.39×10	**1.18**	2.37×10	2.39×10	**2.18**
		Std	3.62×10^{-1}	2.51×10^{-1}	4.98×10^{-1}	4.19×10^{-1}	2.66×10^{-1}	5.94×10^{-1}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
5	0	Median	2.28×10	6.15×10^8	$\mathbf{6.11 \times 10^{-15}}$	2.58×10	$1.56\text{e} \times 10^9$	$\mathbf{1.48 \times 10^{-14}}$
		Std	6.75	1.33×10^8	1.97×10^{-14}	5.94	3.26×10^8	2.11×10^{-14}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
6	0	Median	1.26×10^2	5.97×10^2	$\mathbf{1.23 \times 10^{-2}}$	3.98×10^2	1.16×10^3	1.48×10^{-2}
		Std	5.19×10	6.80×10	1.62×10^{-2}	8.98×10	1.05×10^2	3.42×10^{-2}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
7	$1 - n$	Median	−6.02	−3.81	$\mathbf{-2.34 \times 10}$	−7.81	−4.86	$\mathbf{-3.87 \times 10}$
		Std	1.05	5.51×10^{-1}	1.22	1.11	8.46×10^{-1}	1.79
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
8	$-n(n+4) \times (n-1)/6$	Median	2.09×10^6	3.15×10^6	$\mathbf{-2.73 \times 10^3}$	7.69×10^7	5.99×10^7	$\mathbf{2.02 \times 10^4}$
		Std	6.94×10^5	6.29×10^5	4.56×10^3	1.33×10^7	7.89×10^6	4.81×10^4
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
9	0	Median	5.42	4.54	**1.24**	1.04×10	8.55	**2.77**
		Std	3.19×10^{-1}	3.32×10^{-1}	3.80×10^{-1}	2.41×10^{-1}	4.27×10^{-1}	5.18×10^{-1}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
10	0	Median	2.86×10^2	4.18×10^2	$\mathbf{6.34 \times 10^{-11}}$	5.87×10^2	7.65×10^2	$\mathbf{4.78 \times 10^{-10}}$
		Std	3.40×10	3.49×10	1.82×10^{-1}	3.73×10	5.80×10	7.24×10^{-1}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
11	0	Median	1.09×10^2	7.01×10^2	$\mathbf{7.34 \times 10^{-11}}$	2.52×10^2	1.42×10^3	$\mathbf{3.57 \times 10^{-10}}$
		Std	1.01×10	9.08×10	3.15×10^{-11}	1.80×10	1.21×10^2	1.47×10^{-10}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
12	0	Median	$\mathbf{4.00 \times 10^{-1}}$	2.63×10	1.60	$\mathbf{7.00 \times 10^{-1}}$	3.66×10	2.30
		Std	4.74×10^{-2}	1.75	3.09×10^{-1}	8.30×10^{-2}	1.56	3.87×10^{-1}
			$h = -1$	$h = 1$		$h = -1$	$h = 1$	
13	0	Median	6.11×10	4.92×10	$\mathbf{4.69 \times 10^{-3}}$	1.04×10^2	8.85×10	$\mathbf{1.08 \times 10^{-2}}$
		Std	2.38	3.13	5.22×10^{-4}	5.79	3.54	1.12×10^{-3}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	
14	0	Median	1.26×10^3	1.14×10^{18}	$\mathbf{6.22 \times 10^{-15}}$	1.06×10^4	5.05×10^{18}	$\mathbf{2.35 \times 10^{-12}}$
		Std	1.28×10^2	3.16×10^{17}	3.91×10	4.24×10^3	1.43×10^{18}	1.06×10^{-2}
			$h = 1$	$h = 1$		$h = 1$	$h = 1$	

8. Schwefel's Problem 2.21, defined as
$f(x) = \max\{|x_i|, 1 \leqslant i \leqslant n\}$,
where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.
9. Schwefel's problem 2.22, defined as
$$f(x) = \sum_{i=1}^{n}|x_i| + \prod_{i=1}^{n}|x_i|,$$
where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-10 \leqslant x_i \leqslant 10$.
10. Sphere function, defined as
$$f(x) = \sum_{i=1}^{n}x_i^2,$$
where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.
11. Step function, defined as
$$f(x) = \sum_{i=1}^{n}(\lfloor x_i + 0.5\rfloor)^2,$$
where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.
12. Sum of different power, defined as
$$f(x) = \sum_{i=1}^{n}|x_i|^{i+1}$$
where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-1 \leqslant x_i \leqslant 1$.
13. Sum Squares function, defined as
$$f(x) = \sum_{i=1}^{n}ix_i^2$$
where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-10 \leqslant x_i \leqslant 10$.
14. Shifted Sphere function, defined as
$$f(z) = \sum_{i=1}^{n}z_i^2 + f\_bias_1,$$
where $z = x - o$; $o = (o_1, o_2, \ldots, o_n)$ is the shifted global optimum; global optimum $x^* = o$ and $f(x^*) = f\_bias_1 = -450$ for $-100 \leqslant x_i \leqslant 100$.

Table 3
Comparison with the HS variants on 30-dimensional unimodal functions (Minimum medians are in bold. The function number is in bold if the IFFO is better than all the benchmark algorithms).

Function	Optimum		IFFO	HS	IHS	GHS	SGHS	DLHS
1	0	Median	$\mathbf{6.17 \times 10^{-12}}$	3.61×10^{-3}	1.25×10^{-4}	1.02×10^{-4}	2.95×10^{-8}	2.88×10^{-9}
		Std	3.51×10^{-12}	1.08×10^{-3}	8.97×10^{-4}	2.30×10^{-3}	2.57×10^{-8}	5.09×10^{-9}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
2	0	Median	$\mathbf{6.67 \times 10^{-1}}$	4.17	2.03	6.98×10^{-1}	1.10	8.32×10^{-1}
		Std	5.36×10^{-1}	2.02	1.96	2.26	1.28	1.33
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
3	−1	Median	**−1.00**	**−1.00**	**−1.00**	**−1.00**	**−1.00**	**−1.00**
		Std	0.00	1.19×10^{-5}	3.79×10^{-8}	2.78×10^{-6}	0.00	0.00
				$h = 0$	$h = 0$	$h = 0$	$h = 0$	$h = 0$
4	0	Median	$\mathbf{3.72 \times 10^{-7}}$	1.24×10^{4}	2.89×10^{3}	1.97×10^{3}	3.40×10	7.41×10^{-1}
		Std	5.74×10^{-7}	7.74×10^{3}	1.98×10^{3}	2.13×10^{4}	6.69×10^{2}	2.00×10^{2}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
5	0	Median	3.66×10^{-2}	6.98×10^{-2}	7.23×10^{-2}	$\mathbf{7.57 \times 10^{-3}}$	3.57×10^{-2}	8.33×10^{-2}
		Std	1.50×10^{-2}	2.23×10^{-2}	2.47×10^{-2}	5.03×10^{-3}	1.85×10^{-2}	2.83×10^{-2}
				$h = 1$	$h = 1$	$h = -1$	$h = -1$	$h = 1$
6	0	Median	7.34×10	1.96×10^{2}	1.53×10^{2}	$\mathbf{3.06 \times 10}$	8.67×10	1.57×10^{2}
		Std	2.72×10^{2}	6.19×10^{2}	1.84×10^{2}	5.31×10	9.98×10	3.81×10^{2}
				$h = 1$	$h = 1$	$h = 0$	$h = 0$	$h = 1$
7	0	Median	$\mathbf{3.68 \times 10^{2}}$	4.35×10^{3}	3.40×10^{3}	1.97×10^{3}	1.13×10^{3}	7.62×10^{2}
		Std	2.38×10^{2}	1.05×10^{3}	1.12×10^{3}	8.27×10^{3}	4.36×10^{2}	5.96×10^{2}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
8	0	Median	$\mathbf{2.87 \times 10^{-6}}$	7.00	6.59	9.23	6.91×10^{-1}	6.11
		Std	3.73×10^{-7}	9.66×10^{-1}	1.06	5.71	2.33×10^{-1}	2.20
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
9	0	Median	$\mathbf{2.33 \times 10^{-6}}$	8.59×10^{-2}	2.66×10^{-3}	2.75×10^{-2}	2.08×10^{-4}	6.72×10^{-5}
		Std	4.25×10^{-7}	5.39×10^{-2}	3.53×10^{-2}	4.55×10^{-2}	9.89×10^{-4}	4.06×10^{-5}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
10	0	Median	$\mathbf{4.96 \times 10^{-13}}$	7.07	5.08×10^{-7}	2.51×10^{-3}	2.76×10^{-4}	5.34×10^{-10}
		Std	3.18×10^{-13}	3.24	1.25×10^{-7}	5.36×10^{-2}	5.03×10^{-3}	1.98×10^{-9}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
11	0	Median	**0.00**	3.00	**0.00**	**0.00**	**0.00**	**0.00**
		Std	0.00	2.07	7.18×10^{-1}	0.00	0.00	1.81
				$h = 1$	$h = 0$	$h = 0$	$h = 0$	$h = 0$
12	0	Median	$\mathbf{4.90 \times 10^{-15}}$	6.12×10^{-9}	7.92×10^{-9}	2.11×10^{-5}	1.02×10^{-12}	9.34×10^{-15}
		Std	2.15×10^{-14}	1.08×10^{-8}	1.10×10^{-8}	1.06×10^{-4}	5.13×10^{-12}	5.23×10^{-14}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 0$
13	0	Median	$\mathbf{6.44 \times 10^{-12}}$	6.57×10^{-3}	8.25×10^{-4}	3.90×10^{-4}	3.41×10^{-8}	2.90×10^{-9}
		Std	2.50×10^{-12}	2.17×10^{-3}	3.50×10^{-3}	8.76×10^{-3}	4.55×10^{-5}	4.65×10^{-9}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
14	−450	Median	$\mathbf{-4.50 \times 10^{2}}$	-4.44×10^{2}	$\mathbf{-4.50 \times 10^{2}}$	1.33×10^{3}	$\mathbf{-4.50 \times 10^{2}}$	$\mathbf{-4.50 \times 10^{2}}$
		Std	3.00×10^{-13}	2.78	1.32×10^{-7}	3.72×10^{2}	9.16×10^{-3}	2.52×10^{-9}
				$h = 1$	$h = 0$	$h = 1$	$h = 0$	$h = 0$
15	−450	Median	$\mathbf{4.75 \times 10^{2}}$	4.89×10^{3}	4.32×10^{3}	2.29×10^{4}	7.16×10^{2}	2.52×10^{3}
		Std	4.70×10^{2}	1.30×10^{3}	1.38×10^{3}	5.66×10^{3}	5.29×10^{2}	1.39×10^{3}
				$h = 1$	$h = 1$	$h = 1$	$h = 0$	$h = 1$

15. Shifted Schwefel's problem 1.2, defined as

$$f(z) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} z_j\right)^2 + f\_bias_2,$$

where $z = x - o$; $o = (o_1, o_2, \ldots, o_n)$ is the shifted global optimum; global optimum $x^* = o$ and $f(x^*) = f\_bias_2 = -450$ for $-100 \leqslant x_i \leqslant 100$.

4.2. multimodal problems

1. Ackley's function, defined as

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e,$$

where global optimum $x^* = 0$ with $f(x^*) = 0$ for $-32 \leqslant x_i \leqslant 32$.

2. Alpine function, defined as

$$f(x) = \sum_{i=1}^{n} |x_i \sin(x_i) + 0.1x_i|$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-10 \leqslant x_i \leqslant 10$.

3. Expansion of F10, defined as

$$f(x) = f_{10}(x_1, x_2) + \cdots + f_{10}(x_{i-1}, x_i) + f_{10}(x_n, x_1),$$

where $f_{10}(x, y) = (x^2 + y^2)^{0.25}[\sin^2(50(x^2 + y^2)^{0.1}) + 1]$ with global optimum $x^* = (0, 0, \ldots, 0)$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.

4. Expanded Scaffer's function, defined as

$$f_s(x, y) = 0.5 + \frac{\sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5}{\left(1 + 0.001(x^2 + y^2)\right)^2}$$

Table 4
Comparison with the HS variants on 50-dimensional unimodal functions (Minimum medians are in bold. The function number is in bold if the IFFO is better than all the benchmark algorithms).

Function	Optimum		IFFO	HS	IHS	GHS	SGHS	DLHS
1	0	Median	$\mathbf{4.85 \times 10^{-11}}$	2.44×10	2.28×10	1.27×10^{-1}	1.26×10^{-6}	4.49×10^{-8}
		Std	1.96×10^{-11}	6.18	4.17	3.42×10^{-1}	3.60×10^{-4}	1.91×10^{-3}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
2	0	Median	$\mathbf{9.48 \times 10^{-1}}$	3.47×10^2	3.57×10^2	4.34	5.86	2.54
		Std	1.98	1.44×10^2	1.35×10^2	4.42×10	3.38	2.85
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
3	−1	Median	**−1.00**	-9.84×10^{-1}	-9.79×10^{-1}	**−1.00**	**−1.00**	**−1.00**
		Std	0.00	3.65×10^{-3}	4.05×10^{-3}	1.33×10^{-4}	0.00	0.00
				$h = 1$	$h = 1$	$h = 0$	$h = 0$	$h = 0$
4	0	Median	$\mathbf{1.61 \times 10^{-6}}$	2.83×10^6	3.31×10^5	8.81×10^5	6.72×10^3	1.04×10^3
		Std	1.27×10^{-6}	1.62×10^6	1.26×10^5	1.97×10^6	2.59×10^4	1.17×10^4
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
5	0	Median	1.00×10^{-1}	3.99×10^{-1}	4.42×10^{-1}	$\mathbf{6.99 \times 10^{-2}}$	1.27×10^{-1}	2.35×10^{-1}
		Std	2.13×10^{-2}	1.21×10^{-1}	9.13×10^{-2}	5.21×10^{-2}	4.64×10^{-2}	5.94×10^{-2}
				$h = 1$	$h = 1$	$h = -1$	$h = 1$	$h = 1$
6	0	Median	$\mathbf{9.00 \times 10}$	2.72×10^4	2.65×10^4	1.92×10^2	1.69×10^2	3.08×10^2
		Std	2.46×10^2	8.50×10^3	9.97×10^3	1.23×10^3	3.42×10^2	7.30×10^2
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
7	0	Median	$\mathbf{7.15 \times 10^3}$	3.00×10^4	2.83×10^4	7.34×10^4	8.13×10^3	1.51×10^4
		Std	2.30×10^3	6.24×10^3	5.08×10^3	2.13×10^4	2.32×10^3	4.32×10^3
				$h = 1$	$h = 1$	$h = 1$	$h = 0$	$h = 1$
8	0	Median	2.21×10	2.16×10	2.15×10	2.23×10	**5.02**	1.90×10
		Std	1.25×10	1.74	1.93	8.47	2.38	2.13
				$h = 0$	$h = 0$	$h = 0$	$h = -1$	$h = -1$
9	0	Median	$\mathbf{7.31 \times 10^{-6}}$	9.62	8.14	3.51×10^{-1}	2.56×10^{-2}	5.38×10^{-4}
		Std	1.09×10^{-6}	1.10	1.12	3.80×10^{-1}	2.54×10^{-2}	8.10×10^{-2}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
10	0	Median	$\mathbf{2.95 \times 10^{-12}}$	5.22×10^2	4.77×10^2	6.86×10^{-1}	1.30×10^{-1}	2.13×10^{-6}
		Std	6.62×10^{-13}	1.20×10^2	1.01×10^2	3.00	1.13×10^{-1}	6.63×10^{-4}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
11	0	Median	**0.00**	3.97×10^2	3.92×10^2	**0.00**	**0.00**	4.00
		Std	0.00	9.30×10	1.04×10^2	0.00	0.00	7.26
				$h = 1$	$h = 1$	$h = 0$	$h = 0$	$h = 1$
12	0	Median	$\mathbf{1.56 \times 10^{-14}}$	8.49	6.19	1.18×10^{-2}	5.89×10^{-13}	2.48×10^{-14}
		Std	7.07×10^{-14}	2.65×10	2.98×10	6.31×10^{-2}	1.48×10^{-11}	4.64×10^{-11}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 0$
13	0	Median	$\mathbf{4.77 \times 10^{-11}}$	9.54×10	8.65×10	4.84×10^{-1}	1.04×10^{-3}	3.32×10^{-7}
		Std	2.66×10^{-11}	2.12×10	2.30×10	1.30	1.27×10^{-2}	1.10×10^{-2}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
14	−450	Median	$\mathbf{-4.50 \times 10^2}$	1.17×10^2	7.31×10	1.48×10^4	$\mathbf{-4.50 \times 10^2}$	$\mathbf{-4.50 \times 10^2}$
		Std	8.53×10^{-13}	1.01×10^2	1.14×10^2	2.48×10^3	1.33×10^{-1}	5.81×10^{-4}
				$h = 1$	$h = 1$	$h = 1$	$h = 0$	$h = 0$
15	−450	Median	1.51×10^4	3.59×10^4	3.41×10^4	9.52×10^4	1.09×10^4	2.19×10^4
		Std	5.01×10^3	7.17×10^3	9.38×10^3	1.36×10^4	3.77×10^3	6.80×10^3
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$

Expanded to

$$f(x) = f_s(x_1, x_2) + f_s(x_2, x_3) + \cdots + f_s(x_n, x_1)$$

with global optimum $x^* = (0, 0, \ldots, 0)$ and $f(x^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.

5. Generalized Penalized Function 1, defined as

$$f(x) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n}\mu(x_i, 10, 100, 4)$$

$$y_i = 1 + \frac{1}{4}(x_i + 1), \mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leqslant x_i \leqslant a. \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

with global optimum $x^* = (-1, -1, \ldots, -1)$ and $f(x^*) = 0$ for $-50 \leqslant x_i \leqslant 50$.

6. Griewank function, defined as

$$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

where global optimum $x^* = 0$ with $f(x^*) = 0$ for $-600 \leqslant x_i \leqslant 600$.

7. Inverted Cosine Wave function, defined as

$$f(x) = -\sum_{i=1}^{n-1}\left(\exp\left(-\frac{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}{8}\right) \times \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}\right)\right),$$

where global optimum $x^* = 0$ and $f(x^*) = 1 - n$ for $-5 \leqslant x_i \leqslant 5$.

8. Neumaier 3 Problem, defined as

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$$

Subject to $-n^2 \leqslant x_i \leqslant n^2$, $i = 1, 2, \ldots, n$. The global minima can be expressed as $f(x^*) = -\frac{n(n+4)(n-1)}{6}$, $x^* = i(n + 1 - i)$.

Table 5
Comparison with the HS variants on 30-dimensional multimodal functions (Minimum medians are in bold. The function number is in bold if the IFFO is better than all the benchmark algorithms).

Function	Optimum		IFFO	HS	IHS	GHS	SGHS	DLHS
1	0	Median	$\mathbf{5.13 \times 10^{-7}}$	1.04	4.20×10^{-1}	1.22×10^{-2}	6.11×10^{-5}	1.93
		Std	9.35×10^{-8}	3.41×10^{-1}	5.34×10^{-1}	2.61×10^{-2}	4.66×10^{-3}	4.96×10^{-1}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
2	0	Median	$\mathbf{2.40 \times 10^{-6}}$	1.49×10^{-1}	5.62×10^{-2}	1.83×10^{-3}	4.24×10^{-5}	9.97×10^{-5}
		Std	1.26×10^{-6}	5.62×10^{-2}	2.41×10^{-2}	1.70×10^{-3}	4.74×10^{-4}	7.46×10^{-4}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
3	0	Median	$\mathbf{1.82 \times 10^{-2}}$	3.43×10	3.47×10	1.45×10	8.48	4.33×10
		Std	2.18×10^{-1}	5.55	6.97	1.52×10	1.67	1.31×10
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
4	0	Median	**1.18**	2.17	2.02	3.08	1.23	1.96
		Std	4.98×10^{-1}	5.88×10^{-1}	5.58×10^{-1}	1.74	4.59×10^{-1}	5.56×10^{-1}
				$h = 1$	$h = 1$	$h = 1$	$h = 0$	$h = 1$
5	0	Median	$\mathbf{6.11 \times 10^{-15}}$	7.52×10^{-3}	1.37×10^{-2}	3.69×10^{-5}	8.40×10^{-10}	3.46×10^{-4}
		Std	1.97×10^{-14}	2.17×10^{-2}	3.41×10^{-2}	2.15×10^{-4}	7.20×10^{-6}	6.54×10^{-2}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
6	0	Median	1.23×10^{-2}	1.09	2.03×10^{-2}	$\mathbf{6.03 \times 10^{-3}}$	7.65×10^{-2}	1.30×10^{-2}
		Std	1.62×10^{-2}	3.18×10^{-2}	1.23×10^{-1}	1.42×10–001	6.63×10^{-2}	1.81×10^{-2}
				$h = 1$	$h = 1$	$h = 0$	$h = 1$	$h = 0$
7	$1 - n$	Median	$\mathbf{-2.34 \times 10}$	-2.29×10	-2.27×10	-2.03×10	-2.31×10	-2.34×10
		Std	1.22	1.11	1.45	4.02	1.43	1.23
				$h = 0$	$h = 0$	$h = 1$	$h = 0$	$h = 0$
8	$-n(n+4) \times (n-1)/6$	Median	$\mathbf{-2.73 \times 10^{3}}$	1.59×10^{4}	2.67×10^{4}	1.38×10^{5}	1.03×10^{3}	-4.50×10^{2}
		Std	4.56×10^{3}	1.38×10^{4}	1.53×10^{4}	8.87×10^{4}	4.57×10^{3}	6.29×10^{3}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 0$
9	0	Median	1.24	$\mathbf{7.97 \times 10^{-1}}$	1.02	1.76	8.48×10^{-1}	1.24
		Std	3.80×10^{-1}	3.30×10^{-1}	3.51×10^{-1}	6.49×10^{-1}	3.71×10^{-1}	3.53×10^{-1}
				$h = -1$	$h = 0$	$h = 1$	$h = -1$	$h = 0$
10	0	Median	$\mathbf{6.34 \times 10^{-11}}$	1.03	2.16	1.31×10^{-3}	4.43×10^{-2}	1.00
		Std	1.82×10^{-1}	7.56×10^{-1}	1.35	2.87×10^{-2}	3.01×10^{-1}	1.42
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
11	0	Median	$\mathbf{7.34 \times 10^{-11}}$	1.05	1.77	1.31×10^{-3}	4.71×10^{-7}	8.72×10^{-1}
		Std	3.15×10^{-11}	6.29×10^{-1}	1.49	1.71×10^{-2}	6.09×10^{-7}	1.00
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
12	0	Median	1.60	1.80	1.70	$\mathbf{3.00 \times 10^{-1}}$	1.00	1.80
		Std	3.09×10^{-1}	2.97×10^{-1}	2.36×10^{-1}	3.14×10^{-1}	1.86×10^{-1}	3.27×10^{-1}
				$h = 1$	$h = 1$	$h = -1$	$h = -1$	$h = 1$
13	0	Median	$\mathbf{4.69 \times 10^{-3}}$	2.84	5.87×10^{-1}	2.76×10^{-1}	7.06×10^{-2}	6.30×10^{-1}
		Std	5.22×10^{-4}	3.29×10^{-1}	1.35×10^{-1}	2.90×10^{-1}	1.07×10^{-2}	6.59×10^{-1}
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$
14	0	Median	$\mathbf{6.22 \times 10^{-15}}$	1.35×10^{5}	1.13×10^{3}	1.48×10^{2}	1.33×10^{2}	6.80×10
		Std	3.91×10	2.96×10^{5}	7.70×10^{3}	1.37×10^{3}	1.53×10^{2}	8.51×10
				$h = 1$	$h = 1$	$h = 1$	$h = 1$	$h = 1$

9. Pathologic function, defined as

$$f(x)=\sum_{i=1}^{n-1}\left(0.5+\frac{\sin^2\left(\sqrt{100x_i^2+x_{i+1}^2}\right)-0.5}{1+0.001\left(x_i^2-2x_ix_{i+1}+x_{i+1}^2\right)^2}\right)^2$$

where global optimum $x^*=0$ and $f(x^*)=0$ for $-100\leqslant x_i\leqslant 100$.

10. Rastrigin function, defined as

$$f(x)=\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)+10\right),$$

where global optimum $x^*=0$ with $f(x^*)=0$ for $-5.12\leqslant x_i\leqslant 5.12$.

11. Non-continuous Rastrigin function, defined as

$$f(y)=\sum_{i=1}^{n}\left(y_i^2-10\cos(2\pi y_i)+10\right),$$

where $y_i=\begin{cases}x_i & |x_i|<1/2\\ round(2x_i)/2 & |x_i|\geqslant 1/2\end{cases}$ with global optimum $x^*=0$ and $f(x^*)=0$ for $-5.12\leqslant x_i\leqslant 5.12$.

12. Salomon problem, defined as

$$f(x)=1-\cos\left(2\pi\sqrt{\sum_{i=1}^{n}x_i^2}\right)+0.1\sqrt{\sum_{i=1}^{n}x_i^2}$$

where global minimizer $x^*=0$ with $f(x^*)=0$ for $-100\leqslant x_i\leqslant 100$.

13. Weierstrass problem, defined as

$$f(x)=\sum_{i=1}^{n}\left\{\sum_{k=0}^{k_{\max}}\left[a^k\cos(2\pi b^k(x_i+0.5))\right]\right\}-n\sum_{k=0}^{k_{\max}}\left[a^k\cos(2\pi b^k\cdot 0.5)\right]$$

where $a=0.5$, $b=3$, $k_{\max}=30$ with global optimum $x^*=0$ and $f(x^*)=0$ for $-0.5\leqslant x_i\leqslant 0.5$.

14. Whitley function, defined as

$$f(x)=\sum_{k=1}^{n}\sum_{j=1}^{n}\left(\frac{y_{jk}^2}{4000}-\cos(y_{jk})+1\right),\quad y_{jk}=100(x_k-x_j^2)^2+(1-x_j^2)^2$$

with global optimum $x^*=(1,1,\ldots,1)$ and $f(x^*)=0$ for $-100\leqslant x_i\leqslant 100$.

Table 6
Comparison with the HS variants on 50-dimensional multimodal functions (Minimum *Medians* are in bold. The function number is in bold if the IFFO is better than all the benchmark algorithms).

Function	Optimum		IFFO	HS	IHS	GHS	SGHS	DLHS
1	0	Median	$\mathbf{8.95\times10^{-7}}$	5.31	5.08	1.17×10^{-1}	6.25×10^{-2}	2.93
		Std	1.41×10^{-7}	4.11×10^{-1}	4.00×10^{-1}	2.49×10^{-1}	3.80×10^{-2}	6.01×10^{-1}
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
2	0	Median	$\mathbf{9.04\times10^{-6}}$	1.88	2.00	2.47×10^{-2}	1.64×10^{-3}	1.06×10^{-3}
		Std	3.15×10^{-6}	2.97×10^{-1}	4.14×10^{-1}	3.19×10^{-2}	2.30×10^{-3}	5.10×10^{-3}
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
3	0	Median	$\mathbf{3.52\times10^{-2}}$	1.36×10^{2}	1.30×10^{2}	2.84×10	2.26×10	9.70×10
		Std	6.00	1.04×10	1.11×10	1.86×10	4.18	1.83×10
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
4	0	Median	**2.18**	7.88	7.34	1.03×10	2.79	4.45
		Std	5.94×10^{-1}	8.72×10^{-1}	8.43×10^{-1}	4.05	6.74×10^{-1}	1.11
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
5	0	Median	$\mathbf{1.48\times10^{-14}}$	3.31	3.49	4.19×10^{-3}	1.20×10^{-4}	2.68×10^{-1}
		Std	2.11×10^{-14}	$7.27\times10{-}001$	1.22	2.54×10^{-2}	5.81×10^{-4}	3.96×10^{-1}
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
6	0	Median	1.48×10^{-2}	5.69	5.47	9.68×10^{-1}	2.24×10^{-1}	$\mathbf{7.57\times10^{-3}}$
		Std	3.42×10^{-2}	1.19	1.17	3.46×10^{-1}	1.26×10^{-1}	1.27×10^{-2}
				$h=1$	$h=1$	$h=1$	$h=1$	$h=-1$
7	$1-n$	Median	-3.87×10	-3.54×10	-3.58×10	-2.94×10	-3.82×10	$\mathbf{-3.94\times10}$
		Std	1.79	1.18	1.10	6.20	2.26	1.55
				$h=1$	$h=1$	$h=1$	$h=0$	$h=0$
8	$-n(n+4)\times(n-1)/6$	Median	$\mathbf{2.02\times10^{4}}$	1.15×10^{6}	1.07×10^{6}	5.00×10^{6}	7.31×10^{4}	1.33×10^{5}
		Std	4.81×10^{4}	3.74×10^{5}	5.31×10^{5}	2.74×10^{6}	4.72×10^{4}	8.99×10^{4}
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
9	0	Median	2.77	3.34	3.47	5.90	**2.52**	3.06
		Std	5.18×10^{-1}	3.42×10^{-1}	4.20×10^{-1}	5.40×10^{-1}	4.78×10^{-1}	4.02×10^{-1}
				$h=1$	$h=1$	$h=1$	$h=0$	$h=1$
10	0	Median	$\mathbf{4.78\times10^{-10}}$	3.82×10	3.95×10	1.66×10^{-1}	3.60	9.49
		Std	7.24×10^{-1}	4.02	5.04	3.96×10^{-1}	1.65	5.28
				$h=1$	$h=1$	$h=0$	$h=1$	$h=1$
11	0	Median	$\mathbf{3.57\times10^{-10}}$	3.28×10	3.48×10	3.32×10^{-1}	3.39×10^{-1}	8.51
		Std	1.47×10^{-10}	4.94	3.30	1.02	6.69×10^{-1}	5.22
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
12	0	Median	2.30	4.98	5.10	8.07×10^{-1}	**1.80**	3.25
		Std	3.87×10^{-1}	5.29×10^{-1}	5.22×10^{-1}	3.26×10^{-1}	2.64×10^{-1}	4.52×10^{-1}
				$h=1$	$h=1$	$h=-1$	$h=-1$	$h=1$
13	0	Median	$\mathbf{4.69\times10^{-3}}$	2.84	5.87×10^{-1}	2.76×10^{-1}	7.06×10^{-2}	6.30×10^{-1}
		Std	5.22×10^{-4}	3.29×10^{-1}	1.35×10^{-1}	2.90×10^{-1}	1.07×10^{-2}	6.59×10^{-1}
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$
14	0	Median	$\mathbf{6.22\times10^{-15}}$	1.35×10^{5}	1.13×10^{3}	1.48×10^{2}	1.33×10^{2}	6.80×10
		Std	3.91×10	2.96×10^{5}	7.70×10^{3}	1.37×10^{3}	1.53×10^{2}	8.51×10
				$h=1$	$h=1$	$h=1$	$h=1$	$h=1$

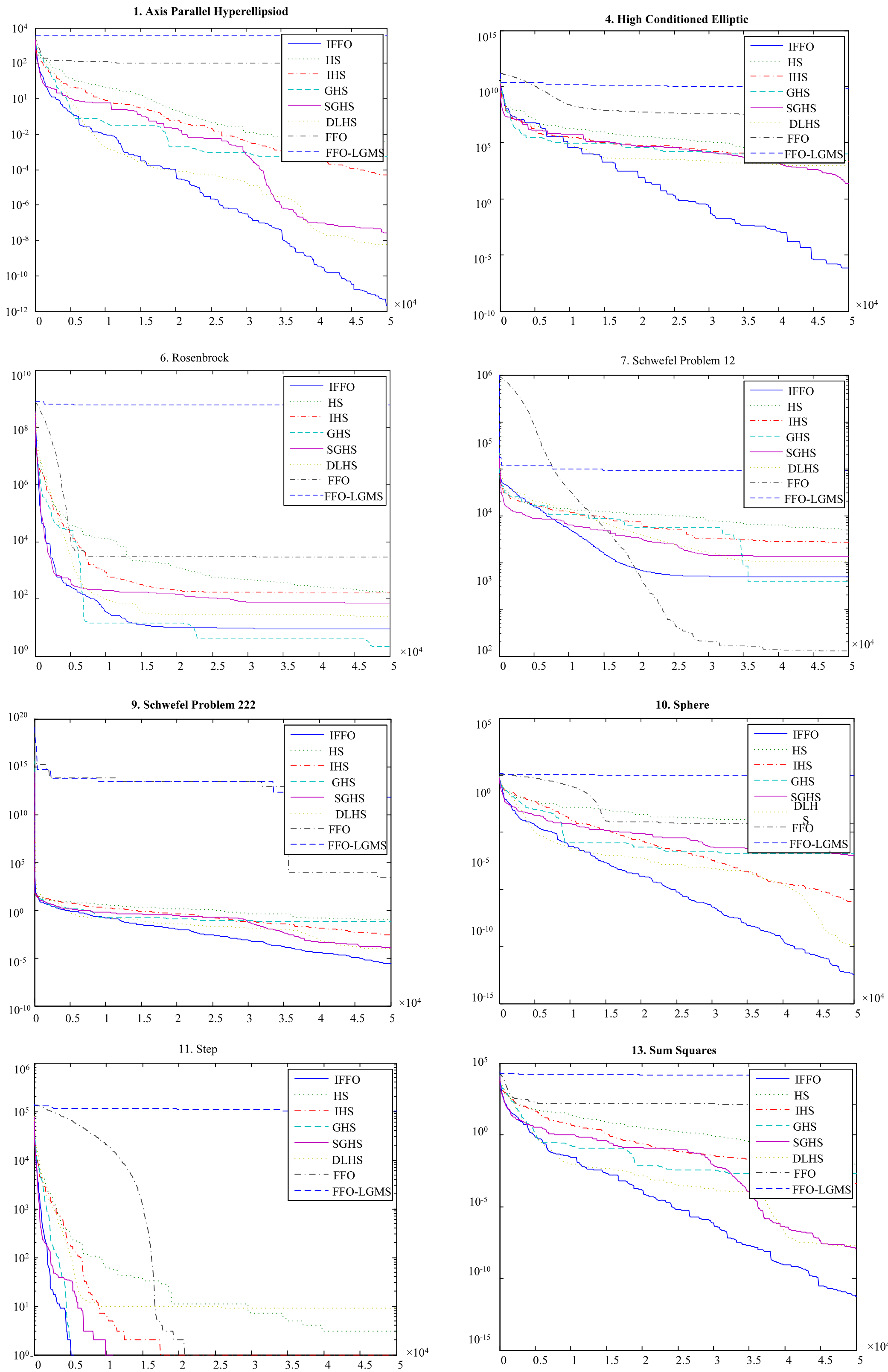

Fig. 4. Typical solution history graph along the iterations for 30-dimensional unimodal functions.

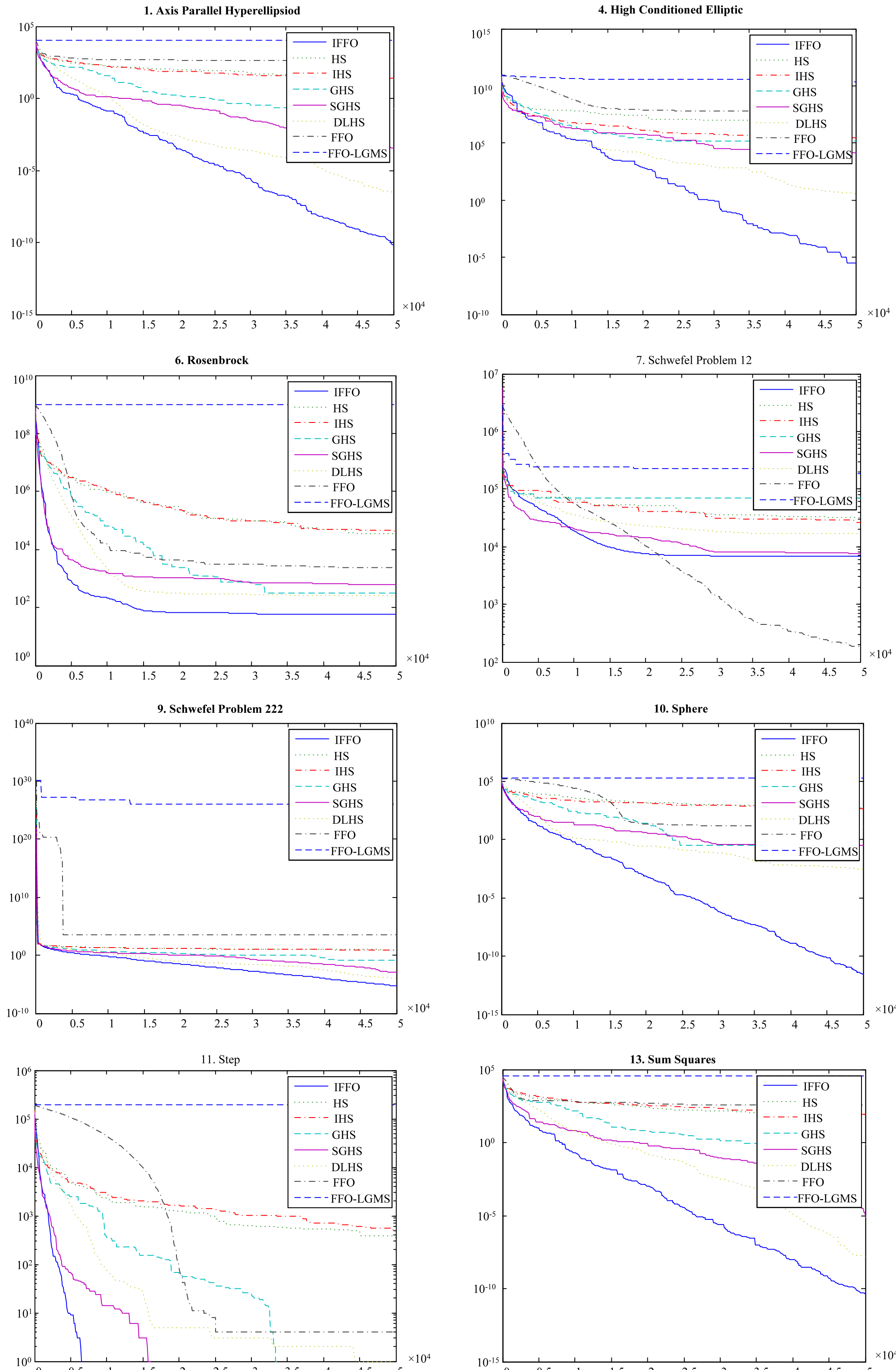

Fig. 5. Typical solution history graph along the iterations for 50-dimensional unimodal functions.

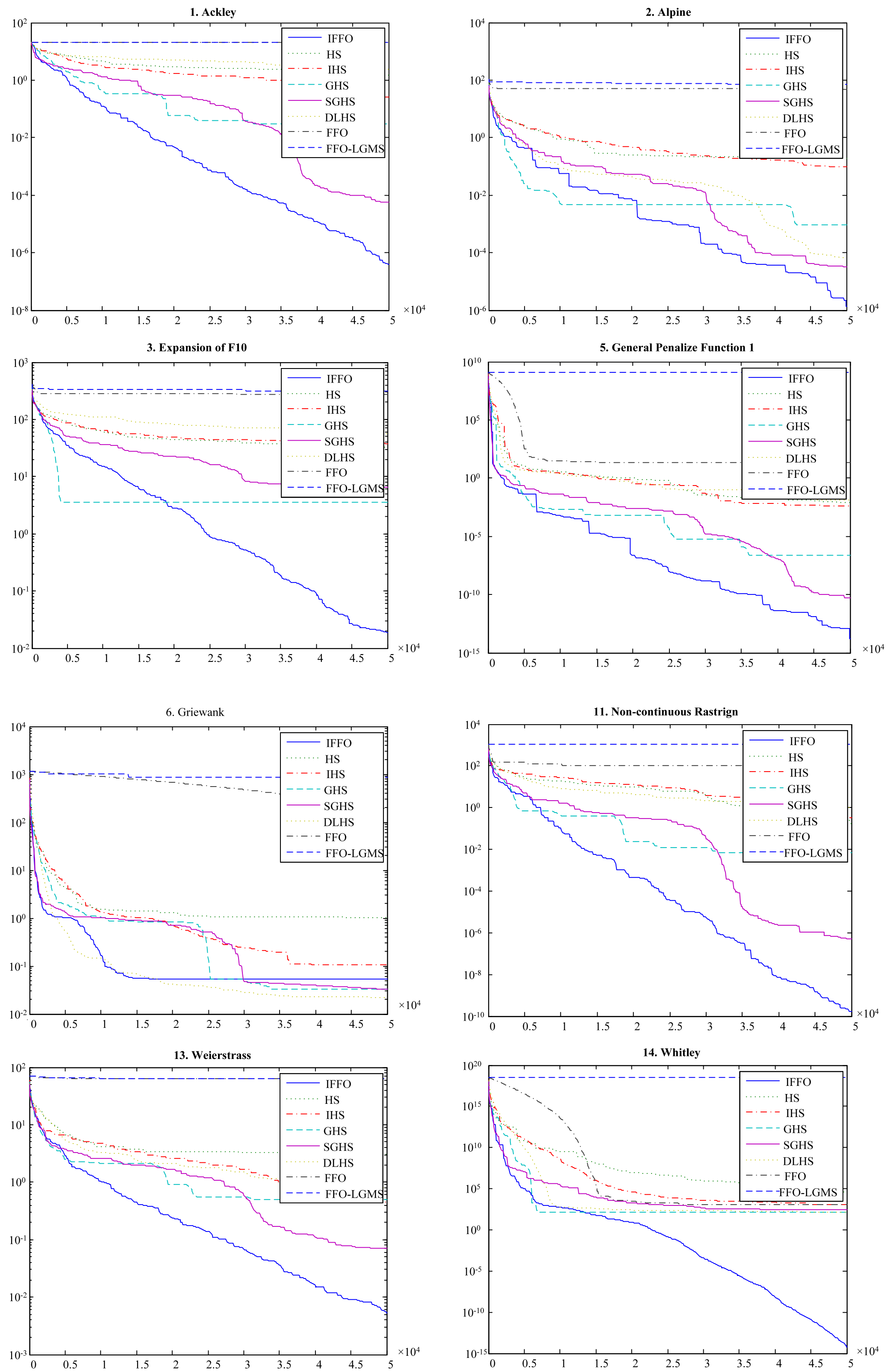

Fig. 6. Typical solution history graph along the iterations for 30-dimensional multimodal functions.

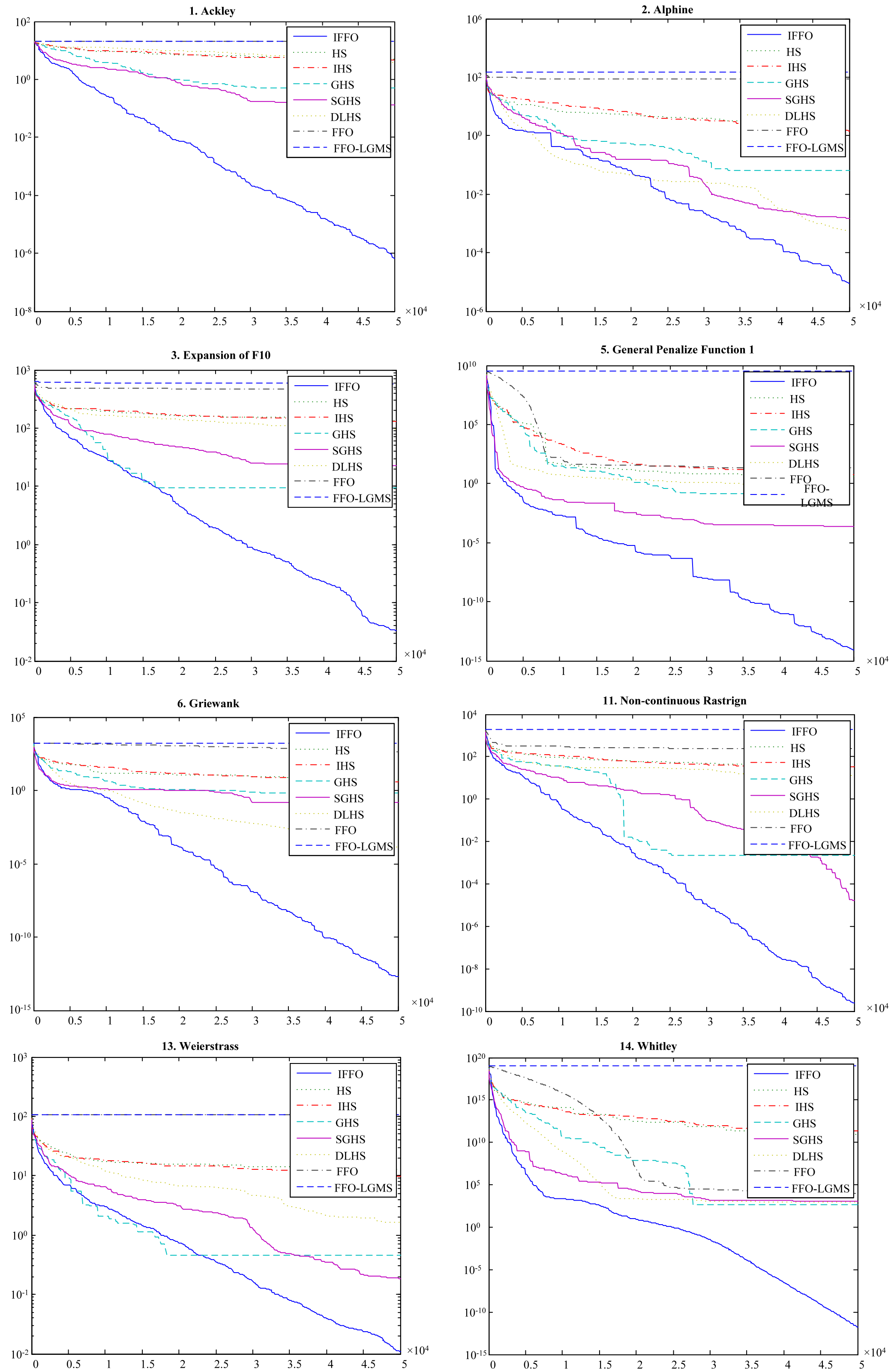

Fig. 7. Typical solution history graph along the iterations for 50-dimensional multimodal functions.

Table 7
The effect of *PS* on the IFFO algorithm (50-dimensional multimodal functions).

	Optimum	*PS* = 5	*PS* = 10	*PS* = 15	*PS* = 20	*PS* = 30	*PS* = 40
1	0	9.43×10^{-7}	8.95×10^{-7}	8.85×10^{-7}	9.48×10^{-7}	9.66×10^{-7}	1.02×10^{-6}
2	0	6.82×10^{-6}	9.04×10^{-6}	8.77×10^{-6}	1.00×10^{-5}	9.54×10^{-6}	9.92×10^{-6}
3	0	3.59×10^{-2}	3.52×10^{-2}	3.60×10^{-2}	3.58×10^{-2}	3.76×10^{-2}	2.79×10^{1}
4	0	2.21	2.18	2.40	2.36	2.94	4.32
5	0	1.81×10^{-14}	1.48×10^{-14}	2.05×10^{-14}	1.46×10^{-14}	2.14×10^{-14}	2.20×10^{-14}
6	0	1.23×10^{-2}	1.48×10^{-2}	1.48×10^{-2}	1.48×10^{-2}	1.97×10^{-2}	1.85×10^{-2}
7	$1-n$	-3.83×10^{1}	-3.87×10^{1}	-3.83×10^{1}	-3.74×10^{1}	-3.76×10^{1}	-3.76×10^{1}
8	$-n(n+4) \times (n-1)/6$	6.58×10^{3}	2.02×10^{4}	3.21×10^{4}	4.67×10^{4}	7.05×10^{4}	7.99×10^{4}
9	0	2.79	2.77	2.99	2.99	3.22	3.55
10	0	5.00×10^{-10}	4.78×10^{-10}	9.95×10^{-10}	9.95×10^{-1}	1.99	2.98
11	0	3.36×10^{-10}	3.57×10^{-10}	3.63×10^{-10}	4.03×10^{-10}	4.96×10^{-10}	6.25×10^{-10}
12	0	2.45	2.30	2.20	2.60	2.45	2.30
13	0	1.09×10^{-2}	1.08×10^{-2}	1.11×10^{-2}	1.10×10^{-2}	1.14×10^{-2}	1.15×10^{-2}
14	0	2.26×10^{-13}	2.35×10^{-12}	2.88×10^{-9}	8.68×10^{-7}	6.38×10^{-3}	1.28×10^{2}

5. Computational results

5.1. Comparison with the variants of the FFO

In this section, we compare the presented IFFO with the basic FFO and a new variant, namely FFO_LGMS, presented by Shan et al. [15] based on the 29 benchmark functions with dimensions equal to 30 and 50 (i.e. $n = 30$ and $n = 50$). Unlike the basic FFO, FFO_LGMS generates a new individual $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ by setting $x_{i,j} = \delta_j + \omega \times rand() \times (UB_j - LB_j)$ for $j = 1, 2, \ldots, n$, where ω is a weight generated by $\omega = \omega_0 \times \alpha^{Iter}$, and ω_0 is the initial weight and α is the weight coefficient. In the experiments, we set $PS = 10$ and $Iter_{\max} = 5000$ for both the FFO and IFFO algorithms. The other parameters of the IFFO are fixed as $\lambda_{\max} = (UB - LB)/2$ and $\lambda_{\min} = 10^{-5}$. For the FFO_LGMS, we set $\omega_0 = 1$, $\alpha = 0.95$ and $PS = 50$ according to Shan et al. [15]. The maximum iteration number of the FFO_LGMS is fixed at 1000. All the three algorithms are implemented using C++ in Microsoft Visual Studio 2010. The algorithms are run on the same Inter (R) Core (TM) i7-2600 CPU @ 3.40 GHz with 8.00 GB RAM in Windows 7 Operating System. Each problem is run 30 independent replications. The median and standard deviations (std) over these 30 replications for the unimodal and multimodal functions are reported in Tables 1 and 2, respectively. The *h* values reported in Tables 1 and 2 are the results of Ranksum tests. An *h* value equal to 1 or −1 indicates that the results obtained by the IFFO is significantly better or worse than those by the FFO, while *h* value equal to zero implies that the results by the two compared algorithms are not statistically different.

As we can see from Table 1, for the 15 unimodal functions with dimensions equal to 30, the IFFO generates 12 significantly better and two significantly worse median values than the FFO, and performs much better than FFO_LGMS for all the functions. With the increase of dimensions, the results yielded by all the algorithms become slightly worse. However, the IFFO outperforms the FFO for 13 functions and FFO_LGMS for 15 functions at dimensions equal to 50. For the 14 multimodal functions in Table 2, it can be also observed that IFFO surpasses FFO/FFO_LGMS at 13/14 functions for dimensions equal to 30 and 50, respectively. Hence, it can be concluded that as a whole the proposed IFFO significantly improves the basic FFO and FFO_LGMS. However, we have to mention that for most of the 29 functions, the IFFO generates small gaps with the true optimal solutions. There is room for the IFFO to be further improved in the future study.

5.2. Comparison of IFFO with harmony search algorithms

This section compares the IFFO with the harmony search (HS) algorithm which is conceptualized by using the improvisation process that occurs when a musician searches for a better state of harmony. In the HS algorithm [18], each solution is called a "harmony" and represented by an *n*-dimension real vector. An initial population of harmony vectors are randomly generated and stored in a harmony memory (HM). Then a new candidate harmony is generated from all of the solutions in the HM by using a memory consideration rule, a pitch adjustment rule and a random re-initialization. Finally, the HM is updated by comparing the new candidate harmony and the worst harmony vector in the HM. The worst harmony vector is replaced by the new candidate vector if it is better than the worst harmony vector in the HM. The above process is repeated until a certain termination criterion is met. For the detail of the HS algorithms, the readers can refer [16–19]. We consider five variants of the HS algorithms, including the basic HS [16], the Improved HS (IHS) [9], the Global-best HS (GHS) [17], the Self-adaptive Global-best HS (SGHS) [18], the local-best harmony search algorithm with dynamic subpopulations (DLHS) [19]. For the algorithm's parameters of the HS variants, we directly take them from the literature.

We code all the five HS variants using C++ in Microsoft Visual Studio 2010 and run them on the aforementioned PC. In the experiments, each problem is run 30 independent replications and each replication is allowed to run 50,000 evaluations of the objective function for both 30-dimensional and 50-dimensional problems. Note that our IFFO algorithm also runs $Iter_{\max} \times PS = 50{,}000$ function evaluations for these benchmark functions. We report the median and standard deviations (std) over these 30 replications and the results of Ranksum tests in Tables 3–6.

It is clear from Tables 3–6 that the IFFO performs steadily well in terms of median values, and outperforms the five HS variants. For the 15 30-deminsional unimodal functions in Table 3, the IFFO produces 14/12/11/9/11 significantly better, and 1/3/3/5/4 equal, and 0/0/1/1/0 significantly worse median values than the HS/IHS/GHS/SGHS/DLHS. For the 15 50-dimensional unimodal functions in Table 4, the IFFO surpasses the HS/IHS/GHS/SGHS/DLHS with providing better solutions at 14/14/11/10/10 functions. For the 15 multimodal benchmark functions, it can be observed from Tables 5 and 6 that the IFFO performs significantly better than the five HS variant regardless of the dimension of decisions involved.

5.3. Typical solution history graph of the IFFO with the competition algorithms

We report the typical solution history graph along the iterations of the compared algorithms in Figs. 4–7. It can be observed that overall the evolution curves of the IFFO descend much faster and reach lower level than that of the five HS variants and the FFO and FFO_LGMS. Thus, it can be concluded that the IFFO significantly outperforms the competition algorithms for function optimization in comparison.

5.4. The effect of population size on the IFFO

This section investigates the effect of *PS* value on the performance of the IFFO. We set *PS* at $PS \in \{5, 10, 15, 20, 30, 40\}$, and fix the other parameters as follow: $\lambda_{\max} = (UB - LB)/2$, $\lambda_{\min} = 10^{-5}$, and $Iter_{\max} = 5000$. We run the IFFO 30 independent replications for each of the 30 benchmark functions with dimension equal to 30 and 50. The median values generated by using different *PS* values for the 15 50-dimensional multimodal functions are summarized in Table 7.

It is can be found from Table 7, that for most benchmark functions, the IFFO with *PS* equal to 5 and 10 is superior to *PS* = 15, 20, 30, and 40, suggesting a small population size is much effective for the IFFO algorithm. The effects of population size on the IFFO algorithm for solving the other functions are very similar.

6. Conclusions

Fruit fly optimization (FFO) is a new evolutionary computation approach. This paper adapts the FFO to high-dimensional continuous function optimization problems and also presents an improved variant. To our best knowledge, this is the first time that the FFO has been proposed for solving high-dimensional functions. To improve the performance of the FFO and eliminate the drawbacks which lie with fixed values of search radius, we change the search radius dynamically with iteration number so that the search radius can be well-adjusted to different evolution phases. Unlike the basic FFO which changes all the decision variants of the swarm location for producing a new solution, we randomly choose one decision variant with uniform distribution to change for enhancing the intensive search. Extensive comparative studies are conducted based on various 30 test functions. The numerical results demonstrate that the proposed IFFO is a powerful search algorithm, and as a whole it performs significantly better than five state-of-the-art harmony search algorithms in solving high-dimensional functions. Our future work will develop a self-adaptive FFO and multiple-population FFO, and generalize the IFFO to solve combinatorial and discrete optimization problems.

Acknowledgements

This research is partially supported by National Science Foundation of China 61174187, Program for New Century Excellent Talents in University (NCET-13-0106), Specialized Research Fund for the Doctoral Program of Higher Education (20130042110035), Science Foundation of Liaoning Province in China (2013020016), Basic Scientific Research Foundation of Northeast University under Grant N110208001, Starting Foundation of Northeast University under Grant 29321006, IAPI Fundamental Research Funds (2013ZCX04-04).

References

[1] W.T. Pan, A new evolutionary computation approach: fruit fly optimization algorithm, in: 2011 Conference of Digital Technology and Innovation Management, Taipei.
[2] W.T. Pan, A new fruit fly optimization algorithm: taking the financial distress model, Knowl.-Based Syst. 26 (2012) 69–74.
[3] 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.
[4] 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.
[5] J. Han, P. Wang, X. Yang, Tuning of PID controller based on fruit fly optimization algorithm, in: International Conference on Mechatronics and Automation (ICMA), 2012, pp. 409–413.
[6] C. Li, S. Xu, W. Li, L. Hu, A novel modified fly optimization algorithm for designing the self-tuning proportional integral derivative controller, J. Converg. Inform. Technol. 7 (2012) 69–77.
[7] L. Wang, X.L. Zheng, S.Y. Wang, A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem, Knowl.-Based Syst. 48 (2013) 17–23.
[8] M.M. Ali, C. Khompatraporn, Z.B. Zabinsky, A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems, J. Glob. Optim. 31 (2005) 635–672.
[9] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.
[10] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102.
[11] 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, May 2005 and KanGAL Report #2005005, IIT Kanpur, India.
[12] F. Kang, J. Li, Z. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, Inform. Sci. 181 (2011) 3508–3531.
[13] D. Karaboga, B. Akav, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[14] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (2008) 64–79.
[15] D. Shan, G. Chao, H. Dong, LGMS_FOA: an improved fruit fly optimization algorithm for solving optimization problems, Math. Prob. Eng. 2013, Article ID 108768.
[16] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization, harmony search theory and practice, Comput. Meth. Appl. Mech. Eng. 194 (2005) 3902–3933.
[17] M.G.H. Omran, M. Mahdavi, Global-best harmony search, Appl. Math. Comput. 198 (2008) 643–656.
[18] Q.K. Pan, P.N. Suganthan, M.F. Tasgetiren, A self-adaptive global best harmony search algorithm for continuous optimization problems, Appl. Math. Comput. 206 (2010) 830–848.
[19] Q.K. Pan, P.N. Suganthan, J. Liang, M.F. Tasgetiren, A local-best harmony search algorithm with dynamic subpopulations, Eng. Optim. 42 (2010) 101–117.