IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received October 5, 2019, accepted November 1, 2019, date of publication November 5, 2019, date of current version November 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951716

Improved Black Hole Algorithm Based on Golden Sine Operator and Levy Flight Operator

W. XIE[1], J. S. WANG[1,2], (Member, IEEE), AND Y. TAO[1]
[1]School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China
[2]National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning, Anshan 114051, China

Corresponding author: J. S. Wang (wang_jiesheng@126.com)

This work was supported in part by the Project by National Natural Science Foundation of China under Grant 21576127, in part by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province under Grant 2017FWDF10, and in part by the Project by Liaoning Provincial Natural Science Foundation of China under Grant 20180550700.

ABSTRACT The black hole (BH) algorithm is a new type of natural heuristic algorithm inspired by the movement law of the ‘‘black hole’’ celestial body in the universe. BH algorithm has received extensive attention due to its advantages such as fewer parameters, simple algorithm structure and strong exploitation. For the shortcomings of poor exploaration and premature convergence of BH algorithm, the improved golden sine (G-S) operator is introduced into BH algorithm to greatly improve the exploaration. Then the Levy flight operator with controlled step size has become a better local search operator from the global search operator, so the improved Levy flight operator is introduced to further improve the exploitation of BH algorithm. Ultimately, the golden sine operator and Levy flight operator based black hole (GSLBH) algorithm is able to balance the exploaration with exploitation. GSLBH, BH, Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WAO), Firefly Algorithm (FA), Golden Sine Algorithm (Gold-SA) were adopted to carry out the simulation experiments with 17 benchmark functions, respectively, and the statistical data results are analyzed and compared. Finally, it can be concluded that the proposed improved black hole algorithm has better exploaration, and the convergence speed and accuracy of the algorithm have been further improved.

INDEX TERMS Black hole algorithm, golden sine operator, levy flight operator, function optimization.

I. INTRODUCTION

As the complexity of various optimization problems in the real world continues to increase, traditional mathematical methods are far from achieving the goal of solving these problems. The natural heuristic algorithms can effectively solve many optimization problems encountered in the actual process, which makes the natural heuristic algorithm become a research hot spot in recent years [1]. Natural heuristics are often a series of new calculations inspired by inspiration from various behaviors, phenomena or processes in nature [2]. The earliest natural heuristic algorithm, genetic algorithm (GA) was proposed by Professor Holland in 1975 [3]. After that, many experts and scholars proposed a series of natural heuristic algorithms through observation and research on various natural phenomena. The ant colony optimization (ACO) algorithm was proposed by Dorigo in 1992. It is well suited to solve large-scale distributed optimization problems without the need to provide a global model or centralized control [4]. In 1995, Dorigo proposed a particle swarm optimization (PSO) algorithm through the study of bird foraging behavior, which is one of the most widely used natural heuristic algorithms [5], [6]. The artificial fish swarm algorithm (AFSA) is a heuristic algorithm proposed by Xiaolei Li et al. in 2002 based on the foraging, clustering and rear-end behavior of fish schools [7]. In 2002, K.M. Passino proposed a bacterial foraging algorithm (BFA) by simulating the behavior of Escherichia coli phagocytizes food in the human intestine [8]. According to the behavior of bees looking for honey sources, Karaboga proposed the artificial bee colony (ABC) algorithm in 2005 [9]. The firefly algorithm (FA) is an algorithm proposed by Yang X. S. in 2008 to solve the function optimization problem [10]. In 2012, Yang et al. proposed a flower pollination algorithm (FPA) in the process of imitating flower pollination [11]. In 2013, Salcedo-Sanz, S et al. proposed the coral reef optimization (CRO) algorithm inspired

The associate editor coordinating the review of this manuscript and approving it for publication was M. Venkateshkumar.

by the behavior of coral herds and coral reefs [12]. In 2015, Mirjalili et al. proposed the ant lion optimization (ALO) algorithm based on the behavior of ant lions preying on ants [13]. According to the characteristics of whale predation behavior, Lewis A. et al. proposed the whale optimization algorithm (WOA) in 2016 [14].

The function optimization problem is a very important type optimization problem, which is to find its maximum or minimum value in the domain of the function. For some low-dimensional functions with simple structure, they can be optimized by traditional mathematical methods, but most of the functions that need to be optimized have the characteristics of complex structure and high dimension, traditional mathematical methods often cannot solve them. Natural heuristic algorithms can replace these mathematical methods to optimize these complex functions and achieve better results. Although a single natural heuristic algorithm can complete the function optimization problem independently, the result of the optimization is often not accurate enough, and even the global optimal solution cannot be found, and it is easy to fall into the local optimum. In order to maintain a balance between the exploitation and the exploaration, many experts and scholars propose improved strategies for the natural heuristic algorithm. Yi Liu initialized the PSO algorithm by chaos optimization, which enhanced the exploaration of the PSO algorithm and increased the convergence speed of the algorithm [15]. Peng Y et al. introduced information entropy into ABC so as to improve the convergence speed and convergence precision of the algorithm [16]. Wang Jing et al. added an adaptive step size strategy to the FA to further enhance exploaration of the FA [17]. Wang X G et al. proposed a WOA based on adaptive weighting and simulated degradation, which allows WOA to easily jump out of local optimal values and accelerate the convergence of the algorithm [18]. Wen Long et al. introduced a modified parameter "C" strategy in the gray wolf optimizer (GWO), which increased the exploaration of the algorithm [19]. Singh D et al. improved the exploitation of flower pollination algorithm (FPA) by using improved distribution parameters [20]. Wen L et al. introduced a nonlinear convergence factor into WOA, reducing the probability that the algorithm falls into local optimum [21].

The black hole (BH) algorithm is a new method proposed by Hatampou et al. to solve the problem of data clustering [22]. It is a new algorithm proposed to simulate the characteristics of "black holes" in the universe, that is to say the celestial bodies near the "black holes" will be collapsed down by "black holes". Then this new natural heuristic algorithm was then further standardized [23]. The execution mechanism of the algorithm mainly includes two parts. The first part is that all search agents in the solution space move to the current global optimal position according to the rules of the black hole algorithm. The second part is that the search agents which enters the " event horizon " of the black hole center will be collapsed down by the "black hole", that is to say that the collapsed search agents disappear from the solution space. In order to ensure that the population of search agents in the solution space is constant, a corresponding number of search agents are randomly generated at any position in the solution space to continue searching for the optimal solution. The algorithm has attracted a lot of attention from many experts and scholars because of its simple structure, few parameters, high local search accuracy and fast convergence [24]–[29]. For the function optimization problem, BH algorithm can optimize some simple low-dimensional functions and achieve better results, but for most high-dimensional complex functions, BH algorithm shows poor performance, is easy to fall into the local optimum, and the algorithm will premature convergence. In order to solve these defects of BH algorithm, many experts and scholars have improved it. Saber Yaghoobi et al. introduced a random vector in search agents movement formula of BH algorithm, which increased the search space of the population, and introduced the hybridization strategy in GA, which increased the diversity of the population [30]. These operations have greatly improved the exploaration and reduced the possibility of search agents falling into local optimum of BH algorithm. Hamid Aslani et al. used chaos theory to establish a chaotic inertia weighting mechanism and applied it to BH algorithm (CIWBH) [31], which increased the exploaration of the algorithm. Wang Tong et al. used Euclidean distance to initialize the population and improve its exploaration and set the upper bound of the black hole radius to avoid the search agents skipping the global optimal solution due to the excessive black hole area [32]. Chen Min-You et al. combined the mechanism of PSO algorithm and BH algorithm and proposed RBH-PSO algorithm to improve the convergence speed and accuracy of the algorithm [33]. Therefore, it can be seen that most scholars improve the BH algorithm by improving its exploaration.

Levy flight is a way of animal motion based on Levy's distribution theory. Studies have shown that many animals and insects follow the typical characteristics of the Levy distribution [34]. As a global search operator, Levy flight is first applied to the cuckoo searching (CS) algorithm, so the cuckoo algorithm has a strong exploaration [35]. Since then, many experts and scholars have applied Levy flight operator to other algorithms to enhance their exploaration [36]–[39]. The golden sine algorithm (Gold-SA) is a mathematical heuristic algorithm proposed by Erkan T. et al. in 2017 inspired by the periodic variation of the sine function [40]. When the independent variable of the sine function changes, the function value will change periodically. This change corresponds to the ordinate change of the point on the unit circle. The phenomenon that the point is continuously scanned on the unit circle is equivalent to the process that the search agents in the natural heuristic algorithm continuously search for the optimal solution in the solution space. Gold-SA can traverse all the values on the sine function according to the relationship between the sine function and the unit circle, that is to say that it finds all the points on the unit circle and the space of the golden section is reduced in the location update process to scan the area that may only produce good results at the same time. Thus, these operations greatly improve the search speed and achieve a good balance between exploaration and exploitation [41].

This paper firstly proposed an improved Gold-SA operator into the standard BH algorithm, which greatly improves the exploaration of BH algorithm, and then proposed a Levy flight operator with control step size, which further improves the exploitation of the algorithm. The improved black hole algorithm based on golden sine operator and levy flight operator (GSLBH) maintains a balance between exploaration and exploitation, and the function optimization ability of the algorithm has been greatly improved. Finally, the performance of the improved algorithm is verified by 17 benchmark functions, and the experimental results are compared with black hole (BH) algorithm, particle swarm optimization (PSO) algorithm, whale optimization (WAO) algorithm, firefly algorithm (FA) and golden sine algorithm (Gold-SA). Experimental results show that the improved algorithm is feasible and efficient. The structure of this paper is organized as follows. The second part introduces the basic black hole (BH) algorithm. The third part introduces the improvement process of GSLBH algorithm. The fourth part gives the simulation results of the GSLBH algorithm to optimize the benchmark functions. The fifth part draws the conclusion.

II. BASIC PRINCIPLE OF BLACK HOLE ALGORITHM

A. "BLACK HOLE" PHENOMENON

Large celestial bodies in the universe can be roughly divided into two kinds of stars and planets. The mass and volume of a star are generally several times or even hundreds of times larger than that of a planet, which makes the star have a strong gravitational force. The planets around the stars are subjected to powerful gravitational force of the stars, causing them to make circular motions around the stars. According to the knowledge of astronomy, after the nuclear energy in the star is completely released, its volume will be drastically reduced due to its powerful gravitational force, eventually forming a "singularity" with infinitesimal volume and infinite mass, this "singularity" is called "black hole". The "black hole" has a very strong gravitational force, so that the light it launches cannot escape, and make a circular motion around the "black hole" [42]. When the celestial bodies around the "black hole" are close to the "black hole", these celestial bodies will be sucked into the "black hole" and eventually become part of the "black hole". This phenomenon is called the process of "black holes" collapsing down other celestial bodies. The gravitational effect of "black holes" also has a certain range of restrictions. The range of gravitational effects of "black holes" is called "event horizon" or "Schwarzschild radius", denoted as R_s, which can be expressed as:

$$R_s = \frac{2GM}{c^2} \tag{1}$$

where, G is the universal gravitational constant, M is the mass of the "black hole", and c is the speed of light. When the distance from the body to the "black hole" is less than R_s, the celestial body will be collapsed down by the "black hole". The black hole model in the universe is shown in Fig. 1 [43]. It can be seen from Fig. 1 that the R_s of the black hole is ellipsoidal rather than spherical.

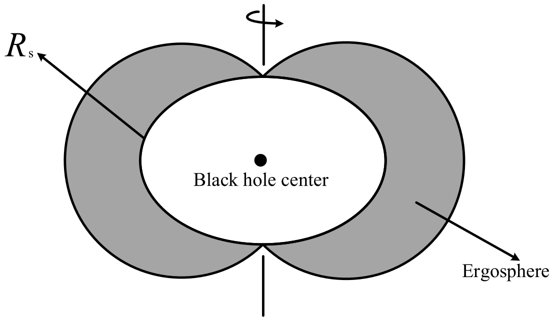


FIGURE 1. "Black hole" model figure.

B. BLACK HOLE ALGORITHM

The black hole algorithm is a natural heuristic algorithm by simulating the "black hole" phenomenon in the universe. If n agents $X = (x_1, x_2, \ldots, x_n)$ are allocated in the solution space and the solution space is p-dimensional, the position of each agent can be expressed as $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}), i = 1, 2, \ldots, n$, the current optimal position is $G_{best} = (g_1, g_2, \ldots, g_p)$, that is to say G_{best} is the "black hole center", and the formula for moving the agents x_i to the optimal position is expressed as:

$$x_i(t+1) = x_i(t) + rand \times (G_{gest} - x_i(t)), \quad G_{gest} \neq x_i(t), \quad i = 1, 2, \ldots, n \tag{2}$$

where, $x_i(t)$ is the current position of agent x_i, $x_i(t+1)$ is the position of agent x_i moving at the next iteration, $rand$ is a random number between (0, 1), and G_{gest} is the global optimal position. All particles in the solution space move to the current global optimal position according to Eq. (2) and the process of particle movement is shown in Fig. 2.

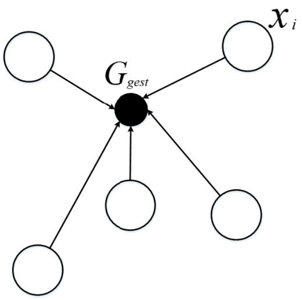


FIGURE 2. Schematic diagram of agent movement to a black hole.

The black hole algorithm also simulates the "black hole" phenomenon in the universe and defines an

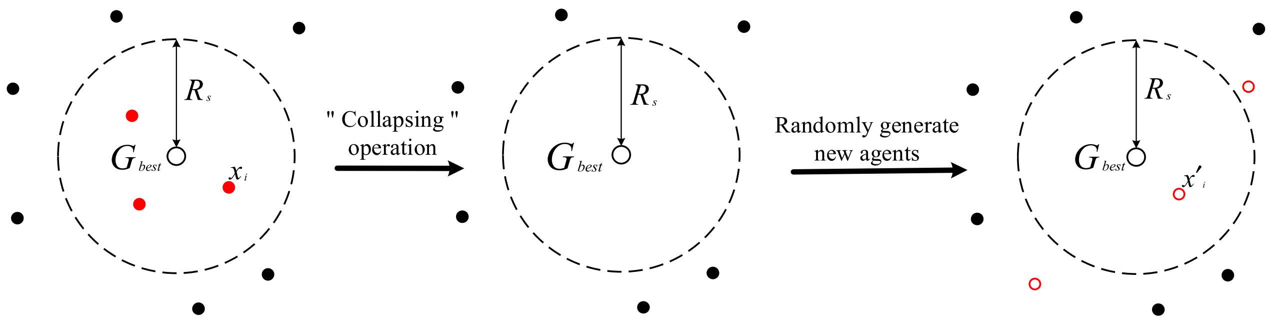


FIGURE 3. Schematic of the " collapsing " operation.

"event horizon" R_S. R_S in the black hole algorithm can be defined as:

$$R_s = \frac{|f_{BH}|}{\sum_{i=1}^{n} |f_i|} \tag{3}$$

where, $|f_{BH}|$ is the current global optimal position, G_{gest} corresponds to the absolute value of the function fitness value, and $|f_i|$ is the absolute value of the function fitness value of the agent x_i. The distance from a particle x_i to the center G_{gest} of a black hole is expressed by the Euclidean distance shown in Eq. (4).

$$d_i = \sqrt{\sum_{m=1}^{p} (g_m - x_{im})^2}, \; i = 1, 2, \ldots, n; \; m=1, 2, \ldots, p \tag{4}$$

where, d_i is the distance of the agent x_i to G_{gest}, g_m is the coordinate value corresponding to the m-th dimension of G_{gest}, and x_{im} is the coordinate value corresponding to the m-th dimension of the agent x_i.

After the agent x_i completes a movement by using Eq. (2), it is necessary to judge the size of d_i and R_S. When d_i is greater than R_S, the agent x_i performs the next operation. When d_i is smaller than R_S, the agent x_i is "collapsed down" by the black hole. The agent x_i "disappears" from the solution space, and the agent is discarded, and the number of agents that are collapsed down is k. In order to ensure that the number of populations in the solution space is constant, it is necessary to add agents equal to the number of agents that are " collapsed down". The black hole algorithm uses an operation of re-allocating k agents $x'_i = (x'_{i1}, x'_{i2}, \ldots, x'_{1p}), i = 1, 2, \ldots, k$ randomly at any position in the solution space. The schematic diagram of the black hole algorithm for "collapsing" operation is shown in Fig. 3. When the algorithm is executed, the "collapsing" operation guarantees the diversity of the population in the later stage to a certain extent. The operation flow chart of the algorithm is shown in Fig. 4.

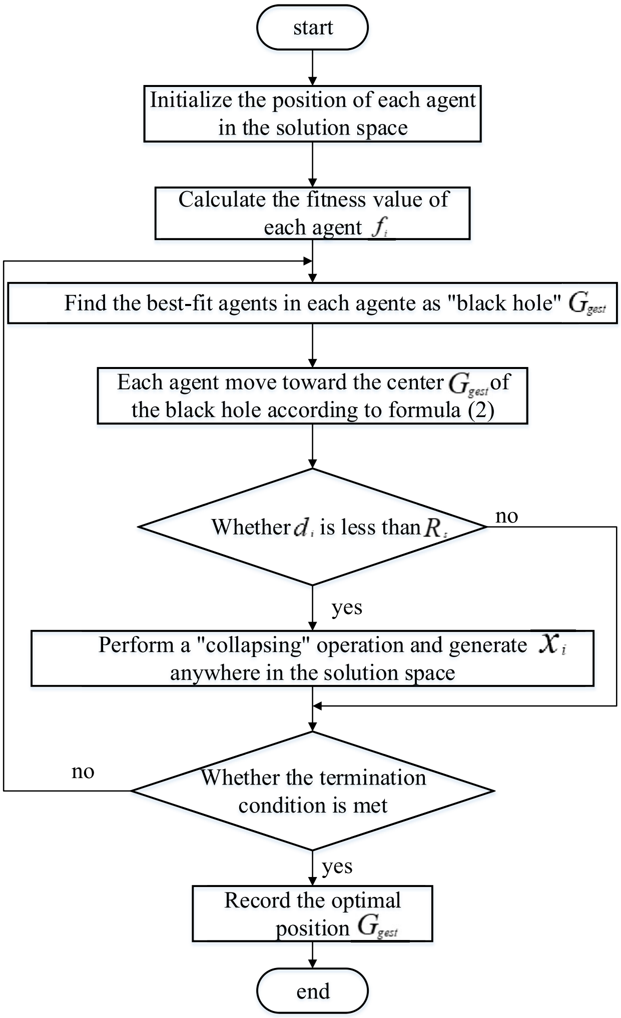

start
Initialize the position of each agent in the solution space
Calculate the fitness value of each agent f_i
Find the best-fit agents in each agente as "black hole" G_{gest}
Each agent move toward the center G_{gest} of the black hole according to formula (2)
Whether d_i is less than R_s
no
yes
Perform a "collapsing" operation and generate X_i anywhere in the solution space
no
Whether the termination condition is met
yes
Record the optimal position G_{gest}
end

FIGURE 4. Flow chart of black hole algorithm.

III. IMPROVED BLACK HOLE ALGORITHM BY INTRODUCING GLOBAL SEARCHING OPERATORS

A. LEVY FLIGHT

Levy flight was proposed by the French mathematician Paul Levy. Since then, many scholars have found that the predation and hunting methods for most of the creature are in line with the Levy flight process. And it is widely believed that Levy flight can improve the efficiency and accuracy of predation. The mechanism of Levy flight is long-term and short-distance migration, which increases the diversity of the population. When the Levy flight occasionally jumps over long distances,

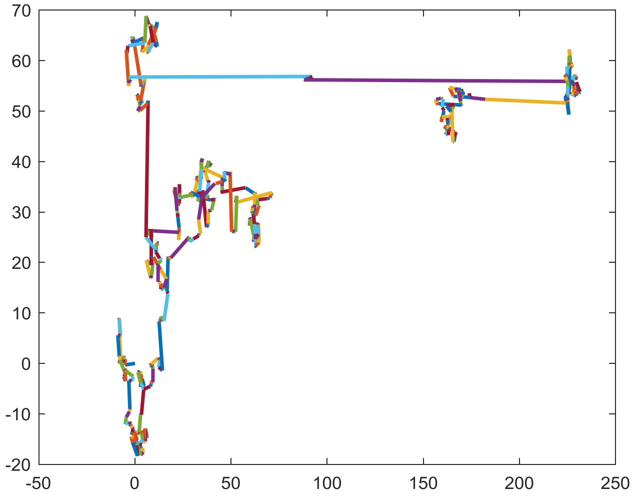


FIGURE 5. Levy flight path of 500 times movements in a 2-dimensional space.

the variability of the direction guarantees a detailed search of the population for nearby areas. The organic combination of short-distance and long-distance jumps greatly enhances the exploaration of the population. Fig. 5 shows the Levy flight motion trajectory that is continuously moved 500 times in a 2-dimensional space, and the characteristics of the Levy flight mechanism described above can be observed from the trajectory.

Each step of Levy flight consists of two control factors. One is the direction of flight, which generally selects the uniform distribution function, and the other is the step size to obey the Levy distribution, where the general step size is Manantea. So the step size can be calculated by:

$$s = \frac{u}{|v|^{1/\beta}} \tag{5}$$

where, both u and v follow Gaussian distribution shown as:

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \tag{6}$$

where

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\beta\, 2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{7}$$

$$\sigma_v = 1 \tag{8}$$

where, β is generally 1.5, and Γ is a standard *Gamma* function, whose complete expression is expressed as:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t}\, dt \tag{9}$$

When $z = n$ is an integer, $\Gamma(n) = (n-1)!$. For s, it obeys the Levy distribution of $|s| \geq |s_0|$, where s_0 is the minimum step size. In principle, $|s_0| \gg 0$, but in the actual application process, the size of s_0 generally takes between 0.1 and 1. Studies have shown that the Levy flight mode can maximize the efficiency of the search target under uncertainty [44]. When performing function optimization, the search formula for Levy flight is described as:

$$x_i(t+1) = x_i(t) + \alpha \otimes s \tag{10}$$

where, $x_i(t+1)$ is the population position after the Levy flight operation, $x_i(t)$ is the current population position, α is a random number subject to uniform distribution, s is the step vector of Levy flight, and $\otimes$ is the multiplication between elements and elements.

B. GOLDEN SINE ALGORITHM (GOLD-SA)

The golden sine algorithm (Gold-SA) is a mathematical heuristic algorithm proposed by Erkan T. et al. in 2017, which is inspired by the periodic variation of the sin function in trigonometric functions, where agents search the solution space according to the golden sine line. As shown in Fig. 6, the definition of the trigonometric function shows that the function value y_0 of the sin function corresponds to the ordinate of the point on the unit circle, and the angle value of θ is a multiple of the period (2π) of the abscissa value of the point on the sin function. When the argument x of the sin function changes, the function value y also changes with the corresponding function relationship. This change of the corresponding relationship is equivalent to the continuous scanning the unit circle by the points on the unit circle.

The phenomenon that the points on the unit circle are continuously scanned on the circle is similar to the process in which the agents continuously search for the optimal solution in the solution space. This similarity promotes the development of Gold-SA. The motion trajectory of the agent searching the optimal solution is shown in Fig. 7 (red curve).

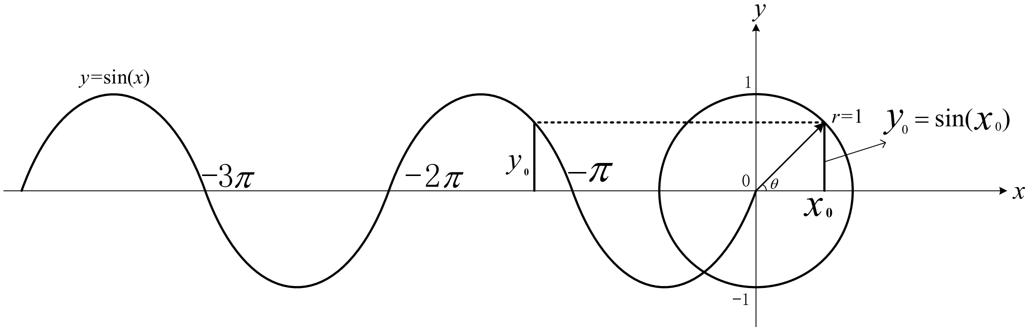


FIGURE 6. Correspondence diagram between sine function and unit circle.

TABLE 1. Parameter settings for adopted optimization algorithms.

Algorithm	Main parameters settings
PSO	Particle number n = 40 ; Learning factor $c_1 = 2$, $c_2 = 2$; Inertia weight $w_{Max}=0.9$, $w_{Min}=0.1$
FA	population size n = 40 ; $\beta_0=1$; $\gamma=1$
WOA	population size n = 40
Gold-SA	population size n = 40
BH	population size n = 40
GSLBH	population size n = 40

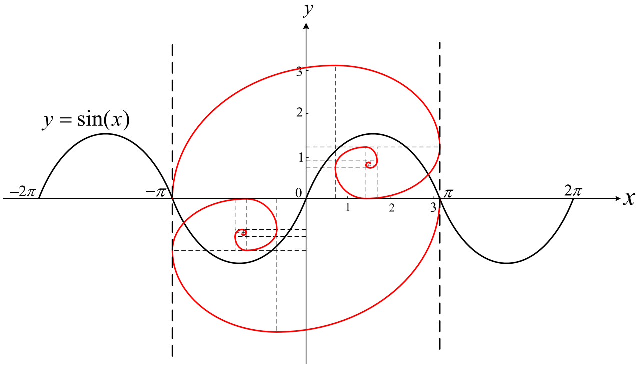


FIGURE 7. Schematic diagram of the movement of agents in Gold-SA.

The movement of agent x_i in the algorithm can be described as:

$$x_i(t+1)=x_i(t)*|\sin(r_1)|-r_2*\sin(r_1)*|m_1*D-m_2*x_i(t)| \tag{11}$$

where, $x_i(t+1)$ is the position after agent x_i completes a search, $x_i(t)$ is the current position of the agent, r_1 is a random number between $[0, 2\pi]$, and r_2 is a random number between $[0, \pi]$. D is the target position and it is the global optimal position in Gold-SA. m_1 and m_2 are coefficients obtained by the golden section method, which can reduce the search space of the agents and improve the search efficiency of the agents. The coefficients allow the agents to move from the current position to the target position.

The key to determining whether the performance of the heuristic algorithm is efficient is whether the algorithm can effectively traverse all locations in the solution space where there may be global optimal solutions. However, the excessive searching space leads to the efficiency of algorithm execution, which is the main problem faced by heuristic algorithms. Whether it can effectively reduce the algorithm searching space while ensuring that the search ability of the algorithm is not reduced becomes the key to solve the problem. Based on this idea, Gold-SA uses a golden section search (GSS) to enable agents in the solution space to search by the best possible path. GSS is an optimization method that can find the maximum or minimum value of simple unimodal function. The name of this method is derived from the golden ratio applied to the method, the value of this ratio is represented as

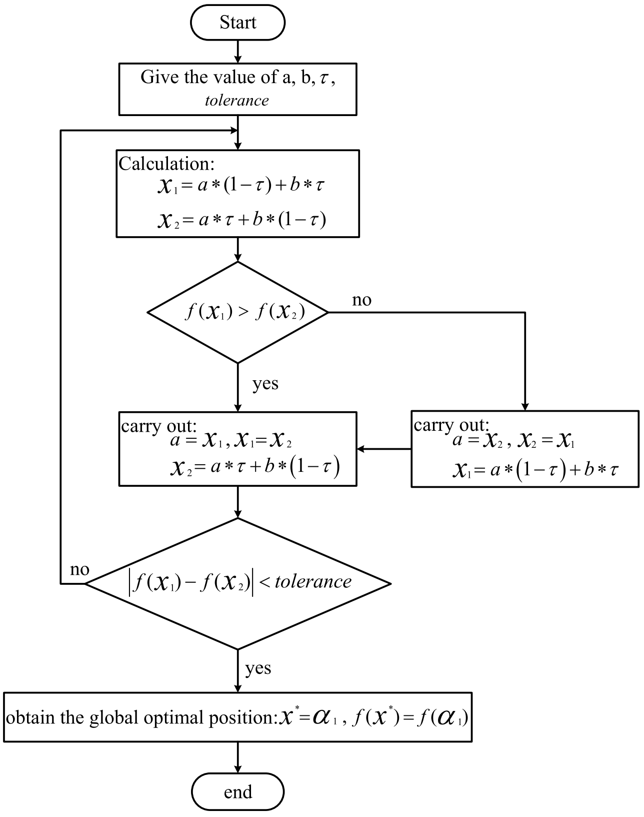

Start
Give the value of a, b, τ, tolerance
Calculation:
$x_1 = a*(1-\tau)+b*\tau$
$x_2 = a*\tau+b*(1-\tau)$
$f(x_1) > f(x_2)$
no
yes
carry out: $a = x_1, x_1 = x_2$
$x_2 = a*\tau+b*(1-\tau)$
carry out: $a = x_2, x_2 = x_1$
$x_1 = a*(1-\tau)+b*\tau$
no
$|f(x_1)-f(x_2)| < tolerance$
yes
obtain the global optimal position: $x^* = \alpha_1$, $f(x^*) = f(\alpha_1)$
end

FIGURE 8. Flowchart of GSS operation.

$$\tau = \frac{1-\sqrt{5}}{2} \tag{12}$$

GSS does not require much gradient information, and it has two advantages over other search methods [45]: 1) Only one new function evaluation is required for each step; 2) There is a continuous reduction factor at each step. Suppose there is a 2-dimensional function $f(x)$. In the 2-dimensional search

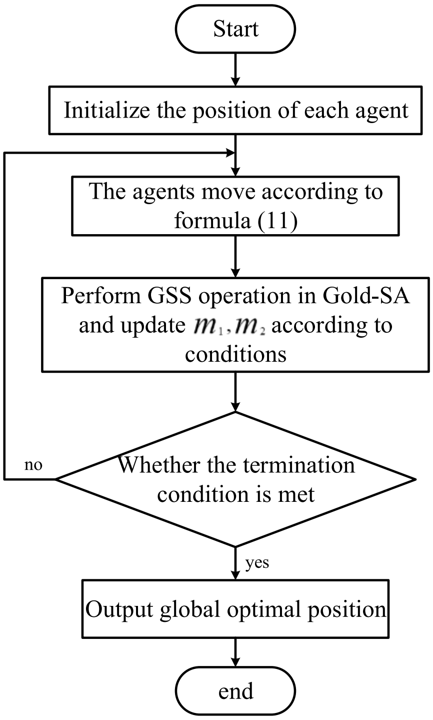

Start
Initialize the position of each agent
The agents move according to formula (11)
Perform GSS operation in Gold-SA and update m_1, m_2 according to conditions
Whether the termination condition is met
no
yes
Output global optimal position
end

FIGURE 9. Flow chart of Gold-SA.

space, the search range of GSS is $[a, b]$, then the operation flow chart of GSS is shown in Fig. 8. In Fig. 8, τ is the golden ratio in Eq. (12).

When the idea of GSS is introduced into Gold-SA, the representation of coefficients m_1 and m_2 is similar to GSS:

$$m_1 = a * \tau + b * (1 - \tau) \tag{13}$$

$$m_2 = a * (1 - \tau) + b * \tau \tag{14}$$

When applying GSS to Gold-SA, the initial values of a and b are $-\pi$ and π, respectively, and the coefficients m_1 and m_2 will change with the change of the target value. The specific process of the change is shown in the following pseudo code.

```
Step1: $if\ \ CurrentValue < TargetValue$
        $b = m_2, \quad m_2 = m_1$
        $m_1 = a * \tau + b * (1 - \tau)$
    $else$
        $a = m_1, \quad m_1 = m_2$
        $m_2 = a * (1 - \tau) + b * \tau$
Step2: $if\ m_1 == m_2$
        $a = random1, \quad b = random2$
        $m_1 = a * \tau + b * (1 - \tau)$
        $m_2 = a * (1 - \tau) + b * \tau$
```

In the above pseudo code, *CurrentValue* is the current position coordinate value of the agent, *TargetValue* is the coordinate value of the ideal position, and *random*1 and *random*2 are random numbers between $[0, \pi]$ and $[-\pi, 0]$, respectively, τ is the golden ratio in Eq. (12), and m_1 and m_2 need to be recalculated according to different actual conditions. The complete Gold-SA flowchart is shown in Fig. 9.

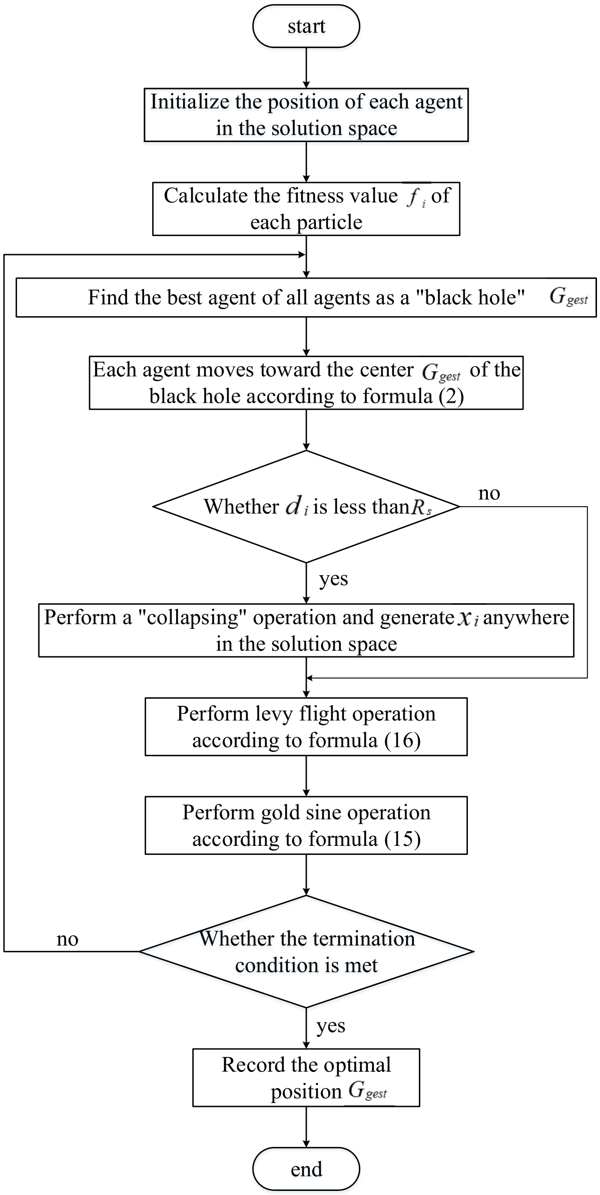

start
Initialize the position of each agent in the solution space
Calculate the fitness value f_i of each particle
Find the best agent of all agents as a "black hole" G_{gest}
Each agent moves toward the center G_{gest} of the black hole according to formula (2)
Whether d_i is less thanR_s
no
yes
Perform a "collapsing" operation and generate x_i anywhere in the solution space
Perform levy flight operation according to formula (16)
Perform gold sine operation according to formula (15)
Whether the termination condition is met
no
yes
Record the optimal position G_{gest}
end

FIGURE 10. Flowchart of GSLBH algorithm.

C. IMPROVED BLACK HOLE ALGORITHM

In the proposed improved BH algorithm, an improved golden sine (GS) operator is first introduced, which is shown as:

$$x_i(t+1) = x_i(t) * |\sin(r_1)| - r_2 * \sin(r_1) * |m_1 * D - m_2 * x_i(t)| \tag{15}$$

where, $m_1 = -\pi + (1 - \tau) * \pi$, $m_2 = -\pi + \tau * \pi$.

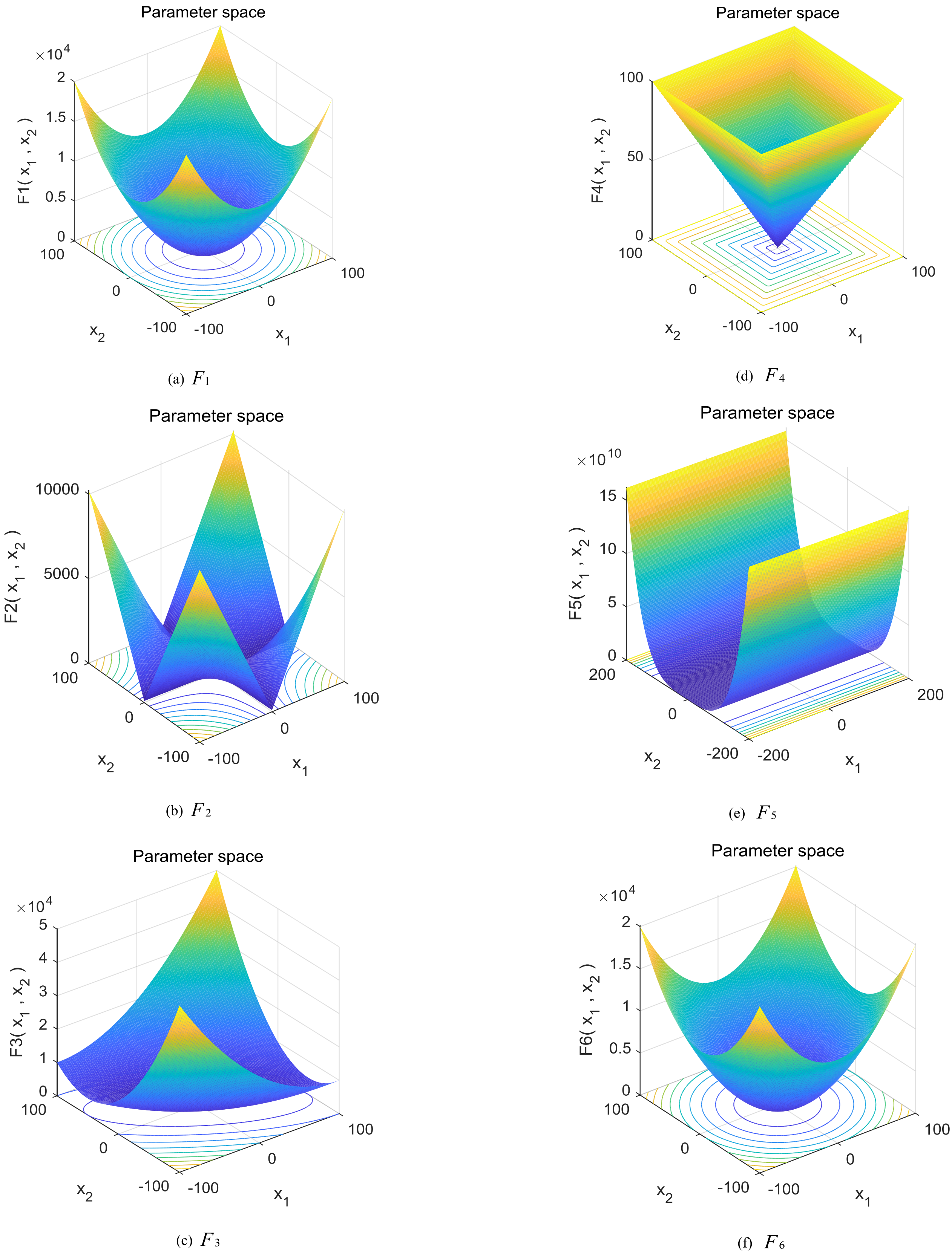
The meaning of other parameters has the same meaning as the parameters in Eq. (11) in Gold-SA. Here m_1 and m_2

TABLE 2. Benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30,100	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30,100	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{j} x_j\right)^2$	30,100	[-100,100]	0
$F_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30,100	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	30,100	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30,100	[-100,100]	0
$F_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	30,100	[-1.28,1.28]	0
$F_8(x)=\sum_{i=1}^{n}-x_i^2\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30,100	[-500,500]	-418.9829× dim
$F_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30,100	[-5.12,5.12]	0
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30,100	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30,100	[-600,600]	0
$F_{12}(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]\right\}$ $+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4},u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	30,100	[-50,50]	0
$F_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]$ $+(x_n-1)^2\left[1+\sin^2(3\pi x_n)\right]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	30,100	[-50,50]	0
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x)=-\sum_{i=1}^{5}\left[(X-a)_i(X-a)_i^2+c_i\right]^{-1}$	4	[0,10]	-10.1532
$F_{16}(x)=-\sum_{i=1}^{7}\left[(X-a)_i(X-a)_i^2+c_i\right]^{-1}$	4	[0,10]	-10.4028
$F_{17}(x)=-\sum_{i=1}^{10}\left[(X-a)_i(X-a)_i^2+c_i\right]^{-1}$	4	[0,10]	-10.5363
$F_{18}(x)=x_1^2+10^6\sum_{i=2}^{n}x_i^2$	30,100	[-10,10]	0
$F_{19}(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	30,100	[-100,100]	0
$F_{20}(x)=\sin^2(\pi w_1)+\sum_{i=1}^{n-1}(w_i-1)^2\left[10+\sin^2(\pi w_i+1)\right]+(w_n-1)^2\left[1+\sin^2(2\pi w_n)\right]$ $w_i=1+\frac{x_i-1}{4},\quad i=1,\ldots,n$	30,100	[-10,10]	0
$F_{21}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30,100	[-32,32]	0
$F_{22}(x)=\left[\frac{1}{n-1}\sum_{i=1}^{n-1}\left(\sqrt{s_i}\left(\sin\left(50.0s_i^{0.2}\right)+1\right)\right)\right]^2\quad s_i=\sqrt{x_i^2+x_{i+1}^2}$	30,100	[-10,10]	0

are set as constant because BH algorithm itself has a certain exploaration. The change of m_1 and m_2 in the process of algorithm execution will cause interference to the normal search of BH algorithm, which will affect the searching accuracy of the algorithm. Therefore, constants are used for m_1 and m_2 in this paper.

In this paper, the introduction of the improved Levy flight operator based on the introduction of the golden sine (GS)



(a) F_1

(d) F_4

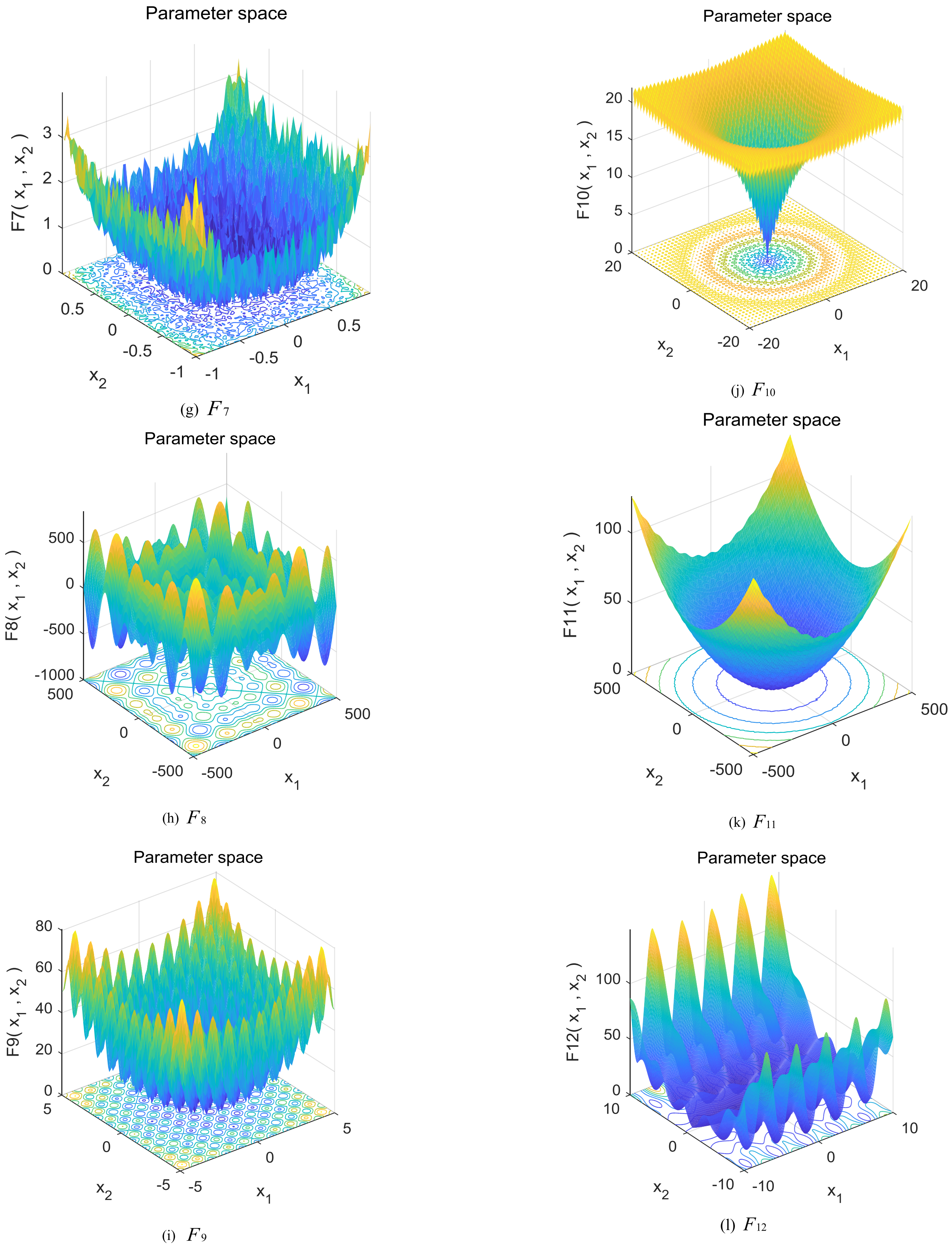
(b) F_2

(e) F_5

(c) F_3

(f) F_6

FIGURE 11. **2-dimensional figures of the adopted benchmark functions.**



(g) F_7

(j) F_{10}

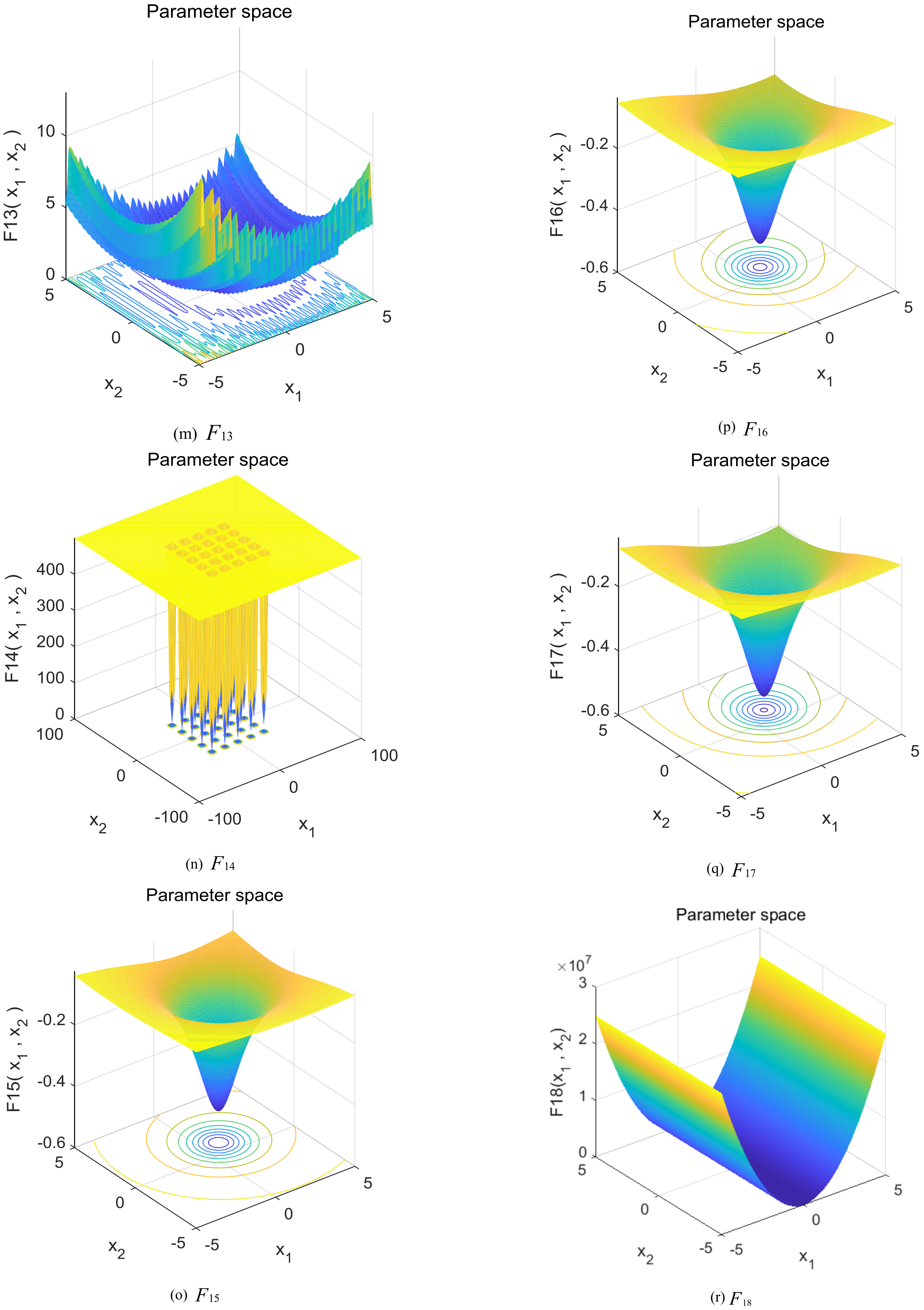
(h) F_8

(k) F_{11}

(i) F_9

(l) F_{12}

FIGURE 11. ***(Continued.)*** **2-dimensional figures of the adopted benchmark functions.**



(m) F_{13}

(p) F_{16}

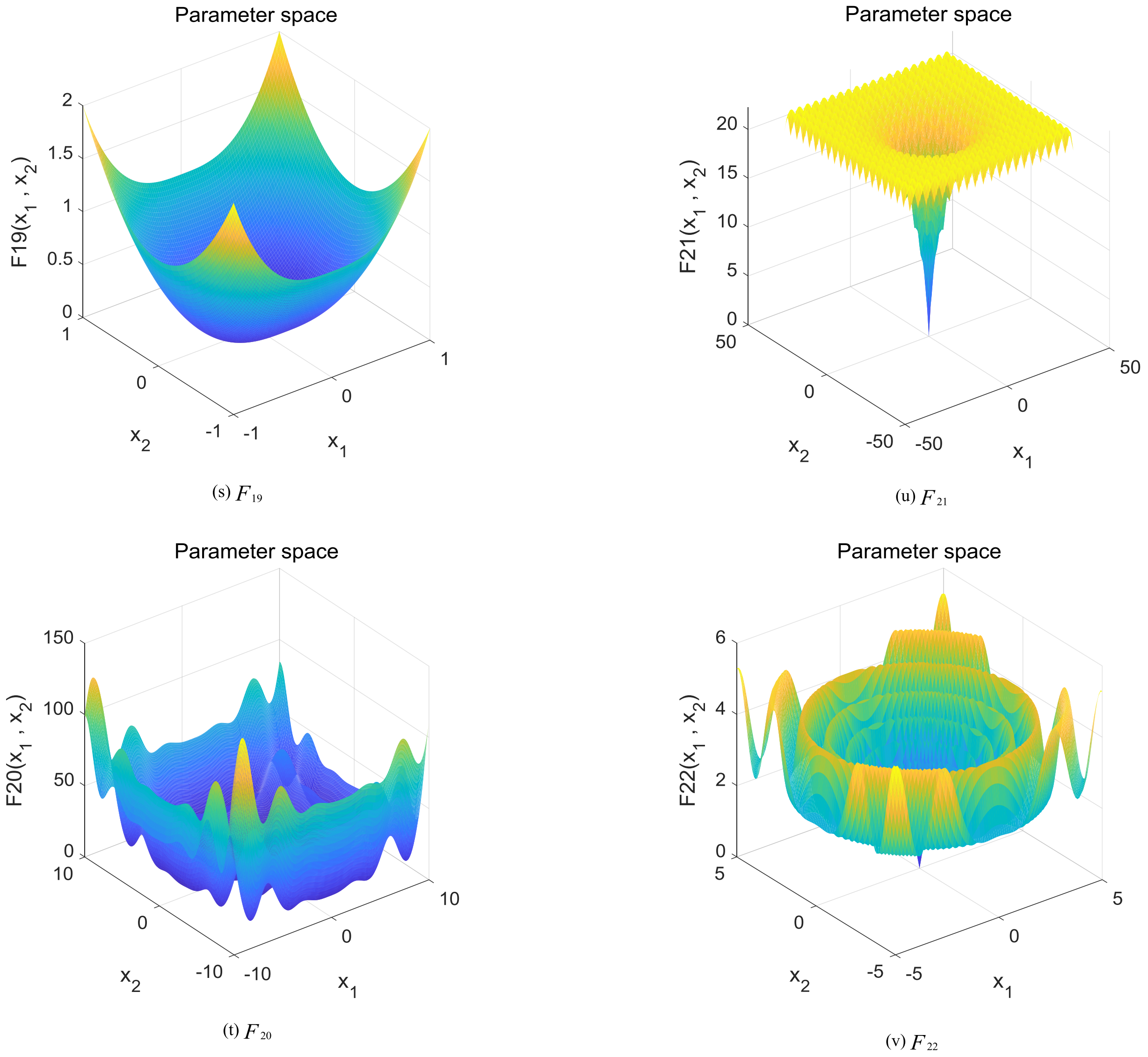
(n) F_{14}

(q) F_{17}

(o) F_{15}

(r) F_{18}

FIGURE 11. ***(Continued.)*** **2-dimensional figures of the adopted benchmark functions.**



(s) F_{19}

(u) F_{21}

(t) F_{20}

(v) F_{22}

FIGURE 11. ***(Continued.)*** **2-dimensional figures of the adopted benchmark functions.**

operator enables the algorithm (GSLBH) to maintain a balance between exploaration and exploitation. The improved Levy operator is a Levy flying operator based on the symbol function *sign* and a smaller step size.

The introduction of the *sign* symbol function allows Levy to traverse to more positions in the solution space. Reducing the step size of Levy flight allows the agent to better traverse a local range in the solution space and enhance the exploitation of the algorithm. The improved Levy operator is shown in Eq. (16).

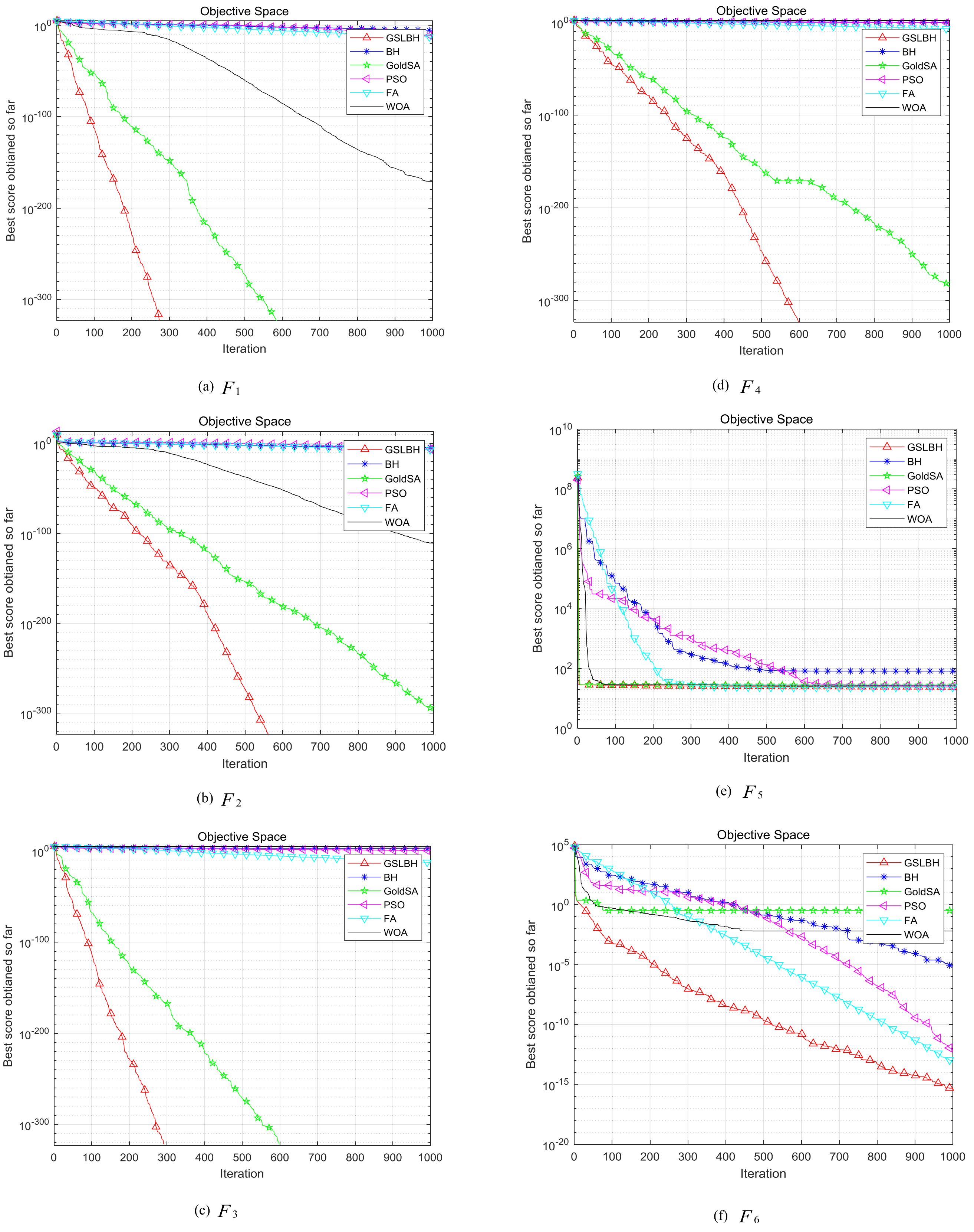
$$x_i(t+1) = x_i(t) + 0.015 * \text{sign}(rand - 1/2) \otimes s \quad (16)$$

where, 0.015 is a step control factor that ensures that the agents are searched in a smaller range; the value of $\text{sign}(rand - 1/2)$ is three random numbers, and their values are -1, 1 and 0, respectively; s has the same meaning as s in Eq. (10) and is a step vector of levy flight; $\otimes$ represents the multiplication between elements.

At the same time, the idea of " selective movement " is also introduced in this algorithm. After the searching agents in the solution space perform a golden sine (GS) operation or an levy flight operation, each agent is compared with the agent position of the previous iteration. If the position after the agent moves is better than the position before the movement, the agent stays at the current position; Otherwise the agent returns to the position before this movement. The operation flowchart of GSLBH algorithm is shown in Fig. 10, and the related pseudo code is described as follows.

Give the number of populations *noP*, the iteration number *iter*, and give the parameters m_1 and m_2 of the golden sine operation

```
Initialize the position of each agent
for i = 1: $iter$
```



(a) F_1

(b) F_2

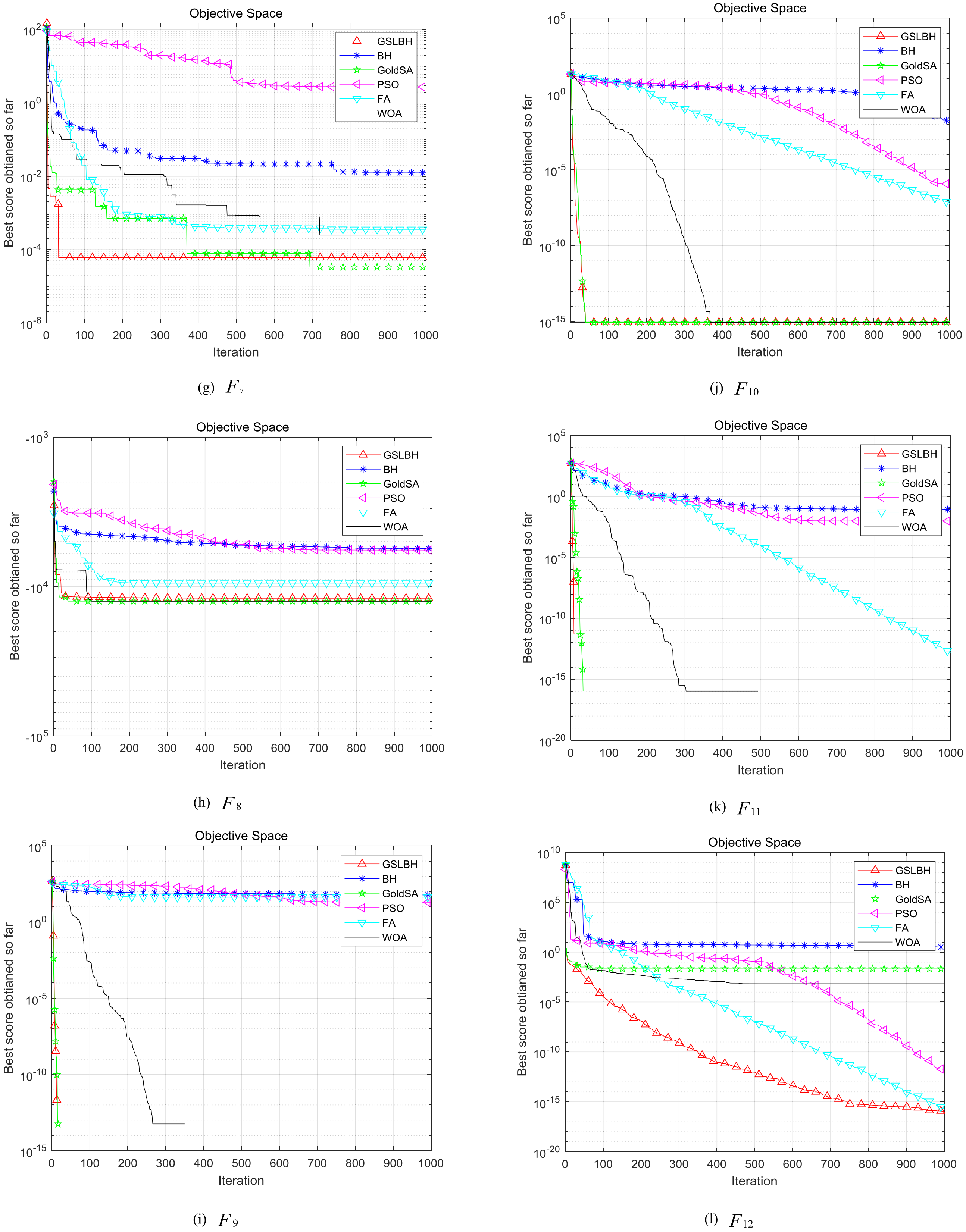
(c) F_3

(d) F_4

(e) F_5

(f) F_6

FIGURE 12. Experimental results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ under 30 dimensions.



(g) F_7

(j) F_{10}

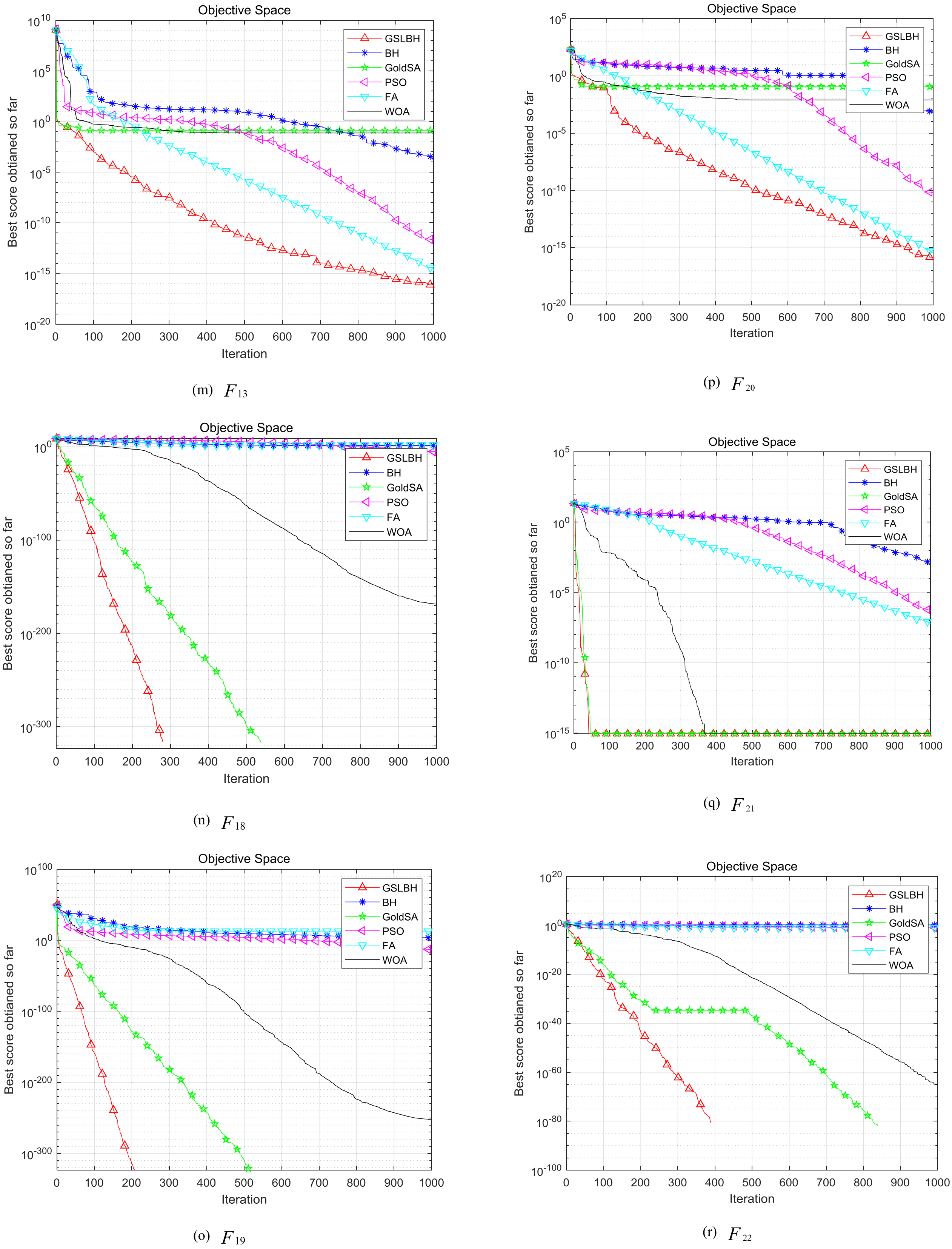
(h) F_8

(k) F_{11}

(i) F_9

(l) F_{12}

FIGURE 12. ***(Continued.)*** **Experimental results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ under 30 dimensions.**



(m) F_{13}

(p) F_{20}

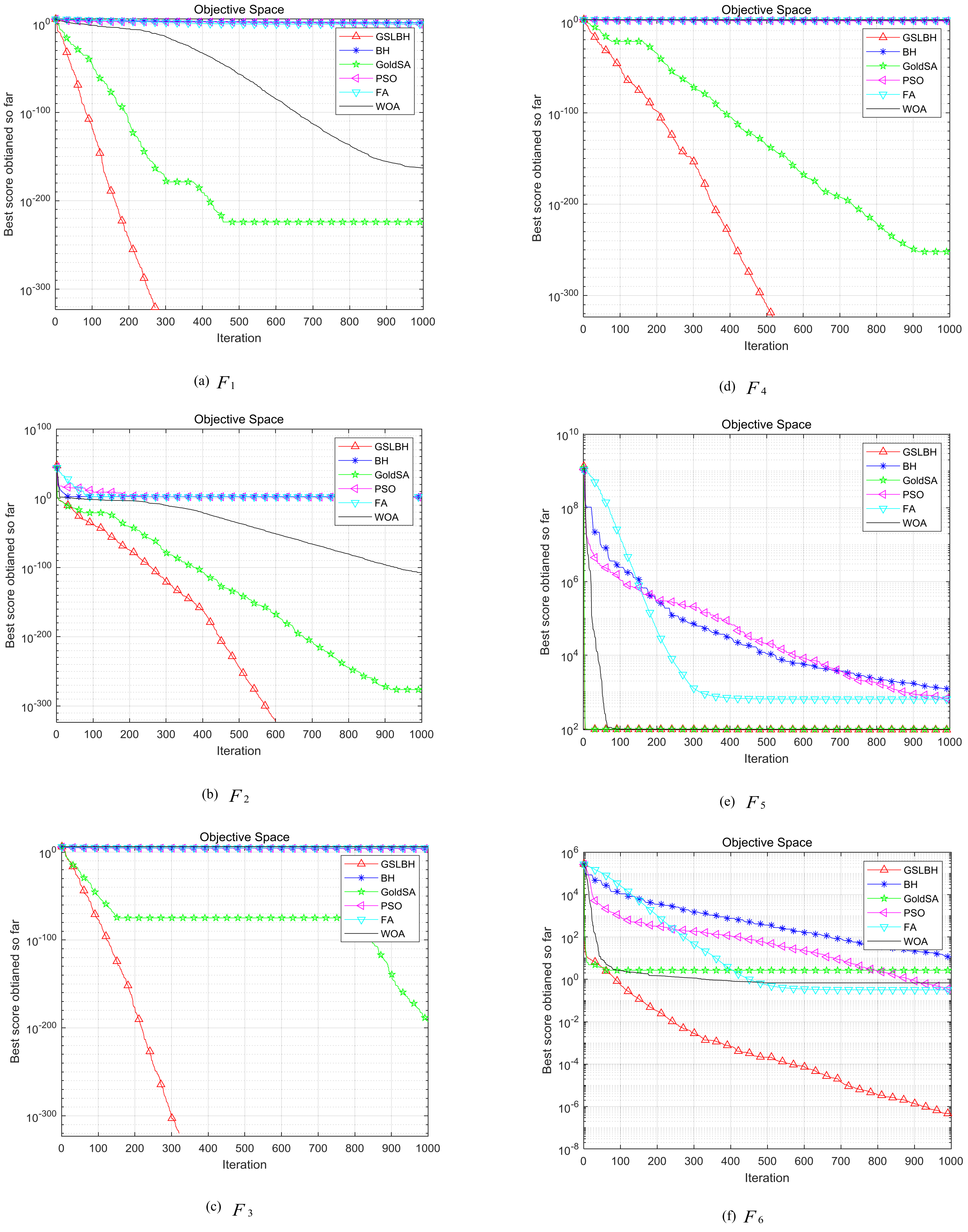
(n) F_{18}

(q) F_{21}

(o) F_{19}

(r) F_{22}

FIGURE 12. ***(Continued.)*** **Experimental results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ under 30 dimensions.**



(a) F_1

(d) F_4

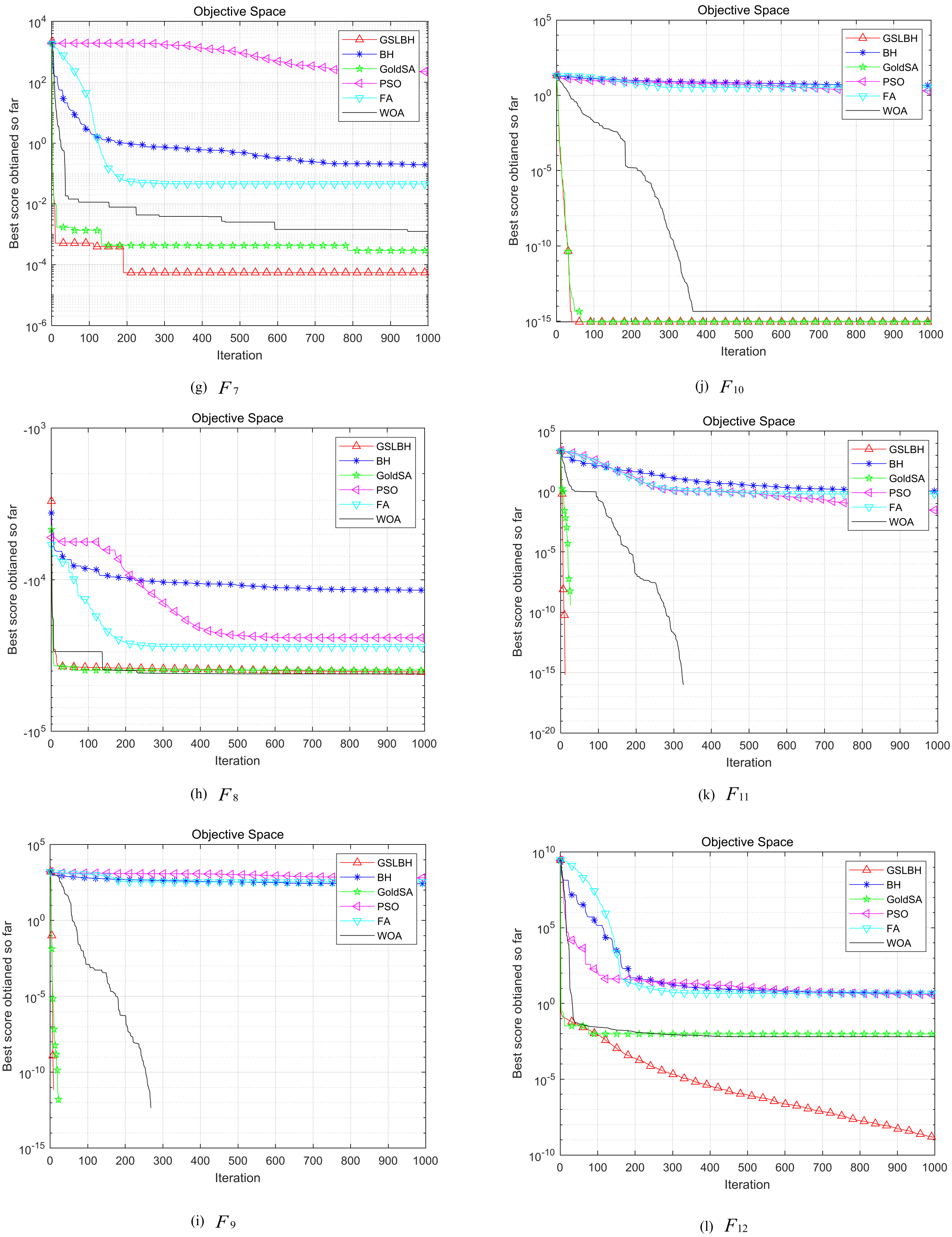
(b) F_2

(e) F_5

(c) F_3

(f) F_6

FIGURE 13. Experimental results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ under 100 dimensions.



(g) F_7

(j) F_{10}

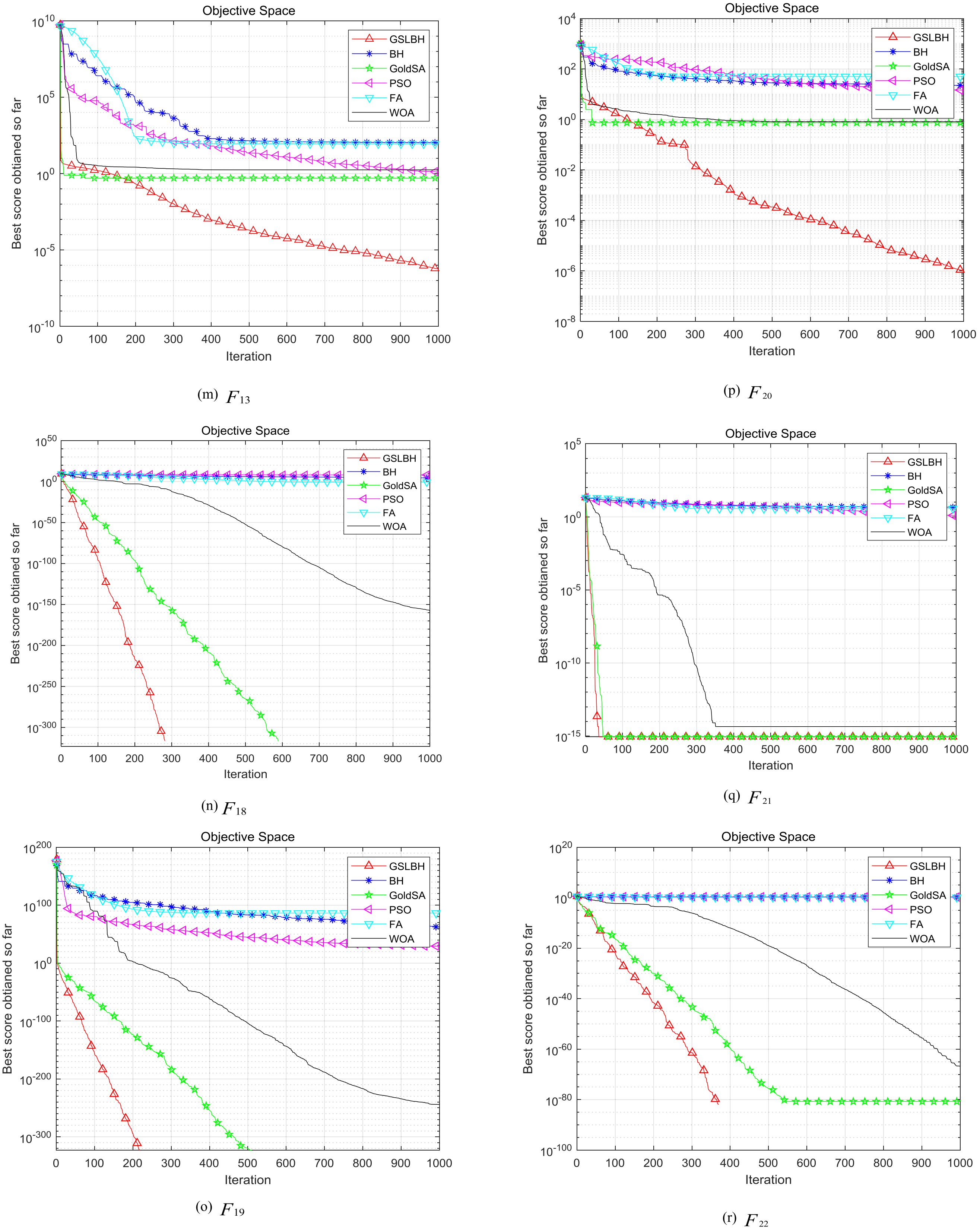
(h) F_8

(k) F_{11}

(i) F_9

(l) F_{12}

FIGURE 13. ***(Continued.)*** **Experimental results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ under 100 dimensions.**



(m) F_{13}

(p) F_{20}

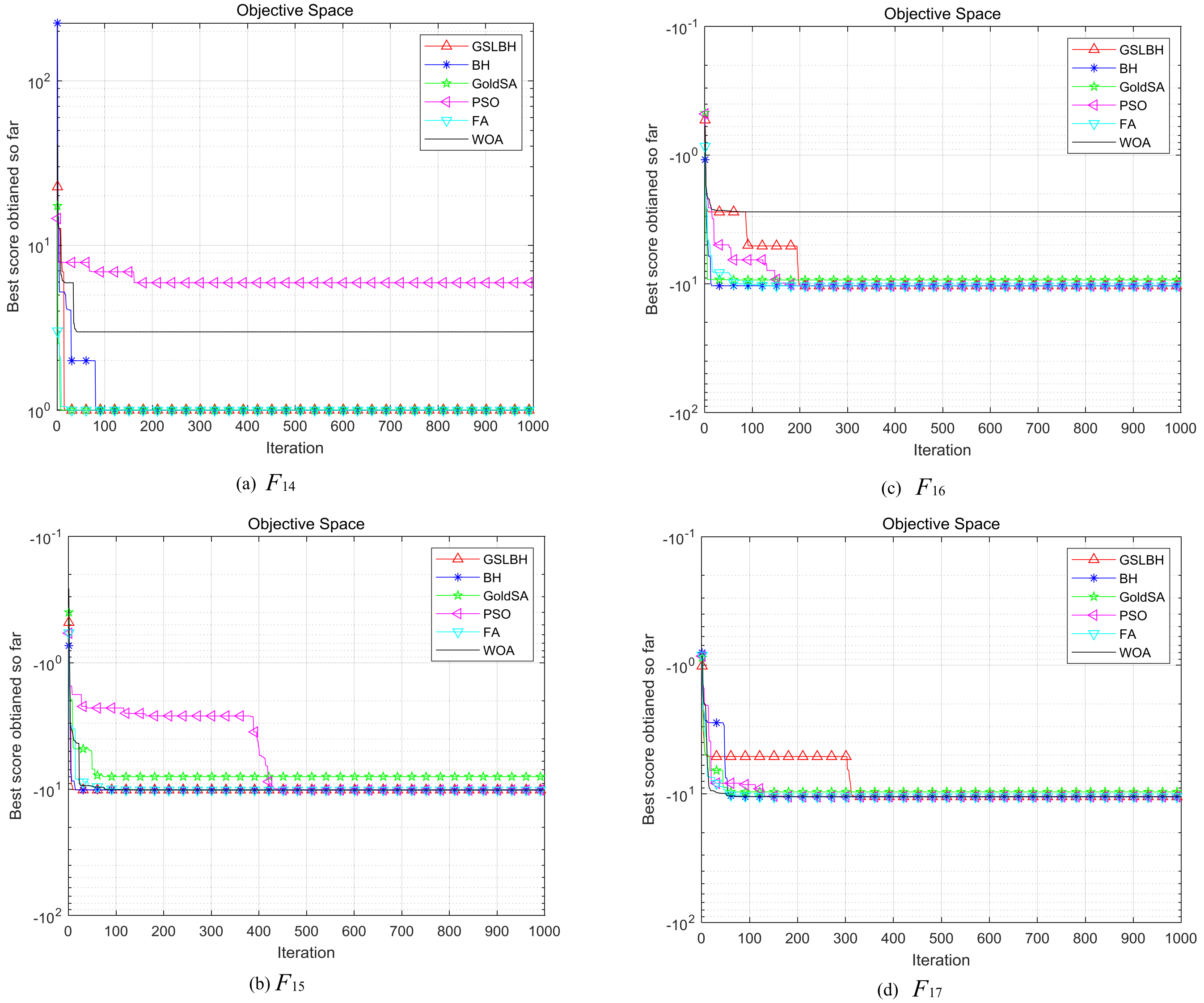
(n) F_{18}

(q) F_{21}

(o) F_{19}

(r) F_{22}

FIGURE 13. ***(Continued.)*** **Experimental results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ under 100 dimensions.**



(a) F_{14}

(c) F_{16}

(b) F_{15}

(d) F_{17}

FIGURE 14. Experimental results of $F_{14} \sim F_{17}$.

```
    Get the global optimal position $G_{gest}$, and the global optimal value $fit_{gest}$
    Record the optimal value for each iteration
    Let each agent move according to Eq. (2)
    for j = 1: $noP$
        Calculate the Euclidean distance $d_i$ of each agent according to the formula to the optimal position $G_{gest}$, and the "event range" $R_s$ of the black hole $G_{gest}$
        if $d_j < R_s$
         Let the agent $x_i$ "disappear" and create a new agent $x'_j$ anywhere in the solution space
        end if
    end for
    Let each agent complete the flight operation of levy according to Eq. (16)
    Agents perform "selective movement" operations
    Let each agent complete the golden sine search according to Eq. (15)
    Agents perform "selective movement" operations
end for
```

IV. SIMULATION EXPERIMENTS AND RESULTS ANALYSIS

A. TEST FUNCTIONS

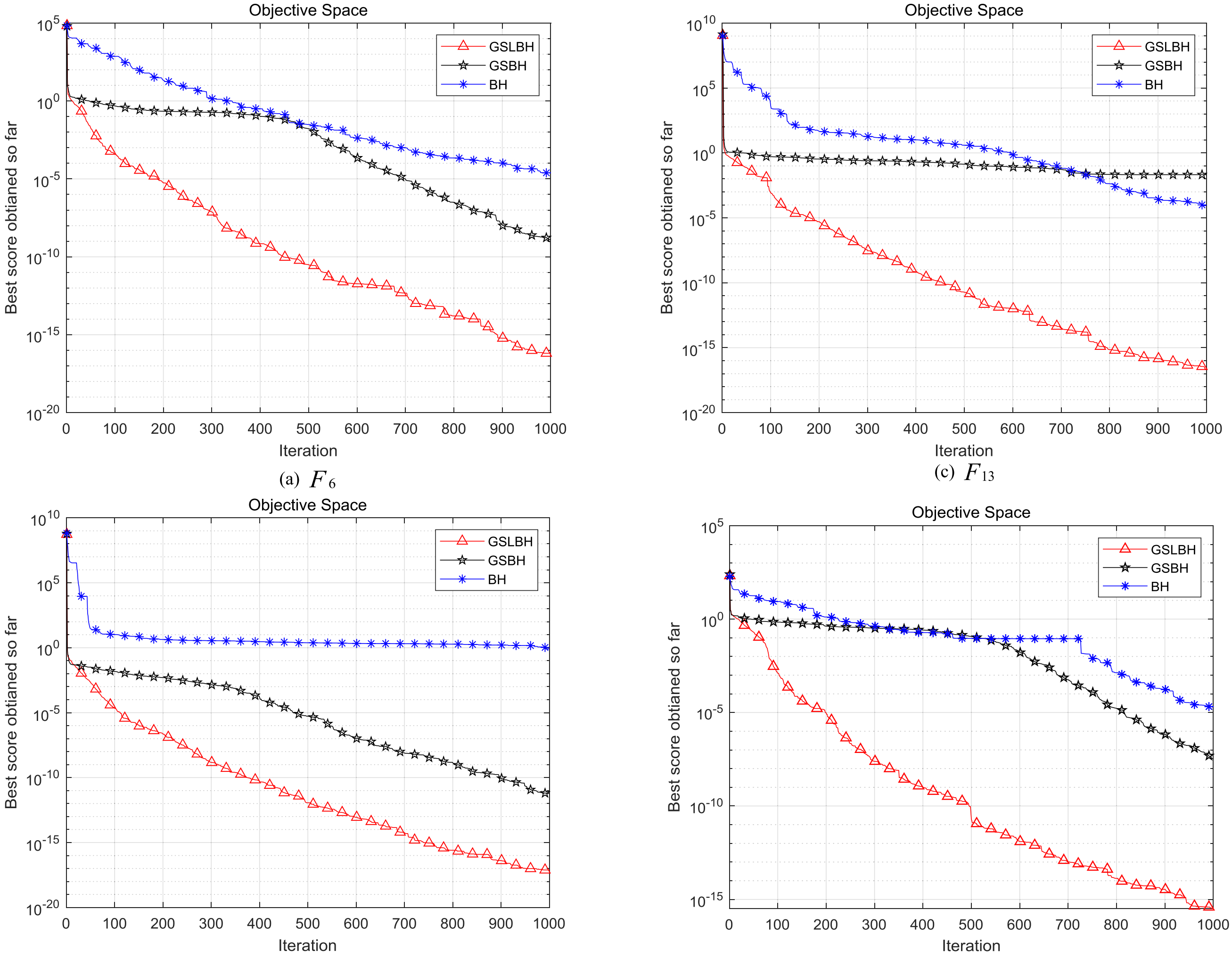
In order to test the optimization performance of the improved black hole algorithm, 17 benchmark functions in CEC 2005 and 5 benchmark functions in the CEC 2017 are adopted to verify the exploaration of the improved algorithms. $F_1 \sim F_{17}$ is benchmark function in CEC 2005, and $F_{18} \sim F_{22}$ is benchmark function in CEC 2017. The expressions of 17 benchmark functions are shown in Table 1. In order to prove the superiority of the improved algorithm from

TABLE 3. Experimental results under 30 dimensions.

Function	Performance	GSLBH	BH	Gold-SA	PSO	FA	WOA
F_{1}	Best	0	5.74E-07	0	1.12E-13	6.53E-14	9.65E-181
	Ave	0	4.39E-06	0	2.21E-10	8.09E-14	5.68E-166
	Std	0	3.13E-06	0	6.16E-10	1.20E-14	0
F_{2}	Best	0	9.21E-07	1.4566e-310	4.42E-07	1.15E-07	3.33E-115
	Ave	0	2.88E-05	3.55E-181	1.30E-05	1.22E-07	1.13E-107
	Std	0	4.70E-05	0	1.47E-05	5.20E-09	3.53E-107
F_{3}	Best	0	76.9725	0	1.526	2.27E-13	10124.2873
	Ave	0	255.3071	3.25E-274	5.5464	1.6647	20855.6133
	Std	0	75.6116	0	3.5761	2.7704	7763.8735
F_{4}	Best	0	0.77741	2.19E-306	0.1598	9.83E-08	0.0025501
	Ave	0	1.0009	5.29E-191	0.32346	1.14E-07	28.5138
	Std	0	0.19832	0	0.14585	9.61E-09	35.942
F_{5}	Best	23.8215	17.1262	4.6566	17.5719	24.3002	26.2675
	Ave	24.2635	75.4138	24.1551	24.5291	25.0953	26.9144
	Std	0.18293	69.9865	8.3513	3.5521	0.49889	0.38733
F_{6}	Best	3.29E-17	1.14E-06	0.16153	7.56E-13	6.40E-14	0.0053457
	Ave	4.47E-16	9.51E-06	0.57137	3.90E-11	7.85E-14	0.011005
	Std	5.09E-16	1.05E-05	0.44344	7.34E-11	9.44E-15	0.0069307
F_{7}	Best	1.70E-05	0.0079374	1.73E-05	0.032304	0.00026007	0.00043796
	Ave	4.95E-05	0.018643	0.0001804	5.4165	0.00055395	0.0012559
	Std	1.95E-05	0.0063923	0.00012826	8.9315	0.00018586	0.0010185
F_{8}	Best	-12569.4865	-6368.5137	-12486.6099	-7417.0686	-10812.6268	-12569.4289
	Ave	-11151.8694	-5927.3548	-11991.2111	-5257.5638	-9936.0063	-12115.1715
	Std	2063.1929	324.2851	431.6857	1428.5213	597.2131	899.321
F_{9}	Best	0	33.8865	0	32.8337	25.8689	0
	Ave	0	45.2712	0	55.5477	33.0326	5.68E-15
	Std	0	8.9032	0	25.0977	5.1122	1.80E-14
F_{10}	Best	8.88E-16	0.00072381	8.88E-16	2.84E-07	6.17E-08	8.88E-16
	Ave	8.88E-16	0.053863	8.88E-16	2.52E-06	6.65E-08	1.60E-15
	Std	0	0.13811	0	2.60E-06	3.68E-09	1.50E-15
F_{11}	Best	0	9.85E-05	0	1.07E-13	1.22E-13	0
	Ave	0	0.050029	0	0.0054197	1.47E-13	0
	Std	0	0.052039	0	0.0063368	1.89E-14	0
F_{12}	Best	1.27E-18	3.91E-05	0.00040711	8.58E-15	1.58E-16	0.00049465
	Ave	1.35E-17	1.1235	0.012766	2.58E-11	2.08E-16	0.0017514
	Std	1.79E-17	0.99364	0.011731	8.12E-11	2.12E-17	0.0022229
F_{13}	Best	3.07E-17	1.54E-06	0.060847	6.00E-14	2.72E-15	0.01279
	Ave	8.82E-15	0.0034748	0.24704	0.0021975	3.00E-15	0.15116
	Std	2.72E-14	0.0054831	0.12248	0.0046327	2.79E-16	0.11692
F_{18}	Best	0	0.0078722	0	7.9588e-07	6.3611	5.0426e-176
	Ave	0	4.2518	0	60.0001	43.8119	2.2201e-159
	Std	0	6.879	0	51.6397	34.5045	7.004e-159

TABLE 3. ***(Continued.)*** **Experimental results under 30 dimensions.**

F_{19}	Best	0	17.0398	0	2.1616e-20	2.246e+08	1.9598e-249
	Ave	0	51228.2139	0	7.1608e-15	2.052e+11	8.6656e-234
	Std	0	111642.0978	0	1.853e-14	4.848e+11	0
F_{20}	Best	7.1783e-17	3.0498e-08	0.002386	1.6203e-12	3.396e-16	0.0040299
	Ave	6.4894e-15	0.0089977	0.15822	0.0089528	4.0404e-16	0.18345
	Std	1.5265e-14	0.028334	0.1333	0.028311	3.9184e-17	0.26761
F_{21}	Best	8.8818e-16	0.00010996	8.8818e-16	3.4978e-07	6.2774e-08	8.8818e-16
	Ave	8.8818e-16	0.0060415	8.8818e-16	2.222e-06	6.8286e-08	3.7303e-15
	Std	0	0.0091606	0	2.2277e-06	3.1242e-09	2.8024e-15
F_{22}	Best	0	0.88076	0	0.094751	0.0022511	6.5319e-67
	Ave	0	1.2381	0	0.20702	0.032863	8.4421e-63
	Std	0	0.21995	0	0.069372	0.043402	2.1176e-62



(a) F_6

(c) F_{13}

(b) F_{12}

(d) F_{20}

FIGURE 15. **The experimental results when the partial function takes 30 dimensions.**

multiple angles, 17 functions are divided into three groups. $F_1 \sim F_7$ are unimodal functions, that is to say that these functions only have one global optimal solution in the solution space, so these single-peak functions are used to test the convergence rate of the algorithm. $F_8 \sim F_{13}$ are multimodal functions, where there ares more than one extreme

TABLE 4. Experimental results under 100 dimensions.

Function	Performance	GSLBH	BH	Gold-SA	PSO	FA	WOA
F_1	Best	0	6.3087	0	0.29839	1.52E-12	6.53E-185
	Ave	0	10.0051	1.56E-97	0.70401	0.053677	7.83E-162
	Std	0	3.9316	4.9379E-97	0.36015	0.10511	2.42E-161
F_2	Best	0	1.438	2.99E-306	22.9139	69.0601	1.16E-111
	Ave	0	167.2917	1.04E-157	51.5085	105.3914	1.43E-103
	Std	0	141.8691	3.3E-157	20.6247	31.7503	4.48E-103
F_3	Best	0	9378.728	0	5170.5509	46655.1338	596884.5355
	Ave	0	11891.8491	3.30E-53	6993.2233	56210.1011	797026.3223
	Std	0	1.83E+03	1.0420E-52	1327.2967	5936.6173	123446.2026
F_4	Best	0	8.139	4.92E-308	3.7641	30.196	25.4511
	Ave	0	9.7826	6.35E-50	6.4912	36.9039	76.538
	Std	0	1.3036	2.0067E-49	1.2154	3.8819	25.0234
F_5	Best	94.8192	730.0654	16.6817	425.2757	471.7171	96.9289
	Ave	95.0983	1103.9708	90.0216	755.3517	810.1985	97.4559
	Std	0.1985	278.9916	25.7693	272.454	268.9258	0.46978
F_6	Best	3.65E-07	5.1579	0.1584	0.1675	1.72E-12	0.53388
	Ave	6.69E-07	9.8969	1.6276	0.4429	0.11507	0.99336
	Std	3.9479E-07	3.7707	1.6107	0.20416	0.24234	0.36291
F_7	Best	8.81E-06	0.13949	2.51E-05	146.5004	0.023981	0.000249
	Ave	0.00010662	0.18468	0.00021798	252.291	0.031749	0.0014037
	Std	6.3647E-05	0.0308	1.6940E-04	111.3933	0.00547	0.0010769
F_8	Best	-40861.8845	-15968.1438	-41486.2692	-21219.3089	-29283.7093	-41884.8607
	Ave	-38154.6327	-13757.0088	-38748.7549	-10953.1421	-28386.866	-39323.1906
	Std	2.46E+03	1.11E+03	2390.9	6923.5488	901.1958	4334.7803
F_9	Best	0	201.726	0	454.8517	264.6585	0
	Ave	0	252.2491	0	508.0508	317.3908	0
	Std	0	42.1997	0	51.4597	29.2078	0
F_{10}	Best	8.88E-16	3.6297	8.88E-16	0.70423	3.1846	8.88E-16
	Ave	8.88E-16	4.4942	8.88E-16	1.961	3.5192	3.73E-15
	Std	0	0.4752	0	0.54359	0.17495	2.25E-15
F_{11}	Best	0	1.1062	0	0.0050616	0.54207	0
	Ave	0	1.167	0	0.012424	0.64348	0
	Std	0	0.0613	0	0.0061232	0.092882	0
F_{12}	Best	1.17E-09	4.1692	0.0019871	0.069355	4.7853	0.0046912
	Ave	3.32E-09	5.3803	0.020081	0.73153	5.7318	0.0096476
	Std	1.9619E-09	0.9732	0.01547	0.55967	0.76789	0.0029744
F_{13}	Best	1.72E-07	100.0548	0.046821	0.17046	65.1132	0.48215
	Ave	4.81E-08	114.123	0.76736	0.58573	92.4424	0.8668
	Std	7.7770E-07	12.1857	0.56572	0.26587	12.6063	0.25718

TABLE 4. ***(Continued.)*** **Experimental results under 100 dimensions.**

F_{18}	Best	0	10691.2075	0	1.2905e+05	2.0751	1.229e-169
	Ave	0	17401.9046	2.8201e-126	1.0606e+07	39.5068	2.3301e-161
	Std	0	6400.6868	8.918e-126	3.2012e+07	31.4131	6.4345e-161
F_{19}	Best	0	1.4572e+55	0	2.7343e+29	1.7506e+79	2.058e-262
	Ave	0	4.2326e+71	0	1.6619e+35	1.4693e+84	1.2915e-210
	Std	0	1.3253e+72	0	2.8116e+35	2.1969e+84	0
F_{20}	Best	4.3734e-07	4.0468	0.12618	13.8157	31.4425	0.63467
	Ave	2.8841e-06	13.3339	0.87948	31.4898	40.8895	0.95943
	Std	3.0858e-06	8.202	0.5456	9.7132	6.6105	0.25793
F_{21}	Best	8.8818e-16	3.3827	8.8818e-16	1.0769	3.2599	8.8818e-16
	Ave	8.8818e-16	4.5236	8.8818e-16	1.6361	3.455	3.0198e-15
	Std	0	0.94053	0	0.34438	0.16595	2.4841e-15
F_{22}	Best	0	1.6163	0	1.1281	1.3804	8.7116e-69
	Ave	0	1.812	1.2995e-62	1.3948	1.4655	5.8591e-65
	Std	0	0.13521	4.1091e-62	0.15991	0.07574	1.1487e-64

TABLE 5. **Experimental results of the combined functions.**

Function	Performance	GSLBH	BH	Gold-SA	PSO	FA	WOA
F_{14}	Best	0.998	0.998	0.998	0.998	0.998	0.998
	Ave	0.998	0.998	0.99801	1.4946	0.998	0.998
	Std	1.94E-14	8.72E-15	2.21E-05	0.70195	1.96E-16	8.85E-11
F_{15}	Best	-10.153	-10.1181	-10.1185	-10.1532	-10.1532	-10.1532
	Ave	-9.1323	-7.1189	-8.2983	-7.6224	-10.1532	-9.642
	Std	2.1489	3.2731	2.1772	2.6677	1.87E-15	1.6117
F_{16}	Best	-10.4027	-10.3744	-10.3961	-10.4029	-10.4029	-10.4029
	Ave	-8.2759	-8.3148	-8.7247	-8.2933	-10.4029	-9.2013
	Std	2.7442	3.3585	2.3314	2.7235	1.67E-15	2.5478
F_{17}	Best	-10.5363	-10.5109	-10.5294	-10.5364	-10.5364	-10.5363
	Ave	-8.3723	-8.1265	-9.2025	-9.6687	-10.5364	-8.3728
	Std	2.7919	3.1414	1.9231	2.7439	2.37E-15	2.7923

points in the multimodal functions, so the algorithm is easy to fall into the local optimum. The test results of the multimodal functions can well explain the function optimization ability of the algorithm. $F_{14} \sim F_{17}$ are combined functions [46]. The combined function is formed by rotation, shifting, and offset of various benchmark functions. $F_{18} \sim F_{22}$ are complex functions in CEC 2017, $F_{18} \sim F_{19}$ are unimodal functions, and $F_{20} \sim F_{22}$ are multimodal functions. Such functions have lower dimensions and a small number of local optimum values. Both the unimodal functions and the multimodal functions are tested in 30-dimensional and 100-dimensional, respectively, and the combined functions are tested in a dimension suitable for itself. The simulation experiments were carried out several times on these three types of functions, and the simulation results were obtained after 1000 iterations of each experiment. The main operating parameters of each algorithm are shown in Table 2. Table 1 also contains the definition fields, dimensions, and minimum values of the three types of benchmark functions. dim in function F_8 represents the dimensions of the variables.

TABLE 6. Experimental results under 30 dimensions.

Function	Performance	GSLBH	GSBH	BH
F_1	Best	0	0	5.74E-07
	Ave	0	0	4.39E-06
	Std	0	0	3.13E-06
F_2	Best	0	0	9.21E-07
	Ave	0	0	2.88E-05
	Std	0	0	4.70E-05
F_3	Best	0	0	76.9725
	Ave	0	0	255.3071
	Std	0	0	75.6116
F_4	Best	0	0	0.77741
	Ave	0	0	1.0009
	Std	0	0	0.19832
F_5	Best	23.8215	25.8404	17.1262
	Ave	24.2635	26.3357	75.4138
	Std	0.18293	0.46046	69.9865
F_6	Best	3.29E-17	7.63E-14	1.14E-06
	Ave	4.47E-16	1.33E-10	9.51E-06
	Std	5.09E-16	2.05E-10	1.05E-05
F_7	Best	1.70E-05	6.29E-07	0.0079374
	Ave	4.95E-05	1.02E-05	0.018643
	Std	1.95E-05	7.45E-06	0.0063923
F_8	Best	-12569.4865	-12569.487	-6368.5137
	Ave	-11151.8694	-12178.628	-5927.3548
	Std	2063.1929	483.782	324.2851
F_9	Best	0	0	33.8865
	Ave	0	0	45.2712
	Std	0	0	8.9032
F_{10}	Best	8.88E-16	8.88E-16	0.00072381
	Ave	8.88E-16	8.88E-16	0.053863
	Std	0	0	0.13811
F_{11}	Best	0	0	9.85E-05
	Ave	0	0	0.050029
	Std	0	0	0.052039
F_{12}	Best	1.27E-18	1.00E-12	3.91E-05
	Ave	1.35E-17	3.27E-08	1.1235
	Std	1.79E-17	6.40E-08	0.99364
F_{13}	Best	3.07E-17	5.10E-07	1.54E-06
	Ave	8.82E-15	0.072025	0.0034748
	Std	2.72E-14	0.073457	0.0054831
F_{18}	Best	0	0	0.0078722
	Ave	0	0	4.2518
	Std	0	0	6.879
F_{19}	Best	0	0	17.0398
	Ave	0	0	51228.2139
	Std	0	0	111642.097
F_{20}	Best	7.1783e-17	0.0029504	3.0498e-08
	Ave	6.4894e-15	0.31387	0.0089977
	Std	1.5265e-14	0.15432	0.028334
F_{21}	Best	8.8818e-16	8.8818e-16	0.00010996
	Ave	8.8818e-16	8.8818e-16	0.0060415
	Std	0	0	0.0091606
F_{22}	Best	0	0	0.88076
	Ave	0	0	1.2381
	Std	0	0	0.21995

TABLE 6. (Continued.) Experimental results under 30 dimensions.

The three-dimensional images of the functions are shown in Fig. 11.

B. SIMULATION EXPERIMENTS AND PERFORMANCE ANALYSIS WITH OTHER ALGORITHMS

In order to verify the performance of the improved algorithm, this paper compares the improved algorithm with the simple black hole (BH) algorithm, Particle Swarm Optimization (PSO) algorithm, Firefly algorithm (FA), Whale Optimization Algorithm (WOA) and Golden Sine Algorithm (Gold-SA) for simulation experiments. Each test function was run 10 times, and the number of iterations per experiment was set to 1000. The 30-dimensional simulation results of $F_1 \sim F_{13}$ and $F_{18} \sim F_{22}$ are shown in Fig. 12, and the 100-dimensional simulation results are shown in Fig. 13. The simulation results of $F_{14} \sim F_{17}$ are shown in Fig. 14. The result of running 10 times when function $F_1 \sim F_{13}$ takes 30 dimensions is put into Table 3, the result of running 10 times when $F_1 \sim F_{13}$ takes 100 dimensions is put into Table 4. The simulation results of $F_{14} \sim F_{17}$ running 10 times are put into Table 5. These tables give the optimal value, mean value, and standard deviation for each experimental result. The optimal values for the individual functions are already described in Table 2.

It can be seen from the above experimental results that the function optimization ability of GSLBH algorithm is better than other optimization algorithms except for a few cases. For the test results of the unimodal function $F_1 \sim F_7$, it can be seen from the results that the convergence of GSLBH algorithm is better than other optimization algorithms. Seen from the table data, whether they are 30-dimensional or 100-dimensional, it is clear that the optimization results of GSLBH algorithm for unimodal functions are mostly better than other algorithms.

The optimization results of GSLBH algorithm on $F_1 \sim F_4$ can converge to the minimum value of 0; For function F_5,

TABLE 7. Experimental results under 100 dimensions.

Function	Performance	GSLBH	GSBH	BH
F_1	Best	0	0	6.3087
	Ave	0	0	10.0051
	Std	0	0	3.9316
F_2	Best	0	0	1.438
	Ave	0	0	167.2917
	Std	0	0	141.8691
F_3	Best	0	0	9378.728
	Ave	0	0	11891.8491
	Std	0	0	1.83E+03
F_4	Best	0	0	8.139
	Ave	0	0	9.7826
	Std	0	0	1.3036
F_5	Best	94.8192	95.6576	730.0654
	Ave	95.0983	96.616	1103.9708
	Std	0.1985	0.83582	278.9916
F_6	Best	3.65E-07	0.0005132	5.1579
	Ave	6.69E-07	0.044183	9.8969
	Std	3.9479E-07	0.063292	3.7707
F_7	Best	8.81E-06	2.48E-06	0.13949
	Ave	0.00010662	2.02E-05	0.18468
	Std	6.3647E-05	1.82E-05	0.0308
F_8	Best	-40861.8845	-41658.738	-15968.1438
	Ave	-38154.6327	-39381.1	-13757.0088
	Std	2.46E+03	2586.7363	1.11E+03
F_9	Best	0	0	201.726
	Ave	0	0	252.2491
	Std	0	0	42.1997
F_{10}	Best	8.88E-16	8.88E-16	3.6297
	Ave	8.88E-16	8.88E-16	4.4942
	Std	0	0	0.4752
F_{11}	Best	0	0	1.1062
	Ave	0	0	1.167
	Std	0	0	0.0613
F_{12}	Best	1.17E-09	8.98E-06	4.1692
	Ave	3.32E-09	0.0007344	5.3803
	Std	1.9619E-09	0.0008045	0.9732
F_{13}	Best	1.72E-07	0.21644	100.0548
	Ave	4.81E-08	1.337	114.123
	Std	7.7770E-07	0.54052	12.1857

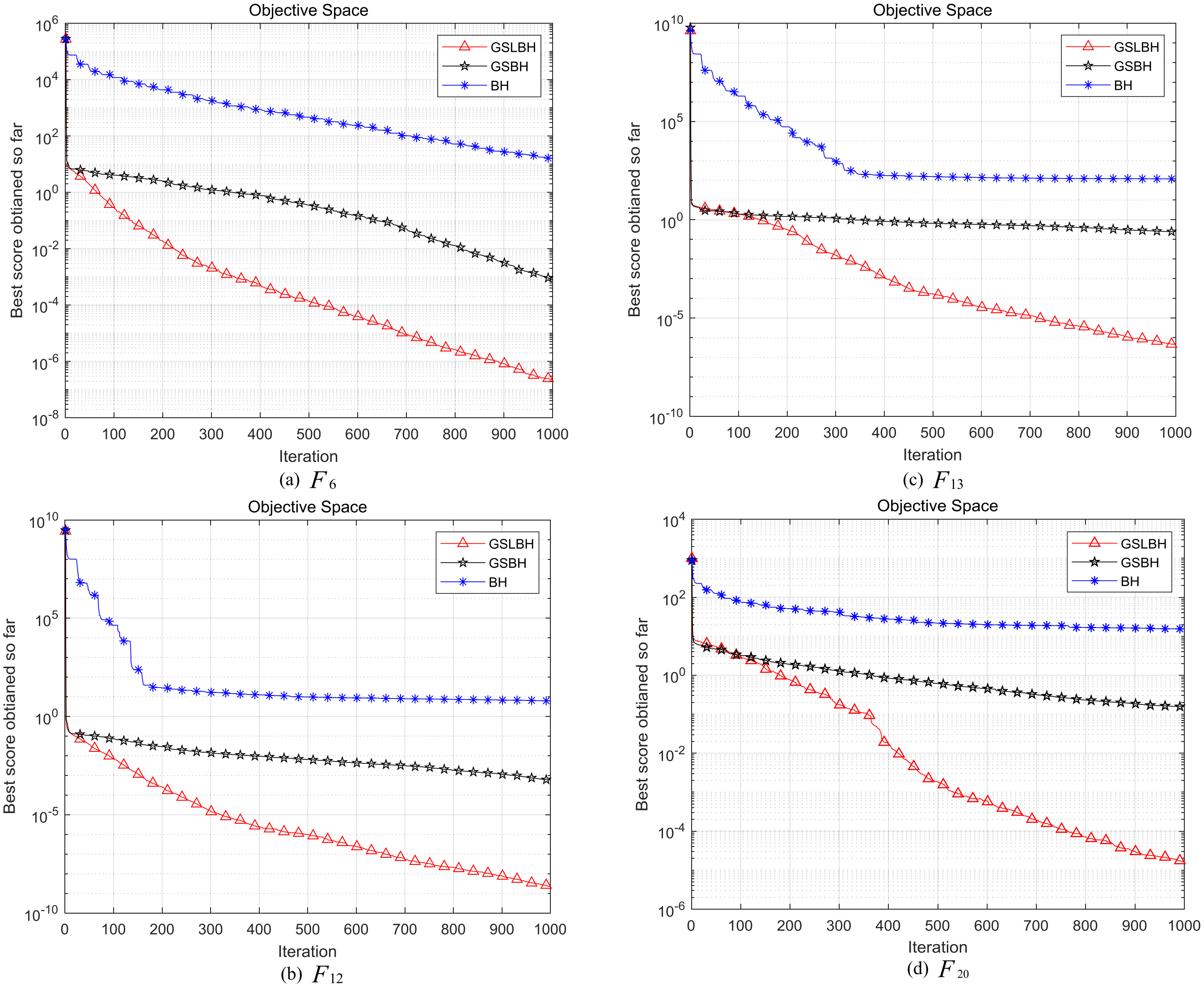
TABLE 7. *(Continued.)* Experimental results under 100 dimensions.

Function	Performance	GSLBH	GSBH	BH
F_{18}	Best	0	0	10691.2075
	Ave	0	0	17401.9046
	Std	0	0	6400.6868
F_{19}	Best	0	0	1.4572e+55
	Ave	0	0	4.2326e+71
	Std	0	0	1.3253e+72
F_{20}	Best	4.3734e-07	0.10479	4.0468
	Ave	2.8841e-06	0.29991	13.3339
	Std	3.0858e-06	0.11532	8.202
F_{21}	Best	8.8818e-16	8.8818e-16	3.3827
	Ave	8.8818e-16	8.8818e-16	4.5236
	Std	0	0	0.94053
F_{22}	Best	0	0	1.6163
	Ave	0	0	1.812
	Std	0	0	0.13521

TABLE 8. Experimental results of combined functions.

Function	Performance	GSLBH	GSBH	BH
F_{14}	Best	0.998	0.998	0.998
	Ave	0.998	1.1968	0.998
	Std	1.94E-14	0.41912	8.72E-15
F_{15}	Best	-10.153	-10.1478	-10.1181
	Ave	-9.1323	-6.5761	-7.1189
	Std	2.1489	2.4501	3.2731
F_{16}	Best	-10.4027	-10.3862	-10.3744
	Ave	-8.2759	-7.7334	-8.3148
	Std	2.7442	2.7897	3.3585
F_{17}	Best	-10.5363	-10.5305	-10.5109
	Ave	-8.3723	-7.8234	-8.1265
	Std	2.7919	2.8424	3.1414

the optimization effect of GSLBH algorithm is not particularly ideal. Although the convergence accuracy of GSLBH algorithm is lower than that of Gold-SA and PSO algorithm (only in 30-dimensional), its optimization effect is better than other algorithms. The optimization effects of GSLBH algorithm on F_6 and F_7 are also better than other algorithms. For the multimodal functions $F_8 \sim F_{13}$, the convergence accuracy of GSLBH algorithm is near the optimal value of the function. For the functions F_9, F_{11}, GSLBH algorithm can find the optimal value. It can be seen from the table data that the optimization results of GSLBH algorithm for



(a) F_6

(c) F_{13}

(b) F_{12}

(d) F_{20}

FIGURE 16. Experimental results of the partial functions under 100 dimensions.

F_8, F_9, F_{10}, and F_{11} are basically the same as those of WOA and Gold-SA. The optimization results of other multimodal functions by adopting GSLBH algorithm are better than other algorithms. For the combined function $F_{14} \sim F_{17}$, GSLBH algorithm does not show a particularly obvious advantage by observing the figure results and table data. In the figure results of F_{16} and F_{17}, the convergence speed of GSLBH algorithm is slower than most algorithms. For the experimental results of $F_{18} \sim F_{22}$, the optimization effect of GSLB is better than that of most other algorithms.Only the optimization effect of Gold-SA is close to that of GSLBH, but from the results of the graph and the data in the table, it can be seen that the optimization effect of GSLBH for F_{20} is obviously better than that of Gold-SA.The table data shows that the optimization results of GSLBH are slightly better than the optimization results of other algorithms.

Seen from all the simulation results, most algorithms have some optimization ability for most benchmark functions in low-dimensional (30-dimensional). However, in the case of high-dimensional (100-dimensional), the optimization performance of the algorithm will drop sharply. Even some algorithms will easily fall into the local optimum and cannot find the optimal value. On the contrary, whether the GSLBH is faced with low-dimensional or high-dimensional conditions, the optimization performance of the algorithm is not greatly affected, and still can show better optimization results.Through the above experimental results, it can be seen that GSLBH has good exploaration and exploitation.Regardless of whether the function is high-dimensional or low-dimensional, GSLBH always obtains a better solution and exhibits certain robustness. Based on all the experimental results, GSLBH is an improved algorithm with strong function optimization ability.

C. SIMULATION EXPERIMENTS AND PERFORMANCE ANALYSIS OF IMPROVED ALGORITHMS

In order to more clearly show the effect of the improved golden sine operator and the improved levy flight operator on the optimal performance of BH, GSLBH, GSBH that only introduces an improved golden sine operator and the standard BH are respectively compared and simulated. Table 6 and Table 7 show the simulation results of the 30-dimensional and 100-dimensional unimodal functions $F_1 \sim F_7$, $F_{18} \sim F_{19}$ and the multimodal functions $F_8 \sim F_{13}$, $F_{20} \sim F_{22}$ respectively. Table 8 shows the simulation results of the combined function. Experimental results of partial functions in 30-dimensional and 100-dimensional are shown in Fig. 15 and Fig. 16. It can be seen from the simulation results that GSLBH is mostly the same as GSBH, but several functions are obviously different, respectively F_6, F_{12}, F_{13}, F_{20}. After the introduction of the levy flight operator, the convergence of the algorithm is greatly enhanced. Especially in the case of high-dimensional, this difference is very obvious. In order to express this difference, the paper gives the figure results of three functions F_6, F_{12}, F_{13} and F_{20} in the case of 30-dimensional and 100-dimensional. In Fig. 15 and Fig. 16, it can be clearly seen that GSLBH is more astringent than GSBH, and the performance of function optimization is better. It can be seen from the comparison experiments of GSLB and GSBH that the experimental results show that the improved golden sine operator plays a decisive role in the exploaration of the black hole algorithm, and the improved levy flight operator further enhances the exploitation of BH.

V. CONCLUSION

As a new type of natural heuristic algorithm, black hole algorithm has received wide attention because of its fewer parameters, simple structure and strong exploitation. However, the shortcomings of this algorithm are also very obvious, it has the disadvantages of poor exploaration and easy to fall into local optimum. This paper first introduces an improved golden sine operator that greatly enhances the exploaration of the black hole algorithm. Then, an improved levy flight local search operator is introduced to make the algorithm maintain balance between exploitation and exploaration. The improved black hole algorithm (GSLBH) was tested with 17 benchmark functions. The experimental results show that the improved golden sine operator plays a decisive role in the exploaration of the black hole algorithm, and the improved levy flight operator further enhances the exploitation of BH. In a word, the introduction of two improved operators greatly enhances the performance of the black hole algorithm, and makes this improved black hole algorithm show better optimization ability for different functions. In the future study, this golden sine algorithm with strong exploaration will be focused on, and this golden sine search idea will be considered being applied to other natural heuristic algorithms, such as ant lion optimization algorithm(ALO), particle swarm optimization (PSO), to better solve the optimization problems.

REFERENCES

[1] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[2] L. L. Li, "Overview of the application of bat algorithm," *Softw. Guide*, vol. 15, no. 12, pp. 170–171, Dec. 2016.

[3] J. H. Holland, "Adaptation in natural and artificial systems," *Quart. Rev. Biol.*, vol. 6, no. 2, pp. 126–137, 1975.

[4] W. D. Ren and W. X. Sun, "Application of an improved ant colony algorithm in TSP problem solving," *Chem. Eng. Trans.*, vol. 51, pp. 373–378, Aug. 2016, doi: 10.3303/CET1651063.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 2002, pp. 1–59.

[6] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, 2003.

[7] L. X. Li, Z. J. Shao, and J. X. Qian, "An optimizing method based on autonomous animals: Fish-swarm algorithm," *Syst. Eng. Theory Pract.*, vol. 22, no. 12, pp. 32–38, Nov. 2002.

[8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002.

[9] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010.

[10] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Beckington, U.K.: Luniver Press, 2008, pp. 81–89.

[11] X. S. Yang, "Flower pollination algorithm for global optimization," in *Proc. 11th Int. Conf. Unconventional Comput. Natural Comput. (UCNC)*, Sep. 2012, pp. 240–249, doi: 10.1007/978-3-642-32894-7_27.

[12] S. Salcedo-Sanz, J. Del Ser, I. Landa-Torres, S. Gil-López, and J. A. Portilla-Figueras, "The coral reefs optimization algorithm: A novel metaheuristic for efficiently solving optimization problems," *Sci. World J.*, vol. 2014, 2014, Art. no. 739768, doi: 10.1155/2014/739768.

[13] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.

[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[15] L. Yi, "Study on an improved PSO algorithm and its application for solving function problem," *Int. J. Smart Home*, vol. 10, no. 3, pp. 51–62, 2016.

[16] Y.-C. Li and Y. Peng, "Improved artificial bee colony algorithm based on information entropy," *Control Decis.*, vol. 30, no. 6, pp. 1121–1125, Aug. 2015.

[17] S. Yu, S. Yang, and S. Su, "Self-adaptive step firefly algorithm," *J. Appl. Math.*, vol. 2013, Sep. 2013, Art. no. 832718.

[18] D. L. Chu, H. Chen, and X. G. Wang, "Whale optimization algorithm based on adaptive weight and simulated annealing," *Acta Electron. Sinica*, vol. 47, no. 5, pp. 992–999, 2019.

[19] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning grey wolf optimizer," *IEEE Access*, vol. 7, pp. 113810–113825, 2019.

[20] D. Singh, U. Singh, and R. Salgotra, "An extended version of flower pollination algorithm," *Arabian J. Sci. Eng.*, vol. 43, no. 12, pp. 7573–7603, 2018.

[21] W. Long, S. Cai, and J. Jiao, "Improved whale optimization algorithm for large scale optimization problems," *Syst. Eng.-Theroy Appl.*, vol. 37, no. 11, pp. 2983–2994, 2017.

[22] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2012.

[23] H. R. E. H. Bouchekara, "Optimal design of electromagnetic devices using a black-hole-based optimization technique," *IEEE Trans. Magn.*, vol. 49, no. 12, pp. 5709–5714, Dec. 2013.

[24] E. Pashaei and N. Aydin, "Binary black hole algorithm for feature selection and classification on biological data," *Appl. Soft Comput.*, vol. 56, pp. 94–106, Jul. 2017.

[25] T. Wang, X. W. Gao, and Z. J. Jiang, "Parameters optimizing of LSSVM based on black hole algorithm," *J. Northeastern Univ. (Natural Sci.)*, vol. 35, no. 2, pp. 170–174, 2014.

[26] P. Veres, T. Bányai, and B. Illés, "Optimization of in-plant production supply with black hole algorithm," *Solid State Phenomena*, vol. 261, pp. 503–508, Aug. 2017.

[27] H. Bouchekara, "Optimal power flow using black-hole-based optimization approach," *Appl. Soft Comput.*, vol. 24, pp. 879–888, Nov. 2014.

[28] D. D. W. Sungkono, "Black hole algorithm for determining model parameter in self-potential data," *J. Appl. Geophys.*, vol. 148, pp. 189–200, Jan. 2018.

[29] Á. G. Rubio, B. Crawford, R. Soto, E. Olguín, S. Misra, A. Jaramillo, S. M. Villablanca, and J. Salas, "Solving the set covering problem with a binary black hole inspired algorithm," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2016, pp. 207–219.

[30] S. Yaghoobi and H. Mojallali, "Modified black hole algorithm with genetic operators," *Int. J. Comput. Intell. Syst.*, vol. 9, no. 4, pp. 652–665, 2016.

[31] H. Aslani, M. Yaghoobi, and M.-R. Akbarzadeh-T, "Chaotic inertia weight in black hole algorithm for function optimization," in *Proc. Int. Congr. Technol., Commun. Knowl.*, Nov. 2015, pp. 123–129.

[32] T. Wang, W. F. Liu, and C. F. Liu, "Optimization algorithm of black-hole based on Euclidean distance," *J. Shenyang Univ. Technol.*, vol. 38, no. 2, pp. 201–205, Mar. 2016.

[33] A. M. Reynolds and M. A. Frye, "Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search," *PLoS ONE*, vol. 2, no. 4, p. e354, Apr. 2007.

[34] A. M. Reynolds and M. A. Frye, "Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search," *PLoS ONE*, vol. 2, no. 4, p. e354, 2007.

[35] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput.*, Dec. 2009, pp. 210–214.

[36] X.-M. Zhang, X. Wang, and Q. Tu, "Particle swarm optimization algorithm based on combining global-best operator and Levy fligh," *J. Univ. Electron. Sci. Technol. China*, vol. 47, no. 3, pp. 421–429, 2018, doi: 10.3969/j.issn.1001-0548.2018.03.016.

[37] H. Yin, K. L. Dong, Z. R. Peng, and S. Y. Li, "Fish swarm algorithm with Lévy flight and firefly behavior," *Control Theory Appl.*, vol. 35, no. 4, pp. 497–505, Apr. 2018.

[38] N. Li, G. Li, and Z. L. Deng, "An improved sine cosine algorithm based on Levy flight," *Proc. SPIE*, vol. 10420, Jul. 2017, Art. no. 104204R.

[39] T. Hassanzadeh, H. Vojodi, and A. M. E. Moghadam, "A multilevel thresholding approach based on Levy-flight firefly algorithm," in *Proc. 7th Iranian Conf. Mach. Vis. Image Process.*, Nov. 2011, pp. 1–5.

[40] E. Tanyildizi and G. Demir, "Golden sine algorithm: A novel math-inspired algorithm," *Adv. Elect. Comput. Eng.*, vol. 17, no. 2, pp. 71–78, 2017, doi: 10.4316/AECE.2017.02010.

[41] J. F. Yu, S. Liu, J. J. Wang, and X. Y. Lu, "An ant lion optimization algorithm combining Levy flight and gold sine," *J. Comput. Appl.*, to be published.

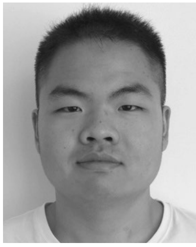
[42] R. Wald, *General Relativity*. Chicago, IL, USA: Univ. Chicago Press, 1984, pp. 299–300.

[43] N. K. Glendenning, *Special and General Relativity*. Berlin, Germany: Springer, 2007.

[44] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Beckington, U.K.: Luniver Press, 2008, pp. 16–17.

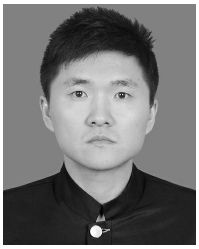
[45] R. K. Arora, *Optimization: Algorithms and Applications*. 2015, pp. 46–47.

[46] J. J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Proc. IEEE Swarm Intell. Symp.*, Jun. 2005, pp. 68–75.



W. XIE is currently pursuing the master's degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China.

J. S. WANG received the B.Sc. and M.Sc. degrees in control science from the University of Science and Technology Liaoning, China, in 1999 and 2002, respectively, and the Ph.D. degree in control science from the Dalian University of Technology, China, in 2006. He is currently a Professor and a Master's Supervisor with the School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interests include modeling of complex industry process, intelligent control, and computer integrated manufacturing.



Y. TAO received the B.Sc. degree in computer science and technology from Anshan Normal University, China, in 2006, and the M.Sc. degree in computer application technology from Northwest Normal University, China, in 2015. He is currently a Lecturer with the School of Computer Science and Software Engineering, University of Science and Technology Liaoning. His main research interests include artificial intelligence, machine learning, and information security.