IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 18, 2019, accepted November 3, 2019, date of publication November 8, 2019, date of current version November 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952468

Improvement and Application of Hybrid Firefly Algorithm

JIQUAN WANG, MINGXIN ZHANG, HAOHAO SONG, ZHIWEN CHENG, TIEZHU CHANG, YUSHENG BI, AND KEXIN SUN
College of Engineering, Northeast Agricultural University, Harbin 150030, China

Corresponding author: Jiquan Wang (wang-jiquan@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018YFD0300105 and Subproject of Special Fund for Agro-scientific Research in the Public Interest (No. 201503116-04).

ABSTRACT Aiming at the problem of poor global search ability and slow convergence speed when solving optimization problems, this paper proposes improved hybrid firefly algorithm (HFA). HFA improves the position updating method, mutation strategy, chaotic search method and evolution strategy of the population. Specifically, the improved position update formula considers both the effect of high-brightness fireflies on position-updated fireflies, and the effects of the optimal firefly on position-updated fireflies. At the same time, a method of adaptive adjustment parameters in the position update formula is presented, which makes the position update method exhibit strong global search ability and local search ability in the initial stage and the later stage of iteration, respectively. In addition, a combined mutation operator is introduced into HFA, which effectively takes the local search and global search ability of the algorithm into account. Since chaotic search exhibits good ergodicity, an operation of randomly moving all fireflies in the population according to chaotic search is given, which enhances the ability of the algorithm to traverse the whole search space, and further improves the global search ability of the algorithm. To verify the effectiveness of HFA, 28 CEC2017 test problems are selected. The calculation results of 28 CEC2017 test problems show that compared with other algorithms, the accuracy of HFA is obviously better than that of other algorithms. Finally, HFA and other intelligent optimization methods in the literatures are used to optimize the structural parameters of cantilever beams. The optimization results show that the weight of the cantilever beam obtained by HFA is obviously smaller than other algorithms. The calculation results of CEC2017 test problems and practical problem show that the solving quality of HFA is obviously better than other algorithms.

INDEX TERMS Firefly algorithm, position update, combined mutation operator, chaotic search, evolution strategy.

I. INTRODUCTION

In 2008, by studying the mutual attraction and movement process of firefly individuals, Yang [1] proposed a novel swarm intelligent optimization algorithm, termed as firefly algorithm (FA). Because the FA has the advantages of simple concept, fewer parameters to be set, less influence of parameters on the algorithm, and easy implementation [2]. Therefore, the FA has been extensively used in process optimization [3]–[5], pipeline scheduling [6], image processing [7], [8], RFID network planning [9], neural network training [10], [11], job shop scheduling [12], and data mining [13], [14]; good results have been obtained as well.

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

The existing simulation results of the firefly algorithm show that the firefly algorithm has high optimization accuracy, and has been applied in many fields, showing a good application prospect.

Over the past decade, many scholars in the world have improved the FA and achieved some research results. According to the mechanism and the technology, the research results of the FA are split into four major aspects: (1) parameter control of FA; (2) hybrid FA; (3) position update method of FA; (4) application research of FA. The research results obtained in the research of firefly algorithm improvement not only improve the optimization performance of FA, but also promote the development of FA algorithm theory and application.

In the parameter control of FA, in 2013, Yang *et al.* [15] proposed a self-tuning algorithm framework to adjust the parameters of FA. The simulation results show that the performance of FA is obviously enhanced. In 2015, Yu *et al.* [16] proposed an improved firefly algorithm that dynamically adjusts the step factor strategy. The calculation results of 16 benchmark functions show that the solution quality of the improved firefly algorithm is significantly improved. In 2016, Wang *et al.* [17] proposed an improved FA based on adaptive adjustment of parameters. The improved FA implements the adaptive adjustment of step factor α and introduces the concept of minimum attraction to ensure that the attraction between fireflies is never 0. The results of 13 benchmark functions show that the performance of the improved firefly algorithm is better than the basic firefly algorithm and five other improved firefly algorithms. In 2019, Zhang *et al.* [18] proposed an improved FA, dynamically adjusting the step factor α. The calculation results of 6 benchmark functions show that the proposed method has superior performance in terms of accuracy, stability and robustness compared with other algorithms.

In the hybrid FA, in 2012, Tilahun and Ong [19] proposed an improved FA and gave a position update method for the brightest firefly in the population. The calculation results of the 7 benchmark functions show that the improved FA, compared with the basic FA, improves the convergence speed significantly. However, since the improved FA generates multiple random directions and performs comparison operations, the computational complexity of the algorithm is greatly increased. In addition, because the step size exerts a great impact upon the convergence speed of FA algorithm, the step size should be adaptive for different optimization problems and different stages of iteration, instead of using a fixed constant. In 2015, Long *et al.* [20] proposed a hybrid firefly algorithm which combines dynamic random local search algorithm with firefly algorithm. The results of four benchmark functions and two engineering optimization problems show that the hybrid firefly algorithm has better optimization performance than other optimization algorithms. In 2018, Verma *et al.* [21] proposed a hybrid FA that introduces Opposition-Based Learning (OBL) into the FA. The calculation results of 11 benchmark functions show that the hybrid firefly algorithm can improve the accuracy of solution significantly. Because the opposite solutions of the algorithm are calculated using all dimensions of the original solutions, some excellent original solutions are easy to lose. In 2018, Zhou *et al.* [22] proposed a hybrid FA that introduces orthogonal centroid opposition-based learning into the FA. The calculation results of 28 benchmark functions show that the quality of the solutions obtained by the hybrid firefly algorithm as compared with those of other algorithms obviously increases. At the later stage of iteration, the algorithm exhibits poor population diversity and small differences between individuals. Thus, it is easy to fall into local optimum. In 2019, Wang and Song [23] proposed a hybrid FA introducing chaotic search into the firefly algorithm. The results of 23 benchmark functions show that the hybrid firefly algorithm can improve the solution quality significantly compared with other algorithms. However, due to the long distance between fireflies at the early iteration stage, the value of the attraction β in the position update formula of the hybrid FA is 0. At this time, the fireflies perform random movement, which makes the convergence speed of the firefly algorithm decrease.

In the position update method of FA, in 2016, Wang *et al.* [24] proposed an improved FA with random attraction, rather than complete attraction. The calculation results of 11 benchmark functions show that the solution quality of the improved FA as compared with those of other algorithms is significantly improved. In 2018, Yelghi and Kose [25] proposed an improved FA that the tidal force formula is introduced to the firefly position update formula, thus better balancing the global search ability and the local search ability of the algorithm. According to the calculation results of 22 benchmark functions, the solution accuracy and convergence speed of the modified FA are better than other algorithms. In 2018, Zhang *et al.* [26] proposed a new firefly position update formula that incorporates accelerated attraction and evasion operations, which better solves the problem of premature convergence of the basic FA. The calculation results of 18 benchmark functions show that the solution accuracy of the improved FA is higher than those of other algorithms for high-dimensional optimization problems.

In the application research of FA, in 2017, He and Huang [27] proposed an improved FA and applied the improved FA in threshold segmentation of multistage color images. The experimental results of 10 color images show that the improved FA has the best effect on threshold segmentation of multistage color images compared with those of other algorithms. In 2018, Liu *et al.* [28] proposed a hybrid non-dominated sorting FA and applied it to the optimization problem of multi-pass grinding process parameters. The experimental results show that the grinding efficiency of hybrid non-dominated sorting FA is up-regulated by 32.4%. In 2018, Wang *et al.* [29] proposed a FA with dynamically adjusted parameter α and applied this method to the parameter optimization of a mixed nonlinear water demand forecasting model. The parameter optimization results show that the parameters optimized using this method have smaller average relative errors than those using other algorithms. In 2018, Zhou *et al.* [30] proposed a FA combining Lévy flight with automatic-learning and applied this method to optimize wireless sensor network configuration, achieving a good effect. In 2019, Singh *et al.* [31] adopted FA optimization technology to optimize the parameters of power system stabilizer, which improved the stability of power system.

In summary, the position update formula of the existing firefly algorithms does not consider the influence of the optimal firefly in the population on the other fireflies. If the attraction term of the optimal firefly to the firefly that performs the position update operation is added to the position update formula, the solution quality of the firefly algorithm can be improved. In literature [23], chaotic search was introduced into FA, so that the optimal firefly in the population could move randomly according to chaotic search. This random movement reduces the possibility of the population falling into local optimum to some extent. Furthermore, chaotic search has a good ergodicity, and it is capable of traversing the entire search space. Therefore, it is considered that all fireflies in the population randomly move according to chaotic search, and the ability of the algorithm to traverse the entire search space is enhanced, thereby strengthening the global search ability of the algorithm. A good position update formula should consider the global search ability of the algorithm and the local search ability of the algorithm. To solve the above problems, an improved hybrid firefly algorithm is proposed in this paper. The calculation results of 28 CEC2017 test problems show that compared with the improved firefly algorithms and other intelligent optimization methods in the existing literatures, the improved hybrid firefly algorithm proposed in this paper is better than other algorithms in solving quality. Finally, the improved hybrid firefly algorithm proposed in this paper is used in the parameter optimization of cantilever beam. The optimization results show that the HFA proposed in this paper obviously better than other algorithms in the literatures.

II. PENALTY FUNCTION METHOD FOR CONSTRAINED OPTIMIZATION

The mathematical model of a constrained optimization problem can be generally expressed as follows:

$$\begin{cases} \min f(X), X = [X_1, X_2, \cdots, X_k, \cdots X_n] \in R \\ s.t. \begin{cases} h_i(X) = 0, & i = 1, 2, \cdots, p \\ g_j(X) \geq 0, & j = 1, 2, \cdots, q \end{cases} \end{cases} \tag{1}$$

where $s.t.$ is the abbreviation of 'subject to', n is the population size, $h_i(X) = 0$ is the i-th equation constraint, p is the number of equation constraints, $g_j(X) \geq 0$ is the j-th inequality constraint, q is the number of inequality constraints, and X_k is a m-dimensional vector $X_k = (x_{k1}, x_{k2}, \cdots, x_{km})$.

Eq. (1) can be expressed as

$$\begin{cases} \min f(X), X = [X_1, X_2, \cdots, X_k, \cdots X_n] \in R \\ s.t.\ R = \{X | h_i(X) = 0, i = 1, 2, \cdots, p; \\ \qquad g_j(X) \geq 0, j = 1, 2, \cdots, q\} \end{cases} \tag{2}$$

Letting X^* be the optimal solution to the constrained optimization problem means $\forall X \in R$: $f(X*) \leq f(X)$. In addition, if $g_j(X*) = 0$, the constraint is referred to as active constraint. Under this concept, all the equation constraints $h_i(X) = 0$, $i = 1, 2, \cdots, p$ are active at X^*.

The penalty function method can be used to convert a constrained optimization problem to an unconstrained optimization problem. For this purpose, the penalty function is constructed by [32]

$$P(X, M) = f(X) + M_1 \sum_{i=1}^{p} [h_i(X)]^2 + M_2 \sum_{j=1}^{q} \left[\min(0, g_j(X))\right]^2 \tag{3}$$

where M_1 and M_2 are the penalty factors, generally chosen as large enough positive constants; the second and third terms on the right are the penalty terms, and $P(X, M)$ is the penalty function.

In Eq. (3), when $X \in R$, there should be no penalty to the feasible points, thus $P(X, M) = f(X)$; when $X \notin R$, for the non-feasible points, M_1 and M_2 should be big, therefore, the value of the second and third terms in equation (3) are large, which is equivalent to the 'penalty' for the infeasible point. Moreover, when X gets farther away from the feasible region, the penalty should be larger. When M_1 and M_2 become sufficiently large, the minimal point $X(M)$ of the unconstrained optimization problem of Eq. (3) is close enough to the minimum point of the original constrained optimization problem. When $X(M) \in R$, it becomes the minimal point of the original constraint problem.

The minimum value of Eq. (3) is given by

$$\min P(X, M) \tag{4}$$

which is equivalent to the minimum value of Eq. (1).

III. BASIC FIREFLY ALGORITHM

In order to simplify the FA, Xinshe Yang proposed three idealized assumptions [1], as following:

(1) All fireflies have no gender distinction and they are all attracted to each other.

(2) The attraction is inversely proportional to the distance. As the distance between fireflies increases, the attraction will gradually decrease.

(3) Each firefly only moves to a firefly that is brighter than it, and if it is the brightest of the population, it will move randomly.

The objective function value of the problem to be optimized determines the brightness of the firefly. For example, for a minimized problem to be optimized, the brightness is inversely proportional to the value of the objective function, i.e., the smaller the value of the objective function, the greater the brightness of the firefly.

Let N is population size, D is the dimension of the variable, and the position of i-th firefly in space is $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})^T (i = 1, 2, \cdots, N)$. For any two different firefly i and firefly j in the population, the distance between them is r_{ij}

$$r_{ij} = X_i - X_j = \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2} \tag{5}$$

where r_{ij} is the distance between firefly i and firefly j, X_i and X_j are the i-th and j-th fireflies in the population, x_{ik} is the

k-th component of the i-th firefly in the population, and x_{jk} is the k-th component of the j-th firefly in the population.

Since the brightness of firefly i decreases with the increase of distance and the absorption of air, the relative brightness of firefly i to firefly j can be defined as

$$I_{ij}\left(r_{ij}\right) = I_0 e^{-\gamma r_{ij}^2} \tag{6}$$

where I_0 is the absolute brightness of firefly i, which is equal to the value of the objective function corresponding to the position of firefly i; γ is the light absorption coefficient, usually $\gamma \in [0.01, 100]$.

The attractiveness of each firefly is defined as follows

$$\beta_{ij}(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \tag{7}$$

where β_0 is the initial attractive value when $r = 0$.

When the light emitted by X_j is brighter than that of X_i, X_i will move towards X_j, and the position update formula of X_i is as follows

$$x_{ik}(t+1) = x_{ik}(t) + \beta_{ij}\left(x_{jk}(t) - x_{ik}(t)\right) + \alpha\,(rand - 0.5) \tag{8}$$

where t is the number of iterations; α is the step factor; *rand* is a random number between [0,1]; The second term $\beta_{ij}(x_{jk}(t)$-$x_{ik}(t))$ is called the attractive term, and the third term α(rand-0.5) is called the random perturbation term.

Let t be the number of iterations and *MaxGen* be the maximum number of iterations. The pseudo code of the basic FA is shown in TABLE 1.

TABLE 1. Pseudo-code of the basic FA.

```
Algorithm 1: Pseudo-code of FA
Begin
Randomly initializes all fireflies in the swarm;
while  t <= MaxGen do
    for  i=1 to N do
      for  j=1 to N do
        if  firefly j is better than firefly i then
            Generate a new firefly according to equation (8);
            Evaluate the new solution;
        end if
      end for
    end for
    Rank the fireflies and find the current best;
    t←t+1;
End while
End
```

IV. THE IMPROVE FIREFLY ALGORITHM: HFA

A. A NEW POSITION UPDATE METHOD

In the basic FA and the improved FA given in literature [23], the variation law of the attraction β_{ij} is illustrated in Figure.1.

Figure.1 shows that the relative attraction β_{ij} decreases with the rise of the distance r_{ij} between X_i and X_j. When $r_{ij} \geq 3$, the value of β_{ij} is close to 0. At this time, the value of the attractive term in the position update formula approaches 0, and the attractive term does not work, and only the random perturbation term is active. Thus, the position

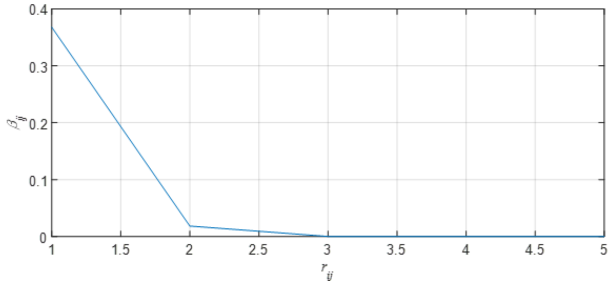


FIGURE 1. Variation law of the relative attractiveness β_{ij}.

update formula is not heuristic, the algorithm degenerates into a random search, and the convergence speed is slower. In the meantime, since the value of the random perturbation term is relatively small, the global search ability of the algorithm is weak, and it is easy to fall into local optimum. Besides, for a given problem to be optimized, If the variable has a larger range of values (e.g., $-100 \leq x_{ik} \leq 100$ $i = 1, 2, \cdots, n$ $k = 1, 2, \cdots, D$), the distance will be relatively large between the fireflies in the population after the random initialization, the β_{ij} goes to 0, and the attractive term does not work. With the rise in the number of iterations, the distance between the fireflies in the population will decrease, and the value of β_{ij} increases. When $r_{ij} < 3$, the attractive term will start to work, and with the rise of the value of the β_{ij}, the role of the attractive term is gradually strengthened, and the heuristic of position update will be enriched, which will improve the convergence speed of the algorithm. A novel position update formula is given for the above problem. The position update formula is written as follows:

$$\begin{aligned} X_i(t+1) = X_i(t) &+ \beta_{ij}R_f\left(X_j(t) - X_i(t)\right) \\ &+ \beta_{i,best}\left(1 - R_f\right)\left(X_{best}(t) - X_i(t)\right) \\ &+ 2r_{ij}\,(rand - 0.5) \end{aligned} \tag{9}$$

where β_{ij} denotes the relative attraction; R_f is the regulating factor; $\beta_{i,best}$ is the relative attraction between the firefly i and the optimal firefly in the t-th generation population; X_{best} represents the optimal firefly in the t-th generation population.

β_{ij}, R_f and $\beta_{i,best}$ are calculated as follows:

$$\beta_{ij} = \beta_{\max} - (\beta_{\max} - \beta_{\min})e^{-\gamma r_{ij}^2} \tag{10}$$

$$\beta_{i,best} = \beta_{min} + (\beta_{\max} - \beta_{\min})e^{-\gamma r_{i,best}^2} \tag{11}$$

$$R_f = \left(e^{-\frac{t}{MaxGen}}\right)^2 \tag{12}$$

where β_{min} denotes the minimum attraction; β_{max} is the maximum attraction. Both of them are constants, usually $\beta_{min} = 0.2$, $\beta_{max} = 1$ [17], and $r_{i,best}$ refers to the distance between firefly i and the optimal firefly in the t-th generation.

Since r_{ij} denotes the distance between firefly i and firefly j in the population, r_{ij} will gradually decrease as the number of iterations increases. Therefore, value of $e^{-\gamma r_{ij}^2}$ will gradually increase as the number of iterations increases. Since β_{min}, β_{max} and γ are constants, it can be known from Eq. (10)

that β_{ij} will gradually decrease as the number of iterations increases. Moreover, Eq. (12) suggests that the value of R_f will decrease as the number of iterations increases. The value of $\beta_{ij}R_f$ will decrease as the number of iterations increases.

Since $r_{i,best}$ is the distance between firefly i and optimal firefly in population, $r_{i,best}$ will decrease as the number of iterations increases. As a result, with the rise in the number of iterations, value of $e^{-\gamma r_{i,best}^2}$ will gradually increase. In the meantime, since β_{min}, β_{max} and γ are constants, it can be known from Eq. (11) that $\beta_{i,best}$ will gradually increase as the number of iterations increases. Furthermore, since the value of R_f gradually decreases as the number of iterations increases, the value of (1-R_f) gradually increases with the rise of the number of iterations. The value of $\beta_{i,best}$(1-R_f) will gradually increase as the number of iterations increases.

In summary, since the value of $\beta_{ij}R_f$ will decrease with the rise of the number of iterations, the value of $\beta_{i,best}$ (1-R_f) will gradually increase as the number of iterations increases. It can be seen from Eq. (9) that the second term $\beta_{ij}R_f$ ($X_j(t)$-$X_i(t)$) in Eq. (9) gradually decreases with the rise of the number of iterations, and the global search ability of the algorithm gradually decreases as the number of iterations increases. Also, since the value of $\beta_{i,best}$(1-R_f) increases with the rise in iterations number, the value of $\beta_{i,best}$(1-R_f) ($X_{best}(t)$-$X_i(t)$) will be close to X_{best}. Besides, under the action of the random perturbation term $2r_{ij}$(rand-0.5), the value of $X_i(t+x1)$ will be locally searched near X_{best} in the later stage of the iteration, i.e., in the later stage of the iteration, the local search ability of the algorithm is enhanced.

B. COMBINED MUTATION OPERATOR

In the meta-heuristic algorithm, the exploitation means the ability of the population to converge to the best solution as soon as possible, and the exploration refers to the ability of the algorithm to explore different areas within the search space [33]. The optimization ability of meta-heuristic algorithm depends on the balance between the exploitation and exploration ability of the algorithm. In order to better balance the exploitation and exploration ability of firefly algorithm and improve the solving quality of firefly algorithm, a combined mutation operator is introduced into HFA. The specific combined mutation operators are as follows:

The i-th firefly with the participation mutation operation is $X_i(t)(i = 1, 2\ldots, N)$, the firefly obtained after the mutation is $X'i(t)(i = 1, 2, \ldots, N)$, $r1$, $r2$, $r3$, $r4$ and $r5$ are random integers randomly selected from the set $[1, 2, \ldots, N]$, and $r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i$. X_{best} is the optimal firefly in the current population. Then each single mutation operator in the combined mutation operator is:

$$X_i'(t) = X_{r1}(t) + F(X_{r2}(t) - X_{r3}(t)) \tag{13}$$

$$X_i'(t) = X_{best}(t) + F(X_{r1}(t) - X_{r2}(t)) \tag{14}$$

$$X_i'(t) = X_i(t) + F(X_{best}(t) - X_i(t)) + F(X_{r1}(t) - X_{r2}(t)) \tag{15}$$

$$X_i'(t) = X_{best}(t) + F(X_{r1}(t) - X_{r2}(t)) + F(X_{r3}(t) - X_{r4}(t)) \tag{16}$$

$$X_i'(t) = X_{r1}(t) + F(X_{r2}(t) - X_{r3}(t)) + F(X_{r4}(t) - X_{r5}(t)) \tag{17}$$

where F is the step size.

The calculation formula of F is

$$F = 0.4 + 0.6 * rand \tag{18}$$

where $rand$ is the random number between [0,1].

Eqs. (13) $\sim$ (17) can be divided into three categories. The first category has Eq. (13) and Eq. (17), which is characterized by the strong global search ability of the mutation operator and not easy to fall into local optimum, but the convergence speed is slow. The second category has the Eq. (14) and Eq. (16), which are characterized by strong local search ability and fast convergence, but easy to fall into local optimum. The third category is only the Eq. (15), which is characterized by a relatively balanced global and local search capability. The above five mutation operators are combined to obtain a combined mutation operator. The combined mutation operator can effectively balance the ability of exploitation and exploration of the algorithm and improve the solving quality of the algorithm. The pseudo code of the combined mutation operator is shown in Table 2.

TABLE 2. The pseudo-code of mutation operator.

```
Algorithm 2: The pseudo-code of mutation operator
for i=1:N
    Calculate the compression factor F by equation (18);
    P_m=rand[0.7,1];
    Select randomly r1, r2, r3, r4, r5, i∈[1,2,…,N],
    and r1≠r2≠r3≠r4≠r5≠i
    if rand< P_m
        if rand<0.15
            X'_i(t)= X_r1 (t)+F(X_r2 (t) − X_r3 (t))
        else if rand<0.4
            X'_i(t)= X_best (t)+F(X_r1 (t) − X_r2 (t))
        else if rand<0.6
            X'_i(t)=X_i(t) +F(X_best (t) − X_i(t))+F(X_r1 (t) − X_r2 (t))
        else if rand<0.85
            X'_i(t)= X_best (t)+F(X_r1 (t) − X_r2(t))+F(X_r3 (t) − X_r4(t))
        else
            X'_i(t)= X_r1 (t)+F(X_r2 (t) − X_r3(t))+F(X_r4 (t) − X_r5(t))
        end if
    else
        X'_i(t)= X_i(t)
    end if
    if f (X'_i(t))<f (X_i(t))
        Replace the i-th solution with the new solution X'_i(t)
    end if
end for
```

In Table 2, P_m is the mutation probability and is a random number between [0.7, 1]. Larger mutation probability can avoid the algorithm falling into local optimum.

C. CHAOTIC SEARCH OF FIREFLIES

Chaotic search is characterized by ergodicity, randomness and regularity. Since chaotic motion can traverse all states in a certain range without repeating according to its own laws. If chaotic variables are used for optimized search, it is obviously better than random search. At the same time, chaotic search overcomes the disadvantage that the traditional optimization method is easy to fall into local optimization. Accordingly, chaotic search has been widely used in meta-heuristic algorithms (e.g., bat algorithm, firefly algorithm, and genetic algorithm), and has achieved good results.

Chaotic mapping is an effective way to achieve chaotic search. The commonest chaotic map is Logistic map. The Logistic map can be described as follows:

$$\begin{aligned} &cs(1) = rand \\ &cs(k+1) = \alpha \times cs(k)(1 - cs(k)), \quad k = 1, 2, \cdots, N \end{aligned} \tag{19}$$

where α denotes a constant. When $\alpha = 4$, the chaotic search is in a completely chaotic state, and the ergodicity of chaotic search is the optimal [23]. $cs(k)$ is the number of chaos generated at the k-th time. N is the population size.

The problem with the Logistic mapping given in the existing literatures is expressed as: When $cs(k) = 0.5$, $cs(k+2) = 0$. If $cs(k+2) = 0$, $cs(k+3), cs(k+4), \ldots, cs(N)$ are all 0. In this case, chaotic search cannot traverse the [0,1] interval. To avoid this phenomenon, whether $cs(k)$ is 0.5 should be judged. If $cs(k) = 0.5$, let $cs(k+1) = eps$ (eps is a sufficiently small positive number), and chaotic numbers are generated continuously according to Eq.(19); if $cs(k) \neq 0$, chaotic numbers are generated continuously according to Eq. (19). In addition, if $cs(k)$ is equal to 0, 0.25, 0.75 and 1, chaotic search cannot traverse the whole search space. The processing principle is the same as the above.

Eq. (19) suggests that the chaotic numbers generated by Logistic mapping can only traverse the [0,1] interval. For a given optimization problem, the search space is not necessarily [0,1]. Thus, it is necessary to map the chaotic sequence generated by the Logistic map to the search space. The method for mapping chaotic sequence to search space is defined as follows:

$$CS(k) = L + (U - L)\frac{cs(k) - \min(cs)}{\max(cs) - \min(cs)} \tag{20}$$

where U denotes the upper limit of the search space; L is the lower limit of the search space; $\min(cs)$ is the minimum value in the chaotic sequence cs; $\max(cs)$ represents the maximum value in the chaotic sequence cs.

The chaotic sequence generated by Eq. (19) can be traversed through the entire search space $[L, U]$ after the transformation of Eq. (20). If all the fireflies in the population are randomly moved by chaotic search, the global search ability of the algorithm will be greatly enhanced. Therefore, each firefly in the population can randomly move according to chaotic search, and the formula for random movement according to chaotic search is as follows:

$$X_i' = (1 - \mu)(X_c - X_i) + \mu CS(k) \tag{21}$$

$$\mu = \frac{1}{t} \tag{22}$$

$$X_c = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{23}$$

where t denotes the current number of iterations; X_c is the center of gravity of the population.

The chaotic search steps of fireflies in the population are as follows:

(1) Generate a chaotic sequence of length N according to Eq. (19);

(2) The generated chaotic sequence is mapped to the search space $[L, U]$ by Eq. (20);

(3) Generate N new solutions according to Eq. (21). After that, for each firefly, the fitness value of the firefly after the chaotic search operation is calculated, and compared with the fitness value of the firefly before the chaotic search, and the better firefly and its fitness are retained.

The pseudo code for chaotic search is shown in Table 3.

TABLE 3. The pseudo-code of chaotic search.

Algorithm 3: The pseudo-code of Chaotic search
generate chaotic sequence cs with length n according to Eq.(19)
mapping chaotic sequence to search space according to Eq.(20)
for l=1 to N do
Generate new solution $X_i'(t)$ according to Eqs.(21)~(23).
If $f(X_i'(t)) < f(X_i(t))$
Replace the i-th solution with the new solution $X_i'(t)$
end *if*
end *for*

D. REMOVING SIMILAR INDIVIDUALS

In the later iteration stage of firefly algorithm, fireflies gradually concentrated near the optimal firefly, and the difference between fireflies decreased gradually, which may lead to the possibility of multiple similar individuals and made the firefly algorithm easy to fall into local optimization. In order to improve the difference between fireflies in the population, this paper proposes a method to remove similar individuals.

To illustrate which fireflies in the population are similar, first give an indicator S that evaluates the similarity of fireflies, namely:

$$S = \|X_i - X_j\| = abs\left[(x_{i1} - x_{j1}), \cdots, (x_{iD} - x_{jD})\right] i \neq j \tag{24}$$

where X_i and X_j are any two different fireflies in the population.

Then the threshold δ is set. When the value of each dimension in S is less than δ, it is considered that X_i and X_j are similar fireflies. If there are two or more similar fireflies, the q similar fireflies with poor fitness are removed and only one similar firefly is retained. In addition, in order to keep the size

TABLE 4. The pseudo-code of removing similar individuals.

Algorithm 4: The pseudo-code of Removing Similar Individuals
for i=1:N
for j=i:N
if $S<\delta$
Initialize a new firefly X_{new}
if $f(X_i)<f(X_j)$
$X_j = X_{new}$
else
$X_i = X_{new}$
end *if*
end *if*
end *for*
end *for*

of the population unchanged, q fireflies are randomly generated. The pseudo code to remove similar firefly operations is shown in Table 4.

E. EVOLUTION STRATEGY FOR IMPROVED HYBRID FIREFLY ALGORITHM

Taking the problem of minimizing as an example, the Evolution Strategy of HFA is as follows:

(1) Parameter setting: the population size N, the solution accuracy ε, the penalty factor M.

(2) Initialization of the population. Generate an initial population according to the following formula

$$X_i = L + rand(1, D).*(U - L), \quad i = 1, 2, \cdots, N \tag{25}$$

where X_i denotes the randomly generated i-th firefly; U and L are the upper and lower vector of the variable; *rand* is the uniformly distributed random vector between [0,1]; D represents the dimension of the variable; ".*" is the multiplication of elements of the same position in two vectors.

(3) Calculate the objective function value of each firefly in the population and find the optimal firefly in the population, denoted as X_{best}.

(4) Each firefly is updated according to Eq. (9), and then the fitness value of the firefly after the position update is calculated, and compared with the fitness value of the firefly before the update, and the better firefly and its fitness value are retained.

(5) According to the mutation probability P_m, the combined mutation operation is performed on all fireflies. For each firefly, the fitness value of the firefly after the mutation operation is calculated, and compared with the fitness value of the firefly before the mutation, and the better firefly individual and its fitness value are retained.

(6) All fireflies perform chaotic search operation according to Eq. (21). For each firefly, calculate the fitness value of the firefly after the chaotic search operation, and compare the fitness values of the firefly before the chaotic search operation, and retain the better firefly and its fitness value.

(7) All fireflies perform removing similar individuals operation.

(8) Judge whether the iteration termination condition is satisfied; if it is satisfied, the iteration is stopped, and the optimal solution and the optimal value are output; if not, the process returns to step (3).

The pseudo code of the HFA is listed in TABLE 5.

TABLE 5. The pseudo-code of HFA.

Algorithm **5:** The pseudo-code of **HFA**
Begin
Initialize the firefly population;
Initialization algorithm parameters;
while $t <= maxgen$ do
for i=1 to N do
for j=1 to N do
if firefly j is better than firefly i then
Calculate the distance r_{ij} between firefly i and j according to Eq. (5);
Calculate the distance r_{best} between the best firefly and j according to Eq. (5);
Calculate β_{ij} and β_{best}. According to Eq. (10) and Eq. (11)
Firefly j moves according to Eq. (9);
end *if*
Update the brightness of fireflies
end *for*
end *for*
All fireflies perform combined mutation operations according to the probability of variation by algorithm 2
All fireflies perform chaotic search operations by algorithm 3
Removing similar individuals from all fireflies by algorithm 4
Sort all fireflies and find out the first N better individuals.
End *while*
Output the best solution and optimal value
End

The flow chart of the HFA is shown in Figure 2.

F. TIME COMPLEXITY

Let $O(f)$ be the computational time complexity of the fitness evaluation function f $(\cdot)$. This paper uses the classic full attraction model. The total number of the movements of the swarm at each iteration in the full attraction model is $N(N+1)/2$. It is clear that the time complexity of the algorithm proposed in this paper is $O(MaxGen*N2*f)$, where $MaxGen$ is the maximum number of generations.

V. SIMULATION EXPERIMENTS AND DATA ANALYSIS

HIFA was compared to eight improved algorithms and basic FA algorithm. The eight improved algorithms that participated in the comparison are as follows: (1) improved FA algorithm [19], [23], [34], [35]; (2) other intelligent algorithms [36]–[39]. When the algorithm reaches the maximum number of iterations, the iteration is stopped and the calculation result is output.

A. TEST ON CEC 2017 OPTIMIZATION PROBLEMS

In order to verify the solving precision of the HFA, 28 CEC2017 test problems are selected, TABLE 6 show these test problems, the detail of which can be found in reference [40].

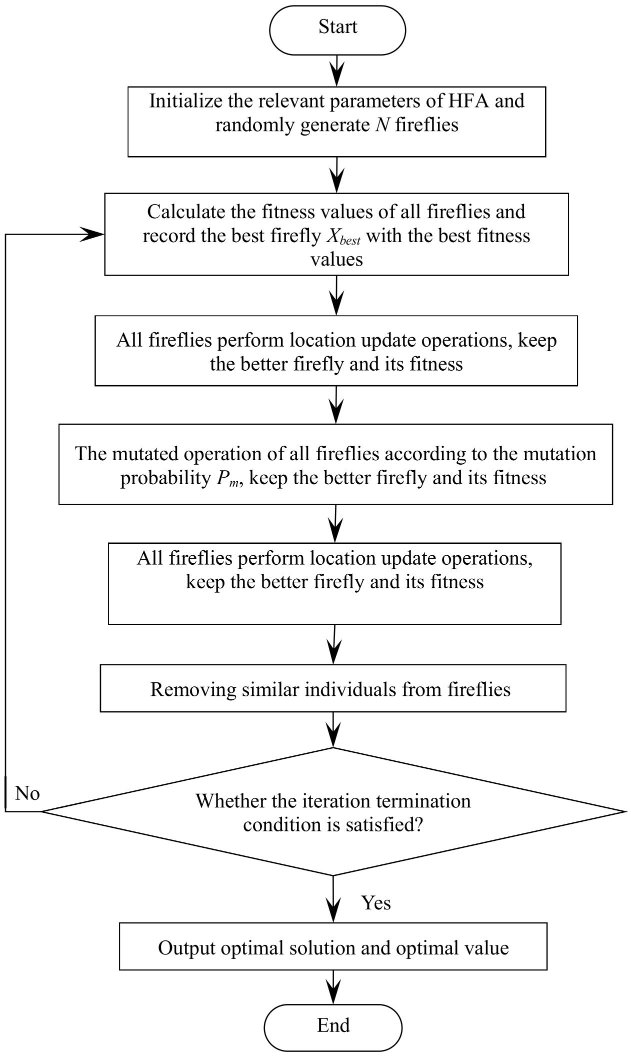


FIGURE 2. The flow chart of the HFA.

In table 6, D is the number of decision variables, I is the number of inequality constraints, and E is the number of equality constraints.

B. ALGORITHM PERFORMANCE EVALUATION INDEX

In this section, in order to measure the performance of HFA, the mean value, standard deviation value, t-test values, and Friedman average ranking used as performance evaluation indicators. The mean value is defined as the average function value achieved by the algorithm out of the total number of independent runs R times. The mean value is abbreviated as *MEAN*, and is represented in Eq. (26).

$$MEAN = \frac{\sum_{i=1}^{R} f_i}{R} \tag{26}$$

The standard deviation is defined as the actual deviation that exists between the average function values achieved in R times runs. The standard deviation is abbreviated as SD, and is represented in Eq. (27).

$$SD = \sqrt{\frac{\sum_{i=1}^{R} (f_i - MEAN)^2}{R}} \tag{27}$$

The t-test values are calculated for every function using its mean and standard deviation in each existing algorithm. The t-test value is abbreviated as T_{val}, and is calculated using the Eq. (28).

$$T_{val} = \frac{MEAN_1 - MEAN_2}{\sqrt{\frac{SD_1 - SD_2}{R} + \varepsilon}} \tag{28}$$

where $MEAN_1$, $MEAN_2$ and SD_1, SD_2 are respectively the mean and standard deviation values of HFA algorithm and other algorithms, R is the number of independent runs of the algorithm, ε is used to avoid zero-division-error, and ε is the smallest constant in the computer. The t-test has been carried out with the significance level of $\alpha = 0.05$. The winning (w), tie (t) and lost (l) cases of t-test values for HFA over existing algorithms are represented in the last row of each table with the attributes named $w/t/l$. For each algorithm that compares with HFA, w represents the number of t-test values less than 0, t represents the number of t-test values equal to 0, and l represents the number of t-test values greater than 0.

In order to check the significant difference of the proposed HFA, the Friedman average ranking test has been used to compare the performance of the HFA and other algorithms. With this test, the relative ranking of each algorithm has been calculated based on the mean values for every function and finally the average ranking and final ranking of all the algorithms are given.

C. PARAMETER SETTINGS

In order to obtain a fair performance comparison basis, the population size $N = 30$, the maximum number of iterations $MaxGen = 20000$, the penalty factor $M = 10^8$, and the number of statistics $tjcs = 100$. The other parameter settings for various algorithms are shown in Table 7.

D. THE CONTRIBUTION OF DIFFERENT STRATEGIES TO IMPROVE THE PERFORMANCE OF FA

The improved position update method based on the basic FA algorithm is abbreviated as FA-P. The chaotic search operation based on the basic FA algorithm is abbreviated as FA-C. The mutation operation based on the basic FA algorithm is abbreviated as FA-M. The basic FA algorithm is abbreviated as FA. In order to verify the contribution of different strategies to improve the performance of FA, this paper selected 28 test problems in CEC2017 with dimension D = 10. The parameter settings and iteration termination conditions are detailed in Section *V. C*. The results are shown in TABLE 8 and TABLE 9. TABLE 8 provides the mean values obtained using the FA, FA-P, FA-C, and FA-M for all 28 test problems. TABLE 9 provides the results of Friedman average ranking test.

TABLE 6. Details of 28 CEC 2017 optimization problems.

Function	Search Range	Type of Objective	Number of Constraints E	Number of Constraints I
C01	$[-100,100]^D$	Non Sepearble	0	1 Separable
C02	$[-100,100]^D$	Non Sepearble Rotated	0	1 Non Separable, Rotated
C03	$[-100,100]^D$	Non Sepearble	1 Separable	1 Separable
C04	$[-10,10]^D$	Sepearble	0	2 Separable
C05	$[-10,10]^D$	Non Sepearble	0	2 Non Separable, Rotated
C06	$[-20,20]^D$	Sepearble	6	0 Separable
C07	$[-20,20]^D$	Sepearble	2 Separable	0
C08	$[-100,100]^D$	Sepearble	2 Non Separable	0
C09	$[-10,10]^D$	Sepearble	2 Non Separable	0
C10	$[-100,100]^D$	Sepearble	2 No Separable	0
C11	$[-100,100]^D$	Sepearble	1 Separable	1 Non Separable
C12	$[-100,100]^D$	Sepearble	0	2 Separable
C13	$[-100,100]^D$	Non Sepearble	0	3 Separable
C14	$[-100,100]^D$	Non Sepearble	1 Separable	1 Separable
C15	$[-100,100]^D$	Sepearble	1	1
C16	$[-100,100]^D$	Sepearble	1 Non Separable	1 Separable
C17	$[-100,100]^D$	Non Sepearble	1	1 Separable
C18	$[-100,100]^D$	Sepearble	0	2 Non Separable
C19	$[-50,50]^D$	Sepearble	0	2 Non Separable
C20	$[-100,100]^D$	Non Sepearble	0	2
C21	$[-100,100]^D$	Rotated	0	2 Rotated
C22	$[-100,100]^D$	Rotated	1 Rotated	3 Rotated
C23	$[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C24	$[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C25	$[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C26	$[-100,100]^D$	Rotated	1 Rotated	1 Rotated
C27	$[-100,100]^D$	Rotated	1 Rotated	2 Rotated
C28	$[-50,50]^D$	Rotated	0	2 Rotated

TABLE 7. Parameters control of different algorithms.

Algorithms	Year	Parameters
HFA	2019	β_0=1, $\beta_{\min}$=0.5, γ=1, a=4
FA	2008	β_0=1, γ=1, α=0.2
RFA	2012	γ=2, $randomdirection$=20
UFA	2019	β_0=1.5, γ_0=0.9, a=4, DPP=3
GDFA	2019	β_0=1, γ=1, a=4, k=5
NEFA	2019	M=6, $\gamma=1/\Gamma^2$, β_0=1
SPPSO	2019	p_1=0.7, p_2=0.2, p_3=0.1
ADE	2018	$threshold$=10^{-3}, $tolerance$=10^{-6}
GA-DEx	2018	μ=0.7, σ=0.1
CSPSO	2019	p_a=0.25, a_{min}=0.01, a_{max}=0.5

TABLE 8. Calculation results of different strategies.

Function	Evaluation indicator	FA	FA-P	FA-C	FA-M
C01	Mean	4.63E+06	1.39E+02	2.04E+03	8.56E-14
C02	Mean	3.94E+06	6.76E+00	2.02E+03	8.95E-14
C03	Mean	8.30E+15	2.51E+04	1.01E+08	8.01E+07
C04	Mean	9.24E+12	6.47E+01	1.69E+02	1.13E+01
C05	Mean	2.64E+18	2.80E-01	1.30E+05	8.99E-13
C06	Mean	6.04E+11	9.68E+08	2.14E+09	1.56E+08
C07	Mean	1.05E+15	2.51E+11	2.26E+11	1.38E+02
C08	Mean	5.02E+19	2.10E+12	4.05E+14	-1.35E-03
C09	Mean	7.57E+24	1.92E+01	1.62E+11	-4.98E-03
C10	Mean	1.18E+20	1.15E+13	4.36E+15	-5.10E-04
C11	Mean	1.25E+19	5.26E+11	1.60E+15	-3.48E+00
C12	Mean	4.28E+18	1.92E+11	5.36E+14	2.16E+01
C13	Mean	3.82E+18	3.84E+13	3.65E+14	7.46E-14
C14	Mean	9.93E+18	3.88E+11	3.85E+14	2.48E+00
C15	Mean	4.80E+18	1.68E+01	2.16E+14	1.27E+02
C16	Mean	4.22E+18	7.48E+01	1.76E+14	3.58E+01
C17	Mean	3.92E+18	8.69E+10	4.31E+14	1.21E+10
C18	Mean	6.10E+30	6.39E+18	4.94E+23	3.66E+01
C19	Mean	5.29E+17	1.77E+16	1.78E+16	1.76E+16
C20	Mean	2.25E+01	1.04E+00	1.46E+00	6.30E-01
C21	Mean	3.86E+18	7.72E+01	3.82E+14	1.39E+01
C22	Mean	3.37E+18	1.80E+12	3.12E+14	7.97E-01
C23	Mean	7.44E+18	3.75E+00	8.41E+14	2.52E+00
C24	Mean	3.07E+18	1.93E+01	1.60E+14	9.59E+00
C25	Mean	3.39E+18	7.23E+01	1.10E+14	2.70E+01
C26	Mean	3.02E+18	1.21E+10	3.02E+14	1.21E+10
C27	Mean	2.75E+30	1.59E+14	5.69E+23	1.00E+11
C28	Mean	5.29E+17	1.77E+16	1.78E+16	1.77E+16

TABLE 9. Statistical analysis of friedman test for the results.

Algorithms	FA	HFA-P	HFA-C	HFA-M
Friedman test	4.00	1.93	2.96	1.11
Ranking	4	2	3	1

As can be seen from Table 8, the mean values of FA-P, FA-C, and FA-M are better than FA. According to the Friedman test shown in Table 9, FA-M (rank = 1.11) ranks first in all algorithms, followed by FA-P (rank = 1.93) and FA-C (rank = 2.96) respectively. Therefore, the mutation operation contributed the most to the improvement of FA performance,

TABLE 10. Calculation results of various algorithms (D = 10).

Function	Evaluation indicator	HFA	FA	NEFA	UFA	RFA	GDFA	SPPSO	ADE	GA-DEx	CSPSO
C01	Mean	3.39E-14	4.63E+06	2.56E+03	1.11E+04	9.00E-02	1.52E+04	3.16E-01	0.00E+00	1.62E+04	8.34E-04
	Std.	1.44E-14	1.27E+06	1.09E+03	8.60E+03	2.37E-02	3.32E+03	1.83E-01	0.00E+00	1.79E+03	3.44E-04
	T_{val}	~	-8.32E+00	-7.42E+00	-4.07E+00	-1.20 E+01	-1.45E+01	-5.48E+00	7.43E+00	-2.86E+01	-7.66E+00
C02	Mean	3.24E-14	3.94E+06	7.67E+03	3.77E+16	4.29E+14	1.71E+04	3.64E-01	0.00E+00	1.71E+04	8.21E-04
	Std.	2.34E-14	1.96E+06	6.10E+03	1.04E+17	8.59E+14	1.14E-04	2.13E-01	0.00E+00	2.42E+01	4.22E-04
	T_{val}	~	-3.19E+00	-3.97E+00	-1.14E+00	-1.58E+00	-4.73E+08	-5.40E+00	4.37E+00	-2.24E+03	-6.16E+00
C03	Mean	2.14E+04	8.30E+15	4.07E+03	1.08E+04	1.00E-01	5.77E+03	2.11E-01	0.00E+00	5.98E+03	7.12E-04
	Std.	9.15E+03	1.66E+16	2.04E+03	1.66E+04	8.44E-03	1.97E+00	8.16E-02	0.00E+00	1.70E+02	3.66E-04
	T_{val}	~	1.58E+00	5.83E+00	1.76E+00	7.38E+00	5.38E+00	7.38E+00	7.38E+00	5.31E+00	7.38E+00
C04	Mean	2.95E+00	9.24E+12	1.29E+03	2.27E+03	1.14E+04	1.86E+02	5.52E+01	0.00E+00	1.71E+02	3.03E+01
	Std.	5.92E+00	5.03E+11	6.20E+02	2.64E+03	1.76E+03	2.45E-02	9.81E+00	0.00E+00	2.81E+00	7.84E+00
	T_{val}	~	-5.49E+01	-6.55E+00	-2.72E+00	-2.05E+01	-9.79E+01	-1.44E+01	1.57E+00	-8.10E+01	-8.79E+00
C05	Mean	4.22E-03	2.64E+18	2.66E+15	8.65E+16	4.55E+17	9.98E+13	8.66E+00	0.00E+00	1.57E+05	3.55E+00
	Std.	1.20E-02	2.52E+17	3.70E+15	1.08E+17	9.11E+17	2.15E+13	1.00E+00	0.00E+00	4.11E+00	1.74E+00
	T_{val}	~	-3.00E+01	-2.27E+00	-2.53E+00	-1.58E+00	-1.47E+01	-2.73E+01	1.12E+00	-1.21E+05	-6.44E+00
C06	Mean	4.27E+07	6.04E+11	4.10E+07	8.48E+02	1.41E+03	1.73E+09	9.31E+02	1.25E+06	2.15E+09	1.15E+03
	Std.	6.64E+07	1.92E+11	1.10E+08	3.81E+02	6.13E+02	1.01E+09	3.61E+02	2.47E+06	1.00E+09	6.09E+02
	T_{val}	~	-6.77E+00	4.11E-02	2.03E+00	2.03E+00	-5.26E+00	2.03E+00	1.97E+00	-6.65E+00	2.03E+00
C07	Mean	-5.22E+01	1.05E+15	4.43E+12	6.13E+09	-4.67E+02	5.58E+05	4.95E+08	1.00E+11	1.46E+12	-9.54E+01
	Std.	4.50E+01	4.83E+14	1.83E+12	1.84E+10	1.16E+02	1.30E+06	1.48E+09	3.62E+10	1.56E+12	6.17E+01
	T_{val}	~	-3.71E+00	-7.64E+00	-1.05E+00	1.05E+01	-1.36E+00	-1.05E+00	-8.75E+00	-2.96E+00	1.79E+00
C08	Mean	-1.35E-03	5.02E+19	1.03E+14	7.58E+14	7.15E+05	2.11E+13	7.55E+04	-1.35E-03	2.11E+13	1.54E+00
	Std.	2.12E-06	3.44E+19	7.48E+13	6.94E+14	1.46E+05	1.22E+06	3.86E+04	1.16E-07	1.55E+09	1.42E+00
	T_{val}	~	-1.46E+00	-4.37E+00	-3.46E+00	-1.55E+01	-5.47E+07	-6.19E+00	3.87E+00	-4.31E+04	-3.42E+00
C09	Mean	-4.98E-03	7.57E+24	3.56E+15	2.37E+22	7.76E+08	1.37E+14	1.03E+01	-2.53E-03	2.64E+14	9.22E+00
	Std.	0.00E+00	1.64E+24	9.54E+15	6.25E+22	1.55E+09	2.02E+14	4.37E+00	3.95E-04	4.47E+14	2.59E+00
	T_{val}	~	-1.15E+01	-1.18E+00	-1.20E+00	-1.58E+00	-2.15E+00	-7.44E+00	-1.96E+01	-1.87E+00	-1.12E+01
C010	Mean	-5.07E-04	1.18E+20	1.92E+16	3.40E+16	1.86E+07	4.72E+16	1.93E+06	-5.10E-04	4.72E+16	4.99E+01
	Std.	1.36E-06	3.57E+19	1.15E+16	4.77E+16	2.67E+06	6.99E+08	2.24E+06	6.20E-08	5.16E+12	3.87E+01
	T_{val}	~	-7.31E+00	-5.27E+00	-2.26E+00	-2.20E+01	-2.14E+08	-2.72E+00	6.47E+00	-2.89E+04	-4.07E+00
C11	Mean	-3.48E+00	1.25E+19	2.40E+14	9.97E+39	1.12E+16	2.01E+13	5.67E+08	7.49E+09	6.23E+39	1.94E+07
	Std.	9.90E-08	3.67E+18	2.12E+14	2.53E+40	9.53E+15	2.46E+11	1.18E+09	9.27E+09	1.62E+40	3.58E+07
	T_{val}	~	-7.60E+00	-3.58E+00	-1.24E+00	-3.71E+00	-2.58E+02	-1.52E+00	-2.56E+00	-1.21E+00	-1.72E+00
C12	Mean	1.06E+01	4.28E+18	4.09E+03	1.52E+04	4.46E+01	1.69E+04	1.49E+04	2.46E+09	5.41E+03	1.42E+04
	Std.	1.13E+01	5.47E+17	2.24E+03	3.82E+03	1.45E+01	3.18E+03	4.76E+03	5.22E+09	1.30E+00	2.72E+03
	T_{val}	~	-2.16E+01	-5.75E+00	-1.26E+01	-5.85E+00	-1.68E+01	-9.87E+00	-1.49E+00	-1.50E+03	-1.64E+01
C13	Mean	4.00E-14	3.82E+18	7.53E+15	6.88E+16	1.29E+01	2.81E+15	1.04E+02	4.36E+18	8.54E+15	1.11E+14
	Std.	3.48E-14	1.03E+18	5.58E+15	8.43E+16	1.24E+00	2.56E+07	7.55E+01	9.21E+17	3.80E+13	1.90E+14
	T_{val}	~	-8.58E+00	-4.27E+00	-2.58E+00	-3.30E+01	-3.48E+08	-4.34E+00	-1.50E+01	-7.10E+02	-1.85E+00
C14	Mean	2.40E+00	9.93E+18	2.04E+01	1.95E+01	2.01E+01	2.10E+01	1.23E+00	2.08E+01	2.11E+01	1.60E+01
	Std.	7.57E-02	1.53E+18	5.49E-01	1.26E+00	1.48E-01	1.51E-01	5.57E-01	5.85E-02	7.52E-02	7.99E+00
	T_{val}	~	-1.74E+01	-1.03E+02	-4.29E+01	-3.36E+02	-3.48E+02	6.58E+00	-6.07E+02	-5.55E+02	-5.38E+00
C15	Mean	1.71E+01	4.80E+18	9.01E+14	1.22E+16	5.57E+15	1.88E+15	2.36E+00	8.64E+00	1.61E+15	9.27E+00
	Std.	2.45E+00	6.73E+17	1.41E+15	8.90E+15	2.83E+15	3.27E+06	1.01E-04	4.64E-04	4.17E+14	1.88E+00
	T_{val}	~	-1.94E+01	-2.01E+00	-4.34E+00	-6.23E+00	-1.81E+09	1.90E+01	1.09E+01	-1.22E+01	8.03E+00
C16	Mean	2.20E+01	4.22E+18	4.28E+14	8.53E+15	4.05E+15	1.88E+15	6.91E+00	6.33E+00	1.84E+15	6.28E+00
	Std.	7.02E+00	3.99E+17	4.69E+14	1.01E+16	1.29E+15	7.53E+05	1.88E+00	1.44E-01	9.05E+13	8.69E-06
	T_{val}	~	-3.03E+01	-2.89E+00	-2.68E+00	-9.93E+00	-7.89E+09	6.56E+00	7.05E+00	-6.44E+01	7.07E+00
C17	Mean	1.21E+10	3.92E+18	1.14E+15	1.60E+15	1.05E+10	2.80E+15	8.50E+09	9.61E+10	2.80E+15	8.10E+09
	Std.	1.59E-01	6.92E+17	7.79E+14	1.94E+15	1.96E+09	3.91E+05	1.20E+09	4.33E-03	8.32E+10	4.32E-03
	T_{val}	~	-1.48E+01	-4.64E+00	-2.61E+00	-1.69E+01	-2.27E+10	9.49E+00	-1.67E+12	-1.07E+05	7.95E+10
C18	Mean	2.92E+01	6.10E+30	3.40E+24	3.36E+28	3.07E+27	2.83E+25	1.16E+13	1.23E+26	5.70E+28	2.58E+18
	Std.	1.27E+01	1.41E+30	6.01E+24	4.00E+28	6.12E+27	1.80E+17	3.38E+13	3.67E+26	6.17E+28	4.85E+18
	T_{val}	~	-1.05E+01	-1.79E+00	-2.66E+00	-1.58E+00	-4.97E+08	-1.08E+00	-1.06E+00	-2.92E+00	-1.68E+00

TABLE 10. *(Continued.)* Calculation results of various algorithms (D = 10).

C19	Mean	1.76E+16	5.29E+17	3.56E+16	3.57E+16	3.53E+16	1.78E+16	1.77E+16	3.70E+17	3.56E+16	3.56E+16
	Std.	4.37E+05	2.74E+13	2.49E+13	2.49E+13	1.40E+13	7.73E+12	2.89E+13	4.39E+13	1.26E+13	4.97E+13
	T_{val}	~	-6.10E+04	-2.28E+03	-2.29E+03	-7.99E+03	-8.58E+01	-1.23E+01	-2.54E+04	-4.51E+03	-1.14E+03
C20	Mean	7.91E-01	2.25E+01	2.14E+00	1.56E+00	7.13E-01	1.23E+00	7.12E-01	9.14E+00	2.14E+00	6.36E-01
	Std.	2.32E-01	9.52E-01	4.40E-01	4.79E-01	1.34E-01	1.38E-03	2.30E-01	4.12E-01	4.22E-01	1.04E-01
	T_{val}	~	-7.15E+01	-8.60E+00	-4.58E+00	-9.64E+06	-5.93E+00	7.57E-01	-5.58E+01	-8.88E+00	1.92E+00
C21	Mean	1.58E+01	3.86E+18	1.27E+04	2.23E+11	1.69E+01	1.32E+11	8.91E+10	3.50E+12	1.60E+04	1.53E+11
	Std.	1.25E+01	4.22E+17	3.44E+03	1.86E+11	5.02E+00	1.39E+11	9.12E+10	1.42E+12	7.09E+00	1.69E+11
	T_{val}	~	-2.58E+01	-1.17E+01	-3.79E+00	-2.64E+05	-3.01E+00	-3.09E+00	-7.81E+00	-3.53E+03	-2.87E+00
C22	Mean	8.07E-01	3.37E+18	6.70E+16	8.28E+17	2.53E+01	2.54E+16	5.16E+02	7.71E+19	7.67E+16	5.11E+14
	Std.	1.59E+00	3.36E+17	6.30E+16	1.24E+18	3.61E+00	8.26E+09	6.29E+02	2.26E+19	2.50E+14	8.61E+14
	T_{val}	~	-2.86E+01	-3.36E+00	-2.10E+00	-1.14E+06	-9.72E+06	-2.59E+00	-1.08E+01	-9.70E+02	-1.87E+00
C23	Mean	2.43E+00	7.44E+18	3.43E+16	8.84E+16	2.44E+00	4.04E+16	2.47E+00	3.79E+07	3.68E+16	1.22E+04
	Std.	1.01E-01	7.20E+17	3.66E+16	9.28E+16	1.23E-01	8.39E+07	3.77E-01	5.67E+07	1.22E+13	3.65E+04
	T_{val}	~	-2.95E+01	-2.96E+00	-3.01E+00	-3.44E+07	-1.52E+09	-3.74E-01	-2.11E+00	-9.51E+03	-1.05E+00
C24	Mean	9.58E+00	3.07E+18	8.64E+15	1.09E+17	1.78E+16	2.24E+16	4.87E+00	2.69E+13	2.08E+16	6.75E+00
	Std.	2.01E+00	5.83E+17	9.06E+15	1.88E+17	6.73E+15	7.24E+06	3.08E+00	8.07E+13	4.04E+15	2.88E+00
	T_{val}	~	-1.35E+01	-3.01E+00	-1.83E+00	-8.37E+00	-9.79E+09	4.05E+00	-1.05E+00	-1.63E+01	2.55E+00
C25	Mean	3.09E+01	3.39E+18	9.07E+15	2.14E+17	6.28E+00	2.24E+16	1.26E+01	2.10E+13	2.25E+16	9.42E+00
	Std.	5.26E+00	8.28E+17	7.02E+15	3.93E+17	7.45E-07	1.92E+07	5.62E+00	6.31E+13	2.28E+14	5.07E+00
	T_{val}	~	-9.78E+00	-4.08E+00	-1.72E+00	-1.41E+06	-3.70E+09	7.55E+00	-1.05E+00	-3.12E+02	9.32E+00
C26	Mean	1.21E+10	3.02E+18	7.69E+15	1.63E+16	8.10E+09	2.54E+16	8.10E+09	9.61E+10	2.54E+16	8.10E+09
	Std.	1.85E-01	1.41E+18	4.87E+15	2.68E+16	1.79E-02	6.67E+06	1.87E-01	6.07E-03	9.18E+11	3.06E-02
	T_{val}	~	-3.61E+00	-4.99E+00	-1.92E+00	-1.38E+11	-1.20E+10	4.82E+10	-1.44E+12	-8.74E+04	6.76E+10
C27	Mean	3.68E+01	2.75E+30	9.74E+26	9.42E+31	5.99E+23	5.65E+27	5.23E+16	1.24E+27	2.75E+31	5.89E+20
	Std.	5.09E-01	9.44E+29	1.30E+27	1.47E+32	1.20E+24	2.63E+19	1.55E+17	3.67E+27	5.04E+31	1.43E+21
	T_{val}	~	-6.06E+00	-2.37E+00	-2.02E+00	-1.58E+00	-6.79E+08	-1.07E+00	-1.07E+00	-1.73E+00	-1.30E+00
C28	Mean	1.77E+16	5.29E+17	3.56E+16	3.57E+16	3.53E+16	1.78E+16	1.77E+16	3.70E+17	3.57E+16	3.56E+16
	Std.	2.44E+13	3.33E+13	1.88E+13	2.69E+13	1.00E+13	2.88E+12	1.74E+13	3.30E+13	3.57E+12	2.96E+13
	T_{val}	~	-5.03E+04	-1.84E+03	-1.56E+03	-2.11E+03	-1.44E+01	-3.24E+00	-2.72E+04	-2.30E+03	-1.47E+03
T-test	*w/t/l*	~	28/0/0	26/0/2	26/0/2	21/0/7	27/0/1	18/0/10	18/0/10	27/0/1	18/0/10

followed by the improved position update method, and finally the chaotic search operation.

E. PERFORMANCE COMPARISON BETWEEN HFA AND OTHER ALGORITHMS

For the convenience of writing, basic FA algorithm is abbreviated as FA, the algorithm in [19] is abbreviated as RFA, the algorithm in [34] is abbreviated as UFA, the algorithm in [23] is abbreviated as GDFA, and the algorithm in [35] is abbreviated as NEFA, the algorithm in [36] is abbreviated as SPPSO, the algorithm in [37] is abbreviated as ADE, the algorithm in [38] is abbreviated as GA-DEx, and the algorithm in [39] is abbreviated as CSPSO. Using 28 test problems in CEC2017, HFA is compared with FA, RFA, UFA, GDFA, NEFA, SPPSO, ADE, GA-DEx and CSPSO.

The experimental results for the $D = 10$, $D = 30$, and $D = 50$ problems are shown in Table 10, Table 11, and Table12, respectively. These tables provide the mean values and standard deviations obtained using the HFA and other algorithms for all 28 test problems. The experimental results of the Friedman average ranking test for the $D = 10$, $D = 30$, and $D = 50$ problems are shown in Table 13, Table 14, and Table 15, respectively.

1) PERFORMANCES FOR THE 10-D PROBLEMS

The experimental results for the $D = 10$ problems are shown in Table 10. As can be seen from Table 10, ADE performs best on seven test problems such as C01, C02, C03, C04, C05, C08, and C10. CSPSO performs best on four test problems such as C16, C17, C20, and C26. SPPSO performs best on three test problems such as C14, C15, and C24. RFA performs best on the C07 and C25 benchmark function. UFA performs best on the C06 benchmark function. HFA performed best on the remaining 11 test problems. At the same time, it can be seen from t-test that the overall performance of HFA is better than the other nine kinds of comparison algorithms. Specifically, compared with FA, HFA has 28 test functions whose values of t-test are better than FA; compared to NEFA, HFA has 26 benchmark functions whose values of t-test are better than NEFA; compared with UFA, HFA has 26 benchmark functions whose values of t-test are better than UFA; compared with RFA, HFA has 21 benchmark functions whose values of t-test are better than RFA; compared with

TABLE 11. Calculation results of various algorithms (D = 30).

Function	Evaluation indicator	HFA	FA	NEFA	UFA	RFA	GDFA	SPPSO	ADE	GA-DEx	CSPSO
C01	Mean	7.05E-12	4.79E+15	2.78E+04	8.73E+04	1.85E+04	6.27E+04	1.99E+02	1.32E+04	6.27E+04	4.95E-01
	Std.	2.59E-12	9.58E+15	8.58E+03	1.05E+05	3.69E+04	1.60E-02	4.52E+01	2.06E+03	2.64E+01	1.11E-01
	T_{val}	~	-1.58E+00	-1.02E+01	-2.62E+00	-1.58E+00	-1.24E+07	-1.39E+01	-2.03E+01	-7.51E+03	-1.40E+01
C02	Mean	7.45E-12	8.13E+05	1.05E+05	3.90E+17	5.59E+17	6.27E+04	2.02E+02	1.22E+04	6.27E+04	3.96E-01
	Std.	1.89E-12	3.64E+05	1.02E+05	6.60E+17	1.12E+18	2.81E-03	4.89E+01	1.82E-12	1.34E-01	8.04E-02
	T_{val}	~	-7.07E+00	-3.26E+00	-1.87E+00	-1.58E+00	-7.07E+07	-1.31E+01	-1.47E+16	-1.48E+06	-1.56E+01
C03	Mean	7.40E+05	1.39E+12	1.43E+05	3.83E+05	9.98E+04	2.55E+05	6.88E+02	1.80E+04	3.10E+05	8.24E-01
	Std.	1.21E+06	1.63E+12	5.23E+04	7.44E+05	8.99E+04	6.24E+04	2.70E+02	3.72E+03	6.10E+01	2.61E-01
	T_{val}	~	-2.71E+00	1.56E+00	7.94E-01	1.67E+00	1.27E+00	1.93E+00	1.89E+00	1.13E+00	1.94E+00
C04	Mean	2.22E+01	5.37E+12	4.47E+03	1.92E+04	6.93E+04	5.11E+02	2.52E+02	1.71E+04	5.11E+02	1.70E+02
	Std.	6.57E+00	3.62E+11	9.71E+02	2.07E+04	4.31E+03	1.45E-05	4.03E+01	3.64E-12	4.25E-02	1.25E+01
	T_{val}	~	-4.69E+01	-1.45E+01	-2.92E+00	-5.09E+01	-2.35E+02	-1.78E+01	-8.20E+03	-2.35E+02	-3.32E+01
C05	Mean	1.72E-03	8.79E+17	3.68E+16	2.66E+18	9.59E+01	2.58E+13	1.28E+02	1.71E+04	4.53E+05	3.92E+01
	Std.	1.85E-03	1.79E+17	5.58E+16	2.12E+18	3.36E+00	2.16E+06	5.30E+01	3.64E-12	7.93E+00	3.27E+01
	T_{val}	~	-1.55E+01	-2.09E+00	-3.97E+00	-9.04E+01	-3.79E+07	-7.62E+00	-2.91E+07	-1.81E+05	-3.79E+00
C06	Mean	1.70E+09	2.62E+11	5.40E+07	3.48E+03	4.07E+03	1.22E+09	2.93E+03	1.79E+05	1.11E+10	3.69E+03
	Std.	8.46E+08	6.31E+10	1.47E+08	9.20E+02	3.46E+02	4.08E+08	6.40E+02	1.34E+05	1.34E+10	1.05E+03
	T_{val}	~	-1.30E+01	6.06E+00	6.35E+00	6.35E+00	1.60E+00	6.35E+00	6.35E+00	-2.22E+00	6.35E+00
C07	Mean	-1.99E+02	1.84E+14	2.27E+14	4.54E+13	-7.61E+02	6.25E+06	2.72E+12	9.68E+10	1.71E+14	-9.39E+01
	Std.	1.54E+02	4.07E+13	7.81E+13	5.35E+13	2.14E+02	1.05E+07	5.54E+12	4.10E+10	7.27E+13	7.38E+01
	T_{val}	~	-1.43E+01	-9.21E+00	-2.68E+00	6.74E+00	-1.89E+00	-1.56E+00	-7.46E+00	-7.43E+00	-1.94E+00
C08	Mean	-1.61E-04	6.54E+18	3.47E+15	4.00E+16	9.83E+06	1.58E+15	2.75E+10	8.74E+16	1.57E+15	3.20E+05
	Std.	4.77E-05	2.22E+18	2.66E+15	3.80E+16	1.49E+06	2.21E+08	1.15E+10	3.97E+16	4.00E+13	1.29E+05
	T_{val}	~	-9.30E+00	-4.12E+00	-3.33E+00	-2.08E+01	-2.26E+07	-7.54E+00	-6.96E+00	-1.24E+02	-7.86E+00
C09	Mean	1.47E+00	1.31E+24	2.27E+19	1.06E+57	3.11E+18	4.20E+16	1.99E+07	6.21E+15	2.71E+23	8.23E+06
	Std.	1.80E+00	3.52E+23	2.72E+19	2.57E+57	6.22E+18	2.41E+15	5.98E+07	1.24E+16	8.14E+23	1.65E+07
	T_{val}	~	-1.17E+01	-2.64E+00	-1.30E+00	-1.58E+00	-5.50E+01	-1.05E+00	-1.58E+00	-1.06E+00	-1.58E+00
C010	Mean	-4.90E-05	2.91E+19	1.68E+18	2.22E+18	1.21E+08	4.60E+17	5.14E+11	2.20E+06	4.62E+17	2.31E+06
	Std.	1.96E-05	1.14E+19	1.19E+18	2.09E+18	1.51E+07	3.00E+10	2.17E+11	2.32E+06	2.09E+15	6.53E+05
	T_{val}	~	-8.03E+00	-4.44E+00	-3.36E+00	-2.52E+01	-4.86E+07	-7.49E+00	-2.99E+00	-7.00E+02	-1.12E+01
C11	Mean	8.61E+09	4.65E+18	4.71E+15	1.43E+108	6.59E+17	4.25E+14	1.10E+10	3.41E+09	2.76E+109	4.25E+08
	Std.	1.93E+10	5.44E+17	2.25E+15	4.28E+108	3.80E+17	1.44E+12	9.99E+09	4.43E+09	8.22E+109	5.61E+08
	T_{val}	~	-2.70E+01	-6.63E+00	-1.05E+00	-5.49E+00	-9.35E+02	-3.52E-01	8.31E-01	-1.06E+00	1.34E+00
C12	Mean	1.44E+01	1.30E+18	2.04E+04	4.11E+08	1.26E+03	1.17E+09	3.33E+09	1.32E+09	1.83E+04	1.19E+09
	Std.	9.93E+00	2.54E+17	2.03E+03	1.17E+09	1.47E+02	3.40E+09	3.79E+09	3.47E+09	1.60E+00	1.49E+09
	T_{val}	~	-1.62E+01	-3.18E+01	-1.11E+00	-2.67E+01	-1.09E+00	-2.78E+00	-1.20E+00	-5.75E+03	-2.52E+00
C13	Mean	8.45E-10	1.35E+18	1.31E+17	5.80E+17	1.70E+12	3.32E+16	3.11E+12	8.34E+04	1.00E+17	4.86E+14
	Std.	4.24E-10	2.81E+17	5.18E+16	3.80E+17	2.20E+11	5.12E+08	1.38E+12	1.46E-11	1.56E+14	8.64E+14
	T_{val}	~	-1.52E+01	-8.01E+00	-4.82E+00	-2.44E+01	-2.05E+08	-7.11E+00	-6.22E+14	-2.03E+03	-1.78E+00
C14	Mean	1.42E+00	2.71E+18	2.11E+01	2.09E+01	2.00E+01	2.13E+01	1.76E+01	2.08E+01	2.14E+01	2.00E+01
	Std.	2.61E-02	8.19E+17	1.31E-01	3.06E-01	2.64E-13	1.13E-01	6.45E+00	4.69E-02	6.20E-02	2.13E-02
	T_{val}	~	-1.05E+01	-4.64E+02	-2.01E+02	-2.25E+03	-5.42E+02	-7.92E+00	-1.14E+03	-9.41E+02	-1.74E+03
C15	Mean	2.25E+01	1.47E+18	2.26E+16	3.53E+17	3.62E+17	2.27E+16	1.18E+01	2.08E+01	5.44E+16	1.49E+01
	Std.	2.88E+00	4.43E+17	1.10E+16	2.90E+17	5.44E+16	1.41E+07	3.14E+00	3.55E-15	8.28E+16	1.69E-06
	T_{val}	~	-1.05E+01	-6.51E+00	-3.85E+00	-2.10E+01	-5.09E+09	7.93E+00	1.82E+00	-2.08E+00	8.28E+00
C16	Mean	1.10E+02	1.14E+18	1.79E+16	2.74E+17	3.56E+17	2.27E+16	5.97E+01	2.08E+01	2.25E+16	1.19E+01
	Std.	1.21E+01	2.43E+17	6.19E+15	2.86E+17	7.35E+16	2.29E+07	1.17E+01	3.55E-15	6.77E+14	1.88E+00
	T_{val}	~	-1.48E+01	-9.13E+00	-3.03E+00	-1.53E+01	-3.12E+09	9.56E+00	2.35E+01	-1.05E+02	2.55E+01

TABLE 11. *(Continued.)* Calculation results of various algorithms (D = 30).

C17	Mean	9.61E+10	1.49E+18	2.95E+16	7.91E+16	9.61E+10	3.22E+16	9.49E+10	9.61E+10	3.22E+16	8.41E+10
	Std.	1.19E-02	5.05E+17	7.80E+15	5.82E+16	8.92E-03	1.18E+07	3.60E+09	4.25E-02	1.04E+12	5.52E-03
	T_{val}	~	-9.32E+00	-1.20E+01	-4.30E+00	2.11E+02	-8.60E+09	1.05E+00	7.08E+00	-9.81E+04	2.89E+12
C18	Mean	1.41E+12	1.53E+30	4.08E+26	1.02E+30	9.11E+25	2.80E+26	1.96E+15	9.61E+10	9.10E+29	5.44E+16
	Std.	2.93E+12	8.41E+29	3.20E+26	6.11E+29	1.82E+26	3.16E+18	2.86E+15	1.53E-05	6.59E+29	1.41E+17
	T_{val}	~	-5.75E+00	-4.03E+00	-5.28E+00	-1.58E+00	-2.80E+08	-2.17E+00	1.41E+00	-4.37E+00	-1.22E+00
C19	Mean	1.83E+17	1.85E+17	3.70E+17	3.70E+17	3.70E+17	1.85E+17	1.85E+17	9.61E+10	3.70E+17	3.70E+17
	Std.	2.87E+07	2.23E+13	1.19E+14	5.77E+13	1.09E+14	1.26E+13	8.17E+13	1.53E-05	5.21E+13	2.82E+14
	T_{val}	~	-3.45E+02	-4.97E+03	-1.03E+04	-5.42E+03	-5.93E+02	-7.49E+01	2.02E+10	-1.14E+04	-2.09E+03
C20	Mean	5.90E+00	1.35E+01	1.06E+01	8.60E+00	2.35E+00	9.29E+00	6.06E+00	4.85E+06	9.70E+00	2.57E+00
	Std.	2.16E+00	1.20E+00	8.07E-01	2.29E+00	1.83E-01	4.45E-05	9.96E-01	1.03E+07	3.66E-01	5.69E-01
	T_{val}	~	-9.73E+00	-6.50E+00	-2.71E+00	5.19E+00	-4.97E+00	-2.15E-01	-1.49E+00	-5.50E+00	4.72E+00
C21	Mean	1.85E+01	9.28E+17	1.02E+11	2.49E+12	4.64E+02	2.90E+12	2.77E+12	3.57E+12	1.86E+10	3.28E+12
	Std.	1.46E+01	3.09E+17	1.84E+11	1.22E+12	1.06E+02	1.08E+12	8.62E+11	1.08E+12	5.04E+07	1.55E+12
	T_{val}	~	-9.48E+00	-1.76E+00	-6.44E+00	-1.32E+01	-8.45E+00	-1.02E+01	-1.04E+01	-1.17E+03	-6.70E+00
C22	Mean	2.76E+09	7.90E+17	2.81E+18	1.23E+19	3.89E+12	9.67E+17	1.58E+13	2.79E+12	2.90E+18	3.36E+15
	Std.	5.33E+09	5.54E+16	1.44E+18	9.39E+18	4.49E+11	3.98E+09	1.01E+13	4.88E-04	5.58E+13	5.58E+15
	T_{val}	~	-4.51E+01	-6.17E+00	-4.15E+00	-2.74E+01	-4.60E+08	-4.94E+00	-1.66E+03	-1.65E+05	-1.91E+00
C23	Mean	1.42E+00	1.89E+18	9.29E+17	3.54E+18	7.25E+08	1.81E+18	3.52E+10	1.85E+07	1.81E+18	1.05E+09
	Std.	2.59E-02	3.51E+17	3.22E+17	2.87E+18	1.81E+08	2.41E+14	2.41E+10	3.82E+07	4.86E+12	3.92E+08
	T_{val}	~	-1.70E+01	-9.11E+00	-3.89E+00	-1.27E+01	-2.38E+04	-4.62E+00	-1.54E+00	-1.18E+06	-8.49E+00
C24	Mean	1.51E+01	9.27E+17	4.53E+17	3.84E+18	3.58E+17	9.07E+17	1.39E+01	7.17E-01	9.07E+17	1.49E+01
	Std.	1.07E+00	1.77E+17	1.77E+17	4.97E+18	8.15E+16	9.30E+08	2.08E+00	1.11E-16	3.80E+10	5.85E-06
	T_{val}	~	-1.65E+01	-8.12E+00	-2.44E+00	-1.39E+01	-3.08E+09	1.58E+00	4.25E+01	-7.54E+07	3.85E-01
C25	Mean	1.32E+02	9.34E+17	6.02E+17	4.30E+18	2.04E+01	9.07E+17	9.87E+01	7.17E-01	9.05E+17	6.60E+01
	Std.	2.03E+01	1.69E+17	2.69E+17	4.91E+18	5.83E+00	6.61E+08	2.14E+01	1.11E-16	5.27E+15	1.30E+01
	T_{val}	~	-1.75E+01	-7.06E+00	-2.77E+00	1.67E+01	-4.34E+09	3.57E+00	2.05E+01	-5.43E+02	8.68E+00
C26	Mean	9.61E+10	8.01E+17	5.88E+17	1.08E+18	9.13E+10	9.62E+17	9.61E+10	9.61E+10	9.62E+17	9.61E+10
	Std.	5.24E-03	2.96E+17	1.84E+17	8.81E+17	5.88E+09	1.41E+09	6.99E-02	1.75E-02	7.76E+11	7.28E-03
	T_{val}	~	-8.55E+00	-1.01E+01	-3.88E+00	2.58E+00	-2.16E+09	1.76E+01	1.30E+01	-3.92E+06	3.57E+02
C27	Mean	2.32E+13	6.04E+29	2.50E+29	1.07E+33	5.38E+15	6.22E+29	1.00E+19	9.61E+10	1.25E+33	6.56E+17
	Std.	2.13E+13	2.65E+29	1.96E+29	1.39E+33	1.09E+15	7.05E+20	2.28E+19	1.53E-05	2.06E+33	8.64E+17
	T_{val}	~	-7.22E+00	-4.03E+00	-2.43E+00	-1.56E+01	-2.79E+09	-1.39E+00	3.42E+00	-1.92E+00	-2.40E+00
C28	Mean	1.85E+17	1.85E+17	3.70E+17	3.70E+17	3.70E+17	1.85E+17	1.85E+17	9.61E+10	3.70E+17	3.70E+17
	Std.	1.28E+14	2.22E+13	4.71E+13	9.29E+13	1.25E+14	3.25E+13	9.38E+13	1.53E-05	2.00E+13	1.68E+14
	T_{val}	~	-1.57E+01	-4.31E+03	-3.72E+03	-3.28E+03	-1.54E+01	-6.70E+00	4.56E+03	-4.54E+03	-2.78E+03
T-test	*w/t/l*	~	28/0/0	26/0/2	26/0/2	21/0/7	26/0/2	20/0/8	15/0/13	27/0/1	18/0/10

GDFA, HFA has 27 benchmark functions whose values of t-test are better than GDFA; compared with SPSO, HFA has 18 benchmark functions whose values of t-test are better than SPPSO; compared to ADE, HFA has 18 benchmark functions whose values of t-test are better than ADE; compared with GA-DEx, HFA has 27 benchmark functions whose values of t-test are better than GA-DEx; compared to CSPSO, HFA has 18 benchmark functions whose values of t-test are better than CSPSO. According to the Friedman test shown in Table 13, HFA(rank = 2.54) ranks first in all algorithms, followed by CSPSO (rank = 3.00) and SPPSO (rank = 3.11) respectively. In summary, for the 10-dimensional problem, the performance of HFA is significantly better than the other nine algorithms.

2) PERFORMANCES FOR THE 30-D PROBLEMS

The experimental results for the D = 30 problems are shown in Table 11. As can be seen from Table 11, ADE performs best on six benchmark functions such as C18, C19, C24, C25, C27, and C28. CSPSO performs best on four benchmark functions, C3, C11, C16, and C17. RFA performs best on the C07, C20, and C26 benchmark functions. SPPSO performs best on C6 and C15 benchmark functions. HFA performs best on the remaining 13 benchmark functions. At the same time, t-test shows that the overall performance of the proposed HFA is better than the other nine comparison algorithms. Specifically, compared with FA, HFA has 28 test functions whose values of t-test are better than FA; compared to NEFA, HFA has 26 benchmark functions whose values of

TABLE 12. Calculation results of various algorithms (D = 50).

Function	Evaluation indicator	HFA	FA	NEFA	UFA	RFA	GDFA	SPPSO	ADE	GA-DEx	CSPSO
C01	Mean	3.26E-10	3.71E+06	1.11E+05	1.63E+05	5.00E+00	4.51E+05	2.89E+03	1.19E+04	4.62E+05	1.57E+01
	Std.	1.43E-10	3.29E+06	6.69E+04	8.66E+04	4.00E+00	4.64E+04	4.49E+02	1.50E+03	5.91E+04	3.69E+00
	T_{val}	~	-3.56E+00	-5.24E+00	-5.97E+00	-3.95E+00	-3.08E+01	-2.04E+01	-2.51E+01	-2.47E+01	-1.35E+01
C02	Mean	4.14E-10	1.01E+14	1.76E+05	2.49E+18	1.19E+01	4.82E+05	2.82E+03	1.05E+04	4.81E+05	1.51E+01
	Std.	2.22E-10	2.03E+14	5.34E+04	2.85E+18	1.56E+00	1.04E-02	4.39E+02	0.00E+00	1.97E+03	4.03E+00
	T_{val}	~	-1.58E+00	-1.04E+01	-2.75E+00	-2.41E+01	-1.46E+08	-2.03E+01	-1.50E+14	-7.72E+02	-1.19E+01
C03	Mean	3.69E+06	1.77E+16	2.47E+05	1.51E+06	1.72E+00	5.19E+05	6.03E+03	1.62E+04	6.18E+05	4.37E+01
	Std.	5.33E+06	2.17E+16	8.10E+04	1.69E+06	8.21E-01	9.99E+04	1.75E+03	4.73E+03	1.19E+04	9.40E+00
	T_{val}	~	-2.58E+00	2.04E+00	1.23E+00	2.19E+00	1.88E+00	2.18E+00	2.18E+00	1.82E+00	2.19E+00
C04	Mean	4.54E+01	9.54E+12	8.66E+03	2.34E+04	1.20E+05	8.56E+02	2.49E+02	4.77E+03	8.56E+02	3.20E+02
	Std.	1.17E+01	5.86E+11	1.93E+03	1.78E+04	4.82E+03	3.35E-06	3.05E+01	8.27E+03	3.61E-02	5.55E+01
	T_{val}	~	-5.14E+01	-1.41E+01	-4.16E+00	-7.85E+01	-2.18E+02	-1.97E+01	-1.81E+00	-2.18E+02	-1.53E+01
C05	Mean	1.90E+01	3.21E+18	1.67E+17	4.24E+18	2.14E+02	3.16E+13	5.17E+01	6.30E+02	6.63E+05	7.68E+01
	Std.	3.80E+01	1.79E+17	1.03E+17	2.95E+18	1.00E+01	8.51E+07	3.45E+01	0.00E+00	2.43E+00	3.89E+01
	T_{val}	~	-5.67E+01	-5.13E+00	-4.55E+00	-1.56E+01	-1.17E+06	-2.01E+00	-5.08E+01	-5.50E+04	-3.36E+00
C06	Mean	1.64E+09	5.80E+11	2.39E+07	4.96E+03	7.12E+03	9.09E+09	5.67E+03	1.05E+07	3.00E+10	5.53E+03
	Std.	7.60E+08	3.83E+11	4.76E+07	1.82E+03	6.13E+02	4.70E+09	2.34E+03	1.38E+07	2.20E+10	1.90E+03
	T_{val}	~	-4.79E+00	6.71E+00	6.82E+00	6.82E+00	-4.95E+00	6.82E+00	6.78E+00	-4.09E+00	6.82E+00
C07	Mean	-3.07E+02	8.42E+14	2.99E+14	2.40E+14	-9.98E+02	7.66E+06	3.35E+13	2.99E+12	4.68E+14	-1.39E+02
	Std.	1.40E+02	4.23E+14	2.78E+14	3.02E+14	3.60E+02	1.10E+07	5.37E+13	7.83E+11	6.90E+13	1.16E+02
	T_{val}	~	-6.29E+00	-3.40E+00	-2.51E+00	5.65E+00	-2.20E+00	-1.97E+00	-1.21E+01	-2.15E+01	-2.94E+00
C08	Mean	5.10E-04	2.52E+19	4.05E+16	2.71E+17	5.81E+07	4.48E+15	2.11E+12	8.95E+16	4.49E+15	1.52E+08
	Std.	1.32E-04	7.97E+18	3.50E+16	5.60E+17	6.17E+06	6.64E+09	5.76E+11	7.30E+16	7.14E+12	6.20E+07
	T_{val}	~	-9.99E+00	-3.66E+00	-1.53E+00	-2.98E+01	-2.13E+06	-1.16E+01	-3.88E+00	-1.99E+03	-7.74E+00
C09	Mean	3.77E+00	8.57E+24	2.77E+20	6.03E+87	2.91E+07	1.90E+17	4.98E+06	2.43E+08	3.16E+48	1.04E+07
	Std.	2.66E+00	4.87E+24	2.58E+20	1.81E+88	2.25E+07	5.32E+11	1.49E+07	2.64E+08	9.43E+48	2.15E+07
	T_{val}	~	-5.56E+00	-3.40E+00	-1.05E+00	-4.10E+00	-1.13E+06	-1.05E+00	-2.91E+00	-1.06E+00	-1.53E+00
C010	Mean	5.94E-05	9.52E+19	3.81E+18	5.57E+18	4.34E+08	1.49E+19	3.81E+13	2.93E+06	1.50E+19	3.18E+08
	Std.	2.34E-05	7.67E+19	1.58E+18	4.16E+18	4.22E+07	4.62E+16	2.07E+13	6.30E+06	3.45E+16	1.56E+08
	T_{val}	~	-3.93E+00	-7.64E+00	-4.24E+00	-3.26E+01	-1.02E+03	-5.82E+00	-1.47E+00	-1.37E+03	-6.43E+00
C11	Mean	8.23E+11	1.68E+19	2.68E+16	6.99E+16	3.91E+18	1.15E+15	4.47E+10	8.59E+09	5.67E+17	8.91E+08
	Std.	8.14E+11	4.72E+18	1.06E+16	9.40E+16	1.38E+18	8.86E+12	3.25E+10	9.43E+09	1.18E+18	1.05E+09
	T_{val}	~	-1.13E+01	-7.96E+00	-2.35E+00	-8.97E+00	-4.09E+02	3.02E+00	3.16E+00	-1.52E+00	3.19E+00
C12	Mean	1.66E+01	4.05E+18	3.85E+04	1.81E+10	3.13E+03	1.65E+10	1.79E+10	8.49E+08	3.28E+04	7.48E+09
	Std.	1.25E+01	8.29E+17	4.23E+03	1.48E+10	6.10E+02	1.35E+10	1.03E+10	2.05E+09	7.30E-01	6.40E+09
	T_{val}	~	-1.54E+01	-2.88E+01	-3.85E+00	-1.61E+01	-3.86E+00	-5.52E+00	-1.31E+00	-8.24E+03	-3.70E+00
C13	Mean	4.29E+00	4.13E+18	5.33E+17	2.77E+18	1.24E+13	1.07E+17	3.28E+13	7.70E+06	3.20E+17	4.40E+15
	Std.	7.52E+00	4.50E+17	1.39E+17	1.68E+18	9.50E+11	7.84E+08	9.23E+12	0.00E+00	1.30E+14	6.63E+15
	T_{val}	~	-2.90E+01	-1.21E+01	-5.22E+00	-4.12E+01	-4.30E+08	-1.12E+01	-3.24E+06	-7.80E+03	-2.10E+00
C14	Mean	1.11E+00	9.21E+18	2.13E+01	2.11E+01	2.00E+01	2.14E+01	2.11E+01	2.08E+01	2.14E+01	2.01E+01
	Std.	1.57E-02	7.98E+17	5.50E-02	3.54E-01	3.07E-13	2.64E-02	2.92E-02	5.43E-02	1.49E-02	3.33E-02
	T_{val}	~	-3.65E+01	-1.11E+03	-1.78E+02	-3.79E+03	-2.09E+03	-1.91E+03	-1.10E+03	-2.97E+03	-1.63E+03
C15	Mean	2.32E+01	3.82E+18	8.25E+16	1.04E+18	1.43E+18	7.47E+16	1.49E+01	2.08E+01	7.94E+16	1.62E+01
	Std.	3.45E+00	4.92E+17	2.92E+16	3.18E+17	1.76E+17	3.25E+07	5.32E-05	3.55E-15	7.75E+15	2.51E+00
	T_{val}	~	-2.46E+01	-8.93E+00	-1.04E+01	-2.57E+01	-7.28E+09	7.58E+00	2.21E+00	-3.24E+01	5.20E+00
C16	Mean	2.26E+02	4.56E+18	8.57E+16	7.92E+17	1.69E+18	7.47E+16	1.33E+02	2.08E+01	7.48E+16	2.07E+01
	Std.	2.11E+01	5.99E+17	2.96E+16	6.23E+17	4.53E+17	4.58E+07	2.04E+01	3.55E-15	1.42E+15	2.88E+00
	T_{val}	~	-2.41E+01	-9.14E+00	-4.02E+00	-1.18E+01	-5.15E+09	1.01E+01	3.08E+01	-1.66E+02	3.05E+01
C17	Mean	2.60E+11	3.12E+18	9.94E+16	2.21E+17	2.60E+11	1.03E+17	2.60E+11	9.61E+10	1.03E+17	2.40E+11
	Std.	1.88E-03	4.70E+17	3.24E+16	1.62E+17	4.71E-03	6.55E+07	6.17E-02	4.65E-02	6.76E+11	1.09E-02
	T_{val}	~	-2.10E+01	-9.69E+00	-4.31E+00	6.38E+02	-4.98E+09	3.55E+01	1.11E+13	-4.83E+05	5.71E+12
C18	Mean	7.45E+01	5.42E+30	9.98E+26	2.48E+30	4.25E+16	1.05E+27	1.38E+16	9.61E+10	3.32E+30	3.59E+17
	Std.	1.10E+02	1.57E+30	7.47E+26	1.55E+30	1.09E+16	2.04E+18	1.24E+16	1.53E-05	1.60E+30	3.07E+17
	T_{val}	~	-1.09E+01	-4.22E+00	-5.05E+00	-1.23E+01	-1.64E+09	-3.51E+00	-2.75E+09	-6.56E+00	-3.69E+00

TABLE 12. ***(Continued.)*** **Calculation results of various algorithms (D = 50).**

C19	Mean	5.22E+17	5.29E+17	1.06E+18	1.06E+18	1.06E+18	5.28E+17	5.28E+17	9.61E+10	1.06E+18	1.06E+18
	Std.	5.94E+09	2.57E+13	3.20E+14	7.44E+13	1.31E+14	4.82E+09	1.68E+14	1.53E-05	1.93E+14	1.11E+15
	T_{val}	~	-6.51E+04	-5.28E+03	-2.28E+04	-1.29E+04	-2.78E+06	-1.12E+02	2.78E+08	-8.78E+03	-1.52E+03
C20	Mean	1.40E+01	2.38E+01	1.91E+01	3.07E+07	4.56E+00	1.82E+01	1.32E+01	4.18E+05	1.90E+01	4.97E+00
	Std.	1.34E+00	2.17E+00	1.45E+00	9.22E+07	2.43E-01	3.33E-06	1.74E+00	8.91E+05	2.18E-01	9.28E-01
	T_{val}	~	-4.88E+01	-8.22E+00	-1.05E+00	2.19E+01	-1.01E+01	1.17E+00	-1.48E+00	-1.18E+01	1.75E+01
C21	Mean	2.06E+01	4.32E+18	1.47E+11	5.84E+12	9.20E+02	6.67E+12	6.43E+12	3.01E+12	1.66E+11	6.79E+12
	Std.	1.40E+01	7.91E+17	1.85E+11	2.00E+12	6.07E+01	1.45E+12	1.52E+12	1.71E+12	5.37E+06	1.42E+12
	T_{val}	~	-1.73E+01	-2.51E+00	-9.22E+00	-4.57E+01	-1.45E+01	-1.34E+01	-5.55E+00	-9.76E+04	-1.51E+01
C22	Mean	2.72E+10	3.56E+18	6.04E+18	3.55E+19	2.48E+13	1.43E+18	9.04E+13	1.48E+12	4.30E+18	6.52E+17
	Std.	4.03E+10	6.35E+17	2.13E+18	2.43E+19	1.02E+12	3.22E+10	2.40E+13	0.00E+00	1.26E+15	1.95E+18
	T_{val}	~	-1.77E+01	-8.97E+00	-4.61E+00	-7.65E+01	-8.79E+07	-1.19E+01	-1.14E+02	-1.08E+04	-1.06E+00
C23	Mean	1.10E+00	7.69E+18	2.58E+18	6.95E+18	4.81E+09	2.77E+18	6.99E+11	1.71E+08	2.77E+18	7.71E+09
	Std.	2.03E-03	8.79E+17	5.31E+17	4.61E+18	4.53E+08	2.86E+09	4.05E+11	1.93E+08	1.64E+14	2.95E+09
	T_{val}	~	-2.77E+01	-1.54E+01	-4.76E+00	-3.35E+01	-3.07E+09	-5.47E+00	-2.80E+00	-5.34E+04	-8.26E+00
C24	Mean	1.81E+01	3.40E+18	1.07E+18	1.02E+19	1.27E+18	1.31E+18	1.49E+01	7.18E-01	1.31E+18	4.84E+13
	Std.	1.40E+00	5.13E+17	5.02E+17	6.38E+18	9.76E+16	1.30E+09	1.65E-04	0.00E+00	6.99E+10	9.70E+13
	T_{val}	~	-2.10E+01	-6.76E+00	-5.05E+00	-4.12E+01	-3.18E+09	7.07E+00	3.90E+01	-5.91E+07	-1.58E+00
C25	Mean	2.38E+02	3.41E+18	1.33E+18	8.26E+18	6.78E+18	1.31E+18	1.94E+02	7.18E-01	1.31E+18	1.28E+02
	Std.	1.70E+01	5.91E+17	3.81E+17	9.02E+18	1.02E+19	7.74E+08	2.83E+01	0.00E+00	5.18E+10	2.56E+01
	T_{val}	~	-1.82E+01	-1.11E+01	-2.89E+00	-2.10E+00	-5.34E+09	4.24E+00	4.43E+01	-7.98E+07	1.13E+01
C26	Mean	2.60E+11	3.53E+18	1.45E+18	2.95E+18	2.60E+11	1.42E+18	2.60E+11	9.61E+10	1.42E+18	2.58E+11
	Std.	3.70E-03	1.88E+17	5.52E+17	2.27E+18	5.37E-03	7.12E+08	6.06E-02	4.23E-02	1.85E+13	6.00E+09
	T_{val}	~	-5.95E+01	-8.33E+00	-4.11E+00	4.79E+02	-6.30E+09	1.07E+01	1.22E+13	-2.42E+05	1.05E+00
C27	Mean	3.99E+10	3.24E+30	1.07E+29	2.05E+33	1.04E+17	8.85E+29	2.99E+19	9.61E+10	1.29E+33	2.42E+18
	Std.	1.20E+11	5.07E+29	1.41E+29	2.26E+33	1.11E+16	2.65E+21	2.40E+19	0.00E+00	9.66E+32	3.00E+18
	T_{val}	~	-2.02E+01	-2.40E+00	-2.87E+00	-2.97E+01	-1.06E+09	-3.93E+00	-1.49E+00	-4.23E+00	-2.55E+00
C28	Mean	5.27E+17	5.29E+17	1.06E+18	1.06E+18	1.06E+18	5.29E+17	5.28E+17	9.61E+10	1.06E+18	1.06E+18
	Std.	3.09E+14	2.55E+13	1.48E+14	8.91E+13	1.57E+14	1.14E+12	1.85E+14	0.00E+00	1.20E+13	4.53E+14
	T_{val}	~	-6.56E+04	-4.90E+03	-5.22E+03	-4.84E+03	-1.63E+01	-8.47E+00	5.40E+03	-5.43E+03	-3.06E+03
T-test	*w/t/l*	~	28/0/0	26/0/2	26/0/2	22/0/6	27/0/1	18/0/10	17/0/7	27/0/1	19/0/9

Note: "~" indicates that the same items do not compare.

t-test are better than NEFA; compared with UFA, HFA has 26 benchmark functions whose values of t-test are better than UFA; compared with RFA, HFA has 21 benchmark functions whose values of t-test are better than RFA; compared with GDFA, HFA has 26 benchmark functions whose values of t-test are better than GDFA; compared with SPSO, HFA has 20 benchmark functions whose values of t-test are better than SPPSO; compared to ADE, HFA has 15 benchmark functions whose values of t-test are better than ADE; compared with GA-DEx, HFA has 27 benchmark functions whose values of t-test are better than GA-DEx; compared to CSPSO, HFA has 18 benchmark functions whose values of t-test are better than CSPSO. According to the Friedman test shown in Table 14, HFA (rank = 2.61) ranks first in all algorithms, followed by CSPSO(rank = 3.25) and SPPSO(rank = 3.61) respectively. In summary, for the 30-dimensional problem, the performance of the HFA proposed in this paper is significantly better than the other nine algorithms.

3) PERFORMANCES FOR THE 50-D PROBLEMS

The experimental results for the D = 50 problems are shown in Table 12. As can be seen from Table 12, ADE performs best on six benchmark functions such as C17, C19, C24, C25, C26, and C28. RFA performs best on three test functions, C03, C07 and C20. CSPSO performs best on two benchmark functions, C11 and C16. SPPSO performs best on the C15 benchmark function. UFA performs best on the C06 benchmark function. HFA performed best on the remaining 15 benchmark functions. At the same time, t-test shows that the overall performance of the proposed HFA is better than the other nine comparison algorithms. Specifically, compared with FA, HFA has 28 test functions whose values of t-test are better than FA; compared to NEFA, HFA has 26 benchmark functions whose values of t-test are better than NEFA; compared with UFA, HFA has 26 benchmark functions whose values of t-test are better than UFA; compared with RFA, HFA has 22 benchmark functions whose values of t-test are better than RFA; compared with GDFA, HFA has 27 benchmark functions whose values of t-test are better than GDFA; compared with SPSO, HFA has 18 benchmark functions whose values of t-test are better than SPPSO; compared with ADE, HFA has 17 benchmark functions whose values of t-test are better than ADE; compared with GA-DEx, HFA has 27 benchmark functions whose values of t-test are better

TABLE 13. Statistical analysis of friedman test for the results (D = 10).

Algorithms	HFA	FA	NEFA	UFA	RFA	GDFA	SPPSO	ADE	GA-DEx	CSPSO
Friedman test	2.54	9.71	5.89	7.71	4.50	6.36	3.11	4.89	7.29	3.00
Ranking	1	10	6	9	4	7	3	5	8	2

TABLE 14. Statistical analysis of friedman test for the results (D = 30).

Algorithms	HFA	FA	NEFA	UFA	RFA	GDFA	SPPSO	ADE	GA-DEx	CSPSO
Friedman test	2.61	8.89	6.57	8.25	4.39	6.29	3.61	3.82	7.32	3.25
Ranking	1	10	7	9	5	6	3	4	8	2

TABLE 15. Statistical analysis of friedman test for the results (D = 50).

Algorithms	HFA	FA	NEFA	UFA	RFA	GDFA	SPPSO	ADE	GA-DEx	CSPSO
Friedman test	2.50	9.14	6.54	8.36	4.21	6.21	3.79	3.43	7.32	3.50
Ranking	1	10	7	9	5	6	4	2	8	3

TABLE 16. Optimization results of various algorithms.

Optimum variables	x_1	x_2	x_3	x_4	x_5	$f(x)$
HFA	6.01031477	4.90165089	4.36435886	3.49442514	2.17522330	**13.0367735**
GDFA	6.042	5.035	4.228	3.421	2.614	13.2861
MVO	6.0239402	5.3060112	4.4950113	3.4960223	2.1527262	13.39959
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	13.4000
GCA-I	6.0100	5.3000	4.4900	3.4900	2.1500	13.4000
GCA-II	6.0100	5.3000	4.4900	3.4900	2.1500	13.4000
CS	6.0089	5.3049	4.5023	3.5077	2.1504	13.9999
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	13.9996

than GA-DEx; compared to CSPSO, HFA has 19 benchmark functions whose values of t-test are better than CSPSO. According to the Friedman test shown in Table 15, HFA ranks first in all algorithms (rank = 2.50), followed by ADE (rank = 3.43) and CSPSO (rank = 3.50) respectively. In summary, for the 50-dimensional problem, the performance of the HFA proposed in this paper is significantly better than the other nine algorithms.

VI. CANTILEVER BEAM OPTIMIZATION DESIGN MODEL AND SOLUTION METHOD

The cantilever beam consists of five square hollow blocks, the first of which is fixed, and the fifth is subjected to vertical loads. Take the cantilever beam design problem given in [23] as an example. The cantilever beam structure is illustrated in Figure.3.

With the minimum weight of the cantilever beam shown in Figure.3 the following mathematical model can be built.

$$\min f(x) = 0.6224 (x_1 + x_2 + x_3 + x_4 + x_5) \quad (29)$$

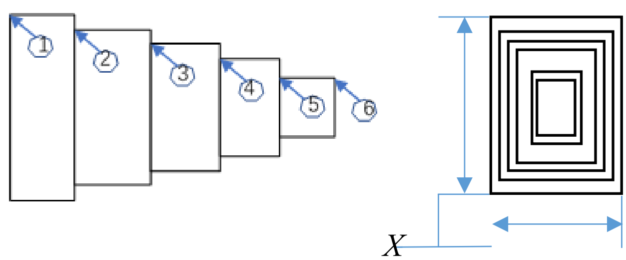


FIGURE 3. Structure design of cantilever beam design.

$$s.t. \begin{cases} g(x) = \dfrac{61}{x_1^3 + x_2^3 + x_3^3 + x_4^3 + x_5^3} \\ 0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100 \end{cases} \quad (30)$$

To further verify the feasibility of the HFA algorithm to solve the cantilever beam parameter optimization problem, the HFA is compared with the algorithms of literature [23], [41], [42], [43] and [44]. Furthermore, for the convenience of writing, the solution method in [23] is abbreviated as GDFA, the solution method in [41] is abbreviated as MVO, the solution methods in [42] are abbreviated as MMA, GCA-I and GCA-II, the solution method in [43] is abbreviated CS, and the solution method in [44] is abbreviated SOS. The mathematical models of cantilever beam parameter optimization problem is solved using HFA, GDFA, MVO, MMA, GCA-I, GCA-II, CS and SOS algorithms, respectively. The calculation results are listed in TABLE 16.

Table16 shows that the weight of the cantilever beam obtained by HFA is significantly smaller than those of other algorithms. Compared with GDFA, MVO, MMA, GCA-I, GCA-II, CS and SOS, the weight of the cantilever beam obtained by HFA decreased by 1.877%, 2.708%, 2.711%, 2.711%, 2.711%, 6.880%, 6.878% respectively.

VII. CONCLUSION

Aiming at the problem that the position update formula of the firefly algorithm in the existing literatures does not consider the attraction of the optimal firefly to all other fireflies in the population, a position update formula considering the attraction of the optimal firefly to all other fireflies in the population is given. And a method to adaptively adjust the parameters in the position update formula is adopted, rendering the position update formula strong global search ability and strong local search ability in the initial stage and the later stage of iteration, respectively. Accordingly, the improved position formula can rise the convergence speed of the HFA.

Some single mutation operators have strong local search ability, and some single mutation operators have strong global search ability. Therefore, a single mutation operator cannot give better consideration to the local and global search ability of the firefly algorithm. In order to better balance the global and local search ability of the firefly algorithm, the mutation operators with strong local search ability and mutation operators with strong global search ability can be combined to obtain the combined mutation operator, and the combined mutation operator is introduced into the improved firefly algorithm to get the hybrid firefly algorithm. Therefore, the HFA global and local search capabilities are stronger and the solution quality is higher.

The problem of Logistic mapping given in existing literatures is that when $cs(k)$ equals 0, 0.25, 0.5, 0.75 and 1, chaotic search cannot traverse the whole search space. In order to avoid this phenomenon, a method of generating chaotic sequence is presented. This method can traverse [0,1] interval. On this basis, a method of mapping chaotic sequence to search space is proposed, which enables the firefly algorithm to traverse the whole search space and enhances the global search ability of the firefly algorithm.

In the literatures that introduces chaotic search in the firefly algorithm, only the optimal firefly in the population is randomly moved according to the chaotic search. This random movement reduces the possibility that the population falls into local optimum to some extent. The problem with this strategy is that only one firefly moves randomly according to chaotic search, which limits its global search ability. To solve this problem, HFA randomly moves all fireflies in the population according to chaos search, so as to further enhance the global search ability of the population.

In order to verify the effectiveness of HFA, 28 test problems in CEC2017 were selected and compared with basic FA algorithm, four improved firefly algorithms, and four other types of intelligent optimization algorithms. The calculation results of 28 CEC2017 test problems show that the solution quality of HFA is better than that of other algorithms, which verifies the efficiency and feasibility of HFA.

Finally, the structural parameters of the cantilever beam are optimized using HFA and various improved algorithms in the literatures. The optimization results show that the solving quality of the HFA is significantly better than other algorithms, which proves that HFA has strong optimization ability.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable and constructive comments that greatly helped improve the quality and completeness of the paper.

REFERENCES

[1] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. London, U.K.: Luniver Press, 2008, pp. 81–96.
[2] S. J. Wang and X. Z. Gao, "A survey of research on firefly algorithm," *Microcomput. Appl.*, vol. 34, no. 8, pp. 8–11, 2015.
[3] R. Dutta, R. Ganguli, and V. Mani, "Exploring isospectral spring–mass systems with firefly algorithm," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 467, no. 2135, pp. 3222–3240, 2011.
[4] P. Aungkulanon, N. Chaiead, and P. Luangpaiboon, "Simulated manufacturing process improvement via particle swarm optimisation and firefly algorithms," in *Proc. World Congr. Eng.*, London, U.K., 2010, pp. 1123–1128.
[5] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emissions load dispatch problem," *Int. J. Combinatorics*, vol. 2011, Nov. 2011, Art. no. 523806.
[6] C. Liu and C. Ye, "Novel bioinspired swarm intelligence optimization algorithm: Firefly algorithm," *Appl. Res. Comput.*, vol. 28, no. 9, pp. 3295–3297, 2011.
[7] M.-H. Horng and T.-W. Jiang, "Multilevel image thresholding selection based on the firefly algorithm," in *Proc. 7th Int. Conf. Ubiquitous Intell. Comput. 7th Int. Conf. Autonomic Trusted Comput.*, Oct. 2010, pp. 58–63.
[8] A. Gálvez and A. Iglesias, "Modified memetic self-adaptive firefly algorithm for 2D fractal image reconstruction," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf.*, Tokyo, Japan, Jul. 2018, pp. 165–170.
[9] C. X. Zhao, C. Wu, J. Chai, X. Wang, X. Yang, J.-M. Lee, and M. J. Kim, "Decomposition-based multi-objective firefly algorithm for RFID network planning with uncertainty," *Appl. Soft Comput.*, vol. 55, pp. 549–564, Jun. 2017.
[10] Y. K. S. S. Rao and B. B. Krishna, "Modeling diesel engine fueled with tamanu oil—Diesel blend by hybridizing neural network with firefly algorithm," *Renew. Energy*, vol. 134, pp. 1200–1212, Apr. 2019.
[11] V. Agarwal and S. Bhanot, "Radial basis function neural network-based face recognition using firefly algorithm," *Neural Comput. Appl.*, vol. 30, no. 8, pp. 2643–2660, 2018.
[12] J. Shen and Y. Zhu, "Chance-constrained model for uncertain job shop scheduling problem," *Soft Comput.*, vol. 20, no. 6, pp. 2383–2391, 2016.
[13] Y. Zeng, Z. Zhang, and A. Kusiak, "Predictive modeling and optimization of a multi-zone HVAC system with data mining and firefly algorithms," *Energy*, vol. 86, pp. 393–402, Jun. 2015.
[14] C. Qi, Q. Chen, A. Fourie, X. Tang, Q. Zhang, X. Dong, and Y. Feng, "Constitutive modelling of cemented paste backfill: A data-mining approach," *Construct. Building Mater.*, vol. 197, pp. 262–270, Feb. 2019.
[15] X.-S. Yang, S. Deb, M. Loomes, and M. Karamanoglu, "A framework for self-tuning optimization algorithm," *Neural Comput. Appl.*, vol. 23, no. 7, pp. 2051–2057, 2013.

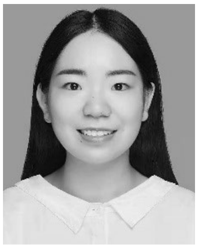
[16] S. Yu, S. Zhu, Y. Ma, and D. Mao, "A variable step size firefly algorithm for numerical optimization," *Appl. Math. Comput.*, vol. 263, pp. 214–220, Jul. 2015.
[17] H. Wang, X. Zhou, H. Sun, X. Yu, J. Zhao, H. Zhang, and L. Cui, "Firefly algorithm with adaptive control parameters," *Soft Comput.*, vol. 21, no. 17, pp. 5091–5102, Sep. 2017.
[18] J. Zhang, Y.-F. Teng, and W. Chen, "Support vector regression with modified firefly algorithm for stock price forecasting," *Appl. Intell.*, vol. 49, no. 5, pp. 1658–1674, 2019.
[19] S. L. Tilahun and H. C. Ong, "Modified firefly algorithm," *J. Appl. Math.*, vol. 2012, Oct. 2012, Art. no. 467631.
[20] W. Long, S. H. Cai, J. Jiao, Y. Chen, and Y. Huang, "Firefly algorithm for solving constrained optimization problems and engineering applications," *J. Central South Univ. Sci. Technol.*, vol. 46, no. 4, pp. 1260–1267, 2015.
[21] O. P. Verma, D. Aggarwal, and T. Patodi, "Opposition and dimensional based modified firefly algorithm," *Expert Syst. Appl.*, vol. 44, pp. 168–176, Feb. 2016.
[22] L. Zhou, L. Ding, and Y. Lei, "An enhanced firefly algorithm with orthogonal centroid opposition-based learning," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2018, pp. 1–8.
[23] C.-F. Wang and W.-X. Song, "A novel firefly algorithm based on gender difference and its convergence," *Appl. Soft Comput.*, vol. 80, pp. 107–124, Jul. 2019.
[24] H. Wang, W. Wang, H. Sun, and S. Rahnamayan, "Firefly algorithm with random attraction," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 1, pp. 33–41, Feb. 2016.
[25] A. Yelghi, C. Káse, "A modified firefly algorithm for global minimum optimization," *Appl. Soft Comput.*, vol. 62, pp. 29–44, Jan. 2018.
[26] L. Zhang, W. Srisukkham, S. C. Neoh, L. C. Peng, and D. Pandit, "Classifier ensemble reduction using a modified firefly algorithm: An empirical evaluation," *Expert Syst. Appl.*, vol. 93, pp. 395–422, Mar. 2018.
[27] L. He and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152–174, May 2017.
[28] Z. Liu, X. Li, D. Wu, Z. Qian, P. Feng, and Y. Rong, "The development of a hybrid firefly algorithm for multi-pass grinding process optimization," *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2457–2472, 2019.
[29] H. Wang, W. Wang, Z. Cui, X. Zhou, J. Zhao, and Y. Li, "A new dynamic firefly algorithm for demand estimation of water resources," *Inf. Sci.*, vol. 438, pp. 95–106, Apr. 2018.
[30] G.-D. Zhou, M.-X. Xie, and T.-H. Yi, "Optimal wireless sensor network configuration for structural monitoring using automatic-learning firefly algorithm," *Adv. Struct. Eng.*, vol. 22, no. 4, pp. 907–918, 2019.
[31] M. Singh, R. N. Patel, and D. D. Neema, "Robust tuning of excitation controller for stability enhancement using multi-objective metaheuristic Firefly algorithm," *Swarm Evol. Comput.*, vol. 44, pp. 136–147, Feb. 2019.
[32] W. L. Winston and J. B. Goldberg, *Operations Research: Applications and Algorithms*. Belmont, CA, USA: Thomson/Brooks/Cole Technometrics, 2004.
[33] A. Farah and A. Belazi, "A novel chaotic Jaya algorithm for unconstrained numerical optimization," *Nonlinear Dyn.*, vol. 93, no. 3, pp. 1451–1480, 2018.
[34] I. Brajević and J. Ignjatović, "An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems," *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2545–2574, 2019.
[35] X. Pan, L. Xue, and R. Li, "A new and efficient firefly algorithm for numerical optimization problems," *Neural Comput. Appl.*, vol. 31, no. 5, pp. 1445–1453, 2019.
[36] O. Zhiyuan, L. Yu, S.-J. Ruan, and T. Jiang, "An improved particle swarm optimization algorithm for reliability-redundancy allocation problem with mixed redundancy strategy and heterogeneous components," *Rel. Eng. Syst. Saf.*, vol. 181, pp. 62–74, Jan. 2019.
[37] H. P. H. Anh, N. N. Son, C. Van Kien, and V. Ho-Huu, "Parameter identification using adaptive differential evolution algorithm applied to robust control of uncertain nonlinear systems," *Appl. Soft Comput.*, vol. 71, pp. 672–684, Oct. 2018.
[38] M. Z. Ali, N. H. Awad, P. N. Suganthan, A. M. Shatnawi, and R. G. Reynolds, "An improved class of real-coded genetic algorithms for numerical optimization," *Neurocomputing*, vol. 275, pp. 155–166, Jan. 2018.
[39] C. Rui, Y.-X. Su, D.-H. Zhang, X.-X. Chi, and H.-J. Zhang, "A hybridization of cuckoo search and particle swarm optimization for solving optimization problems," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 653–670, 2019.
[40] G. H. Wu, R. Mallipeddi, and P. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization," Ph.D. dissertation, Nat. Univ. Defense Technol., Changsha, China, 2016.
[41] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.
[42] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method," *Int. J. Numer. Methods Eng.*, vol. 39, no. 5, pp. 829–846, 1996.
[43] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, 2013.
[44] M.-Y. Cheng and D. Prayogo, "Symbiotic Organisms Search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.

JIQUAN WANG received the B.E. degree in agricultural electrification and automation and the M.E. degree in agricultural electrification and automation from the Northeast Agricultural University, China, in 1996 and 2004, respectively, and the Ph.D. degree in agricultural equipment engineering technology from Shenyang Agricultural University, China, in 2011.

He is currently an Associate Professor with the College of Engineering, Northeast Agricultural University. His current research interests include genetic algorithm theory and its application, neural network theory and its application, parallel computing, and image recognition. He has published over 40 articles in domestic and international academic journals and conference proceedings.

MINGXIN ZHANG received the B.E. degree in information management and information system from the Hebei University of Engineering, China, in 2018. He is currently pursuing the master's degree with the College of Engineering, Northeast Agricultural University.

His current research interest includes genetic algorithm theory and its application.



HAOHAO SONG received the B.E. degree in logistics management from the North China University of Water Resources and Electric Power, China, in 2019. She is currently pursuing the master's degree with the College of Engineering, Northeast Agricultural University.

Her current research interest includes genetic algorithm theory and its application.

ZHIWEN CHENG received the B.E. degree in logistics management from the Hefei University of Technology, China, in 2017. He is currently pursuing the master's degree with the College of Engineering, Northeast Agricultural University.

His current research interest includes genetic algorithm theory and its application.

TIEZHU CHANG received the B.E. degree in industry engineering from Northeast Agricultural University, China, in 2019, where he is currently pursuing the master's degree with the College of Engineering.

His current research interest includes genetic algorithm theory and its application.

YUSHENG BI received the B.E. degree in industry engineering from Northeast Agricultural University, China, in 2018, where he is currently pursuing the master's degree with the College of Engineering.

His current research interest includes genetic algorithm theory and its application.

KEXIN SUN received the B.E. degree in industry engineering from Northeast Agricultural University, China, in 2018, where he is currently pursuing the master's degree with the College of Engineering.

His current research interest includes genetic algorithm theory and its application.