A Hybrid Harmony Search Algorithm based on Firefly Algorithm and Boltzmann Machine

Jiao Yang, Xiaoxia Zhang, Shuai Fu ,Yinyin Hu

***Abstract*—Harmony Search Algorithm (HS) is a new meta-heuristic algorithm based on music creation process. The algorithm has been widely used in many fields because of its simple and flexible structure and fewer controlled parameters. However, HS is easy to be affected by the initial harmony memory and falls into local optimization. In addition, there are also problems, such as low searching accuracy and slow convergence speed of the optimal solution. Therefore, this paper proposes a hybrid harmony search algorithm based on firefly algorithm and boltzmann machine (HSFA-BM). First, in order to avoid the possibility of harmony search algorithm falling into local optimum, boltzmann machine (BM) mechanism is used to update the harmony memory in this paper. Second, in order to enhance the ability of the algorithm to exploit optimal solutions, this paper uses the movement mode of fireflies to generate new solutions. The improved algorithm enhances the ability of global optimization and search precision, and speeds up the convergence speed. In this paper, six classical test functions are used to test the effectiveness of the algorithm, and a very obvious optimization result is obtained.**

***Index Terms*—harmony search, meta-heuristic algorithm, optimization problem, firefly algorithm, boltzmann machine**

I. Introduction

In this era of technology explosion, the development of all walks of life has shown an amazing speed, and the process of development is the process of constantly optimizing their own systems and technologies. Optimization is to quickly and efficiently search for the best or nearly the best solution from a number of alternative solutions under a certain number of constraints. The optimization process is not limited to industry, domain or discipline. As long as it is a problem-solving process can be optimized, such as task scheduling problem optimization, spatial routing problem optimization, resource constraint problem optimization and so on.

Manuscript received April 15, 2021; revised September 23, 2021. This work was supported by Project of Liaoning Xincheng Co., Ltd (Grant No. L20170989).

J. Yang is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan, 114051, China(e-mail: yangjiao0816@163.com).

X.X. Zhang is a Professor of School of computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan, 11041, China (corresponding author, phone:86-0412-5929812; e_mail:aszhangxx@163.com).

S. Fu is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan, 114051, China(e-mail: L-fushuai@163.com).

Y.Y. Hu is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan, 114051, China(e-mail: 2799386740@qq.com).

Optimization can simplify a lot of problems. In the early days, linear programming, dynamic programming and integer programming were common methods to solve optimization problems. These methods can get the optimal solution for solving small scale problems. Due to the increasing complexity of the problem, it is difficult for traditional methods to get the optimal solution within an acceptable time. It is difficult to solve highly complex large-scale mathematical models by clinging to traditional methods.

In the face of the above problems, the meta-heuristic algorithm emerges as the times require. The self-optimization mechanism of the meta-heuristic algorithm and the powerful processing power of computer enable people to find the best solution. Meta-heuristic algorithm is inspired by natural phenomena or natural laws. It usually finds a near-optimal solution within an acceptable search time. Recently, many meta-heuristic algorithms have appeared in public view, such as Flower Pollination Algorithm (FPA)，harmony search algorithm (HS), fruit fly optimization algorithm (FFO) and firefly algorithm (FA), etc.

HS [1] simulates the music playing procedure, and the tones of instruments are constantly altered until a pleasant harmony is completed. Though HS algorithm has been around for a short time, in recent years, with its unique characteristics and advantages, HS algorithm has shown its effectiveness and superiority in the fields of construction, engineering, robot, telecommunications, health and energy etc. Many applications of HS algorithm are described in [2,3].

Although the HS algorithm's advantages and characteristics make it widely used, the harmony search algorithm also has many loopholes when solving complex problems. In order to adapt to different practical problems and heighten robustness, various transformations [4] of harmony algorithm have been published one after another. The idea of self-adaptive global best harmony search (SGHS) algorithm [5] is to improve the global optimal HS algorithm. The exploratory harmony search (EHS) algorithm [6] eliminates the limitations of BW tuning by setting a direct proportional relationship between parameters. for the sake of the equilibrium between diversity and reinforcement, the intelligent tuning harmony search (ITHS) algorithm [7] automatically selects the appropriate tone variation method according to harmony memory during search process. The idea of the improved global optimal harmony search (IGHS) algorithm [8] is to modify pitch adjustment steps of HS. The modified neighborhood heuristic harmony search (MHSNH) algorithm [9] adopts two heuristic techniques to improve the initial harmony memory and solve the flow-shop scheduling problem. An adaptive hybrid harmony search algorithm [10] improves algorithm performance by automatically changing algorithm parameters and hybrid Jaya algorithm (JA).

In this paper, a hybrid harmony search algorithm based on firefly algorithm and Boltzmann machine (HSFA-BM) is presented to solve many disadvantages of harmony search algorithm, such as there is the possibility of falling into local optimality, weak ability to exploit the optimal solution and slow convergence speed. HSFA-BM algorithm has two main aspects of innovation. First, the Boltzmann machine principle can accept the solution with bad quality, so that the algorithm can make a big jump and enhance ability to jump out of local optimality. Second, use the way fireflies fly to create new solutions. The brightness and attraction of the firefly control the direction and distance of the solution, respectively. This algorithm fusion method improves the efficiency of HS algorithm to exploit the optimal solution and enhances the convergence speed.

The other chapters of this paper are as follows. Section 2 introduces the basic harmony search algorithm. Section 3 introduces the basic firefly algorithm and Boltzmann machine mechanism. Section 4 introduces the detailed process of the harmony search algorithm based on Boltzmann machine and hybrid firefly algorithm. Section 5 demonstrates the advantages of HSFA-BM algorithm from multiple angles through experimental comparison. Section 6 is summary and outlook.

II. Harmony search algorithm

A. The principle of harmony search algorithm

The HS algorithm simulated the procedure of musicians repeatedly adjusting the tones of each instrument to obtain a perfect harmony when they improvised music. It is just like the process in which every decision variable is constantly adjusted to get the best solution in engineering optimization.

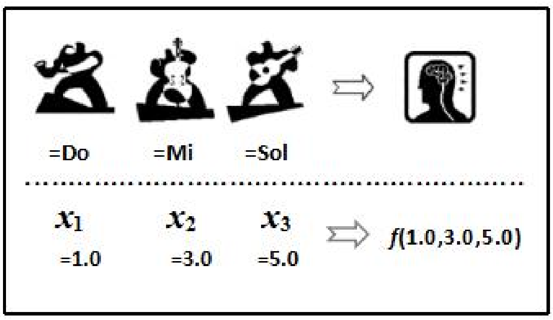

Fig. 1 Analogy between music improvisation and engineering optimization.

Fig. 1 shows the corresponding relationship between improvisation harmonies and production optimization. The figure vividly shows that the instrument is like the determination value, and the different tone of the ozinho is like a decision variable taking different values. There was a remarkable resemblance between the process by which one perceives a pleasing music and the process by which the objective function achieves the optimal solution in optimization.

B. Algorithm steps of harmony search algorithm

The core idea of harmony search algorithm: First, initialize a preset size of the harmony memory. Then, the program constantly generates new harmonies. New harmonies come from harmony memory or within permissible limits(It is usually based on probability to determine how to generate new harmonies.). Suppose that the new solution is superior to the inferior solution in the harmony memory, the worst solution in the harmony memory is replaced with the newly generated solution. Repeat the above steps until the stop condition is met. The specific work is as follows.

Step 1: Initialize the main parameters of the algorithm. Harmony Memory Size (*HMS*), Harmony Memory considering rate (*HMCR*), Pitch Adjusting rate (*PAR*), Pitch Adjusting bandwidth (*BW*), Maximum number of cycles (stop standard, *maxIter*), Upper and lower bounds on the components of the solution (*low*[*D*], *high* [*D*]), where *D* is the dimension of the solution.

Step 2: Initialize the harmony memory. Formula (1) is initial harmonic memory $P(0)=\{X_i\,(0) \mid low[j] < x_{i,j}\,(0) < hight[j],\ i=1,2,\ldots,HMS;\ j=1,2,\ldots,D\}$. Where $P(0)$ is the harmony memory of the 0 generation. $X_i\,(0)$ represents the *i*th solution in the 0 generation harmony memory. $X_{i,j}\,(0)$ is the *j*th component of the *i*th harmony in the 0 generation harmony memory.

$$P(0)=\begin{bmatrix} X_1(0) \\ X_2(0) \\ \ldots \\ X_{(HMS-1)}(0) \\ X_{(HMS)}(0) \end{bmatrix}=\begin{bmatrix} x_{1,1}(0) & x_{1,2}(0) & \ldots & x_{1,D}(0) \\ x_{2,1}(0) & x_{2,2}(0) & \ldots & x_{2,D}(0) \\ \ldots & \ldots & \ldots & \ldots \\ x_{(HMS-1),1}(0) & x_{(HMS-1),2}(0) & \ldots & x_{(HMS-1),D}(0) \\ x_{(HMS),1}(0) & x_{(HMS),2}(0) & \ldots & x_{(HMS),D}(0) \end{bmatrix} \quad (1)$$

Step 3: Generate a new solution. There are three ways to produce a melodious tune. In many studies, these three schemes are formalized into a quantitative optimization process to generate new harmonic vectors $X_{new} = \{ x_{(new)1}, x_{(new)2}, \ldots, x_{(new)D} \}$. The first way is Memory consideration as formula (2). The musician plays a melody from his memory (That is, harmony memory, HM).

$$x_{(new)i} = \mathrm{HM}[Rand\_int][i] \quad (2)$$

Here $x_{(new),i}$ is the *i*th component of the new solution，and i=1,2,...,D. Rand_int can generate a random integer greater than or equal to 1 and less than or equal to *HMS*.

The second method is pitch adjustment. Playing a piece of music similar to what the musician remembers, as formula (3).

$$x_{(new)i} = x_{(\mathrm{new})i} \pm Rand * BW \quad (3)$$

Here i=1,2,...D. Rand can randomly output a uniformly distributed number greater than 0 and less than 1. Formula (3) essentially creates a new harmony by slightly modifying existing solutions. The third way is random selection. The musician played a brand new tune.

Step 4: Regenerate the harmony memory. Suppose that the fitness value of X_{new} is perfect than the inferior harmony in the HM, then X_{new} is used instead of the inferior harmony vector in harmony memory.

Step 5: Determine the termination conditions. Suppose that the stop condition (maximum number of cycles) is met, no calculation operation is performed, on the contrary, Step 3 will continue.

The third way of generating new harmonies is by random generation. This approach is conducive to expanding the search space to conduct a wide area search. Pitch adjustment surfaces appear to perform the same function as randomly generating new harmonies. However, the tone variation is based on the harmonies in the HM to produce new harmonies by tiny adjustment, which is equivalent to the neighborhood solution of the known solution. It has strong limitations. So Pitch adjustment is a local search. Random generation of new solutions is beneficial to explore a new search area, then to enrich the diversity of solution. The pseudo-code of the harmony search algorithm is shown in Fig. 2.

```
Algorithm1    Harmony Search (HS)
Input:    HMCR, PAR, BW, HMS.
Output:   bestHarmony.
Initialize Harmony Memory : HM.
for i =1 to HMS do
for j =1 to D do
   HM[i][j]=Rand * (high[ j ]-low[ j ])+low[ j ];
end
end
While Stopping Criteria Not Reached do
   for i =1 to D do
     if Rand < HMCR then
        x(new) i = HM [ Rand_int ][i];
     if Rand < PAR then
        x(new) i =x(new) i ±Rand * BW;
     end if
     else
        x(new) i =Rand * (high[ j ]-low[ j ])+low[ j ];
     end if
   end for
   if f(Xnew) < f(Xworst) then
     Update HM by replacing Xworst by Xnew.
   end if
end while
```

Fig.2 Pseudo code of harmony search.

In Fig. 2, X_{new} is an improvised new harmony. X_{worst} represents the inferior quality harmony in the current harmony memory. Rand can randomly output a uniformly distributed number greater than 0 and less than 1. Rand_int can generate a random integer greater than or equal to 1 and less than or equal to *HMS*.

C. Advantages and disadvantages of harmony search algorithm

HS is a powerful meta-heuristic algorithm. HS algorithm has many advantages over other optimization algorithms. First, the algorithm structure is simple and flexible. Second, fewer controlled parameters. Third, the algorithm does not need the gradient information and initial conditions of the objective function. Fourth, the calculation is simple. Therefore, the ability of harmony search algorithm to combine with many meta-heuristic algorithms is greatly improved. In [11], harmony search algorithm is mixed with particle swarm optimization algorithm to successfully get out of dynamic parameter value estimation problem. In [12], the combination of genetic algorithm, simulated annealing algorithm and artificial immune system makes the harmony quality in the harmony memory further improved.

Algorithms are combined to complement each other's strengths. The above successful cases show that algorithm fusion is a measure that achieves the desired end to improve algorithm efficiency. In this paper, the harmony search algorithm for low accuracy and slow search speed shortcomings also used this method. This method is to embed the good mechanism of firefly algorithm into harmony search algorithm. The inspiration comes from the global optimal harmony search algorithm (GHS). This core concept of GHS is to replace this pitch adjustment part of the basic HS with the finest solution in harmony memory. Since *BW* is limited to the optimization of the algorithm, this approach destroys the diversity of solution. Therefore, this paper will replace the pitch adjustment mode with the movement mode of fireflies to produce a new solution. The way fireflies approach the optimal solution is controlled by the dual constraints of spatial distance and interindividual attraction, so that the new solution can be accurately valued near the optimal solution. This method not only increases the diversity of solution, makes the convergence precision better and enhances the rate of convergence. The principle of boltzmann machine is used for control, so that the algorithm does not always accept the good solution, but also accepts the bad solution with a certain probability, so that the algorithm jumps out of the local optimal.

III. FIREFLY ALGORITHM AND BOLTZMANN MACHINE PRINCIPLE

A. Firefly algorithm

Firefly algorithm (FA) [13] belongs to a population intelligent algorithm. The core idea is to simulate the glowing courtship behavior of fireflies in nature.

The generating mechanism of the new solution of firefly algorithm consists of two elements: luminance and attractiveness. Luminance is used to represent the quality of the firefly, then the movement direction of the individual is controlled. Attractiveness is used to control the distance the individual moves. Each firefly continues to fly toward the brightest individual in the population in its search area. The global or local optimal solution is found through location iteration. If there are no brighter individuals in the search range, the fireflies make random movements. These two elements are constantly updated for optimization. The algorithm is described as follows:

Suppose $X_i = (x_{i1}, x_{i2},..., x_{iD})$ is the ith solution in the colony, where i =1,2... N, and N represents the colony size.

D is this dimension of the solution. The relative luminance of fireflies as formula (4).

$$I_i = I_0 * e^{-g\, r_{ij}} \tag{4}$$

In which I_0 is maximum luminance. ϒ is the absorption coefficient that the strength of the light (usually a constant), and r_{ij} represents the spatial interval between solution X_i and solution X_j as formula (5).

$$r_{ij} = \left\| X_i - X_j \right\| = \sqrt{\sum_{d=1}^{D} \left(x_{id} - x_{jd}\right)^2} \tag{5}$$

For any two fireflies X_i and X_j (i=1, 2, ..., N; j=1, 2, ..., N; $i \neq j$), formula (6) is the attraction between them.

$$b\,(r_{ij}) = b_0 * e^{-g\, r_{ij}^2} \tag{6}$$

where β_0 is maximum attract degree (usually a constant).

For firefly X_i and firefly X_j, the poorer individual will move to the better one. Assuming that X_j is superior to X_i, then X_i moves towards X_j, and the mode of movement is defined as formula (7).

$$x_{id}(t+1) = x_{id}(t) + b\,(r_{ij}) * (x_{jd}(t) - x_{id}(t)) + a * e \tag{7}$$

Where d= 1, 2, ..., D, $\alpha \in$ [0,1] is a step size, ε is a random number greater than or equal to -0.5 and less than or equal to 0.5, and t is the current number of cycles. Fig. 3 is the pseudo code for the standard firefly algorithm. T_{max} is the maximum number of iterations (algorithm termination condition).

```
Algorithm2   Firefly Algorithm(FA)
Population initialization, set t = 0.
While t < T_max do
    for i = 1 to N do
    for j = 1 to N do
        if X_j better than X_i then
            Move X_i towards X_j according to formula (3);
            Calculate the fitness value of X_i;
        end if
    end for
    end for
        t++;
end while
output result.
```

Fig. 3 Basic Firefly Algorithm Pseudocode.

B. Boltzmann machine principle

Boltzmann machine (BM) is a random neural network [14]. In this network, there are only two output states, namely 0 or 1 in unipolar binary. State 1 means the neuron is in the on state, and state 0 means the neuron is disconnected. Through continuous feedback and update of the network, the state of each neuron is the optimal solution of the problem when the network finally reaches the BM distribution state (particle stable state in physics).

The topology of BM network is special, which is between the full interconnection structure and the hierarchical structure. Formally, it has a symmetric weight, namely $\omega_{ij} = \omega_{ji}$, and ω_{ii}=0. But in terms of the function of neurons, there are input nodes, hidden nodes and output nodes. Typically, input and output nodes are called visible nodes, while hidden nodes are called invisible nodes. During training, input and output nodes receive training set samples, while hidden nodes mainly play an auxiliary role to realize the connection between input and output, so that the training set can be reproduced in the visible unit. The connection form of the hetero-associative BM network is shown in Fig. 4, and there is no obvious hierarchy among the three types of nodes.

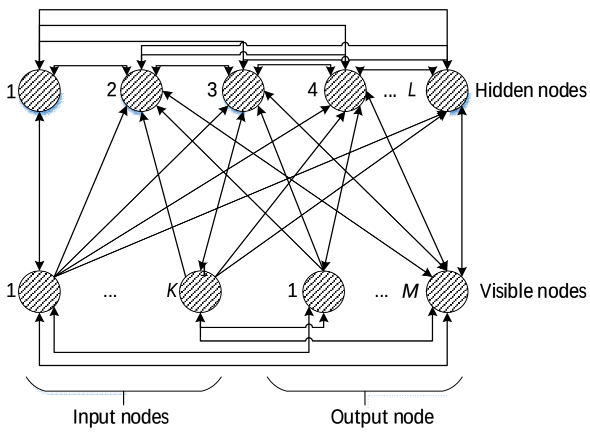

Fig. 4 Hetero-associative BM network.

Assume that $S=(s_1,s_2,\ldots,s_i,\ldots,s_N)$ represents the structure of BM, where $s_i \in \{0,1\}$ represents the status of neuron i. N is the sum of neurons. The net input of a single neuron in BM network is defined as formula (8).

$$\mathrm{net}_j = \sum_{i=1}^{N-1} (w_{ij} s_i - T_j) \tag{8}$$

Where ω_{ij} represents the weight between neuron i and neuron j, and T_j represents the threshold of neuron j. The difference between BM and other networks is that the above net input can not directly obtain the determined output state through the sign transfer function, and the actual output state will occur according to a certain probability. The net input of a neuron can obtain the transition probability of the output state through the s-shaped function(Sigmoid function). The transition probability is defined as formula (9).

$$P_j(1) = \frac{1}{1 + e^{-net_j / T}} \tag{9}$$

Where $P_j(1)$ represents the probability that the output state of neuron j is 1, and T is the temperature. Formula (10) is the probability of state 0.

$$P_j(0) = 1 - P_j(1) \quad (10)$$

Formula (11) is the network state described by the energy function of BM network.

$$E(t) = -\frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N} T_i x_i \quad (11)$$

Suppose that BM network works in an asynchronous manner. According to the deduced formula of Hopfield neural network, the *j*th neuron changes state every time is defined as formula (12).

$$\Delta E(t) = -\Delta x_j(t) net_j(t) \quad (12)$$

Various situations of formula (11) are discussed below:

If $net_j > 0$ and $P_j(1) > 0.5$ can be obtained according to formula (9), neuron j has a large probability of $x_j = 1$. If the original $x_j = 1$ then $\Delta x_j = 0$, thus $\Delta E = 0$; If the original $x_j = 0$ and $\Delta x_j = 1$, thus $\Delta E < 0$;

If $net_j < 0$ and $P_j(1) < 0.5$ can be obtained according to formula (9), neuron j has a large probability of $x_j=0$. If the original $x_j = 0$, then$\Delta x_j = 0$, thus$\Delta E = 0$; If the original $x_j =1$, $\Delta x_j = -1$, thus $\Delta E < 0$;

As can be seen from the above discussion, as the status of BM network changes, the energy of BM mesh in the sense of probability always changes on a decreasing direction. This shows that even though the mesh energy evolves in a decreasing direction, it can not be ruled out that A small number of neurons status maybe switch with low probability, thereby temporarily increasing the mesh energy. Just because of this possibility, BM mesh enhances the possibility of deviating from local optimality. This characteristic of BM network is the essential difference of energy change from other networks. Due to the working mode of random value of neuron state according to probability, BM network has the ability of constantly jumping out of the trough of higher position (local minimum) to search for the new trough of lower position. Based on the working mode of BM network where the state of neurons randomly switchs according to probability, this paper uses HS algorithm to accept the poor solution according to probability in the updating link of solution, so that the algorithm search mechanism constantly jumps out of the local optimum.

IV. A HYBRID HARMONY SEARCH ALGORITHM BASED ON FIREFLY ALGORITHM AND BOLTZMANN MACHINE

In the process of algorithm optimization, the balance between diversification and Intensification is an important factor to measure the performance of meta-heuristic algorithm. For the sake of heightening the performance of the algorithm and prevent falling into local optimum, a proper balance must be struck between the two contradictory characteristics of diversification and Intensification.

In the HS algorithm, if the solution in the initial harmony memory is close to a local optimal solution, there is serious premature convergence, which leads to the algorithm falling into local optimal solution. Based on the idea of balance above, in this paper, the basic HS algorithm is improved by using this method that the states of the neurons in BM network are switched randomly according to probability. The method of HS algorithm updating harmony memory is replaced by the method of obtaining the value of neuron state randomly according to probability, so that the algorithm can accept the bad solution on the basis of percentages. Its percentage is limited by the current temperature and the number of cycles of the program.

The temperature parameter is kept at a higher position and the number of iterations is small at the beginning, which is conducive to accepting the bad solution, thus increasing the ability to deviate from the local optimum. As the algorithm continues to run, the temperature parameter gradually decreases and the number of iterations gradually increases, which leads to the gradual reduction of the acceptance probability of bad solutions, and gradually shifts the focus to the exploitation of good solutions. This improved algorithm improves the convergence accuracy and speed of the algorithm, and avoids the algorithm falling into local optimum.

The above improvements greatly heighten the global optimization performance of the HS algorithm. Although the basic HS possesses good local optimization performance, it can not match the improved global search ability. Based on the inspiration of GHS, this paper uses the movement mode of fireflies to generate new solutions. The basic HS is to generate a new solution with a fixed step size *BW*. And the way that GHS generates new solutions is by constantly taking the best solution in HM. Both of these methods have limitations and are not conducive to the exploitation of the global optimum solution. FA can ameliorate the moving tendency of poor solution to the better solution, thus improving the convergence accuracy and speed.

The specific algorithm steps of HSFA-BM are ut infra:

Step 1: Initialize the main arguments of the algorithm. Arguments of the HS, maximum T_0 and minimum T_{end} of the temperature parameters in BM.

Step 2: Initialize harmony memory.

Step 3: Generate a new solution. There are three ways to produce a new solution $X_{new} = \{x_{(new)1}, x_{(new)2}, \ldots, x_{(new)D}\}$ according to probability.

Memory consideration: Formula (2) is a way to generate new solutions in the harmony memory.

Pitch adjustment: Formula (13) adjusts the harmony using the flight mode of fireflies.

$$\begin{aligned} x_{(new)i} &= \mathrm{HM}\,[worstIndex\,][i] + attraction\ * \\ &\{\mathrm{HM}\,[bestIndex\,][i] - \mathrm{HM}\,[worstIndex\,][i]\} + rand\,() \\ &* BW \end{aligned} \quad (13)$$

Variable '*attraction*' is attraction between fireflies as formula (6).

Random selection: Play a new piece of music as shown in formula (14).

$$x_{(new)i} = Rand * (hight\ [i] - low\ [i]) + low\ [i] \tag{14}$$

Step 4: Regenerate the HM. Suppose that the fitness value of X_{new} is perfect than the inferior harmony vector X_{worst} in the HM, then X_{new} is used to replace the X_{worst} in HM. On the contrary, the poor solution is adopted with a certain percentage, and the adoption rates is shown in formula (15).

$$\mathrm{P} = e^{-d\,c/T_0} \tag{15}$$

Where, *dc* is the fitness value difference between X_{new} and X_{worst}.

Step 5: Determine the termination conditions. If the algorithm reaches the stop condition, then the computation stops. Otherwise, repeat step 3.

```
Algorithm3:   HSFA-BM
1:  Input:     HMCR, PAR, BW, HMS, T0, Tend.
2:  Output:   bestHarmony.
3:  Initialize Harmony Memory : HM.
4:  While  Stopping Criteria Not Reached  do
5:  While  T0 <= Tend  do
6:    for  t =1 to L  do
7:    for  i =1 to D  do
8:      if  Rand < HMCR  then
9:          x(new) i = HM [ Rand_int ][i];
10:     if  Rand < PAR  then
11:         x(new) i = HM [worstIndex][i]+attraction*
12:       (HM [bestIndex][i]-HM [worstIndex][i])+BW * Rand;
13:     end if
14:     else
15:         x(new) i = Rand * (high [ j ]-low[ j ])+low [ j ];
16:     end
17:   end for
18:   if  dc<0 || Math.exp(-dc/T0) >= Rand  then
19:     Update HM by replacing Xworst by Xnew.
20:   end if
21:   T0 = q*T0;
22: end while
23: end while
```

Fig. 5 HSFA-BM algorithm pseudo code.

Intelligent system hybridization can learn from each other in function. they can deal with many complex, imprecise and fuzzy problems in real life. The pseudo code of HSFA-BM algorithm on the minimization problem is shown in Fig. 5. In algorithm 3, *High*[*j*] and *Low*[*j*] are the upper and lower boundary of the *j*th component of a solution. *Attraction* is the attraction among fireflies. HSFA-BM algorithm generates a new harmony in steps 7-17. Steps 11-12 are how the firefly moves. In Step 18, the algorithm always accepts excellent quality harmonies, but poor quality harmonies are also accepted with a probability that depends on the difference between the fitness value of the new solution and the fitness value of the inferior solution.

V. SIMULATION EXPERIMENT AND RESULT ANALYSIS

In this experiment, six typical continuous functions are tested to verify the effectiveness of HSFA-BM algorithm. Table I shows the main parameters and their values in the algorithm. With the number of cycles increases, the quality of harmony in harmony memory will gradually improve. For the sake of making the harmony memory play its due role, the HS algorithm sets up the harmony memory considering rate (*HMCR*). Yang[15] suggested $HMCR \in [0.7, 0.95]$. Main parameters of pitch adjustment are pitch bandwidth (*BW*) and pitch adjustment rate (*PAR*). Yang [15] recommended $PAR \in [0.1, 0.5]$. *HMS* expresses the capacity of the HM. If the setting of *HMS* is too small, the multiplicity of solutions will be destroyed and the convergence accuracy will be affected. The convergence rate will be affected if the *HMS* is set too large. A reasonable *HMS* should be set according to the size of the problem.

Table I
Main parameters of the algorithm

	name	symbol	values
1	Harmony memory considering rate	*HMCR*	$9.5*10^{-1}$
2	pitch adjustment rate	*PAR*	$3*10^{-1}$
3	bandwidth	*BW*	$1*10^{-3}$
4	harmony memory size	*HMS*	5

The functions in Table II are divided into two categories based on their peaks. The first is the unimodal functions including f_1 and f_5, and the second is the multiimodal functions including f_2 through f_4 and f_6. The image of the unimodal function is characterized by only one peak, so the convergence rate of the algorithm can be detected. Because of the existence of many local minima, the multimodal function can detect the global searching ability of the algorithm. When *D*=10, the results of HS, HS-BM and HSFA-BM algorithms running 10 times on 6 functions are shown in Table III. Table IV shows the experimental results when *D*=20. Table V shows the experimental results when *D*=30. Best is the best fitness value. Worst is the worst fitness value. The search accuracy of the algorithm can be seen from both. Average is the average of the algorithm. Std is the standard deviation of the algorithm, which shows the stability of the algorithm. When *D*=10, the convergence curves of HS, HS-BM and HSFA-BM algorithms on the six functions are shown in Fig. 6 to Fig. 11. The number of program runs is displayed on the horizontal axis. The best solution for each generation is shown on the vertical axis. The three convergent curves in each function represent three algorithms.

The experiments prove that HS-BM is better than HS and HSFA-BM is superior to HS-BM in terms of convergence speed, convergence accuracy and out-of-local optimal performance. Table III shows the following information. The accuracy of the optimal value of the

HS-BM algorithm is 4-10 orders of magnitude higher than that of the HS algorithm. The accuracy of the optimal value of the HSFA-BM algorithm is 2-5 orders of magnitude higher than that of the HS-BM algorithm. The accuracy of the worst value of HS-BM algorithm is 5-12 orders of magnitude higher than that of HS algorithm. The accuracy of the worst value of HSFA-BM algorithm is 2 to 5 orders of magnitude higher than that of HS-BM algorithm. The average accuracy of the HS-BM algorithm is 5-10 orders of magnitude higher than that of the HS algorithm. The average accuracy of HSFA-BM algorithm is 2-5 orders of magnitude higher than that of HS-BM algorithm. The standard deviation accuracy of the HS-BM algorithm is 6-10 orders of magnitude higher than that of the HS algorithm. The standard deviation accuracy of HSFA-BM algorithm is 2-5 orders of magnitude higher than that of HS-BM algorithm. The addition of BM greatly improves each index of optimization, indicating that the algorithm falling into local optimum has a serious impact on the output of the final result. After the mixed FA algorithm, the improvement range of the optimized indicators is consistent, indicating that the FA algorithm has a consistent influence over the convergence precision and speed. It can be seen from Table III, Table IV and Table V that the search accuracy of the algorithm decreases with the increase of the problem dimension. However, in the same dimension, the search accuracy of the algorithm still has the characteristics that HS-BM is better than HS and HSFA-BM is better than HS-BM, but the improved algorithm has a better effect on the low-dimensional problems. For the convergence curve, it is obvious from the overall view that the convergence accuracy is HS-BM superior to HS, and HSFA-BMsuperior to HS-BM, which indicates that the improved algorithm improves the convergence accuracy. HSFA-BM always gives priority to convergence to a certain precision, and the convergence speed is relatively fast. At the beginning of the iteration, HS-BM algorithm always gives priority to better solutions than HS algorithm, indicating that BM mechanism improved the global search ability and could obtain better solutions in a wider range. Compared with HS-BM algorithm, HSFA-BM algorithm always gets better solutions because FA has a strong ability to exploitation optimal solutions. In the middle of iteration, except for f_2 function, the curve of HS algorithm always converges after a sharp drop, which just shows that it is difficult for harmony search algorithm to escape local optimum. The reason why the other two improved algorithms always jump to find the global optimal solution is that the BM mechanism can accept the inferior solution with a certain probability, expand the search scope, explore more new areas, and make the algorithm separate from the local optimalizing situation, so that the algorithm keeps getting nearer to the global optimal solution. At the end of the the algorithm, the HS algorithm and the HS-BM algorithm have converged, while the solution of the HSFA-BM algorithm is still changing as the FA adjusts to the exploitation of the optimal solution. Firefly's courtship flight can accurately grasp the direction and distance, so the process of the new solution generated by the firefly algorithm is the course of rapidly approaching the best solution. For the sake of evaluating the superiority of HSFA-BM algorithm more comprehensively, table VI

Table II
SIMULATION TESTING FUNCTIONS

Function name	Expression	Range	*f*min
Sphere	$f_1(\mathrm{x})=\sum_{i=1}^{D} x_i^2$	[-5.12,5.12]	0
Griewank	$f_2(\mathrm{x})=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	0
Ackley	$f_3(\mathrm{x})=-20exp(-0.02\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-exp(\frac{1}{D}\sum_{i=1}^{D}cos(2px_i))+20+e$	[-30,30]	0
Rastrigin	$f_4(x)=\sum_{i=1}^{D}(x_i^2-10cos(2px_i)+10)$	[-5.12,5.12]	0
Step	$f_5(x)=\sum_{i=1}^{D}[x_i+0.5]^2$	[-100,+100]	0
H14	$f_6(x)=\exp(0.5*\sum_{i=1}^{D}x_i^2)$	[-1.28,1.28]	0

Table III
THE SIMULATION RESULTS OF THREE ALGORITHMS(D=10)

Function name	Algorithm	Best	Worst	Average	Std
	HS	2.67E-06	3.32E+02	5.31E-03	9.67E-03
f_1	HS-BM	2.50E-11	2.43E-10	1.20E-10	7.36E-11
	HSFA-BM	1.48E-15	1.92E-14	9.30E-15	6.24E-15
	HS	8.99E-02	2.88E-01	1.77E-01	7.08E-02
f_2	HS-BM	1.60E-12	3.91E-11	1.72E-11	1.13E-11
	HSFA-BM	1.11E-16	2.66E-15	9.44E-16	7.81E-16
	HS	2.61E-03	7.90E-01	2.30E-01	2.36E-01
f_3	HS-BM	3.51E-07	2.68E-06	1.25E-06	6.75E-07
	HSFA-BM	8.12E-09	1.78E-08	1.27E-08	2.61E-09
	HS	3.49E-03	5.08E+00	5.90E-01	1.50E+00
f_4	HS-BM	6.12E-09	6.92E-08	2.78E-08	2.07E-08
	HSFA-BM	1.94E-13	2.16E-12	9.61E-13	6.25E-13
	HS	1.86E-02	4.24E+00	1.44E+00	1.56E+00
f_5	HS-BM	2.04E-11	3.28E-10	1.26E-10	8.79E-11
	HSFA-BM	2.64E-15	1.42E-14	4.83E-15	3.19E-15
	HS	3.87E-06	3.00E-04	9.99E-05	9.82E-05
f_6	HS-BM	1.37E-11	1.55E-10	5.21E-11	4.13E-11
	HSFA-BM	8.88E-16	5.33E-15	3.20E-15	1.40E-15

Table IV
THE SIMULATION RESULTS OF THREE ALGORITHMS(D=20)

Function name	Algorithm	Best	Worst	Average	Std
	HS	5.58E-05	5.55E-02	2.03E-02	2.26E-02
f_1	HS-BM	3.59E-08	1.25E-07	6.57E-08	3.06E-08
	HSFA-BM	2.17E-10	1.84E-09	6.31E-10	4.70E-10
	HS	5.82E-01	8.69E-01	7.22E-01	9.30E-02
f_2	HS-BM	2.27E-09	9.38E-09	5.70E-09	2.36E-09
	HSFA-BM	5.29E-11	2.00E-10	1.20E-10	4.22E-11
	HS	3.10E-02	9.36E-01	3.85E-01	2.37E-01
f_3	HS-BM	1.05E-05	3.41E-05	2.51E-05	7.30E-06
	HSFA-BM	1.67E-06	3.62E-06	2.83E-06	5.70E-07
	HS	1.41E-02	2.06E+01	3.89E+00	5.95E+00
f_4	HS-BM	2.96E-06	1.65E-05	8.67E-06	3.79E-06
	HSFA-BM	3.48E-08	1.13E-07	6.80E-08	2.34E-08
	HS	1.37E-01	4.18E+01	1.21E+01	1.33E+01
f_5	HS-BM	4.90E-09	8.04E-08	4.70E-08	2.18E-08
	HSFA-BM	2.59E-10	1.41E-09	5.48E-10	3.26E-10
	HS	3.76E-06	3.27E-03	1.03E-03	1.16E-03
f_6	HS-BM	2.50E-08	8.83E-08	6.55E-08	1.92E-08
	HSFA-BM	4.73E-10	3.19E-09	1.23E-09	7.55E-10

shows that when D=10, the optimal value of HSFA-BM is Contrasted with the other three meta-heuristic algorithms. Through this analysis of the results, it is proved that the effect of HSFA-BM algorithm is still significant compared with other algorithms. The three meta-heuristic algorithms are PSO, SA and DE. The numerical results have shown that the HSFA-BM algorithm jumps breaks away from local optimum, improves the astringency and the convergence speed.

VI. SUMMARY AND OUTLOOK

This paper puts forward a hybrid harmony search algorithm based on firefly algorithm and boltzmann

Table V
THE SIMULATION RESULTS OF THREE ALGORITHMS(D=30)

Function name	Algorithm	Best	Worst	Average	Std
f_1	HS	1.08E-03	1.03E-01	3.18E-02	3.77E-02
	HS-BM	2.26E-07	6.07E-07	4.53E-07	1.12E-07
	HSFA-BM	2.51E-08	5.88E-08	4.04E-08	1.02E-08
f_2	HS	8.85E-01	9.85E-01	9.42E-01	3.19E-02
	HS-BM	1.61E-08	4.38E-08	2.83E-08	7.28E-09
	HSFA-BM	2.14E-09	4.34E-09	3.12E-09	7.52E-10
f_3	HS	5.57E-03	7.96E-01	2.99E-01	2.90E-01
	HS-BM	5.44E-05	7.32E-05	6.43E-05	5.41E-06
	HSFA-BM	2.28E-05	3.26E-05	2.80E-05	2.93E-06
f_4	HS	4.25E-03	8.11E+00	1.74E+00	2.45E+00
	HS-BM	1.11E-04	1.88E-04	1.43E-04	2.19E-05
	HSFA-BM	1.36E-05	2.82E-05	2.08E-05	4.40E-06
f_5	HS	1.05E-01	5.07E+01	1.93E+01	1.74E+01
	HS-BM	4.54E-07	8.56E-07	6.96E-07	1.25E-07
	HSFA-BM	7.51E-08	9.78E-08	8.84E-08	7.08E-09
f_6	HS	2.00E-06	7.09E-03	1.98E-03	2.16E-03
	HS-BM	3.01E-07	4.79E-07	3.75E-07	5.81E-08
	HSFA-BM	4.36E-08	6.86E-08	5.39E-08	7.21E-09

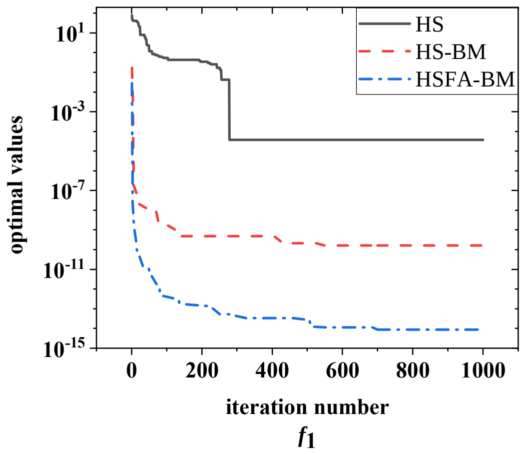

Fig. 6 Three algorithms test the simulation results of f_1 function.

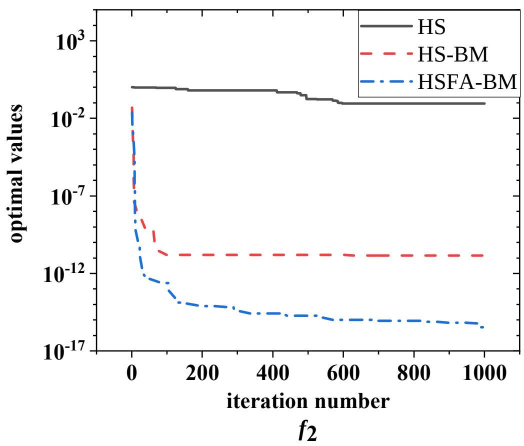

Fig. 7 Three algorithms test the simulation results of f_2 function.

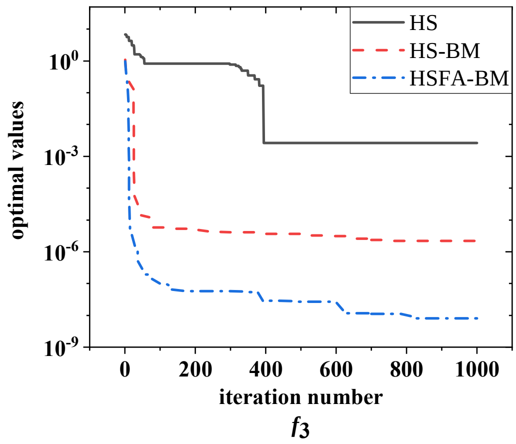

Fig. 8 Three algorithms test the simulation results of f_3 function.

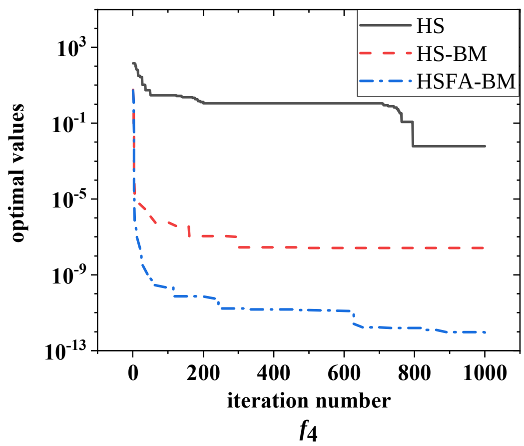

Fig. 9 Three algorithms test the simulation results of f_4 function.

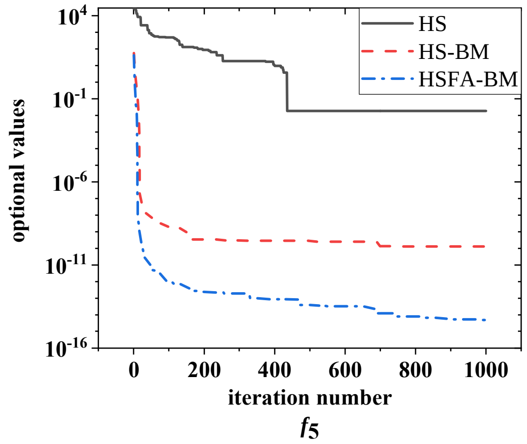

Fig. 10 Three algorithms test the simulation results of f_5 function.

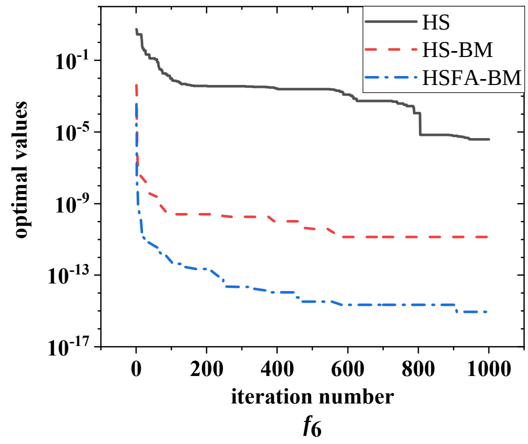

Fig. 11 Three algorithms test the simulation results of f_6 function.

Table VI
MANY RESULTS OF ALGORITHMS COMPARISON

Function name	*f*min	HSFA-BM	PSO	SA	DE
f_1	0	1.48E-15	8.47E-11	4.21E+0	2.59E+2
f_2	0	1.11E-16	3.30E-1	2.86E-1	1.66E+0
f_3	0	8.12E-09	2.52E-1	3.71E+0	1.08E+1
f_4	0	1.94E-13	1.62E-1	7.67E+1	3.97E+2
f_5	0	2.64E-15	7.21E-8	7.38E+3	9.64E+4
f_6	0	8.88E-16	1.58E-11	3.30E-1	3.23E+3

machine, namely HSFA-BM, which solves the problems of HS algorithm's dependence on the initial HM, poor convergence accuracy and slow convergence speed, etc. According to boltzmann machine principle, harmony search algorithm accepts the poor solution with a certain probability when updating the harmony memory, thus enhancing the ability of the algorithm to break away from the local optimum. The harmony search algorithm uses the firefly's flight mode to generate new solutions. Firefly's courtship flight can accurately grasp the direction and distance, so the process of the new solution generated by the firefly algorithm can quickly approach the optimal solution. The convergence precision and speed of the algorithm are improved.

In this paper, HSFA-BM is used to solve continuous problems and has achieved excellent results, which can continue to verify the ability to solve discrete problems. Some probability parameters in HSFA-BM are fixed values that can be set to dynamically adaptive values. HSFA-BM's incipient HM is generated at random, which is not universal. We can try to generate initial harmony with chaotic sequence.

REFERENCES

[1] Z.W. Geem, J.H. Kim and G. Loganathan, “A new heuristic optimization algorithm: harmony search,” Simulation, vol. 76, no. 2, pp60-68, 2001

[2] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. Del Ser, M.N. Bilbao, S. Salcedo-Sanz and Z.W. Geem, “A survey on applications of the harmony search algorithm,” Engineering *Applications* of Artificial Intelligence, vol. 26, no. 8, pp1818–1831, 2013

[3] A. Assad and K. Deep, “Applications of harmony search algorithm in data mining: A survey, in: Proceedings of Fifth International Conference on Soft Computing for Problem Solving,” Springer, vol. 437, pp863–874, 2016

[4] O. Moh'd Alia and R. Mandava, “The variants of the harmony search algorithm: an overview,” Artificial Intelligence Review, vol. 36, pp49-68, 2011

[5] Q.K.. Pan, P.N. Suganthan, M.F. Tasgetiren and J.J. Liang, “A self-adaptive global best harmony search algorithm for continuous optimization problems,” Applied Mathematics and Computation, vol. 216, no. 3, pp830–848, 2010

[6] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham and B.K. Panigrahi, “Exploratory power of the harmony search algorithm: analysis and improvements for global numerical optimization,” IEEE Transactions on Systems, Man, and Cybernetics, *Part B(Cybernetics)*, vol. 41, no. 1, pp89–106, 2011

[7] P. Yadav, R. Kumar, S.K. Panda and C. Chang, “An intelligent tuned harmony search algorithm for optimisation,” Information Sciences, vol. 196, pp47–72, 2012

[8] M. El-Abd, “An improved global-best harmony search algorithm,” Applied mathematics and computation, vol. 222, pp94–106, 2013
[9] I.A. Doush, et al., “Flow shop scheduling with blocking using modified harmony search algorithm with neighboring heuristics methods,” Applied Soft Computing Journal, vol.85, 2019
[10] Y. Melda, K.A. Ece and B. Gebrail, et al., “Adaptive-Hybrid Harmony Search Algorithm for Multi-Constrained Optimum Eco-Design of Reinforced Concrete Retaining Walls,” Sustainability, vol.13, no.4, pp1639-1639, 2021
[11] M.S. Rosle, M.S. Mohamad and Y. W. Choon, “A Hybrid of Particle Swarm Optimization and Harmony Search to Estimate Kinetic Parameters in Arabidopsis thaliana,” Processes, vol.8, no.8, 2020
[12] Y.C. Lee and A.Y. Zomaya, “Interweaving heterogeneous metaheuristics using harmony search,” in: Parallel & Distributed Processing, 2009. IPDPS , 2009. IEEE International Symposium on, IEEE, pp1-8, 2009
[13] K.K. Chakravarthi, L. Shyamala and V. Vaidehi, “Cost-effective workflow scheduling approach on cloud under deadline constraint using firefly algorithm,” Applied Intelligence, vol.51, no.3, pp1629-1644,2021
[14] I. Argatov Ivan and S. Chai Young , “Artificial neural network modeling of sliding wear,” Proceedings of the Institution of Mechanical Engineers, vol.235, no.4, pp748-757,2021
[15] X.S. Yang, “Harmony Search as a Metaheuristic Algorithm, in: Music-inspired harmony search algorithm,” Springer, vol. 191, pp1–14, 2009