mization capability is enhanced. The value of the variable $\alpha \otimes Levy(\lambda)$ will also tend to decrease when the global search is weakened in the later iterations and the value of $\alpha \otimes Levy(\lambda)$ is close to 0. It indicates that $T^t X \to HM^{best}\ (t \to \infty)$. In addition, according to the properties of the Cauchy point sequence, the position of the current population is composed of n-dimensional decision variables $X_j^{\ 1} = \left(X_{j_1}^1, X_{j_2}^1, X_{j_3}^1, \ldots, X_{j_{n-2}}^1, X_{j_{n-1}}^1, X_{j_n}^1\right)$ in HS-CS, and the calculation of the position of the next generation is based on the position of the previous generation, and so on. Hence, a set of matrices $X_j = \left(X_j^{\ 1}, X_j^{\ 2}, \ldots, X_j^{\ n-1}, X_j^{\ n}\right)$ that conforms to the Cauchy point column property could be obtained, and all iteratively computed positions are within the given boundary constraints $[LB_j, UB_j] \subset A$. Due to the global search updating strategy (equation (11)) and the boundary constraints, the $X_{j_0}^{t+1}$ can be represented as below:

$$X_{j_0}^{t+1} = \begin{cases} UB_j, if\left(T^t X^t\right)_{j_0} > UB_j, \\ LB_j, if\left(T^t X^t\right)_{j_0} < LB_j, \\ HM^{best,t} + \alpha(HM^{best,t} - HM^{worst,t}) \otimes Levy(\lambda), \\ \qquad otherwise \end{cases} \quad \text{and}$$

$X_{j_0}^{t+1}$, $X_{j_0}^{t} \in [LB_j, UB_j] \subset A$. The $f\left(X_{j_0}\right) = \hat{X} \in \mathbf{R}$ is fulfilled, that is $\lim_{n\to\infty} f\left(X_n\right) = f\left(X_{j_0}\right)$. It also indicates that the equation 19 fulfills $d\left(X_n, X_{j_0}\right) = \left|f\left(X_n\right) - f\left(X_{j_0}\right)\right| \to 0$, $(n \to \infty)$. Hence, $X_n \to X_{j_0}$ is obtained. It further proves that (A, d) is a complete metric space.

The last task is to prove that (A, d) is a separable metric space. Assuming $V = \bigcup_{k=1}^{\infty} \bigcup_{i=1}^{N_k} X_i^{(k)}$, it easy to conclude that the V a countable set and $V \subset A$. For any $X_n \in A$, because of the $B_1 = \bigcup_{i=1}^{N_1} B\left(X_i^{(i)}, 1\right) \supset A$, $X_{i_1}^{(i)}$ is chosen to ensure $X_n \in B\left(X_{i_1}^{(i)}, 1\right)$, where i_1 is a number from 1 to N_1. Due to $B_2 \supset A$, select the $X_{i_2}^{(i)}$, chosen to ensure $X_n \in B\left(X_{i_2}^{(i)}, 1/2\right)$. Due to the same approach, an infinite sequence $\left\{X_{i_k}^{(k)}\right\}_{k=1}^{\infty}$ can be gained to make $d\left(X_n, X_{i_k}^{(k)}\right) < 1/k$. Furthermore, $X_{i_k}^{(k)} \to X_n$, $(k \to \infty)$, might be concluded because $k \to \infty$ and because of the positive definiteness of d. The V is a dense subset in A. It also proved that the (A, d) is a separable metric space. To sum up, the (A, d) is a complete and separable metric space. □

Definition 1: The random operator $\varphi : \Omega \times A \to A$ is called a random compression operator if there is a non-negative real random variable such that Equation (20) satisfies the following conditions.

$$\rho\left(\left\{\delta : d\left(\varphi(\delta, X_i), \varphi(\delta, X_{i+1})\right) \le H(\delta) d\left(X_i, X_{i+1}\right)\right\}\right) = 1 \quad , X_i ,\ X_{i+1} \in A \tag{20}$$

Theorem 2: The map φ formed by each iteration update is a random compression operator in the fine-tuning and optimization (local search) stage of the HS-CS.

Proof: The HS-CS may produce better individual conditions than the previous generation in each iteration because of the appropriate fine-tuning selection operator and greedy selection strategy that are adopted. There is a non-negative real random variable $H(\delta)$ which fulfills:

$$\left|f\left(X_{i+1}\right) - f\left(X_{i+2}\right)\right| \le H(\delta)\left|\left(z - f\left(X_i\right)\right) - \left(z - f\left(X_{i+1}\right)\right)\right| \quad , 0 \le H(\delta) < 1 .$$

Furthermore,

$$\begin{aligned} d\left(\varphi(\delta, X_i), \varphi(\delta, X_{i+1})\right) &= d\left(X_{i+1}, X_{i+2}\right) \\ &= \left|\left(z - f\left(X_{i+1}\right)\right) - \left(z - f\left(X_{i+2}\right)\right)\right| \\ &= \left|f\left(X_{i+1}\right) - f\left(X_{i+2}\right)\right| \\ &\le H(\delta)\left|\left(z - f\left(X_i\right)\right) - \left(z - f\left(X_{i+1}\right)\right)\right| \\ &= H(\delta) d\left(X_i, X_{i+1}\right) \end{aligned}$$

$$\Omega_0 = \left\{\delta : d\left(\varphi(\delta, X_i), \varphi(\delta, X_{i+1})\right) \le H(\delta) d\left(X_i, X_{i+1}\right)\right\} \subseteq \Omega$$

$$\rho(\Omega_0) = 1$$

Hence, the mapping $\varphi : \Omega \times A \to A$ formed by each iteration update is a random compression operator. □

According to the above proven theorems, the

convergence of the proposed HS-CS algorithm is further verified using a convergence proof framework of the stochastic optimization algorithm proposed by Solis and Wets et al. (1981).

The following two assumptions are given at first.

Assumption 1: Assume that if the conditions $f\left(D(z,\xi)\right)\le f(z)$ and $\xi\in S$ $\left(R^n,B,\mu_k\right)$ hold, then $f\left(D(z,\xi)\right)\le f(\xi)$, where D is the function that generates the solution of the problem, ξ is a random variable generated from the probability space $\left(R^n,B,\mu_k\right)$, $S\subseteq R^n$ represents the constraint space of the problem, μ_k is the probability measure on B, and B is the σ domain of R^n subsets.

Assumption 2: For any Borel subset T of S, if the measure $v(T)>0$, then $\prod_{k=0}^{\infty}\left(1-\mu_k(T)\right)=0$ holds where, $v(T)$ is the n-dimensional closure in the subset T, and $\mu_k(T)$ is the probability of T obtained by μ_k.

Theorem 3: Assuming that the objective function f to be solved by the HS-CS algorithm is a measurable function, its solution space S is a measurable subset, and $\left\{HM_{best}^{(t)}\right\}_{t=0}^{\infty}$ is the solution sequence generated by the algorithm, then $\lim_{k\to\infty}P\left[HM_{best}^{t}\in R_{\varepsilon}\right]=1$ is established where, $P\left[HM_{best}^{t}\in R_{\varepsilon}\right]$ is the probability of $HM_{best}^{t}\in R_{\varepsilon}$, and R_{ε} is the set of global optimal points.

Proof: The iterative function D of HS-CS can be defined below.

$$D\left(HM_{best}^{t},X_i^{t}\right)=\begin{cases}HM_{best}^{t}, when & f\left(HM_{best}^{t}\right)\le f\left(X_i^{t}\right),\\ X_i^{t}, & otherwise\end{cases}$$

It is easy to prove that the iterative function satisfies Hypothesis 1.

In order to satisfy Assumption 2, the union of the sample space of a harmonic library of size N must contain S, which is $S\subseteq\bigcup_{i=1}^{N}M_i^{t}$, where M_i^{t} is the support set of the sample space of the *t-th* generation. According to the HS-CS algorithm under the two evolutionary equations 8 and 11, all individuals in the population will converge to the optimal position but it may not be the global optimal position.

Taking equation 11 as an example, because of $X_i\to HM_{\text{worst}}\Big/1+\alpha\otimes Levy(\lambda)$, $\lim_{t\to\infty}M_i^{t}=0$ when $t\to\infty$. As the iteration t grows, the closures $v\left[M_i^{t}\right]$ of each M_i^{t} and $v\left[\bigcup_{i=1}^{N}M_i^{t}\right]$ of its union $\bigcup_{i=1}^{N}M_i^{t}$ both gradually decrease. Therefore, there is a $\exists t_1$ such that $v\left[\bigcup_{i=1}^{N}M_i^{t}\cap S\right]<v(S)$ while $t>t_1$. That is, the evolution equation 11 does not satisfy hypothesis 2.

Therefore, under the condition that Assumption 2 is not satisfied, the HS-CS regenerates new individuals by setting a perturbation strategy. If a candidate solution is selected in a poor set of individuals, then there must be an integer t_2. It further indicates that $\beta\supseteq S$ while $t>t_2$, where β is the union of support sets that generate individuals after the perturbation strategy. Then, for the HS-CS algorithm, there is $\exists t'$, such that $\bigcup_{i=1}^{N}M_i^{t}\bigcup\beta\supseteq S$ while $t>t'$.

To sum up, assuming the Borel subset of S is $D=M_i^{t}$, there exists $v[D]>0$, $\mu_t[D]=\sum_{i=1}^{N}\mu_i^{t}[D]=1$. It is further obtained that $\prod_{t=0}^{\infty}\left(1-\mu_t(D)\right)=0$. It is easy to conclude that the HS-CS fulfills the assumption 2, so that $\lim_{k\to\infty}P\left[HM_{best}^{t}\in R_{\varepsilon}\right]=1$.

Hence, the HS-CS algorithm converges to the global optimal solution with probability 1. □

4 Function optimization problem using HS-CS

To verify the feasibility and robustness of the proposed HS-CS algorithm, two experiments are performed from two different aspects. One is the function optimization utilizing 12 benchmark functions, and the other is the fuzzy weight production rule extraction by the HS-CS algorithm and BPNN over IRIS.

This experiment runs under the environment of Intel (R) Core (TM) I5-10200H CPU @2.40GHz, 16G memory and the Windows10 operating system. The programming language is Python 3.7.

4.1 Function optimization

In this experiment, 12 classical benchmark functions, which include Sphere, LevyN13, Alpine, Rastrigin, Girewank, Ackley, Step, Schwefel 2.22, Bohachevsky, Quatric, RoseBroken, and Schwefel (the function datasets at http://www.sfu.ca/~ssurjano/) are selected to test the performance of the proposed HS-CS algorithm. These benchmark functions can be divided into two classes: the unimodal functions and the multimodal functions. The details of those selected benchmark functions are listed below.

Table 2. 12 benchmark functions

Name	Functions	Dim	Search space	f^*	Group
Sphere	$F_1=\sum_{i=1}^{D}x_i^2$	D	[-5.12,5.12]	0	Unimodal
LevyN13	$F_2=\sin^2(3\pi x_1)+(x_1-1)^2\left[1+\sin^2(3\pi x_2)\right]+(x_2-1)^2\left[1+\sin^2(2\pi x_2)\right]$	D	[-10,10]	0	Multi-modal
Alpine	$F_3=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	D	[-10,10]	0	Multi-modal
Rastrigin	$F_4=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	D	[-600,600]	0	Multi-modal
Girewank	$F_5=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	D	[-600,600]	0	Multi-modal
Ackley	$F_6=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	D	[-32,32]	0	Multi-modal
Step	$F_7=\sum_{i=1}^{D}\left[x_i+0.5\right]^2$	D	[-100,100]	0	Unimodal
Schwefel 2.22	$F_8=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	D	[-10,10]	0	Unimodal
Bohachevsky	$F_9=\sum_{i=1}^{D-1}\left[x_i^2+2x_{i+1}^2-0.3\cos(3\pi x_i)-0.4\cos(4\pi x_{i+1})+0.7\right]$	D	[-100,100]	0	Unimodal
Quatric	$F_{10}=\sum_{i=1}^{D}ix_i^4+rand()$	D	[-100,100]	0	Unimodal
RoseBroken	$F_{11}=\sum_{i=1}^{D}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	D	[-30,30]	0	Unimodal
Schwefel	$F_{12}=418.9829D-\sum_{i=1}^{D}x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	D	[-500,500]	0	Multi-modal

Where, ***f**** means the theoretical global optimal solution of the current function, ***Dim*** represents the dimension of the test function, and D is the value of the variable dimension. Furthermore, D is set as 10, 30, and 50 respectively. The purpose of setting different values of dimension is to verify whether these algorithms can jump out of the local search state quickly and effectively while falling into a local extremum in a high-dimensional solution problem.

4.2 Parameter design and experiment results

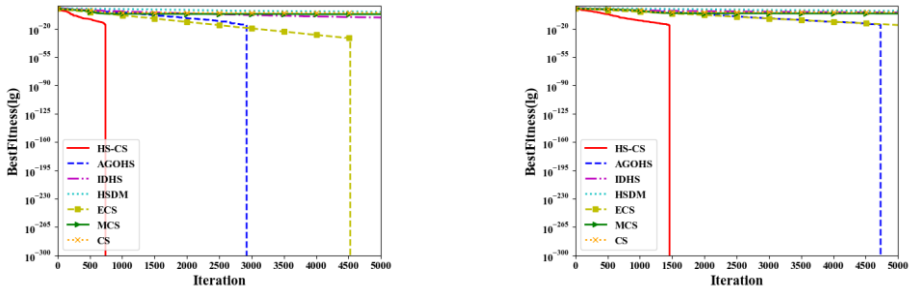
In this experiment, the HS-CS algorithm is compared with the other three improved HS algorithms and three CS algorithms. The other three HS algorithms are (i) global optimal adaptive harmony search algorithm (AGOHS) (Li et al. 2020), (ii) improved differential harmony search algorithm (IDHS)(Wang et al., 2019), and (iii) harmony search with differential mutation based on pitch adjustment (HSDM)(Qin et al., 2011)). The three CS algorithms are: (i) novel enhanced cuckoo search algorithm (ECS) (Kamoona and Patra, 2019) (ii) modified cuckoo search (MCS) (Ong and Zainuddin, 2019), and (iii) cuckoo search (CS) (Yang and Deb, 2009)). In addition, to ensure the fairness of all algorithms in numerical experiments, the population size and the maximum iteration, times were set to 100 and 5000, respectively. The related parameter setting of these algorithms is shown in Table 3.

To reduce the randomness of the algorithm and maintain the fairness of the algorithm, the optimization operation of each function is independently run 30 times (*Num=30*).

Figures 1 and 2 show part of the experimental results. Furthermore, the entire experiment results and corresponding analysis are given in the supplementary material.

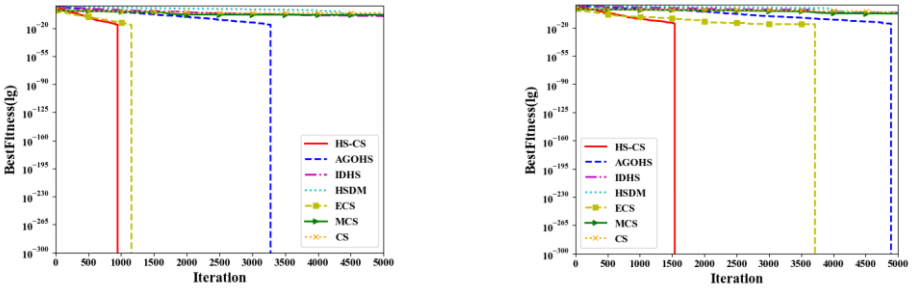
Table 3 The parameter settings of the examined algorithms

Algorithms	Population	Iteration	Parameters
HS-CS	100	5000	$HMCR_{min}=0.8$, $HMCR_{max}=0.9$, $PAR_{min}=0.1$, $PAR_{max}=0.9$, $\alpha_0=0.01$
AGOHS	100	5000	$HMCR_{min}=0.8$, $HMCR_{max}=0.9$, $PAR_{min}=0.1$,$PAR_{max}=0.9$, $F=N(0.5,0.3)$
IDHS	100	5000	$HMCR_{min}=0.8$, $HMCR_{max}=0.9$,$PAR_{min}=0.1$, $PAR_{max}=0.9$
HSDM	100	5000	$HMCR=0.98$
ECS	100	5000	$\alpha=0.01$, $P_a=0.25$, $\lambda=1.5$
MCS	100	5000	$\alpha=0.01$, $P_a=0.25$, $\lambda=1.5$
CS	100	5000	$\alpha=0.01$, $P_a=0.25$, $\lambda=1.5$



(a) 30 dimensions (b) 50 dimensions

Fig.1 Optimize the convergence curve of Step Function



(a) 30 dimensions (b) 50 dimensions

Fig.2 Optimizing the convergence curve of the Girewank Function

As a unimodal function, the Step function is usually used to test the convergence accuracy and speed of the algorithm. The HS-CS algorithm integrated with Levy flight to broaden the search scope of the population, enhances the position update strategy and improves the global exploration ability. As shown

in the Figure 1, whether the Step function is 30-dimension or 50-dimension, the proposed algorithms all exhibit the best convergence curves. In 30-dimension, although the convergence accuracy of the AGOHS and ECS algorithm is similar to the proposed algorithm, they require more iterations. When the dimensionality of the test function increases, the AGOHS algorithm requires more iterations to achieve the same convergence accuracy as the HS-CS algorithm. In the set number of iterations, the ECS algorithm fails to achieve appreciable results. Therefore, the HS-CS algorithm also has certain advantages over the selected algorithms in terms of stability.

As a multimodal function, the Girewank function is usually used to test the ability of the algorithm to jump out of the local optimum. As shown in Figure 2, when the dimension of the test function is low, since the search mechanism of Levy flight of the CS was adopted by the ECS and HS-CS, the convergence efficiency of those algorithms was similar. The AGOHS algorithm requires more iterations to obtain similar convergence accuracy to the HS-CS algorithm. As the dimension of the test function increases, the convergence efficiency of the AGOHS and ECS algorithm decreases significantly. In the search process, the HS-CS algorithm indicated the direction of "pitch adjusting and select the best". The Levy flight in CS operator is adopted to find candidate individuals when updating the HM, which enriches the number of alternative solutions and strengthen the disturbance to avoid falling into stagnation prematurely in the search process. Thus, the HS-CS algorithm still maintains a good convergence efficiency.

On the other hand, the further analysis of the experiment using the Wilcoxon rank sum test, the mean value (*MEAN*) and the standard deviation (*SD)* is discussed in detail in the supplementary material. Combining the analysis above and in the supplementary material, the HS-CS algorithm exhibited good performance in dealing with high-dimensional optimization problems, proving that it has obvious fast convergence and self-adaptive ability.

5 Weighted fuzzy production rule extraction model based on HS-CS

To further verify the practicability of the HS-CS algorithm, another typical application-weighted fuzzy production rule extraction from the IRIS dataset utilizing the HS-CS and BPNN is demonstrated here. The entire extraction process can be separated into the following main modules: (i) data fuzzification, (ii) BPNN generation and (iii) BPNN framework optimization using the HS-CS algorithm , the importance index matrix obtained from the trained BPNN, and the weighted fuzzy production rules extraction.

5.1 Experimental data

In the IRIS plant classification dataset, there are three classification problems: Setosa, Versicolor, and Virginica. Each category contains 50 samples, a total of 150 sample examples. Moreover, each sample contains 4 attributes, such as Sepal Length (SL), Sepal Width (SW), Petal Length (PL), Petal Width (PW). The measurement unit of each attribute is cm.

5.2 The Weighted fuzzy production Rules extraction implementation analysis

According to our presented framework (Li et al., 2020), the entire extraction process includes the following key steps:

First, IRIS dataset fuzzification. For each example attribute of the IRIS dataset, three fuzzy sets (semantic attribute values) are used to mark each example attribute, which are large (LG), medium (MED), and small (SM).

Then, the corresponding BPNN is established by the IRIS dataset. In this module, 12 semantic attribute values of the IRIS dataset are used to generate rule preconditions, and the 3 classification results are used to build 12 input nodes of input layer, 3 output nodes of output layer, 1 hidden layer and 4 hidden nodes generated by experience. The Sigmoid function is used to obtain BPNN as an activation function. Meanwhile, the fuzzy data set is randomly divided into two parts, in which 80% of the examples are used as the training set and the remainder are used as the test set.

Thirdly, the HS-CS is employed to optimize the related parameters and the objective function of the obtained BPNN. The parameters of the HS-CS set as follows: HMS=100, $HMCR_{max}=0.9$, $HMCR_{min}=0.8$, $PAR_{min}=0.1$, $PAR_{max}=0.9$, $\alpha=0.01$, the dimension of each harmony $D=12$ (reflects the number of input nodes), the learning rate is 0.02, the penalty factor is

0.001, and the momentum is 0.9. Furthermore, the threshold value is set as 0.5 after training the BPNN is completed by the HS-CS algorithm. The connection weights connected to the threshold are "pruned" for removing redundant paths and obtaining a simple BPNN structure.

Fourthly, the corresponding importance index matrix $W_{n\times m}=OW_{n\times k}*IW_{k\times m}$ can be gained while the BPNN optimization and training is completed. Among which, $OW_{n\times k}$ represents the connection weight matrix between the hidden layer and the output layer in the BPNN, k represents the number of nodes in the hidden layer, n represents the number of output nodes in the output layer. $IW_{k\times m}$ represents the connection weight matrix from the input layer to the hidden layer, and m represents the number of input nodes.

Through the above steps, the importance index matrix of the obtained BPNN by the HS-CS is as follows:

$$W_{3\times12}=\begin{pmatrix} 0 & -0.72 & -10.92 & 0 & 0 & 0.44 & 21.81 & -1.25 & -1.36 & 11.49 & -20.79 & -1.4 \\ 0 & -6.53 & 5.95 & 0 & 0 & 4 & -12.89 & 22.29 & -22.77 & -0.66 & 29.8 & -12.79 \\ 0 & 5.75 & 9.82 & 0 & 0 & -3.52 & -16.26 & -14.82 & 34.18 & -4.11 & -16.48 & 11.26 \end{pmatrix}$$

In the $W_{3\times12}$ importance index matrix, 3 rows and 12 columns represent 3 categories and 12 semantic attribute values corresponding to IRIS, respectively. Moreover, the attribute value does not play a role in the IRIS classification if the corresponding values in columns 1, 4, and 5 are all 0.

In the importance index matrix, the classification result corresponding to each row may be the same or multiple rules. If the attribute value element of the unified attribute is not 0, multiple rules will be generated, that is there is NOT OR in the preceding one. If the attribute value element is negative, the corresponding NOT antecedent is generated.

Based on the above requirements, the weighted fuzzy production rules can be automatically extracted from the importance index matrix. This paper only lists the first row of the matrix produces 18 classification rules as Iris-Setosa and the entire generated local weighted fuzzy production rules could be found in the supplementary material.

The first row of the matrix produces 18 classification rules as Iris-Setosa.

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is SM [21.81] and PW is SM [11.49] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is SM [21.81] and PW is NOT MED [20.79] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is SM [21.81] and PW is NOT LGR [1.4] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is NOT MED [1.25] and PW is SM [11.49] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is NOT MED [1.25] and PW is NOT MED [20.79] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is NOT MED [1.25] and PW is NOT LGR [1.4] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is NOT LGR [13.36] and PW is SM [11.49] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is NOT LGR [13.36] and PW is NOT MED [20.79] THEN Iris-Setosa

IF SL is NOT MED [0.72] and SW is LGR [0.44] and PL is NOT LGR [13.36] and PW is NOT LGR [1.4] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is SM [21.81] and PW is SM [11.49] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is SM [21.81] and PW is NOT MED [20.79] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is SM [21.81] and PW is NOT LGR [1.4] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is NOT MED [1.25] and PW is SM [11.49] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is NOT MED [1.25] and PW is NOT MED [20.79] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is NOT MED [1.25] and PW is NOT LGR [1.4] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR

[0.44] and PL is NOT LGR [13.36] and PW is SM [11.49] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is NOT LGR [13.36] and PW is NOT MED [20.79] THEN Iris-Setosa

IF SL is NOT LGR [10.92] and SW is LGR [0.44] and PL is NOT LGR [13.36] and PW is NOT LGR [1.4] THEN Iris-Setosa

To highlight the advantages of the proposed HS-CS algorithm, two other algorithms are used to extract the weighted fuzzy production rules from the IRIS dataset under the same experimental environment, respectively the standard HS (Geem and Kim, 2001) and the AGOHS (Li et al., 2020).

The extracted rules are classified and verified, and then applied to the training set and the test set for comparison. The related data are listed in Table 4.

Table 4 Comparison of IRIS classification accuracy results

Method	Training Set	Testing Set
BPNN Trained by HS-CS	96.45%	97.37%
Accuracy of obtained weighted fuzzy production rules	95.54%	97.37%
BPNN Trained by AGOHS	96.32%	97.36%
Accuracy of obtained weighted fuzzy production rules	66.07%	68.42%
BPNN Trained by HS	96.39%	97.36%
Accuracy of obtained weighted fuzzy production rules	65.72%	68.42%

Table 4 indicates that the proposed HS-CS algorithm has more accuracy than the AGOHS and HS algorithm while optimizing BPNN and extracting the weighted fuzzy generation rules from the BPNN. Meanwhile, the accuracy using the HS-CS on the testing set reaches 97.37%. It further verifies that the HS-CS algorithm has the advantage of enhancing the learning and generalization ability of BPNN. On the other hand, the execution time of implementing the IRIS classification of HS, HS-CS and AGOHS is 35.51s, 40.15s and 370.23s, respectively. Although the execution time of HS is less than that of HS-CS, the accuracy of HS in data classification is less than that of HS-CS. Within a reasonable execution time range, HS-CS can achieve a better classification effect.

5.3 Comparisons of the loss degree and precision curves of the optimized BPNN

Figures 3-4 show the convergence curve and convergence precision curve of the loss function of BPNN optimized by the different HS algorithms in the training process of the IRIS dataset.

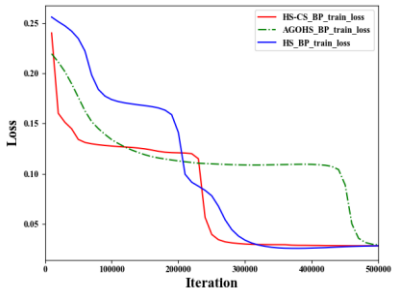


Fig.3 Convergence curve of the loss function of BPNN with different HS algorithms

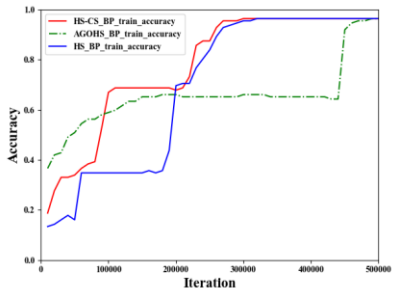


Fig.4 Convergence precision curve of the loss function of BPNN with different HS algorithms

From the convergence diagram and convergence precision diagram of the loss degree function above, it is further indicated that the convergence speed and loss value of the BPNN optimization by the proposed HS-CS algorithm network have obvious convergence advantages and lower loss value compared with the AGOHS and standard HS algorithms. It also illustrates that the HS-CS algorithm plays a certain role in optimized BPNN training process.

To sum up, in the process of fuzzy weighted rule extraction using the HS-CS algorithm, the optimized BPNN by the HS-CS extracts the fuzzy weighted production rules from the IRIS dataset in a relatively short execution time and achieves a better classification effect. The main functions of the HS-CS are to improve the convergence speed of BPNN and reduce the loss during training.

6 Conclusion

In this study, a modified HS-CS is presented to solve the problems of the standard HS which is easily trapped into the local extreme value and is weak in global search ability. In particular, the HS-CS algorithm makes use of Levy flight to explore the solution space, which further enriches the population density

of the HS algorithm, enhances the global search ability, and avoids the algorithm getting trapped in a local optimum. On the other hand, "pitch adjusting and selecting the best" in the improvisation stage and the inertial weight operator is constructed and the degree of the internal individuals in HM are selected to improve the efficiency and convergence accuracy of the basic HS algorithm.

Two experiments are implemented to verify the effectiveness of the proposed HS-CS algorithm. Firstly, the existence of the proposed algorithm limit is proved by differential equation, and the global optimal algorithm is verified by random functional analysis and random search theory. Secondly, the HS-CS and other six algorithms are used to settle the function optimization issue of the 12 selected classical benchmark functions in different high dimensions. The results showed that the HS-CS algorithm can still maintain the speed and precision of rapid convergence in the process of solving the optimization of high-dimensional functions and has strong robustness. Finally, the HS-CS is applied to the IRIS classification for extracting the weighted fuzzy production rules. Through data analysis, it is easy to find that weighted fuzzy production rules obtained by combining BPNN and HS-CS have a high precision because the HS-CS algorithm can enhance the learning and generalization ability of the BPNN effectively.

This research is mainly intended to optimize the solution of a single objective problem for the HS-CS without comprehensive consideration of the impact of multiple factors on the accuracy of the algorithm. Therefore, the influence of multiple factors will be considered in the HS in future work, and the characteristics of the Pareto solution and HS will be used to solve the real-life multi-objective optimization problems, such as obstacle avoidance for multiple unmanned aerial vehicles (UAVs) in coordinated formation.

Contributors

Kaiqing ZHOU designed the research. Xiaoqiang YE and Kaiqing ZHOU processed the data. Shaoqiang Ye and Fangling Wang drafted the manuscript. Kaiqing ZHOU and Azlan Mohd ZAIN helped organize the manuscript. Azlan Mohd ZAIN and Yusliza YUSOFF revised and finalized the paper.

Compliance with ethics guidelines

Yeshao Qiang, Kaiqing ZHOU, Azlan Mohd ZAIN, Fangling Wang and Yusliza YUSOFF declare that they have no conflict of interest.

References

Abbasi M, Abbasi E, Mohammadi-Ivatloo B, 2021. Single and multi-objective optimal power flow using a new differential-based harmony search algorithm. Journal of Ambient Intelligence and Humanized Computing, 12(1): 851-871. https://doi.org/10.1007/s12652-020-02089-6

Al-Shaikh A, Mahafzah B A, Alshraideh M, 2021. Hybrid harmony search algorithm for social network contact tracing of COVID-19. Soft Computing, p.1-23. https://doi.org/10.1007/s00500-021-05948-2

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N, 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review, 42(1), 21-57. https://doi.org/10.1007/s10462-012-9328-0

Costa A, Fernandez-Viagas V, 2022. A modified harmony search for the T-single machine scheduling problem with variable and flexible maintenance. Expert Systems with Applications, 198: 116897. https://doi.org/10.1016/j.eswa.2022.116897

Geem Z W, Kim J H, Loganathan G V, 2001. A new heuristic optimization algorithm: harmony search. simulation, 76(2): 60-68. https://doi.org/10.1177/003754970107600201

Gong J, Zhang Z, Liu J, et al., 2021. Hybrid algorithm of harmony search for dynamic parallel row ordering problem. Journal of Manufacturing Systems, 58: 159-175. https://doi.org/10.1016/j.jmsy.2020.11.014

Gupta S, 2022. Enhanced harmony search algorithm with non-linear control parameters for global optimization and engineering design problems. Engineering with Computers, 38(4): 3539-3562. https://doi.org/10.1007/s00366-021-01467-8

Jagatheesan K, Anand B, Samanta S, et al., 2017. Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm. IEEE/CAA Journal of Automatica Sinica, 6(2): 503-515. https://doi.org/10.1109/JAS.2017.7510436

Kamoona A M, Patra J C, 2019. A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images. Applied Soft Computing, 85: 105749. https://doi.org/10.1016/j.asoc.2019.105749

Li H C, Zhou K Q, Mo L P, et al., 2020. Weighted Fuzzy Production Rule Extraction Using Modified Harmony Search Algorithm and BP Neural Network Framework. IEEE Access, 8: 186620-186637. https://doi.org/10.1109/ACCESS.2020.3029966

Mirjalili S, 2015. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-based systems, 89: 228-249. https://doi.org/10.1016/j.knosys.2015.07.006

Mousavi S M, Abdullah S, Niaki S T A, et al., 2021. An intelligent hybrid classification algorithm integrating fuzzy rule-based extraction and harmony search optimization: Medical diagnosis applications. Knowledge-Based Systems, 220: 106943. https://doi/org/10.1016/j.knosys.2021.106943

Ong P, Zainuddin Z, 2019. Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. Applied Soft Computing, 80: 374-386. https://doi.org/10.1016/j.asoc.2019.04.016

Ouyang H, Gao L, Li S, 2018. Amended harmony search algorithm with perturbation strategy for large-scale system reliability problems. Applied Intelligence, 48(11): 3863-3888. https://doi.org/10.1007/s10489-018-1175-5

Qin A K, Forbes F, 2011. Harmony search with differential mutation based pitch adjustment. Proceedings of the 13th annual conference on Genetic and evolutionary computation. p.545-552. https://doi.org/10.1145/2001576.2001651

Qin F, Zain A M, Zhou K Q, 2022. Harmony Search Algorithm and Related Variants: A Systematic Review. Swarm and Evolutionary Computation, 101126. https://doi.org/10.1016/j.swevo.2022.101126

Shaffiei Z A, Abas Z A, Yunos N M, et al., 2019. Constrained self-adaptive harmony search algorithm with 2-opt swapping for driver scheduling problem of university shuttle bus. Arabian Journal for Science and Engineering, 44(4): 3681-3698. https://doi.org/10.1007/s13369-018-3628-x

Singh N, Kaur J, 2021. Hybridizing sine–cosine algorithm with harmony search strategy for optimization design problems. Soft Computing, 25(16): 11053-11075. https://doi.org/10.1007/s00500-021-05841-y

Solis F J, Wets R J B, 1981. Minimization by random search techniques. Mathematics of operations research, 6(1): 19-30. https://doi.org/10.1287/moor.6.1.19

Tang J, Liu G, Pan Q, 2021. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. IEEE/CAA Journal of Automatica Sinica, 8(10): 1627-1643. https://doi.org/10.1109/JAS.2021.1004129

Tu Q, Chen X, Liu X, 2019. Multi-strategy ensemble grey wolf optimizer and its application to feature selection. Applied Soft Computing, 76: 16-30. https://doi.org/10.1016/j.asoc.2018.11.047

Valaei M R, Behnamian J, 2017. Allocation and sequencing in 1-out-of-N heterogeneous cold-standby systems: multi-objective harmony search with dynamic parameters tuning. Reliability Engineering & System Safety, 157: 78-86. https://doi.org/10.1016/j.ress.2016.08.022

Wang L, Hu H, Liu R, et al., 2019. An improved differential harmony search algorithm for function optimization problems. Soft Computing, 23(13): 4827-4852. https://doi.org/10.1007/s00500-018-3139-4

Wang Y, Gao S, Zhou M, et al., 2020. A multi-layered gravitational search algorithm for function optimization and real-world problems. IEEE/CAA Journal of Automatica Sinica, 8(1): 94-109. https://doi.org/10.1109/JAS.2020.1003462

Yang X S, Deb S, 2009. Cuckoo search via Lévy flights. 2009 World congress on nature & biologically inspired computing (NaBIC). Ieee, p.210-214. https://doi.org/10.1109/NABIC.2009.5393690

Ye S Q, Zhou K Q, Zhang C X, et al., 2022. An Improved Multi-Objective Cuckoo Search Approach by Exploring the Balance between Development and Exploration. Electronics, 11(5): 704. https://doi.org/10.3390/electronics11050704

Zhao Z, Liu S, Zhou M C, et al., 2020. Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem. IEEE/CAA Journal of Automatica Sinica, 8(6): 1199-1209. https://doi.org/10.1109/JAS.2020.1003539

Zhu Q, Tang X, Elahi A, 2021. Application of the novel harmony search optimization algorithm for DBSCAN clustering. Expert Systems with Applications, 178: 115054. https://doi.org/10.1016/j.eswa.2021.115054