Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A modified harmony search algorithm and its applications in weighted fuzzy production rule extraction*#

Shaoqiang YE[1], Kaiqing ZHOU‡[1], Azlan Mohd ZAIN[2], Fangling WANG[1], Yusliza YUSOFF[2]

[1]*School of Communication and Electronic Engineering, Jishou University, Jishou 416000, China*

[2]*Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310, Johor, Malaysia*

E-mail: shaoq_ye@163.com; kqzhou@jsu.edu.cn; azlanmz@utm.my; fanglingong@163.com; yusliza@utm.my

Received Aug. 3, 2022; Revision accepted Jan. 8, 2023; Crosschecked Oct. 30, 2023

Abstract: Harmony search (HS) is a form of stochastic meta-heuristic inspired by the improvisation process of musicians. In this study, a modified HS with a hybrid cuckoo search (CS) operator, HS-CS, is proposed to enhance global search ability while avoiding falling into local optima. First, the randomness of the HS pitch disturbance adjusting method is analyzed to generate an adaptive inertia weight according to the quality of solutions in the harmony memory and to reconstruct the fine-tuning bandwidth optimization. This is to improve the efficiency and accuracy of HS algorithm optimization. Second, the CS operator is introduced to expand the scope of the solution space and improve the density of the population, which can quickly jump out of the local optimum in the randomly generated harmony and update stage. Finally, a dynamic parameter adjustment mechanism is set to improve the efficiency of optimization. Three theorems are proved to reveal HS-CS as a global convergence meta-heuristic algorithm. In addition, 12 benchmark functions are selected for the optimization solution to verify the performance of HS-CS. The analysis shows that HS-CS is significantly better than other algorithms in optimizing high-dimensional problems with strong robustness, high convergence speed, and high convergence accuracy. For further verification, HS-CS is used to optimize the back propagation neural network (BPNN) to extract weighted fuzzy production rules. Simulation results show that the BPNN optimized by HS-CS can obtain higher classification accuracy of weighted fuzzy production rules. Therefore, the proposed HS-CS is proved to be effective.

Key words: Harmony search algorithm; Cuckoo search algorithm; Global convergence; Function optimization; Weighted fuzzy production rule extraction

https://doi.org/10.1631/FITEE.2200334 **CLC number:** TP181

1 Introduction

Swarm intelligence (SI) optimization is a probabilistic search algorithm that operates by simulating the cooperative behavior of social biological groups. It has always been one of the research hotspots in the artificial intelligence domain (Tang et al., 2021). There are many complex and challenging optimization problems in this area. The traditional mathematical optimization schemes, such as the quasi-Newton method and gradient descent method, can hardly solve large-scale complex optimization problems. Due to the limitation of traditional schemes, the SI algorithm,

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 62066016), the Natural Science Foundation of Hunan Province, China (No. 2020JJ5458), the Research Foundation of Education Bureau of Hunan Province, China (Nos. 22B0549 and 22B1046), the Fundamental Research Grant Scheme of Malaysia (No. R.J130000.7809.5F524), the UTMFR Grant (No. Q.J130000.2551.20H71), and the Research Management Center (RMC) of Universiti Teknologi Malaysia (UTM)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2200334) contains supplementary materials, which are available to authorized users

ORCID: Shaoqiang YE, https://orcid.org/0000-0002-8117-9764; Kaiqing ZHOU, https://orcid.org/0000-0001-5779-7135; Azlan Mohd ZAIN, https://orcid.org/0000-0003-2004-3289; Fangling WANG, https://orcid.org/0000-0003-2362-7765; Yusliza YUSOFF, https://orcid.org/0000-0003-3213-1921

using its stochastic walk search characteristics in solving a variety of optimization problems, has become more and more popular and is widely used in various applications such as cyber-physical social systems and feature selection (Tu et al., 2019). Over the past few decades, many SI algorithms have been developed, including the universal gravitational force between objects which was inspired by the gravitational search algorithm (Wang YR et al., 2021). Further SI algorithms have been developed, including the memetic algorithm (MA) which is based on a simulation of cultural evolution (Zhao et al., 2021), the flickering behavior among fireflies which simulates the firefly algorithm (FA) (Jagatheesan et al., 2019), the artificial bee colony (ABC) algorithm which is simulated by bee division and cooperative behavior (Karaboga et al., 2014), the moth-flame optimization (MFO) by simulation of the special navigation of moths at night (Mirjalili, 2015), and the harmony search (HS) algorithm inspired by the improvisation process in which musicians repeatedly adjust the pitch of different instruments to create the most beautiful music (Geem et al., 2001).

HS has the advantages of fewer parameters, simple operation, and easy implementation compared with peer algorithms. However, due to the disadvantages of low optimization speed, relatively poor population diversity, and easy premature convergence, scholars have proposed various improvements to it. Valaei and Behnamian (2017) proposed a new parameter adjustment method, using a dynamic adjustment Taguchi method to improve the pitch adjusting rate (PAR) and to obtain an improved multi-objective HS algorithm. Ouyang et al. (2018) employed a perturbation strategy, key parameter adjustment, and a global selection strategy to balance the capabilities of exploration and exploitation in an amended HS (AHS). AHS was applied to the reliability problem of large-scale systems of non-convex integer nonlinear programming. The search efficiency and convergence performance of the AHS algorithm were then evaluated for performance verification. Shaffiei et al. (2019) proposed a constrained self-adaptive HS with 2-opt (CSAHS-2opt) for solving the driver scheduling problem of the university shuttle bus (DSPUSB). The bandwidth (also called fret width) value was dynamically changed and determined based on the current solution of each driver every week and, when generating a schedule under some constraints, the proposed algorithm could obtain the optimal scheduling scheme. Li et al. (2020) proposed a global optimal adaptive HS algorithm (AGOHS) to obtain weighted fuzzy generation rules from datasets. This algorithm realizes rule extraction by optimizing the back propagation neural network (BPNN) and shows high rule extraction accuracy. Abbasi et al. (2021) used the mutation operation of differential evolution (DE) to replace the bandwidth to improve the local exploitation ability. At the same time, a differential-based HS (DH/best) algorithm was proposed through dynamic distance adjustment between the harmonies in the harmony memory (HM). Harmony was used to enhance search ability in the process of algorithm update. When optimizing the IEEE 118-bus and 57-bus systems, the DH/best algorithm achieved the lowest actual power loss, improved the power of voltage, and reduced the minimizing active power of generation units. Singh and Kaur (2021) embedded the HS algorithm in a sine–cosine algorithm (SCA) to make up for the poor global optimization and low convergence speed of the SCA and proposed a hybrid SCA with HS (HSCAHS). Al-Shaikh et al. (2023) integrated the hill climbing algorithm with HS, and proposed a hybrid harmony search contact tracing (HHS-CT) algorithm for social network contact tracing of the COVID-19 infection. The algorithm could accurately track contacts through social networks and prevent further spread of the epidemic. To overcome the uncertain factors of medical datasets and improve the efficiency of medical diagnosis, Mousavi et al. (2021) used the Taguchi method to adjust the parameters of HS, so that the algorithm could find the best rule in a fuzzy rule based system. Meanwhile, external cross-validation (CV) and internal CV were used to verify and classify the obtained rules to obtain better classification results. Zhu et al. (2021) proposed the K-density based spatial clustering of application with noise (K-DBSCAN) clustering concept. However, the DBSCAN clustering algorithm has difficulty in predicting appropriate clustering parameters; therefore, HS was used to optimize clustering parameters. The experimental results demonstrated that the K-DBSCAN optimized by HS has good clustering performance and high clustering accuracy when dealing with four different datasets. To

optimize the dynamic parallel row ordering problem (DPROP) on the production line, Gong et al. (2021) proposed a hybrid tabu search and HS. The results showed that the algorithm obtained the best solution when solving the DPROP with 30–40 large components. To overcome the defects of the traditional HS, Gupta (2022) proposed a modified HS algorithm (MHSA). A new calculation formula was reconstructed for the search strategy, harmony memory consideration rate (HMCR), and PAR of HS, to enhance search efficiency and accuracy. The algorithm obtained the optimal value in engineering structure design and the performance was far better than those of other comparison algorithms. Costa and Fernandez-Viagas (2022) proposed a self-adaptive HS (SAHS) mechanism to solve the single machine scheduling problem with flexible/variable maintenance in a specific time window.

To sum up, although HS has made some achievements in theoretical and practical application research, there are some shortcomings of the HS algorithm, such as poor convergence accuracy and ease of falling into a local extremum state, in solving high-dimensional problems (Qin F et al., 2022). Therefore, in this paper we propose an improved HS algorithm. The local search strategy and updating mechanism in HS are improved by introducing cuckoo search (CS) (Yang and Deb, 2009; Ye et al., 2022). The proposed algorithm guides the iterative direction of the HS algorithm, prevents the algorithm from falling into stagnation, and enhances its population diversity and convergence efficiency. The specific improvement methods are as follows:

At the optimization stage of HS, due to the randomness of the HS pitch disturbance adjusting method, the exploitation optimization ability is insufficient because of poor solution accuracy and low convergence speed. To improve the optimization ability of HS, an adaptive inertia weight is constructed based on the relationship of information transfer between individuals in HM. Then the optimal solution in HM is used to replace the bandwidth for ensuring that convergence is towards the best position closely and quickly. Thus, these strategies guarantee that the proposed approach has the property to improve the search accuracy and efficiency of local optimization of HS in the later iterations.

At the global exploration stage of HS, new individuals are randomly generated, which causes some defects, such as weak global search ability and poor population diversity. To enhance the global search ability and population diversity of HS, the CS operator is employed in the random generation of new harmony (1−HMCR) to expand and update the solution vector of harmony. In addition, the CS operator is used to select candidate individuals from the poor HM, which enlarges the number of alternative solutions and avoids the HS from falling into the stagnation state of a local optimum during updating HM, thus further improving the accuracy of HS.

To strengthen the adaptability of HS, HMCR and PAR are adjusted adaptively. The former increases linearly and the latter decreases linearly as the number of iterations increases. This ensures that HS can quickly search the global optimal solution in the search region and enriches the adaptability of HS in the search process.

Furthermore, to provide the feasibility and robustness of the proposed HS-CS algorithm (a modified HS with a hybrid CS operator), three theorems are presented and proved to reveal that HS-CS is a global convergence meta-heuristic algorithm.

Finally, two experiments are conducted to verify the feasibility and robustness of HS-CS. Twelve classic functions are selected as benchmarks to reveal the highlights of the proposed HS-CS algorithm. As in our previous work, HS-CS is also used to extract the weighted fuzzy production rules from a given dataset to demonstrate some highlights of the proposed HS-CS, such as high convergence speed, high precision, and self-adaptation ability.

2 Standard HS and CS algorithms

The inspiration, critical operations, and implementations of the standard HS and CS are discussed in detail in this section.

2.1 Standard HS algorithm

HS is a meta-heuristic algorithm that simulates the process of musicians to create the most beautiful music through repeatedly adjusting the tones of different musical instruments. The core operation is to

obtain the next generation of harmony solution vectors based on HMCR, PAR, and random selection. The primary thinking of the entire HS could be summarized as follows:

First, harmony solution vectors which keep the same volume as harmony memory size (HMS) will be created randomly and be stored in HM. Moreover, the harmony solution vectors will be selected from HM with the probability of HMCR, and the new harmony solution vectors will be generated with the probability of 1−HMCR.

Second, the selected harmony solution vector from HM will be tweaked by the probability of PAR to generate a new harmony solution vector. Meanwhile, the selected harmony solution vector in HM will not be processed with the probability of 1−PAR.

Finally, a judgment will be used to evaluate the new harmony solution, whether the vector is better than the worst harmony solution vector in HM or not. If yes, the worst harmony solution vector will be replaced by the new harmony solution vector. Otherwise, the HM will be unchanged until it fulfills the termination condition.

According to the above three main strategies, the implementation steps of the standard HS can be classified into the following five main steps:

Step 1: definition

In this step, the minimum optimization problem function is first defined as

$$\min f(x), \tag{1}$$

where $x=\left(x_1, x_2, \cdots, x_n\right)^{\mathrm{T}} \in \mathbb{R}^n$ is the decision vector in the n-dimensional decision space. In addition, four parameters are defined in this step. These parameters and the corresponding meanings are listed in Table 1.

Step 2: initialization

In this step, the HM will be initialized using Eq. (2):

$$\mathrm{HM}=\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{\mathrm{HMS}} \end{bmatrix}=\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{\mathrm{HMS},1} & x_{\mathrm{HMS},2} & \cdots & x_{\mathrm{HMS},n} \end{bmatrix}, \tag{2}$$

Table 1 Four main parameters and the corresponding meanings in harmony search (HS)

Parameter	Meaning
Harmony memory size (HMS)	The number of harmony solution vectors
Harmony memory consideration rate (HMCR)	The probability of taking a harmony from HM
Pitch adjusting rate (PAR)	The probability of fine-tuning a harmony taken from HM
Bandwidth (bw)	The magnitude of a local disturbance

where X_i is the i^{th} harmony in HM, and $x_{i,j}$ represents the j-dimensional vector (j=1, 2, ⋯, n) of the i^{th} harmony vector (i=1, 2, ⋯, HMS).

HM is a matrix for storing HMS solutions. In HM, each vector can be considered as a solution, and each harmony vector is composed of n variables. HMS harmony variables should be generated to form the initial HM within the scope of the feasible solution space through the principle of random generation or under certain rules.

Step 3: improvisation

In this step, new solutions will be created to ensure that the current solution will approach the potential optimal solution step by step by operating the following sub-steps:

Step 3.1: memory consideration and random selection. Generate a random number $r_1 \in [0,1]$. If $r_1 \le$ HMCR, then a harmony solution vector will be randomly selected from HM. Otherwise, if $r_1 >$HMCR, a new harmony solution vector will be randomly generated and move to step 4.

Step 3.2: pitch disturbance adjusting. Generate a random number $r_2 \in [0,1]$. If $r_2 \le$PAR, a harmony solution vector will be randomly selected from HM to implement the fine-tuning of the perturbation based on the value of bandwidth (bw) for generating a new harmony solution vector. Otherwise, no modification is made.

The pitch disturbance adjusting formula is shown in Eq. (3):

$$X_{\mathrm{new}}^t=\begin{cases} X_i^t \pm \mathrm{rand}(\,)\cdot \mathrm{bw}, & r_2 \le \mathrm{PAR}, \\ X_i^t, & r_2 > \mathrm{PAR}. \end{cases} \tag{3}$$

Step 4: update

In this step, the objective function of each harmony calculation is used as the criterion to decide the

further operation of the newly obtained harmony. If the new one is better than the worst one ($f(X_{\text{new}})< f(X_{\text{worst}})$) in HM, the new one will be used, instead of the worst one in HM. Otherwise, the new one will be discarded.

Step 5: adjustment

In this step, an evaluation will be executed to check whether it fulfills the termination condition. If the termination condition of the maximum iteration is fulfilled, the algorithm will be automatically terminated and the result will be generated. Otherwise, we move to step 3 for implementing one more iteration.

2.2 Standard CS algorithm

Yang and Deb (2009) proposed CS to simulate cuckoos' breeding behavior of laying eggs in other birds' nests.

The core operation of CS is to randomly simulate the female cuckoo laying her eggs in the host's nest. The female cuckoo employs an aggressive strategy of brood parasitism and then lets the host bird hatch her chicks. If the host bird recognizes an alien egg in the nest, it either rejects the egg or abandons the nest and builds a new nest somewhere else to raise another brood. Based on that, the implementations of the standard CS can be classified into the following five main steps:

Step 1: set parameters (including population size, search domain, dimension, and a maximum number of iterations), initialize the position of the bird's nest randomly, and define the objective function.

Step 2: calculate and compare the fitness value of each bird's nest position to obtain the current optimal fitness value.

Step 3: update the population position by calculating the fitness value using the Levy flight. The function of the fitness value is used to compare with the position of the bird's nest of the previous generation, update the position with a better fitness value, and keep it to the next generation.

The Levy flight is defined as follows:

$$x_i^{t+1} = x_i^t + \alpha \otimes \text{Levy}(\lambda), i \in [1, N], \tag{4}$$

where x_i^{t+1} and x_i^t represent the positions of the i^{th} individual in the $(t+1)^{\text{th}}$ and t^{th} generations respectively, α is defined as the step size scaling factor, $\otimes$ means entrywise multiplication, and Levy(λ) is a Levy distribution function which describes the step length of a random walk. The expression of Levy(λ) is shown in Eq. (5):

$$\text{Levy}(\lambda) \sim u = t^{-\lambda}, 1<\lambda\leq 3. \tag{5}$$

Step 4: generate a uniformly distributed random number $r\in[0,1]$ and compare it with the discovery probability p_a. If $r>p_a$, the bird's nest position x_i^{t+1} is randomly updated, and vice versa. Then the position of the newly obtained nest is compared with that of the previous generation, and the global optimal position pb_i^* will be selected.

Step 5: check whether $f(\text{pb}_i^*)$ fulfills the termination condition. If yes, pb_i^* is outputted as the global optimal solution gb; otherwise, move to step 3 for executing a new iteration.

3 Proposed HS-CS algorithm

Due to the drawbacks of falling into a local extremum in the improvisation stage, the standard HS is difficult to jump out of the current state, which will further affect the overall convergence performance. By combining the CS operator with the tuning pitch adjustment operator, an improved HS-CS algorithm is proposed to overcome this shortage. The modification strategies can be summarized based on the following aspects:

1. Because the Levy distribution is a heavy-tailed distribution, and its tail is wider than that of a Gaussian distribution, the algorithm which uses the Levy flight mechanism will have a stronger global search ability and robustness disturbance while a large step size is generated occasionally, or a small step size is generated by rotating 90° suddenly in the search process. Therefore, the Levy flight is selected to update the population for enhancing the global search ability of the HS-CS algorithm.

2. Three main strategies are employed to strengthen the ergodicity and the search ability of HS; these are to select the harmony solution vector randomly by the constructed inertia weight operator in HM with the probability of HMCR, to select the new harmony solution vector randomly in the solution

space with the probability of 1−HMCR, and to update the harmony solution vector using the Levy flight strategy in the CS operator.

3. The CS operator is used to find potential individuals from HM as candidate individuals, expand the population density, and improve the efficiency of HS for avoiding easy-to-premature convergence and falling into a stagnant state in the final stage of the standard HS.

3.1 Implementation steps and the corresponding flowchart of the HS-CS algorithm

Based on the above modifications, the proposed hybrid HS-CS algorithm can be separated into the following seven steps:

Step 1: set the related parameters of the proposed HS-CS algorithm, including constraints, the maximum number of iterations $T_{\max}$, current iteration number t, and search space $[\mathrm{LB}, \mathrm{UB}]$.

Step 2: initialize HM. Randomly generate the harmony solution vectors of the HMS group within the scope of the search space and create the harmony solution vectors by Eq. (6):

$$\mathrm{HM}_{i,j} = \mathrm{LB}_i + (\mathrm{UB}_i - \mathrm{LB}_i)\cdot \mathrm{rand}(), \quad i \in [1, \mathrm{HMS}], j \in [1, n], \tag{6}$$

where LB and UB are the lower and upper limits of $\mathrm{HM}_{i,j}$ respectively, and $\mathrm{rand}() \in [0, 1]$ is a random number.

Step 3: improvisation

Step 3.1: generate a random number $r_1 \in [0, 1]$. If $r_1 > \mathrm{HMCR}$, move to step 3.2; otherwise, the latest solution can be obtained by Eq. (7) through selecting the values in HM:

$$X_{\mathrm{new}}^{t+1}(i) = \mathrm{HM}_i^{\mathrm{worst},t} + \omega\cdot(\mathrm{HM}_i^{\mathrm{best},t} - \mathrm{HM}_i^{\mathrm{worst},t}), \tag{7}$$

where $\mathrm{HM}_i^{\mathrm{best},t}$ and $\mathrm{HM}_i^{\mathrm{worst},t}$ are the best and worst values of the i^{th} HM in the t^{th} generation respectively, $X_{\mathrm{new}}^{t+1}(i)$ represents the i^{th} new solution in the $(t+1)^{\mathrm{th}}$ generation, and ω is the inertia weight which is related to the solution in HM. Moreover, the current optimal solution in HM is better, and the corresponding value of ω is greater. Here, ω can be calculated using Eq. (8):

$$\omega = \begin{cases} (\omega_2 - \omega_1)/(2\omega_2 - \omega_1), & \text{if } 2\omega_2 - \omega_1 \neq 0, \\ \mathrm{rand}(0, 1), & \text{otherwise}, \end{cases} \tag{8}$$

where $\omega_1 = \min(\mathrm{HM}_i^{\mathrm{best},t}, \mathrm{HM}_i^{\mathrm{worst},t})$ and $\omega_2 = \max(\mathrm{HM}_i^{\mathrm{best},t}, \mathrm{HM}_i^{\mathrm{worst},t})$.

Step 3.2: generate a random number $r_2 \in [0, 1]$. If $r_2 > \mathrm{PAR}$, move to step 4. Otherwise, assign the best harmony in HM to the current new solution by Eq. (9) and move to step 5:

$$X_{\mathrm{new}}^{t+1}(i) = \mathrm{HM}_i^{\mathrm{best},t}. \tag{9}$$

Step 4: transmit the newly generated harmony obtained in step 3.1 to the CS operator for updating the harmony solution vector. The new harmony solution vector can be calculated by Eq. (10):

$$X_{\mathrm{new}}^{t+1} = \mathrm{HM}^{\mathrm{best},t} + \alpha(\mathrm{HM}^{\mathrm{best},t} - \mathrm{HM}^{\mathrm{worst},t}) \otimes \mathrm{Levy}(\lambda), \tag{10}$$

where $\alpha = |\alpha_0(2\mathrm{rand}() - 1)|$, $\alpha > 0$, and $\alpha_0 > 0$.

Step 5: determine whether the newly generated harmony solution vector X_{new} is better than the worst solution X_{worst} in HM. If yes, move to step 6; otherwise, move to step 7.

Step 6: update HM and the HM population by Eq. (11):

$$(X_{\mathrm{new}}^{t+1})' = \mathrm{HM}^{\mathrm{worst},t} + \alpha(\mathrm{HM}^{\mathrm{best},t} - \mathrm{HM}^{\mathrm{worst},t}) \otimes \mathrm{Levy}(\lambda). \tag{11}$$

Step 7: determine whether the termination condition is fulfilled or not. If yes, output the optimal solution; otherwise, dynamically adjust the probabilities of HMCR and PAR using Eqs. (12) and (13) respectively, and move to step 3.

$$\mathrm{HMCR}(t) = \mathrm{HMCR}_{\min} + (\mathrm{HMCR}_{\max} - \mathrm{HMCR}_{\min})\cdot t/T_{\max}, \tag{12}$$

$$\mathrm{PAR}(t) = \mathrm{PAR}_{\max} - (\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})t/T_{\max}, \tag{13}$$

where $\mathrm{HMCR}_{\max}$ and $\mathrm{HMCR}_{\min}$ represent the maximum and minimum values of HMCR respectively, and $\mathrm{PAR}_{\max}$ and $\mathrm{PAR}_{\min}$ represent the maximum and minimum values of PAR respectively.

3.2 Time complexity of the HS-CS algorithm

In heuristic algorithms, the calculation of the objective function takes most of the time of the whole algorithm. Assuming that the dimension of the objective

function f is Dim and that the population number is HMS, the time complexity of the HS-CS algorithm can be analyzed below:

In the initial population stage, the time complexity is O(HMS×Dim), and the time complexity of evaluating the objective function of the individual in the population is $O(f(\text{Dim}))$.

In the "pitch adjusting and selecting the best" stage of the improvisation process, the time complexity is O(HMS×Dim), and the "memory consideration" stage is known from Eq. (11), the corresponding time complexity of which is $O(\text{HMS}\times(\text{Dim}+O(\text{Levy})))$, where O(Levy) represents a random number subject to the Levy distribution and its magnitude of computational complexity is a constant order.

In the population update stage, the time complexity is $O(\text{HMS}\times(\text{Dim}+O(\text{Levy})))$.

According to the corresponding time complexity of each stage, the worst-case time complexity of the HS-CS algorithm can be calculated below, while the number of the current iterations is one:

$O(\text{HMS}\times\text{Dim})+O(\text{HMS}\times(\text{Dim}+f(\text{Dim})))+O(\text{HMS}\times(\text{Dim}+f(\text{Dim})))+O(\text{HMS}\times(\text{Dim}+f(\text{Dim})))\approx O(\text{HMS}\times(\text{Dim}+f(\text{Dim})))$.

Therefore, when the HS-CS algorithm reaches the maximum number of iterations $T_{\max}$, the time complexity is $O(T_{\max}\times\text{HMS}\times(\text{Dim}+f(\text{Dim})))$.

3.3 Convergence analysis of HS-CS

Convergence is one of the most important properties for ensuring the performance of a meta-heuristic algorithm. To provide the feasibility and robustness of the proposed HS-CS algorithm, three theorems are presented and proved to reveal HS-CS as a global convergence meta-heuristic algorithm based on the following methodologies. At first, a differential equation is used to protect that the limit exists in the iterative process of HS-CS. Then, stochastic functional analysis is used to prove the convergence in the iterative process of the solution of the HS-CS algorithm. At the same time, global convergence is finally proved using random functional analysis and the characteristics of the HS-CS algorithm. Moreover, to confirm it, the update mechanism of HS-CS (Eq. (10)) is considered as a linear dynamic system, and a second-order linear differential equation is used to analyze the change of the global optimal position.

Assuming that Eq. (10) is a one-dimensional second-order linear differential equation, Eq. (14) can be obtained by iterative recursive calculation:

$$X_{\text{new}}^{t+2}+\left(\alpha\otimes\text{Levy}(\lambda)\right)X_{\text{new}}^{t+1}-\left(1-\left(\alpha\otimes\text{Levy}(\lambda)\right)\right)\cdot X_{\text{new}}^{t}=2\left(\alpha\otimes\text{Levy}(\lambda)\right)\text{HM}^{\text{best}}. \tag{14}$$

Eq. (14) conforms to the expression of the second-order constant-coefficient homogeneous linear differential equation. Since the variable $\alpha\otimes\text{Levy}(\lambda)$ is a random number, a characteristic of Eq. (15) can be obtained to normalize the equation and to ensure $X_{\text{new}}^{t+2}=\beta^2$, $\alpha\otimes\text{Levy}(\lambda)=\varepsilon$:

$$\beta^2+\varepsilon\beta-(1-\varepsilon)=0. \tag{15}$$

The discriminant of Eq. (15) can be calculated as $\Delta=(\varepsilon-2)^2\geq 0$. To obtain the general solution of Eq. (15), two cases are discussed below.

Case 1 When $\Delta>0$, there are two different real roots β_1, β_2 ($\beta_1\neq\beta_2$), and the general solution of Eq. (15) can be obtained as $y=\text{e}^{\beta_2 t}\left[\frac{c_1}{\beta_1-\beta_2}\text{e}^{(\beta_1-\beta_2)t}+C_2\right]+y^*=C_1\text{e}^{\beta_1 t}+C_2\text{e}^{\beta_2 t}+y^*$ using the general solution of the first-order differential Eq. (16) below:

$$y=\text{e}^{\beta_2 t}\left[C_1\int\text{e}^{(\beta_1-\beta_2)t}\,\text{d}t+C_2\right], \tag{16}$$

where $C_1=\frac{c_1}{\beta_1-\beta_2}$, C_1 and C_2 are arbitrary constants, and the characteristic roots are $\beta_1=-1$ and $\beta_2=1-\varepsilon$. If and only if $\beta_2<0$, there exist limit y and $\lim\limits_{t\to\infty}C_1\text{e}^{\beta_1 t}+C_2\text{e}^{\beta_2 t}+y^*\to y^*$. Hence, $\Delta\geq 0$ could be gained while $\varepsilon\in(0,1)$. Furthermore, the sequence $\left\{X_{\text{new}}^{t+1}\right\}$ of Eq. (11) could converge while $t\to\infty$.

Case 2 When $\Delta=0$, there are two same real roots β_1 and β_2 ($\beta_1=\beta_2$). The general solution of Eq. (15) can be calculated as $y=C_1\text{e}^{\beta_1 t}+C_2\text{e}^{\beta_2 t}+y^*=\text{e}^{\beta_1 t}(C_1+C_2)+y^*$. If and only if $\varepsilon=2$, there exist limit $\Delta=0$ and $\beta_1\equiv-1$. Therefore, the requirement for the existence of the limit $\lim\limits_{t\to\infty}y\to y^*$ is satisfied under this condition.

To sum up, the existence of the limits of the constant-coefficient differential equations which are transformed from HS-CS is verified. According to the conditions of known limit existence, the convergence

of HS-CS in each process of iterative search can be proved step by step below.

The iterative process of solving HS-CS can be regarded as a mapping of the solution space. Due to this mapping relationship, the global and local search convergence properties of HS-CS may be analyzed using the stochastic functional analysis theory based on Solis and Wets (1981).

Theorem 1 Assuming that set A is a non-empty set, there exists $X=(X_1, X_2, \cdots, X_n)\in A$ for any $X_j\in[\mathrm{LB}_j, \mathrm{UB}_j]\subset A$. Notion d is a mapping from Cartesian products to non-negative real functions, $d: A\times A \to \mathbb{R}$. Then d can be expressed as

$$\begin{aligned} d(X_i, X_j) &= \left|\left(z-f(X_i)\right)-\left(z-f(X_j)\right)\right| \\ &= \left|f(X_i)-f(X_j)\right|, \end{aligned} \tag{17}$$

where $\forall X_i, X_j \in A$, z is an integer large enough.

Hence, (A, d) is a complete and separable metric space.

Proof First, it is necessary to prove that (A, d) is a metric space. According to the definition of metric space, A is a non-empty set and d is a real function on $A\times A$.

$\forall X_i, X_j, X_z\in A$, $d(X_i, X_j)$ is needed to fulfill the following requirements:

(1) $\forall X_i, X_j \in A$, there exists $d(X_i, X_j)=0$ if and only if $X_i=X_j$. Meanwhile, there is $X_i\neq X_j$ while $d(X_i, X_j)>0$. The positive-definite condition is met.

(2) $\forall X_i, X_j \in A$, there exists $d(X_i, X_j)=\left|\left(z-f(X_i)\right)-\left(z-f(X_j)\right)\right|=\left|\left(z-f(X_j)\right)-\left(z-f(X_i)\right)\right|=d(X_j, X_i)$. The symmetry condition is fulfilled.

(3) $\forall X_i, X_j, X_z \in A$, there exists

$$\begin{aligned} d(X_i, X_j) &= \left|\left(z-f(X_i)\right)-\left(z-f(X_j)\right)\right| \\ &= \left|\left(\left(z-f(X_i)\right)-\left(z-f(X_z)\right)\right)\right. \\ &\quad \left.+\left(\left(z-f(X_z)\right)-\left(z-f(X_j)\right)\right)\right| \\ &\le \left|\left(z-f(X_i)\right)-\left(z-f(X_z)\right)\right| \\ &\quad +\left|\left(z-f(X_z)\right)-\left(z-f(X_j)\right)\right| \\ &= d(X_i, X_z)+d(X_z, X_j). \end{aligned}$$

The triangle inequality conditions are also fulfilled.

When $d(X_i, X_j)$ meets the three mentioned conditions, it is concluded that (A, d) is a metric space.

Next is to prove that (A, d) is a complete metric space. Assume that A is a finite state space. Then each individual and the number of states in A are finite. According to the characteristics of greedy algorithm optimization and boundary constraints in the HS-CS iteration process, the position of the population individual in the current generation is relatively optimal compared to the position of the population individual in the previous generation, and the position of each individual cannot exceed the boundary in each iteration operation. The optimal individual in the current population is represented by the Cauchy sequence $\{X_n\}$ using the greedy algorithm and the property of boundary constraints. Furthermore, suppose that the point sequence $\{X_n\}$ is any Cauchy point column in A. Then $\forall\eta>0$, $\exists N\in\mathbb{N}$ ($\mathbb{N}$ is the set of natural numbers), such that there is $d(X_n, X_m)<\eta$ while $n, m>N$. Let $f(X_n)$ and $f(X_m)$ be denoted by a_n and a_m, respectively. Then, $\{a_n\}$ is a Cauchy point column in $\mathbb{R}$ and is convergent, marked as X: $\lim\limits_{n\to\infty} a_n=\hat{X}\in\mathbb{R}$; that is, $\lim\limits_{n\to\infty} f(X_n)=\hat{X}$.

$\forall\hat{X}$, there is $X_{j_0}\in A$ to fulfill $f(X_{j_0})=\hat{X}$. According to the population global search iterative formula

$$\begin{aligned} &\forall X_0\in\mathbb{R}, X_{\mathrm{new}}^{t+1}=\mathrm{HM}^{\mathrm{best},t}+(\alpha\otimes\mathrm{Levy}(\lambda)) \\ &\quad \cdot\left(\mathrm{HM}^{\mathrm{best},t}-\mathrm{HM}^{\mathrm{worst},t}\right), \\ &\quad 0<\alpha\otimes\mathrm{Levy}(\lambda)<1, t=0, 1, \cdots \end{aligned}$$

and supposing that there is an operator T^t in $\mathbb{R}$ and $T^tX=\mathrm{HM}^{\mathrm{best}}+(\alpha\otimes\mathrm{Levy}(\lambda))\cdot(\mathrm{HM}^{\mathrm{best}}-\mathrm{HM}^{\mathrm{worst}})$, the converted population global search iterative formula can be demonstrated as

$$\forall X_0\in\mathbb{R}, X^{t+1}=T^tX^t, t=0, 1, \cdots. \tag{18}$$

With the iterative progress of HS-CS, the global search efficiency of the population gradually changes, i.e., from strong search to a new strategy (the global search ability is weakened while the local optimization ability is enhanced). The value of the variable $\alpha\otimes\mathrm{Levy}(\lambda)$ will also tend to decrease when the global search ability is weakened in the later iterations and the value of $\alpha\otimes\mathrm{Levy}(\lambda)$ is close to 0. It indicates that $T^tX\to$

$\mathrm{HM}^{\mathrm{best}}$ ($t\to\infty$). In addition, according to the properties of the Cauchy point sequence, the position of the current population is composed of n-dimensional decision variables $X_j^1=\left(X_{j_1}^1, X_{j_2}^1, \cdots, X_{j_n}^1\right)$ in HS-CS, 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 $X_j=\left(X_j^1, X_j^2, \cdots, X_j^n\right)$ that conforms to the Cauchy point column property could be obtained, and all iteratively computed positions are within the given boundary constraints $[\mathrm{LB}_j, \mathrm{UB}_j]\subset A$. Using the global search updating strategy (Eq. (10)) and the boundary constraints, $X_{j_0}^{t+1}$ can be represented as

$$X_{j_0}^{t+1}=\begin{cases}\mathrm{UB}_j, \text{ if } \left(T^tX^t\right)_{j_0}>\mathrm{UB}_j,\\ \mathrm{LB}_j, \text{ if } \left(T^tX^t\right)_{j_0}<\mathrm{LB}_j,\\ \mathrm{HM}^{\mathrm{best},t}+\alpha\left(\mathrm{HM}^{\mathrm{best},t}-\mathrm{HM}^{\mathrm{worst},t}\right)\otimes\mathrm{Levy}(\lambda),\\ \quad \text{otherwise,}\end{cases}$$

and $X_{j_0}^{t+1}, X_{j_0}^t\in[\mathrm{LB}_j, \mathrm{UB}_j]\subset A$. $f\left(X_{j_0}\right)=\hat{X}\in\mathbb{R}$ is fulfilled; that is, $\lim\limits_{n\to\infty} f\left(X_n\right)=f\left(X_{j_0}\right)$. It also indicates that Eq. (18) 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 is easy to conclude that V is a countable set and $V\subset A$. For any $X_n\in A$, because $B_1=\bigcup_{i=1}^{N_1}B\left(X_i^{(i)},1\right)\supset A$, $X_{i_1}^{(i)}$ 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 $X_{i_2}^{(i)}$ to ensure $X_n\in B\left(X_{i_2}^{(i)}, 1/2\right)$. Using 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. V is a dense subset in A. It also proves that (A, d) is a separable metric space. To sum up, (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 the following condition is satisfied:

$$\rho\left(\left\{\delta: d\left(\varphi\left(\delta, X_i\right), \varphi\left(\delta, X_{i+1}\right)\right)\le H(\delta)d\left(X_i, X_{i+1}\right)\right\}\right)=1,$$
$$X_i, X_{i+1}\in A. \tag{19}$$

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

Proof 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

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

Furthermore,

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

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

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

According to Theorems 1 and 2, 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 (1981).

The following two assumptions are given at first:

Assumption 1 Assuming that if conditions $f\left(D\left(z, \xi\right)\right)\le f(z)$ and $\xi\in S\left(\mathbb{R}^n, B, \mu_k\right)$ hold, then $f\left(D\left(z, \xi\right)\right)\le f(\xi)$, where D is the function that generates the solution of the problem, ξ is a random variable generated from the probability space $\left(\mathbb{R}^n, B, \mu_k\right)$, $S\subseteq\mathbb{R}^n$ represents the constraint space of the problem, μ_k is the probability measure on B, and B is the σ domain of $\mathbb{R}^n$ 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, that its solution space S is a measurable subset, and that $\left\{\mathrm{HM}^{\mathrm{best},t}\right\}_{t=0}^{\infty}$ is the solution sequence generated by the algorithm, then $\lim_{k\to\infty}P\left[\mathrm{HM}^{\mathrm{best},t}\in R_\varepsilon\right]=1$ is established, where $P\left[\mathrm{HM}^{\mathrm{best},t}\in R_\varepsilon\right]$ is the probability of $\mathrm{HM}^{\mathrm{best},t}\in R_\varepsilon$ and R_ε is the set of global optimal points.

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

$$D\left(\mathrm{HM}^{\mathrm{best},t},X_i^t\right)=\begin{cases}\mathrm{HM}^{\mathrm{best},t}, \text{ if } f\left(\mathrm{HM}^{\mathrm{best},t}\right)\le f\left(X_i^t\right),\\ X_i^t, \qquad \text{otherwise.}\end{cases}$$

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

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 (Eqs. (7) and (10)), all individuals in the population will converge to the optimal position, but it may not be the global optimal position.

Taking Eq. (10) as an example, because of $X_i\to\mathrm{HM}^{\mathrm{worst}}/\left(1+\alpha\otimes\mathrm{Levy}(\lambda)\right)$, $\lim_{t\to\infty}M_i^t=0$ 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, $\exists t_1$ such that $v\left[\bigcup_{i=1}^{N}M_i^t\cap S\right]<v(S)$ while $t>t_1$. That is, Eq. (10) does not satisfy Assumption 2.

Therefore, under the condition that Assumption 2 is not satisfied, 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, $\exists t'$ such that $\bigcup_{i=1}^{N}M_i^t\cup\beta\supseteq S$ while $t>t'$.

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

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 using 12 benchmark functions, and the other is the weighted fuzzy production rule extraction by the HS-CS algorithm and BPNN over IRIS.

These experiments are run under the environment of Intel® Core™ I5-10200H CPU @2.40 GHz, 16 GB memory, and the Windows 10 operating system. The programming language is Python 3.7.

4.1 Function optimization

In this experiment, 12 classical benchmark functions, including Sphere, LevyN13, Alpine, Rastrigin, Griewank, Ackley, Step, Schwefel 2.22, Bohachevsky, Quartic, Rosenbrock, and Schwefel (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: unimodal functions and multimodal functions. The details of these selected benchmark functions are listed in Table 2.

4.2 Parameter design and experimental results

In this experiment, the HS-CS algorithm is compared with three other HS algorithms and three CS algorithms. The three other HS algorithms are global optimal adaptive harmony search algorithm (AGOHS) (Li et al., 2020), improved differential harmony search algorithm (IDHS) (Wang L et al., 2019), and harmony search with differential mutation based on pitch adjustment (HSDM) (Qin AK and Forbes, 2011). The three CS algorithms are novel enhanced cuckoo search algorithm (ECS) (Kamoona and Patra, 2019), modified cuckoo search (MCS) (Ong and Zainuddin, 2019), and cuckoo search (CS) (Yang and Deb, 2009). In addition,

Table 2 Twelve benchmark functions

Name	Function	Dim	Search space	f^*	Class
Sphere	$F_1 = \sum_{i=1}^{D} x_i^2$	D	[−5.12, 5.12]	0	Unimodal
LevyN13	$F_2 = \sin^2(3\pi x_1) + (x_1 - 1)^2[1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + \sin^2(2\pi x_2)]$	D	[−10, 10]	0	Multimodal
Alpine	$F_3 = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	D	[−10, 10]	0	Multimodal
Rastrigin	$F_4 = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	D	[−600, 600]	0	Multimodal
Griewank	$F_5 = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	D	[−600, 600]	0	Multimodal
Ackley	$F_6 = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left[\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right] + 20 + \mathrm{e}$	D	[−32, 32]	0	Multimodal
Step	$F_7 = \sum_{i=1}^{D} (x_i + 0.5)^2$	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	$F_9 = \sum_{i=1}^{D-1} [x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7]$	D	[−100, 100]	0	Unimodal
Quartic	$F_{10} = \sum_{i=1}^{D} ix_i^4 + \mathrm{rand}()$	D	[−100, 100]	0	Unimodal
Rosenbrock	$F_{11} = \sum_{i=1}^{D} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	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	Multimodal

f^*: theoretical global optimal solution of the current function; Dim: dimension of the test function; D: value of the variable dimension. D is set as 10, 30, or 50. 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

to ensure the fairness of all algorithms in the experiment, the population size and the maximum number of iterations are set to 100 and 5000, respectively. The related parameter setting of these algorithms is shown in Table 3.

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

Figs. 1 and 2 show part of the experimental results. Furthermore, the entire experimental results and corresponding analysis are given in the supplementary materials.

As a unimodal function, the Step function is usually used to test the convergence accuracy and speed of the algorithm. The HS-CS algorithm is integrated with Levy flight to broaden the search scope of the population, enhance the position update strategy, and improve the global exploration ability. As shown in Fig. 1, no matter whether the Step function is with dimension 30 or 50, the proposed algorithm always exhibits the best convergence curve. When the dimension is 30, although the convergence accuracies of AGOHS and ECS are similar to that of the proposed algorithm, they require more iterations. When the dimensionality of the test function increases, AGOHS requires more iterations to achieve the same convergence accuracy as the HS-CS algorithm. In the set number of iterations, ECS 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 Griewank function is usually used to test the ability of the algorithm to jump out of local optima. As shown in Fig. 2, when the dimension of the test function is low, since the search mechanism of Levy flight of CS is adopted by ECS and HS-CS, the convergence efficiencies of these algorithms are similar. AGOHS requires more iterations to obtain similar convergence accuracy to the HS-CS algorithm. As the dimension of the test function increases, the convergence efficiencies of AGOHS and ECS decrease significantly. In the search process, the HS-CS algorithm indicates the direction of "pitch adjusting and selecting the best." The Levy flight in the CS operator is adopted to find candidate individuals when updating HM, which increases the number of alternative solutions and strengthens the disturbance to avoid falling into stagnation prematurely in the search process. Thus, the HS-CS algorithm still maintains a good convergence efficiency.

Further analysis of the experiment using the Wilcoxon rank-sum test, the mean value (MEAN), and the standard deviation (SD) is given in detail in the

Table 3 Parameter settings of the examined algorithms*

Algorithm	Parameter(s)
HS-CS	$HMCR_{min}$=0.8, $HMCR_{max}$=0.9, PAR_{min}=0.1, PAR_{max}=0.9, α_0=0.01
AGOHS	$HMCR_{min}$=0.8, $HMCR_{max}$=0.9, PAR_{min}=0.1, PAR_{max}=0.9, F=N(0.5, 0.3)
IDHS	$HMCR_{min}$=0.8, $HMCR_{max}$=0.9, PAR_{min}=0.1, PAR_{max}=0.9
HSDM	HMCR=0.98
ECS	α=0.01, P_a=0.25, λ=1.5
MCS	α=0.01, P_a=0.25, λ=1.5
CS	α=0.01, P_a=0.25, λ=1.5

* Population: 100; maximum number of iterations: 5000

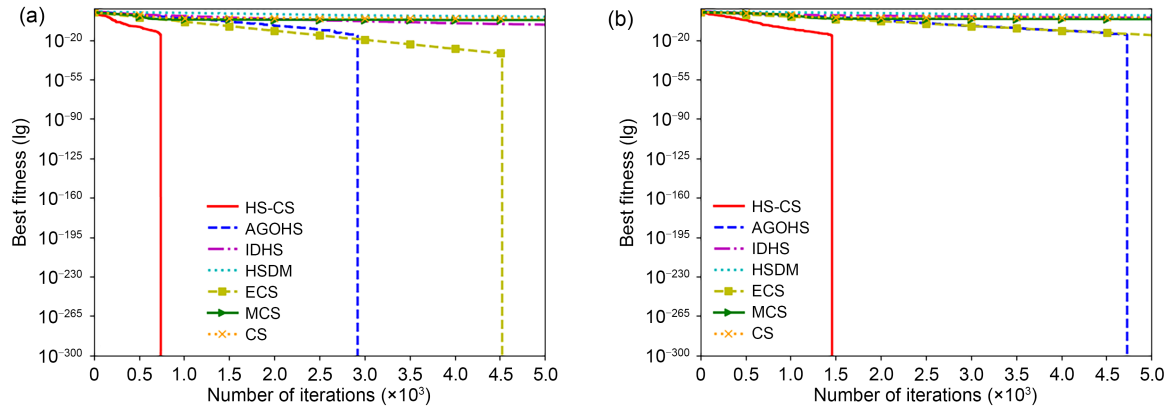

Fig. 1 Optimizing the convergence curve of the Step function with dimension 30 (a) and 50 (b)

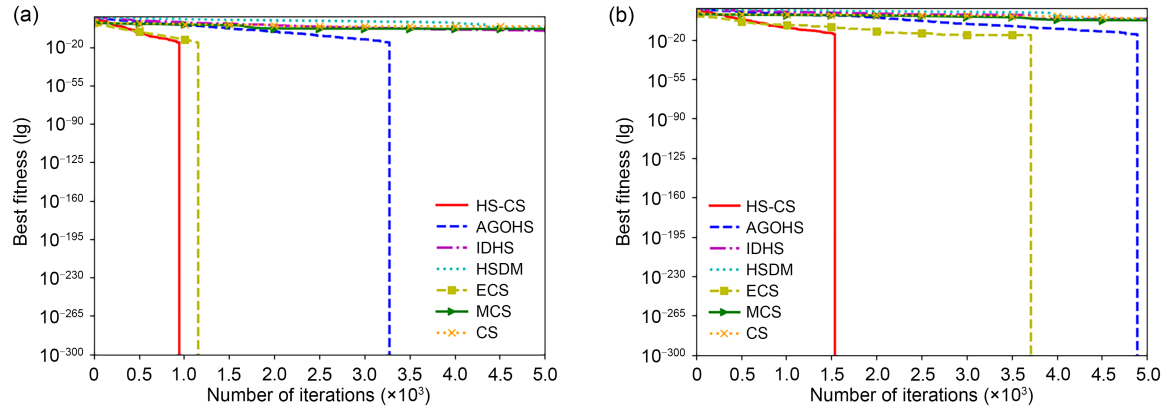

Fig. 2 Optimizing the convergence curve of the Griewank function with dimension 30 (a) and 50 (b)

supplementary materials. Combining the analysis above and that in the supplementary materials, the HS-CS algorithm exhibits good performance in dealing with high-dimensional optimization problems, proving that it has good convergence efficiency and self-adaptability.

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 using HS-CS and BPNN is demonstrated here. The entire extraction process can be separated into the following main modules: (1) data fuzzification, (2) BPNN generation, and (3) BPNN framework optimization using the HS-CS algorithm, the importance index matrix obtained from the trained BPNN, and the weighted fuzzy production rule 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, thus a total of 150 samples. Moreover, each sample contains four attributes, i.e., sepal length (SL), sepal width (SW), petal length (PL), and petal width (PW). The measurement unit of each attribute is cm.

5.2 Implementation analysis of weighted fuzzy production rule extraction

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

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

Second, 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 three classification results are used to build 12 input nodes of the input layer, three output nodes of the output layer, one hidden layer, and four hidden nodes generated by experience. The sigmoid function is used to obtain BPNN as an activation function. Meanwhile, the fuzzy dataset is randomly divided into two parts, in which 80% of the examples are used as the training set and the remainder as the test set.

Third, HS-CS is employed to optimize the related parameters and the objective function of the obtained BPNN. The parameters of HS-CS are set as follows: HMS=100, $\mathrm{HMCR_{max}}$=0.9, $\mathrm{HMCR_{min}}$=0.8, $\mathrm{PAR_{min}}$=0.1, $\mathrm{PAR_{max}}$=0.9, α=0.01, the dimension of each harmony D=12 (reflecting 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 BPNN training 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.

Fourth, the corresponding importance index matrix $W_{n\times m}=\mathrm{OW}_{n\times k}\cdot\mathrm{IW}_{k\times m}$ can be gained while the BPNN optimization and training are completed, among which $\mathrm{OW}_{n\times k}$ represents the connection weight matrix between the hidden layer and output layer in BPNN, k represents the number of nodes in the hidden layer, n represents the number of output nodes in the output layer, $\mathrm{IW}_{k\times m}$ 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 HS-CS is given in Eq. (20).

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

In the importance index matrix, the classification results corresponding to each row may follow the same rule or multiple rules. If the attribute value element of the unified attribute is not 0, multiple rules will be generated; that is, there is no OR in the preceding one. If the attribute value element is negative, the corresponding NOT antecedent is generated.

$$W_{3\times12}=\begin{pmatrix}0 & -0.72 & -10.92 & 0 & 0 & 0.44 & 21.81 & -1.25 & -1.36 & 11.49 & -20.79 & -1.4\\ 0 & -6.53 & 5.95 & 0 & 0 & 4 & -12.89 & 22.29 & -22.77 & -0.66 & 29.8 & -12.79\\ 0 & 5.75 & 9.82 & 0 & 0 & -3.52 & -16.26 & -14.82 & 34.18 & -4.11 & -16.48 & 11.26\end{pmatrix}. \tag{20}$$

Based on the above requirements, the weighted fuzzy production rules can be automatically extracted from the importance index matrix. In this paper, we list only the first row of the matrix, producing 18 classification rules as Iris-setosa (Table 4). The entire generated local weighted fuzzy production rules could be found in the supplementary materials.

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

The extracted rules are classified and verified, and then applied to the training set and test set for comparison. The related data are listed in Table 5.

Table 5 indicates that the proposed HS-CS algorithm has higher accuracy than AGOHS and HS while optimizing BPNN and extracting the weighted fuzzy generation rules from BPNN. Meanwhile, the accuracy obtained using HS-CS on the test 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 35.51, 40.15, and 370.23 s, respectively. Although the execution time of HS is less than that of HS-CS, the accuracy of HS in data classification is lower 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

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

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 optimized by the proposed HS-CS algorithm have obvious advantages compared with AGOHS and the standard HS algorithm. It is also illustrated that the HS-CS algorithm plays a certain role in the training process of the optimized BPNN.

To sum up, in the process of weighted fuzzy rule extraction using the HS-CS algorithm, the BPNN optimized by HS-CS extracts the weighted fuzzy production rules from the IRIS dataset in a relatively short

Table 4 Classification rules as Iris-setosa

No.	IF
1	SL is NOT MED [0.72], SW is LGR [0.44], PL is SM [21.81], and PW is SM [11.49]
2	SL is NOT MED [0.72], SW is LGR [0.44], PL is SM [21.81], and PW is NOT MED [20.79]
3	SL is NOT MED [0.72], SW is LGR [0.44], PL is SM [21.81], and PW is NOT LGR [1.4]
4	SL is NOT MED [0.72], SW is LGR [0.44], PL is NOT MED [1.25], and PW is SM [11.49]
5	SL is NOT MED [0.72], SW is LGR [0.44], PL is NOT MED [1.25], and PW is NOT MED [20.79]
6	SL is NOT MED [0.72], SW is LGR [0.44], PL is NOT MED [1.25], and PW is NOT LGR [1.4]
7	SL is NOT MED [0.72], SW is LGR [0.44], PL is NOT LGR [13.36], and PW is SM [11.49]
8	SL is NOT MED [0.72], SW is LGR [0.44], PL is NOT LGR [13.36], and PW is NOT MED [20.79]
9	SL is NOT MED [0.72], SW is LGR [0.44], PL is NOT LGR [13.36], and PW is NOT LGR [1.4]
10	SL is NOT LGR [10.92], SW is LGR [0.44], PL is SM [21.81], and PW is SM [11.49]
11	SL is NOT LGR [10.92], SW is LGR [0.44], PL is SM [21.81], and PW is NOT MED [20.79]
12	SL is NOT LGR [10.92], SW is LGR [0.44], PL is SM [21.81], and PW is NOT LGR [1.4]
13	SL is NOT LGR [10.92], SW is LGR [0.44], PL is NOT MED [1.25], and PW is SM [11.49]
14	SL is NOT LGR [10.92], SW is LGR [0.44], PL is NOT MED [1.25], and PW is NOT MED [20.79]
15	SL is NOT LGR [10.92], SW is LGR [0.44], PL is NOT MED [1.25], and PW is NOT LGR [1.4]
16	SL is NOT LGR [10.92], SW is LGR [0.44], PL is NOT LGR [13.36], and PW is SM [11.49]
17	SL is NOT LGR [10.92], SW is LGR [0.44], PL is NOT LGR [13.36], and PW is NOT MED [20.79]
18	SL is NOT LGR [10.92], SW is LGR [0.44], PL is NOT LGR [13.36], and PW is NOT LGR [1.4]

SL: sepal length; SW: sepal width; PL: petal length; PW: petal width; LGR: large; MED: medium; SM: small

Table 5 Comparison of IRIS classification accuracy results

Method	Accuracy (%)	
	Training set	Test set
BPNN trained by HS-CS	96.45 (95.54)	97.37 (97.37)
BPNN trained by AGOHS	96.32 (66.07)	97.36 (68.42)
BPNN trained by HS	96.39 (65.72)	97.36 (68.42)

Values in brackets indicate the corresponding accuracies of the obtained weighted fuzzy production rules

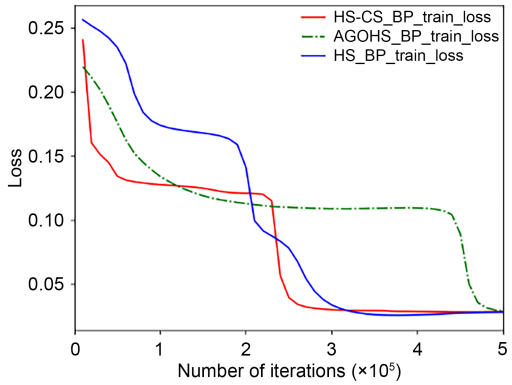

Fig. 3 Convergence curve of the loss function of BPNN with different HS algorithms (BPNN: back propagation neural network; HS: harmony search)

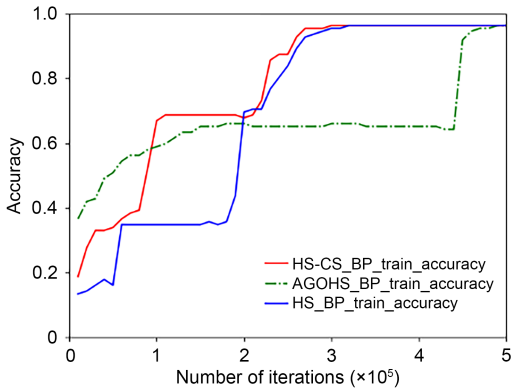

Fig. 4 Convergence precision curve of the loss function of BPNN with different HS algorithms (BPNN: back propagation neural network; HS: harmony search)

execution time and achieves a better classification effect. The main functions of HS-CS are to improve the convergence speed of BPNN and reduce the loss during training.

6 Conclusions

In this paper, a modified HS-CS algorithm is presented to solve the problems of the standard HS which is easily trapped into local optima and is weak in global search ability. In particular, the HS-CS algorithm uses 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 local optima. On the other hand, "pitch adjusting and selecting the best" in the improvisation stage and the inertial weight operator are constructed and the degree of the internal individuals in HM is 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. First, the existence of the limit of the proposed algorithm is proved by differential equations, and the global optimal algorithm is verified by random functional analysis and random search theory. Second, HS-CS and six other algorithms are used to settle the function optimization issue of the 12 selected classical benchmark functions in different high dimensions. The results show 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, 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 abilities of BPNN effectively.

This research is intended mainly to optimize the solution of a single objective problem for 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 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 in coordinated formation.

Contributors

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

Compliance with ethics guidelines

Shaoqiang YE, Kaiqing ZHOU, Azlan Mohd ZAIN, Fangling WANG, and Yusliza YUSOFF declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abbasi M, Abbasi E, Mohammadi-Ivatloo B, 2021. Single and multi-objective optimal power flow using a new differential-based harmony search algorithm. *J Amb Intell Human Comput*, 12(1):851-871. https://doi.org/10.1007/s12652-020-02089-6

Al-Shaikh A, Mahafzah BA, Alshraideh M, 2023. Hybrid harmony search algorithm for social network contact tracing of COVID-19. *Soft Comput*, 27:3343-3365. https://doi.org/10.1007/s00500-021-05948-2

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

Geem ZW, Kim JH, Loganathan GV, 2001. A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2): 60-68. https://doi.org/10.1177/003754970107600201

Gong JH, Zhang ZQ, Liu JQ, et al., 2021. Hybrid algorithm of harmony search for dynamic parallel row ordering problem. *J Manuf Syst*, 58:159-175. https://doi.org/10.1016/j.jmsy.2020.11.014

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

Jagatheesan K, Anand B, Samanta S, et al., 2019. Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm. *IEEE/CAA J Autom Sin*, 6(2): 503-515. https://doi.org/10.1109/JAS.2017.7510436

Kamoona AM, Patra JC, 2019. A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images. *Appl Soft Comput*, 85:105749. https://doi.org/10.1016/j.asoc.2019.105749

Karaboga D, Gorkemli B, Ozturk C, et al., 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev*, 42(1):21-57. https://doi.org/10.1007/s10462-012-9328-0

Li HC, Zhou KQ, Mo LP, 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. *Knowl-Based Syst*, 89:228-249. https://doi.org/10.1016/j.knosys.2015.07.006

Mousavi SM, Abdullah S, Niaki STA, et al., 2021. An intelligent hybrid classification algorithm integrating fuzzy rule-based extraction and harmony search optimization: medical diagnosis applications. *Knowl-Based Syst*, 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. *Appl Soft Comput*, 80:374-386. https://doi.org/10.1016/j.asoc.2019.04.016

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

Qin AK, Forbes F, 2011. Harmony search with differential mutation based pitch adjustment. Proc 13th Annual Conf on Genetic and Evolutionary Computation, p.545-552. https://doi.org/10.1145/2001576.2001651

Qin F, Zain AM, Zhou KQ, 2022. Harmony search algorithm and related variants: a systematic review. *Swarm Evol Comput*, 74:101126. https://doi.org/10.1016/j.swevo.2022.101126

Shaffiei ZA, Abas ZA, Yunos NM, et al., 2019. Constrained self-adaptive harmony search algorithm with 2-opt swapping for driver scheduling problem of university shuttle bus. *Arab J Sci Eng*, 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 Comput*, 25(16):11053-11075. https://doi.org/10.1007/s00500-021-05841-y

Solis FJ, Wets RJB, 1981. Minimization by random search techniques. *Math Oper Res*, 6(1):19-30. https://doi.org/10.1287/moor.6.1.19

Tang J, Liu G, Pan QT, 2021. A review on representative swarm intelligence algorithms for solving optimization problems: applications and trends. *IEEE/CAA J Autom Sin*, 8(10):1627-1643. https://doi.org/10.1109/JAS.2021.1004129

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

Valaei MR, Behnamian J, 2017. Allocation and sequencing in 1-out-of-*N* heterogeneous cold-standby systems: multi-objective harmony search with dynamic parameters tuning. *Reliab Eng Syst Saf*, 157:78-86. https://doi.org/10.1016/j.ress.2016.08.022

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

Wang YR, Gao SC, Zhou MC, et al., 2021. A multi-layered gravitational search algorithm for function optimization and real-world problems. *IEEE/CAA J Autom Sin*, 8(1): 94-109. https://doi.org/10.1109/JAS.2020.1003462

Yang XS, Deb S, 2009. Cuckoo search via Lévy flights. World Congress on Nature & Biologically Inspired Computing, p.210-214. https://doi.org/10.1109/NABIC.2009.5393690

Ye SQ, Zhou KQ, Zhang CX, 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 ZY, Liu SX, Zhou MC, et al., 2021. Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem. *IEEE/CAA J Autom Sin*, 8(6):1199-1209. https://doi.org/10.1109/JAS.2020.1003539

Zhu QD, Tang XM, Elahi A, 2021. Application of the novel harmony search optimization algorithm for DBSCAN clustering. *Expert Syst Appl*, 178:115054. https://doi.org/10.1016/j.eswa.2021.115054

List of supplementary materials

1 Further discussion of the proposed HS-CS algorithm
2 Function optimization using HS-CS
3 Entire weighted fuzzy production rules extracted using HS-CS
Fig. S1 Flowchart of the HS-CS algorithm
Figs. S2–S13 Optimizing the convergence curves of F1–F12 in different dimensions
Table S1 Experimental statistics of mean and standard deviation
Tables S2–S4 Wilcoxon rank-sum test results of six algorithms referring to HS-CS in the 10-, 30-, and 50-dimensional cases
Table S5 Classification rules as Iris-versicolor
Table S6 Classification rules as Iris-virginica