Neural Computing and Applications (2022) 34:13599–13624
https://doi.org/10.1007/s00521-022-07146-z

ORIGINAL ARTICLE

Rules embedded harris hawks optimizer for large-scale optimization problems

Hussein Samma[1,3] · Ali Salem Bin Sama[2,4]

Received: 15 August 2021 / Accepted: 28 February 2022 / Published online: 31 March 2022

Abstract
Harris Hawks Optimizer (HHO) is a recent optimizer that was successfully applied for various real-world problems. However, working under large-scale problems requires an efficient exploration/exploitation balancing scheme that helps HHO to escape from possible local optima stagnation. To achieve this objective and boost the search efficiency of HHO, this study develops embedded rules used to make adaptive switching between exploration/exploitation based on search performances. These embedded rules were formulated based on several parameters such as population status, success rate, and the number of consumed search iterations. To verify the effectiveness of these embedded rules in improving HHO performances, a total of six standard high-dimensional functions ranging from 1000-D to 10,000-D and CEC'2010 large-scale benchmark were employed in this study. In addition, the proposed Rules Embedded Harris Hawks Optimizer (REHHO) applied for one real-world high dimensional wavelength selection problem. Conducted experiments showed that these embedded rules significantly improve HHO in terms of accuracy and convergence curve. In particular, REHHO was able to achieve superior performances against HHO in all conducted benchmark problems. Besides that, results showed that faster convergence was obtained from the embedded rules. Furthermore, REHHO was able to outperform several recent and state-of-the-art optimization algorithms.

Keywords Rule-based optimizer · Harris hawks · Large-scale optimization

1 Introduction

In the era of big data, a lot of real-world, large-scale optimization problems have been existed, such as multi-policy insurance investment planning [1], scheduling [2], and gene biomarker discovery [3]. Tackling these problems using metaheuristic algorithms is considered a difficult task. This is due to the growth in dimension space, i.e., ''*curse of dimensionality*'' [4]. To mitigate these difficulties, researchers suggested several ideas, such as splitting the dimensionality using a divide-and-conquer scheme [5], introducing dynamic balancing between exploration and exploitation [6], or using the concept of population clustering [7].

Recently, many optimizers were introduced in the literature, such as Harris Hawks Optimizer (HHO) [8], Fitness Dependent Optimizer (FDO) [9], Learner Performance-based Behavior (LPB) [10], Child Drawing Development Optimizer (CDDO) [11], and Donkey and Smuggler Optimizer (DSO) [12]. Among them, HHO was given a lot of attention. This is due to its simplicity and efficiency in dealing with various real-world problems such as image segmentation [13], features selection [14], tracking maximum power in solar systems [15], prediction of solar systems productivity [16], designing load frequency of renewable energy plan [17], forecasting of air pollution [18], and predicting food liking [19]. However,

✉ Hussein Samma
hussein.samma@utm.my

1 School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, 81310 UTM Johor, Malaysia

2 Computer and Information Science Department, College of Shari'a & Islamic Studies, Al-Imam Mohammad Ibn Saud Islamic University, Al-hassa, Kingdom of Saudi Arabia

3 Faculty of Computer and Information Technology, University of Shabwah, Shabwah, Republic of Yemen

4 Department of Geology Engineering, Faculty of Oil & Minerals, Aden University, Aden, Republic of Yemen

HHO lacks efficient exploration/exploitation balancing ability. This is because it uses timely depended on energy escape parameters which control the switching from exploration to exploitation mode [8]. As such, when HHO is trapped in local optima during the exploitation phase, it will be hard to escape and return to exploration mode. To overcome these drawbacks, researchers suggested various versions of HHO, which could be categorized as hybridized-based methods and modified-based methods.

The idea of hybridizing HHO with other metaheuristic algorithms was studied by many researchers [2–26]. Abd Elaziz et al. introduced a hybrid model that combines HHO with a moth-flame optimizer [20]. The main goal was to enhance the exploration ability of HHO by using moth-flame to generate the initial population. In addition, a chaos map was embedded for further enhancement. Results show a superior impact on the performances of the hybrid optimizer as compared with the standard HHO. However, the main challenge is related to the increase in complexity due to the large number of parameters that need to be tuned. Further hybrid work was given by ElSayed et al. [21]. Basically, their work integrates HHO with Sequential Quadratic Programming (SQP). SQP was used as a local search optimizer to refine the best solution found by HHO at run-time. Reported results on the problem of finding the best relay directions in power systems indicated better performances were reported from the proposed hybrid approach. A memetic-based HHO scheme was suggested by Li et al. [22]. The key idea of the proposed memetic scheme is to enhance the local search capability of HHO by embedding several elite evolutionary strategies. Conducted analysis in their study on scheduling problems showed further HHO improvements were achieved due to the incorporated local search strategies. The hybridization of Grasshopper Optimizer GO with HHO was discussed by Singh et al. [23]. Their model was applied for the problem of optimal placement of multiple optical network units. The outcomes of GO-HHO demonstrated the superiority of the hybrid model as compared with individual optimizers, i.e., HHO and GO. The idea of evolving multiple HHO populations with quantum particles was given by Ilker et.al [20]. Mainly, their proposed approach was designed to tackle dynamic optimization problems that encompass multiple local optima. Conducted experiment on CEC 2009 showed that multiple HHO populations produced better outcomes in terms of convergence rate and fitness value. The fusion of HHO with both sine–cosine and simulated annealing was discussed in [25] and [26], respectively. Both models were applied for the problem of features selection, and their analysis showed great improvements in tackling feature selection challenges. Very recent hybrid studies which integrate HHO with other metaheuristic optimization algorithms were discussed by Abba et al. [27], Ebrahim et al. [28], Bandyopadhyay et al. [29], Suresh et al. [30], and Mossa et al. [31]. In [27], the hybrid of PSO with HHO for renewable energy load demand forecasting was presented. The synergy of sine–cosine with HHO was discussed in [28] for optimizing the fuel cell−based electric power system. Bandyopadhyay et al. [29] presented the integration of simulated annealing with HHO for deep features selection of COVID-19 from CT-scan images. The hybrid of chaotic multi-verse optimizer with HHO was given in [30] for the problem of medical diagnosis. The issue of parameters estimation of proton exchange membrane fuel cell using a hybrid atom optimizer with HHO was investigated in [31]. Nevertheless, the main challenge of hybrid-based methods is related to the increase of fitness evaluation cost needed for each optimizer. In addition, hybridizing several optimizers raises the difficulties of simultaneously managing them at run-time [32].

A modified-based HHO methods were presented in many studies [33–38]. The idea of modifying HHO by embedding salp optimizer operations was adopted by Abdelaziz et al. [33]. The main aim of their study is to enhance the exploration capability of HHO. In their work, they split the population into two sub-populations, and one half has been evolved under salp operations, and the other half has been executed under the control of HHO operations. The modified model in [24] was applied for the multi-level image thresholding problem, and results showed that embedded salp operations enhanced HHO exploration performances. Similarly, enhancing HHO exploration ability by incorporating differential evolutionary operators was suggested by Wunnava et al. [34]. Their proposed approach was applied for the multi-level image thresholding problem. Nevertheless, incorporating additional operations into HHO raises the challenge of increasing model complexity, which will increase the cost of fitness computation needed for these additional operations. Additional work was proposed by Yousri et al. [35] for improving the effectiveness of HHO in performing the exploration phase. Particularly, they have embedded fractional calculus (FOC) memory which is used to control the movement velocity of HHO agents. As such, FOC helps in avoiding possible premature convergence. Conducted experiments clearly showed better performances were achieved from embedding FOC. Additional HHO modifications were propped by Chen et al. [36]. They incorporated both opposition technique and chaotic local search into HHO. Reported results indicated better HHO improvements due to the enhancement in HHO population diversity. Further modifications were presented by Li et al. [37] for enhancing HHO population quality. Particularly, Li et al. added horizontal and vertical crossover operations into HHO, and results indicated further HHO exploration enhancements. Finally, researchers in [38] suggested the

concept of information exchange to enhance HHO exploration ability. Very recent approaches were done by several researchers where they proposed many modification schemes to improve HHO. For instance, a multi-strategy approach was given by Li et al. [39]. The main idea of their approach is to incorporate different enhancement strategies namely opposition-based learning, logarithmic spiral, and a modified Rosenbrock local search. Other researchers suggested of embedding two different opposition-based schemes, namely selective, leading opposition, and the dynamic opposition technique. Enhancing HHO by implementing different random distribution functions which control the random movement of HHO agents was given by Akdag et al. [40]. Specifically, they have investigated seven types of random distribution functions, including chi-square, normal, exponential, Rayleigh, student's distribution, F-distribution, and lognormal. Reported results clearly showed further improvements were achieved, especially for engineering design problems. Another enhanced version of HHO was discussed by Houssein et al. [41]. The key concept of their enhanced approach is to incorporate genetic operators to enhance exploitation ability in the selection of chemical molecular descriptors problems. An additional recent modified approach was illustrated by Krishna et al. [42]. Basically, they focused on enhancing HHO search capability when dealing with constrained engineering design problems. As such, to boost the exploitation performances of HHO, they have added pattern search algorithm during the exploitation phase of HHO. A chaotic guided HHO algorithm was demonstrated by Singh et al. [43] for data clustering. They have implemented a logistic chaotic map which was executed in the exploration phase of HHO. Despite the slow convergence of the enhanced HHO algorithm in [43],but the results clearly showed an improvement in the achieved clustering performances.

Motivated by HHO popularity, simplicity, and efficiency, this study aims to further improve HHO performances when dealing with large-scale problems that encounter a lot of local optima points. It should be noted that previously mentioned studies mainly focused on enhancing HHO exploration by incorporating chaotic re-initialization schemes [43], embedding opposition-based schemes [36, 39], or using other search operations inside HHO [33, 33–35, 40]. Others suggested using an external local search algorithm with HHO to improve exploitation performance [39, 42]. Despite that, there is still room for improvements by utilizing HHO population status at runtime search progress. Knowing the population status will play a vital role in making the decision about the appropriate time to switch from exploration to exploitation and vice versa. In addition, the idea of controlling the amount of jump during the exploitation phase has been utilized in this work. Therefore, this study formulates several rules that will be embedded into HHO to make adaptive switching of exploration/exploitations. An additional rule was embedded to control the amount of jump during the exploitation phase. The main contributions of this work are outlined as follows.

1. It monitors and utilizes population statuses for adaptive exploration/exploitation switching.
2. It uses agent location information to control the amount of jump needed at exploitation mode.
3. It evaluates the performances on multimodal standard benchmark function, large-scale CEC'2010 benchmark, and one real-world high-dimensional wavelength selection problem.

A table that summarizes all previously discussed HHO variants in terms of their type, authors, techniques, and used benchmarks is given in Table 1. The remaining part of this paper is organized as follows. Section 2 overview the standard HHO algorithm. The proposed embedded rules are explained in Sect. 3. A series of experiments that have been conducted to evaluate the effectiveness of the proposed approach are given in Sect. 4. A summary of the research findings is presented in Sect. 5.

2 Harries Hawks optimizer

HHO is a recent optimizer that has been introduced by Heidari et al [8]. It is inspired by the attacking behavior of Harris Hawks birds, and it consists of three main phases, which are exploration, transition, and exploitation. These phases are explained as follows.

2.1 Exploration phase

In the exploration phase, HHO agents are going to perform discovering of the search space. Basically, there are two strategies that have been formulated for the exploration phase. The first one moves the hawk randomly in the search space. The second strategy is guided by both the best solution X^t_{rabbit} and the mean location of the population X^t_m. These exploration strategies are defined as follows.

$$X_i^{t+1} = \begin{cases} X^t_{\text{rand}} - r_1 \cdot \left| U^t_{\text{rand}} - 2 \cdot r_2 \cdot X^t_i \right| & p \geq 0.5 \\ (X^t_{\text{rabbit}} - X^t_m) - r_3 \cdot (lb + r_4 \cdot (ub - lb)) & p < 0.5 \end{cases} \quad (1)$$

where X_i^{t+1} is the next position of ith hawk at search iteration t. Variables $r_1, r_2, r_3,$ andr_4 are random values in the range of $(0, 1)$. Variables ub and lb are upper and lower bound of the search problem. Variable p is a random value used to control the switching between exploration

Table 1 Related work on HHO

Type	Ref.	Authors	Method	Benchmarks
Hybrid	[20]	Abd Elaziz et al.	It incorporated moth-flame, fractional-order, and chaotic maps to enhance HHO exploration	It uses 13 feature selections from UCI dataset and various engineering design problems
	[21]	ElSayed et al.	It integrated sequential quadratic programming (SQP) with HHO as a single model	It was applied for two problems of relays optimal coordination finding
	[22]	Li et al.	It proposed a memetic technique to enhance HHO local search capability	It was evaluated with 29 numerical optimization test functions and the problem of resource-constrained project scheduling and QoS-aware web service
	[23]	Singh et al.	It combined grasshopper with HHO as a single model	It has been applied for the problem of ONUs placement in Fiber-Wireless (FiWi)
	[20]	Ilker et al.	It evolved multiple HHO populations with quantum particles	It was evaluated with 23 dynamic test functions from CEC 2009 benchmark
	[25]	Kashif et al.	It hybridized the sine–cosine algorithm with HHO	It uses 29 test functions of CEC'17 test suite and 16 datasets for the problem of feature selection
	[26]	Abdel-Basset et al.	It incorporated simulated annealing for HHO search refinement	It has been evaluated with 24 standard datasets and 19 artificial datasets for feature selection problems
	[27]	Abba et al.	It integrated PSO with HHO for renewable energy load demand forecasting	It was evaluated using a lab collected data including solar radiation, temperature, and wind speed to predict load demand
	[28]	Ebrahim et al.	It combined since cosine with HHO for finding the optimal control parameters in fuel cell-based electric power system	It has been evaluated with the standard 23 benchmarks and applied for real-time control of energy consumption
	[29]	Bandyopadhyay et al.	It integrated simulated annealing with HHO for performing search refinement	It was assessed using real-world engineering design problems and COVID-19 deep features selection from CT-scan images
	[30]	Suresh et al.	It incorporated a chaotic multi-verse optimizer into HHO	It was evaluated using two public dataset for medical classification problems including PIMA Indian Diabetic and Wisconsin Breast Cancer
	[31]	Mossa et al.	It combined atom optimizer with HHO	It was applied for parameter estimation of Proton exchange membrane fuel cell. It was tested using three case studies including BCS 500-W PEM, 500 W SR-12PEM, and 250 W stack
Modified	[33]	AbdElaziz et al.	It incorporated salp operations into HHO for enhancing exploration performances	It has been applied for 36 functions from IEEE CEC 2005 benchmark and 11 Gy-scale image segmentation problems
	[34]	Wunnava et al.	It embedded differential evolutionary operators	It was evaluated with 500 images from Berkeley BSDS benchmark for multi-level image thresholding
	[35]	Yousri et al.	It added fractional-order calculus (FOC) memory to guide HHO during search progress	It uses 28 functions from CEC2017 benchmarks problems
	[36]	Chen et al.	It included two schemes which are opposition-based and chaotic local search to enhance HHO exploitation	It has been applied for photovoltaic cells design of three problems which are Shell st40, Shell sm55, and Shell kc200gt photovoltaics
	[37]	Li et al.	It incorporated horizontal and vertical crossover operations	It has been applied for photovoltaic parameter estimation of three models, which are SDM, DDM, and PV
	[38]	Qu et al.	It utilizes an information-sharing scheme to exchange agent's locations, etc	It was evaluated with 28 functions of CEC-2017 real-parameter numerical optimization problems
	[39]	Li et al.	It embedded several search strategies to enhance HHO. These strategies are logarithmic spiral and opposition technique to improve the exploration performances. Rosenbrock local search was added to enhance the exploitation ability	It was tested using IEEE CEC2014 benchmark and other engineering and real-world design problems
	[40]	Akdag et al.	It incorporated various random distribution functions to enhance HHO	It has been applied to optimize IEEE 30-bus power system. In addition, it was evaluated with the standard benchmark problems

Table 1 continued

Type	Ref.	Authors	Method	Benchmarks
	[41]	Houssein et al.	It embedded genetic operators to enhance HHO exploitation performances	It was evaluated using two chemoinformatics dataset namely QSAR Biodegradation and MAO
	[42]	Krishna et al.	It triggered local pattern search algorithm during the exploitation phase of HHO	It was tested using 23 standard CEC2005 benchmark and other engineering design problems
	[43]	Singh et al.	It embedded logistic chaotic map to enhance the initialization and exploration ability of HHO	It was evaluated using 12 UCI machine learning repository clustering problems

strategies. It should be noted that the mean position of the population X_m^t is computed as follows.

$$X_m^t = \frac{1}{N}\sum_{i=1}^{N} X_i^t \tag{2}$$

2.2 Transition phase

In HHO, they considered rabbit energy as the primary indicator used to switch from exploration to exploitation mode. This variable is computed according to the following formula.

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{3}$$

where E is escaping energy decreases linearly based on time iterations t. T is the maximum allocated iterations. Variable E_0 is the initial state energy of each individual, and it is varied randomly in the range $E_0 \in (-1, 1)$. E_0 is updated based on the following equation.

$$E_0 = 2\text{rand} - 1 \tag{4}$$

The computed escaping energy E in Eq. (4) will have the following plot in Fig. 1. It can be seen that HHO will be in the exploration phase when $|E| \geq 1$, while exploitation mode will occur when $|E| < 1$ as demonstrated in Fig. 1.

2.3 Exploitation phase

The main idea of the exploitation phase is to exploit the current best location X_{rabbit}^t and to search around it. In HHO, they proposed four different strategies for performing search exploitation, namely soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives. In order to select which exploitation strategy will be executed, both escaping energy $|E|$ and a random variable r is used to control. In particular, the escaping energy $|E|$ will control the switching between soft/hard besiege strategy. When $|E| \geq 0.5$ indicating that the rabbit X_{rabbit}^t still has energy and soft besiege strategy should be applied; otherwise, the rabbit should be exhausted, and hard besiege strategy should be used. In addition, each strategy could run progressive rapid dives when it is applied. As such, the implemented random variable r will control the activation of this property. As such, when $r < 0.5$, progressive rapid dives will be used; otherwise, it will be off. These strategies are explained as follows.

2.3.1 Soft besiege

As mentioned earlier, soft besiege is activated when $r \geq 0.5$ and $|E| \geq 0.5$, which means that the rabbit X_{rabbit}^t still has energy, trying to escape by a random jump. As such, it will

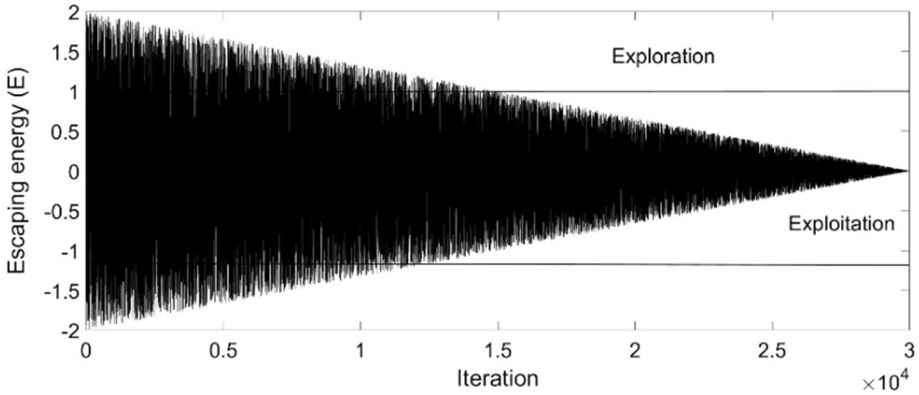


Fig. 1 The behavior of escaping energy E

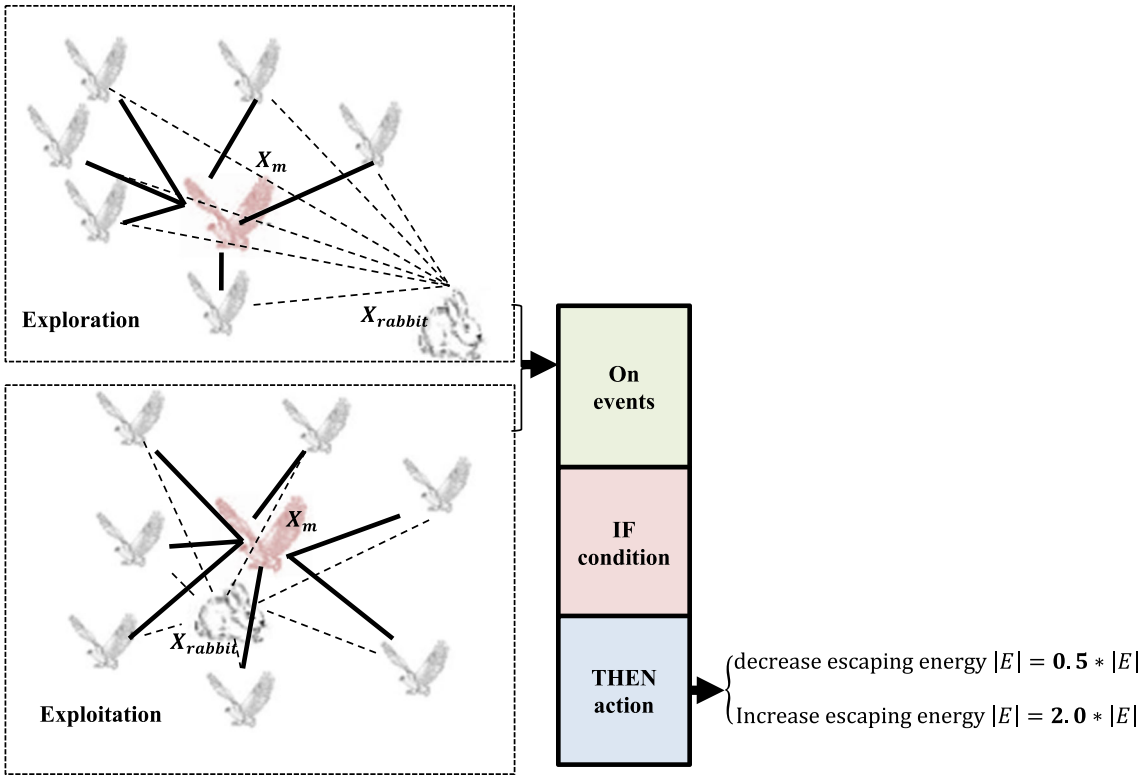


Fig. 2 The proposed embedded rules

be softly encircled by the hawks and attacked according to the following equations.

$$X_i^{t+1} = \Delta X^t - E\left|JX_{\text{rabbit}}^t - X_i^t\right| \tag{5}$$

$$\Delta X^t = X_{\text{rabbit}}^t - X_i^t \tag{6}$$

where ΔX^t is the difference between rabbit location X_{rabbit}^t and current hawk location X_i^t. J is a random factor that mimics rabbit movement, and it is defined as follows.

$$J = 2(1 - r_5) \tag{7}$$

where r_5 is a random value in the range $r_5 \in (-1, 1)$.

2.3.2 Hard besiege

This strategy is applied when the rabbit X_{rabbit}^t is exhausted, and it has only a tiny escaping energy $|E| < 0.5$. Therefore, hawks will attack the rabbit according to the following equation.

$$X_i^{t+1} = X_{\text{rabbit}}^t - E|\Delta X^t| \tag{8}$$

2.3.3 Soft besiege with progressive rapid dives

This strategy is applied when the rabbit still has energy $|E| \geq 0.5$ but $r < 0.5$. However, they suggested in [8] a more intelligent mathematical formula that mimics actual rabbit motion. Specifically, a zigzag motion pattern was formulated using the levy flight (LF) function defined as follows.

$$\text{LF}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{9}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \tag{10}$$

where u, v are random values from $0 to 1$. β is a constant value set as 1.5 as suggested in [8].

Therefore, the hawk will update his position using the following formulas.

$$Y = X_{rabbit}^t - E\left|JX_{rabbit}^t - X_i^t\right| \tag{11}$$

$$Z = Y + S \times LF(D) \tag{12}$$

$$X_i^{t+1} = \begin{cases} Y \text{ if } \text{fitness}(Y) > \text{fitness}\left(X_i^{t+1}\right) \\ Z \text{ if fitness}(Z) > \text{fitness}\left(X_i^{t+1}\right) \end{cases} \tag{13}$$

where S is 1D random values and D is the problem dimension.

2.3.4 Hard besiege with progressive rapid dives

This strategy is very similar to soft besiege with progressive rapid dives, but the hawk will a bit decrease the jump distance because the rabbit is exhausted $|E| < 0.5$. As such, the hawk will update his position considering the mean position X_m^t instead of using X_i^t as defined in Eq. (11). This strategy has been formulated according to the following equations.

$$Y = X_{\text{rabbit}}^t - E\left|JX_{\text{rabbit}}^t - X_m^t\right| \tag{14}$$

$$Z = Y + S \times LF(D) \tag{15}$$

$$X_i^{t+1} = \begin{cases} Y \text{ if } \text{ fitness } (Y) > \text{fitness } \left(X_i^{t+1}\right) \\ Z \text{ if } \text{ fitness } (Z) > \text{fitness } \left(X_i^{t+1}\right) \end{cases} \tag{16}$$

3 The proposed embedded rules

As explained earlier that HHO depends on escaping energy factor $|E|$ for the transition from exploration to exploitation. However, $|E|$ is a time dependent parameter that decreases gradually over time, as shown in Fig. 1. As such, HHO has a lower chance to try to exploit the discovered region and switch to exploitation mode due to $|E| > 1$. As a result of this, HHO will have a slow convergence rate due to the prevention of switching to exploitation mode at the beginning of search progress. Another limitation of HHO is that when $|E|$ becomes very small and the hawks trapped in local optima region, it will be hard to escape and return to exploration stage. To overcome this shortcoming of HHO, this research introduces embedded rules that will improve balancing between exploration and exploitation and help HHO to escape from possible local optima traps by amplifying escaping energy $|E|$.

The main idea of the proposed rules is given in Fig. 2, where embedded rules are going to be triggered according to the occurrence of some events. The first rule is given in Fig. 3. This rule has been formulated to allow HHO to escape from local optima. Three events are used to control the trigger of this rule which are population success, current search time, and a random value r. pop_success factor will take a value of zero or one. It will be zero when the best population location X_{rabbit}^t was not changed; otherwise, it will be one. The second variable t represents the current search time step. It is used to prevent the rule to be triggered during the exploration phase where the escaping energy is already greater than one, i.e., $|E| \geq 1$. The variable r is a random value that randomizes the trigger of RULE 1.

The second rule was designed to help HHO to switch from exploration to exploitation mode at the beginning of the search process. RULE 2 is shown in Fig. 4. It is indicated that RULE 2 is controlled by three parameters which are the location of the rabbit $\boldsymbol{X}_{\text{rabbit}}$ with respect to the mean location $\boldsymbol{X_m}$, current search time step t, and a random value r. It should be noted that the last two parameters (i.e., t and r) are the same in RULE 1; however, the first event is used to check the current population status. Referring to Fig. 2, it can be seen when the hawks encircle the rabbit, then the location of $\boldsymbol{X}_{\text{rabbit}}$ will be close to $\boldsymbol{X_m}$. To check this status, the fitness value of $\boldsymbol{X_m}$ is computed and compared to other hawks. If it belongs to the top 10% of the population, then it means $\boldsymbol{X_m}$ location is near to $\boldsymbol{X}_{\text{mean}}$.

The third rule is formulated to control the amount of hawk jump needed during the exploitation mode. RULE 3

Fig. 3 RULE 1 for switching from exploitation to exploration

RULE 1: IF $pop\_success = 0$ **AND** $t < 0.5\,T$ **AND** $r \geq 0.5$

THEN increase the escaping energy **|E| = 2.0*|E|**

Fig. 4 RULE 2 for switching from exploration to exploitation

RULE 2: IF $\boldsymbol{X_{rabbit}}$ location is near to $\boldsymbol{X_m}$ **AND** $t > 0.5\,T$ **AND** $r \geq 0.5$

THEN decrease the escaping energy **|E| = 0.5* |E|**

Fig. 5 RULE 3 to control the amount of hawk jump during exploitation

RULE 3: IF $|E| < 1$ **AND** $r \geq 0.5$

THEN IF $|X_{rabbit}^t - X_i^t| < |X_i^t - \boldsymbol{X_m}|$ **THEN** |E| = 0.5* |E| **, ELSE** |E| = 2* |E|

is defined in Fig. 5, and it will be activated only during the exploitation phase when $|E| < 1$. RULE 3 is triggered based on hawk location with respect to X^t_{rabbit}, and X^t_i as given in Fig. 5. Rule 3 imply that hawks located closely to X^t_{rabbit} need a small jump; however, those far away hawks need a larger jump to reach the location of X^t_{rabbit}. This jump is reflected by the amount of escaping energy that will influence Eqs. (5), (8), (11), and (14).

The complete steps of the proposed REHHO algorithm are given in algorithm 1 and Fig. 6.

Fig. 6 Flowchart of the proposed REHHO algorithm ▶

Algorithm 1: Rules Embedded Harries Hacks Optimizer (REHHO)

1. Randomly initialize HHO agents $X_i (i = 1,2, \ldots \ldots N)$ according to search space, Set the maximum number of iterations T
2. **While** t < T
3. Compute fitness value for each HHO agent
4. Set the best position $\boldsymbol{X_{rabbit}}$
5. **For** (each hawk) **do**
6. Calculate escaping energy $\boldsymbol{E}$ using Eq.(3)
7. **Apply embedded rules (RULE 1 and RULE 2) that control escaping energy**
8. **If** $(|E| \geq 1)$ **then**
9. Execute exploration phase and update hawk location using Eq.(1)
10. **else if** $(|E| < 1)$
11. **Apply embedded rule (RULE 3) that controls the amount of hawk jump**
12. **if** $(|E| \geq 0.5 \; and \; r \geq 0.5)$ **then**
13. Execute soft besiege and update hawk using Eq.(5)
14. **else if** $(|E| < 0.5 \; and \; r \geq 0.5)$ **then**
15. Execute hard besiege and update hawk using Eq.(8)
16. **else if** $(|E| \geq 0.5 \; and \; r < 0.5)$ **then**
17. Execute soft besiege with progressive rapid dives using Eq.(13)
18. **else if** $(|E| < 0.5 \; and \; r < 0.5)$ **then**
19. Execute hard besiege with progressive rapid dives using Eq. (16)
20. **end If**
21. **end for**
22. **end while**
23. Return $\boldsymbol{X_{rabbit}}$

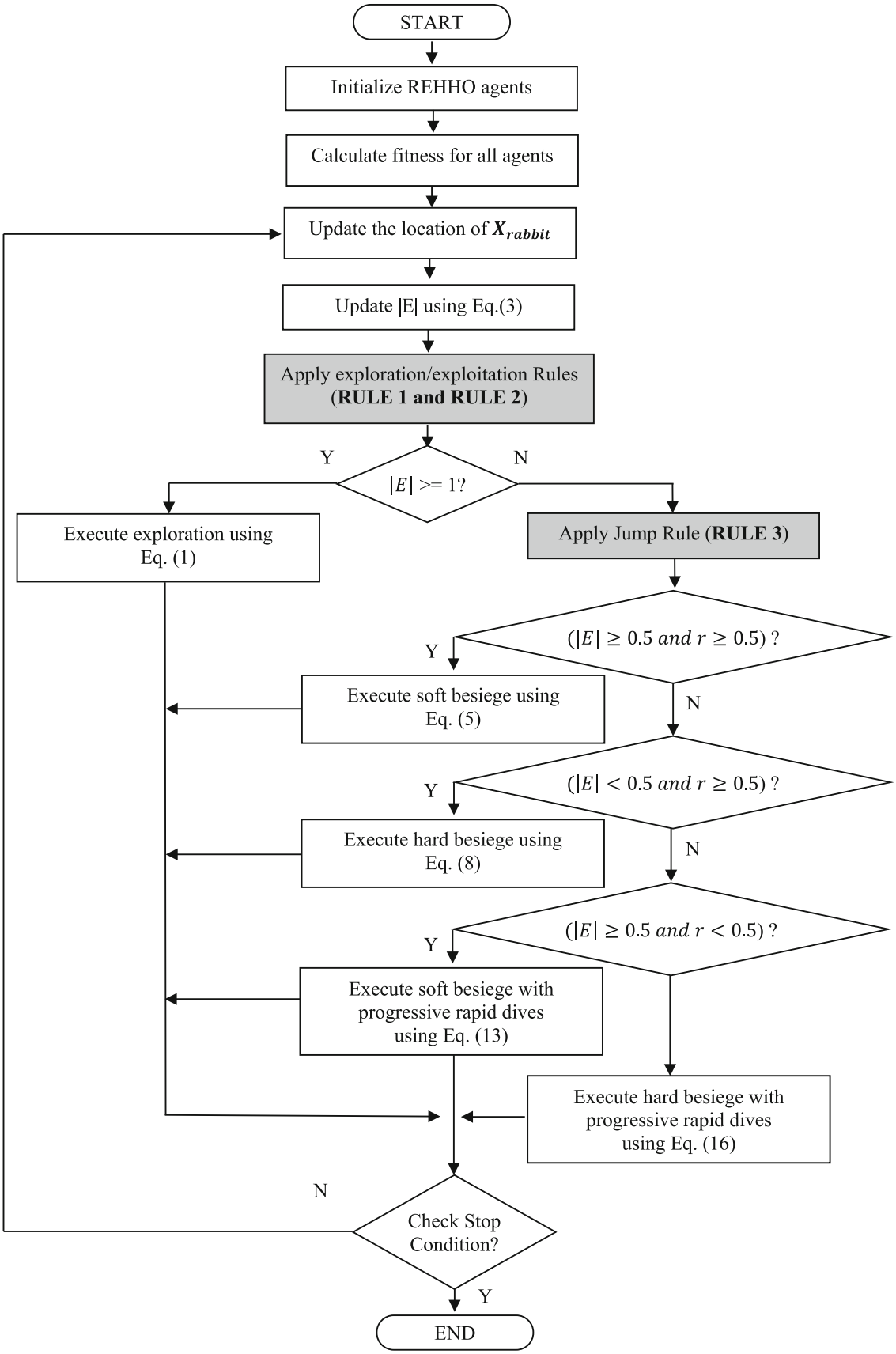
4 Experimental results

4.1 The standard benchmark problems

This section evaluates the performances of REHHO as compared with HHO using six multimodal benchmark problems, which are Schwefel, Rastrigin, Ackley, Griewank, Penalized, and Penalized 2. These problems were executed with various high-dimensional ranging from 1000- D to 10,000-D. These experiments were epeated 30 times, and the maximum number of iterations was set to 10^4.

$$F1(X) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{|x_i|}\right), \quad search \;\; range \; [-500, 500], \quad D = 1000, \; f_{\min} = -418.9829 * n \tag{17}$$

The mathematical formula of Schwefel function is given in Eq. (17). This function has multiple local optima, as can be seen in Fig. 7. Therefore, finding the global optima of


START
Initialize REHHO agents
Calculate fitness for all agents
Update the location of X_{rabbit}
Update |E| using Eq.(3)
Apply exploration/exploitation Rules (RULE 1 and RULE 2)
Y
$|E|$ >= 1?
N
Execute exploration using Eq. (1)
Apply Jump Rule (RULE 3)
Y
$(|E| \geq 0.5\ and\ r \geq 0.5)$?
Execute soft besiege using Eq. (5)
N
Y
$(|E| < 0.5\ and\ r \geq 0.5)$?
Execute hard besiege using Eq. (8)
N
Y
$(|E| \geq 0.5\ and\ r < 0.5)$?
Execute soft besiege with progressive rapid dives using Eq. (13)
Execute hard besiege with progressive rapid dives using Eq. (16)
N
Check Stop Condition?
Y
END

this function under high dimensions is a quiet challenge. Nevertheless, REHHO was able to reach the minimum value faster than HHO, as shown in Fig. 7. This is due to the dynamic of REHHO and its ability to perform earlier exploitation searches as compared with HHO. Further analysis is conducted by evaluating the performances when the dimension of the problem is increased to 5000-D and 10,000-D, respectively. Therefore, the best, medium, worst, mean, and standard deviation of REHHO and HHO algorithms are reported in Table 2. It can be seen that both algorithms reported almost the same mean values in all variations of dimensions which are 1000-D, 5000-D, and 10,000-D. However, REHHO was able to reach the global optima earlier than HHO, as indicated in Fig. 7.

$$F_2(X) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right], \quad search\ \ range\ [-5.12, 5.12], \quad D = 1000,\ f_{\min} = 0 \tag{18}$$

Rastrigin function is given in Eq. (18), and it differs from the Schwefel function where the minimum value is located at point zero, as shown in Fig. 8. The convergence curve of both functions is almost similar. This is due to the nature of the optimized problem, where it is a bit easier as compared with the shifted Schwefel function discussed earlier. Moreover, the variation in the dimensions does not affect the performances of both algorithms, as indicated in Table 3. In other words, REHHO and HHO were able to achieve the global optima in all runs with zero standard deviation, which showed the stability of both algorithms.

Table 2 Results of Schwefel functions

Dim	Fitness	Algorithm	
		REHHO	HHO
1000 − D	Best	− 4.1898e + 05	4.1898e + 05
	Median	− 4.1898e + 05	− 4.1898e + 05
	Worst	− 4.1898e + 05	− 4.1898e + 05
	Mean	− 4.1898e + 05	− 4.1898e + 05
	Std	0	0
5000 − D	Best	− 2.0949e + 06	− 2.0949e + 06
	Median	− 2.0949e + 06	− 2.0949e + 06
	Worst	− 2.0949e + 06	− 2.0949e + 06
	Mean	− 2.0949e + 06	− 2.0949e + 06
	Std	0	0
10,000 − D	Best	− 4.1898e + 06	− 4.1898e + 06
	Median	− 4.1898e + 06	− 4.1898e + 06
	Worst	− 4.1898e + 06	− 4.1898e + 06
	Mean	− 4.1898e + 06	− 4.1898e + 06
	Std	0	0

$$F_3(X) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e, \quad search\ \ range\ [-32, 32]\ D = 1000,\ f_{\min} = 0 \tag{19}$$

Equation (19) shows the mathematical formula of the Ackley function. The 2-D plot of this function is shown in Fig. 9. The plotted convergence curve indicated that REHHO is able to converge faster due to the flattening

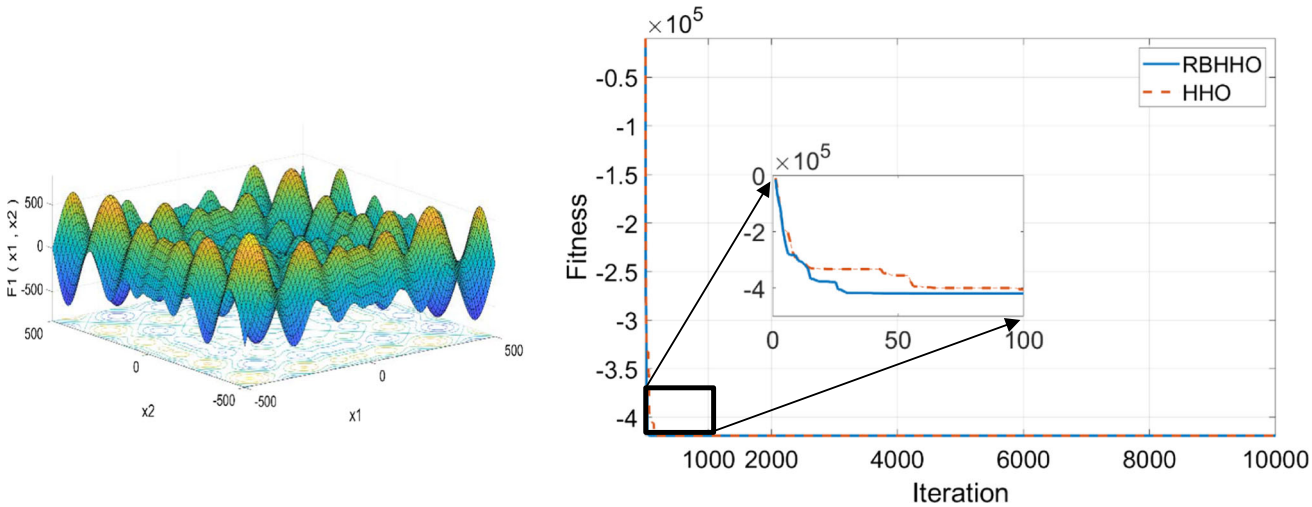


Fig. 7 The Schwefel function with its convergence curve (1000-D)

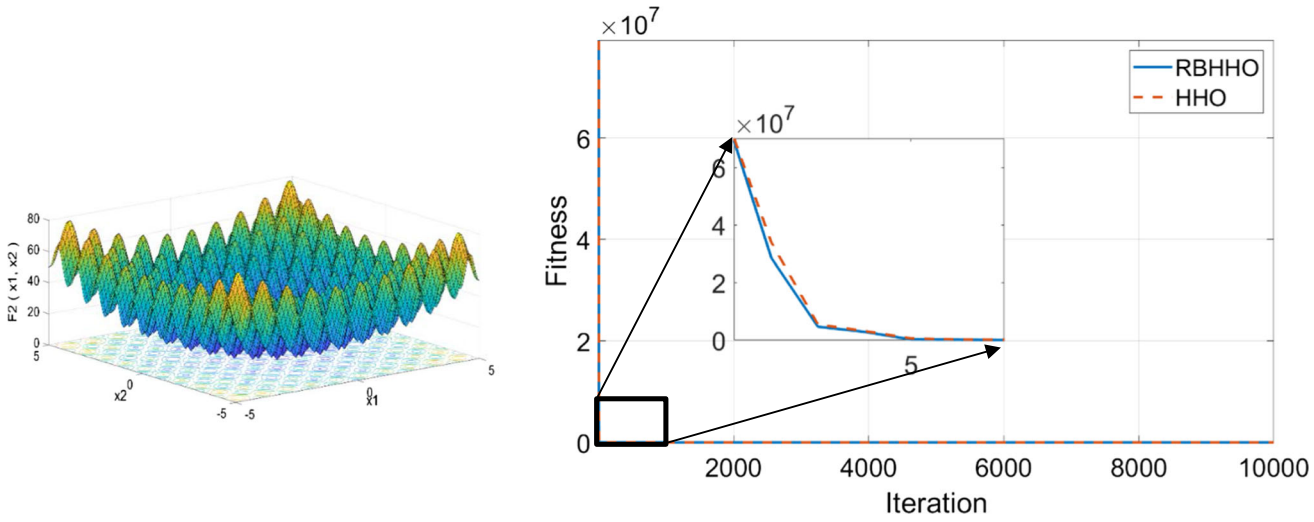


Fig. 8 The Rastrigin function with its convergence curve (1000-D)

Table 3 Results of rastrigin function

Dim	Fitness	Algorithm	
		REHHO	HHO
1000–D	Best	0	0
	Median	0	0
	Worst	0	0
	Mean	0	0
	Std	0	0
5000–D	Best	0	0
	Median	0	0
	Worst	0	0
	Mean	0	0
	Std	0	0
10,000–D	Best	0	0
	Median	0	0
	Worst	0	0
	Mean	0	0
	Std	0	0

nature of this problem. It should be noted that both algorithms were able to reach global optima with less than 50 iterations.

As can be seen in Table 4, increasing the dimensionality of this problem does not influence the performance of both algorithms. This is due to the simplicity of the tacked Ackley function.

$$F_4(X) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,\ search\ \ range\ [-600, 600],\ D = 1000, f_{\min} = 0 \tag{20}$$

The Griewank function is given in Eq. (20). This problem is considered more challenging as compared with Ackley, where it has a lot of local optima, as indicated in Fig. 10. The convergence of the Griewank function confirms the ability of REHHO to switch faster and perform exploitation searches. Nevertheless, both REHHO and HHO were able to reach the global minima, as shown in Table 5.

The Penalized function is defined as follows.

$$F_5(X) = \frac{\pi}{D}\left\{10\sin(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2 \left[1 + 10sin^2(\pi y_{i+1})\right] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$$

$$y_i = 1 + \frac{x_i + 1}{4}$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < a \end{cases} \tag{21}$$

search range $[-50, 50]$,

Unlike previously discussed functions, the Penalized function is considered more challenging. The convergence curve of this function is given in Fig. 11, and it is clearly seen that REHHO converges a bit faster. More importantly, increasing the dimensions of Penalized function from 1000-D to 10,000-D does not influence the performances of REHHO, as given in Table 6.

The Penalized 2 function is defined as follows:

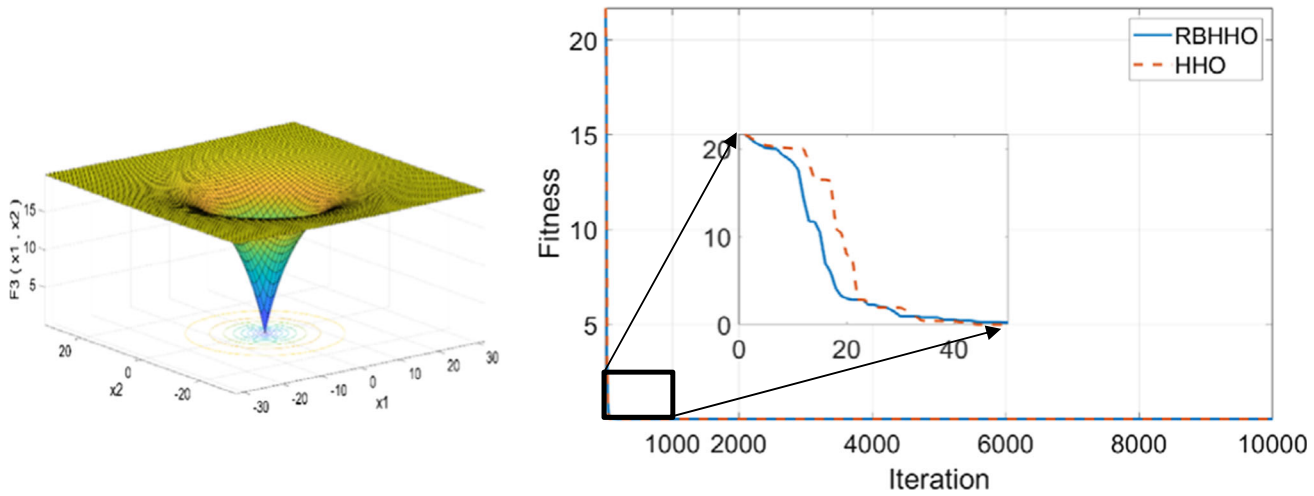


Fig. 9 The Ackley function with its convergence curve (1000-D)

Table 4 Results of Ackley function

Dim	Fitness	Algorithm	
		REHHO	HHO
1000-D	Best	8.8818e − 16	8.8818e − 16
	Median	8.8818e − 16	8.8818e − 16
	Worst	8.8818e − 16	8.8818e − 16
	Mean	8.8818e − 16	8.8818e − 16
	Std	0	0
5000-D	Best	8.8818e − 16	8.8818e − 16
	Median	8.8818e − 16	8.8818e − 16
	Worst	8.8818e − 16	8.8818e − 16
	Mean	8.8818e − 16	8.8818e − 16
	Std	0	0
10,000-D	Best	8.8818e − 16	8.8818e − 16
	Median	8.8818e − 16	8.8818e − 16
	Worst	8.8818e − 16	8.8818e − 16
	Mean	8.8818e − 16	8.8818e − 16
	Std	0	0

$$F_6(X) = 0.1\left\{ sin^2(3\pi x_1) + \sum_{i=1}^{D}(x_i - 1)^2 \times \left[1 + sin^2(3\pi x_i + 1)\right] + (x_D - 1)^2\left[1 + sin^2(2\pi x_D)\right]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4), \text{ search range } [-50, 50], \tag{22}$$

The conducted analysis on the Penalized 2 function is given in Fig. 12 and Table 7. As can be seen, REHHO outperformed HHO with a better mean fitness value and faster convergence curve.

4.2 2010 large scale global benchmark problems

This experimental section examines the effectiveness of the embedded rules in enhancing HHO (i.e., REHHO algorithm) on large-scale benchmark problems. Specifically, a set of 20 large-scale functions from 1000-D CEC'2010 benchmark functions [44] has been used. The details of these functions are given in Table 8. As can be seen, CEC'2010 benchmark consists of 5 groups of functions, namely, separable, single-group $\boldsymbol{m}-$ non-separable functions, $\boldsymbol{D}/2\boldsymbol{m}$ group $\boldsymbol{m}$-non-separable functions, $\boldsymbol{D}/\boldsymbol{m}$ group $\boldsymbol{m}$-non-separable functions, and fully separable functions. Each experiment was repeated 30 times, and the population size was 30 agents with a maximum number of fitness evaluations 3×10^4.

Table 9 depicts the results of REHHO and HHO on CEC'2010 largescale problems. It should be noted that the separable functions F1, F2, and F3 have a smaller number of local optima as compared with other categories. Nevertheless, the reported results showed that REHHO achieved the best results. This is due to the ability of REHHO to perform fast convergence and switch earlier to exploitation mode. In other words, the implemented adaptive switching scheme using the embedded rules helps REHHO to reach faster to global optima point. The results of single-group $\boldsymbol{m}-$ non-separable functions are given by functions F4, F5, F6, F7, and F8. Similarly, REHHO reports the best mean value in all conducted functions.

For the category of $\boldsymbol{D}/2\boldsymbol{m}$ group $\boldsymbol{m}-$ non-separable functions, which is considered more complex, REHHO is still able to perform well due to its ability to switch back from exploitation mode to exploration mode when

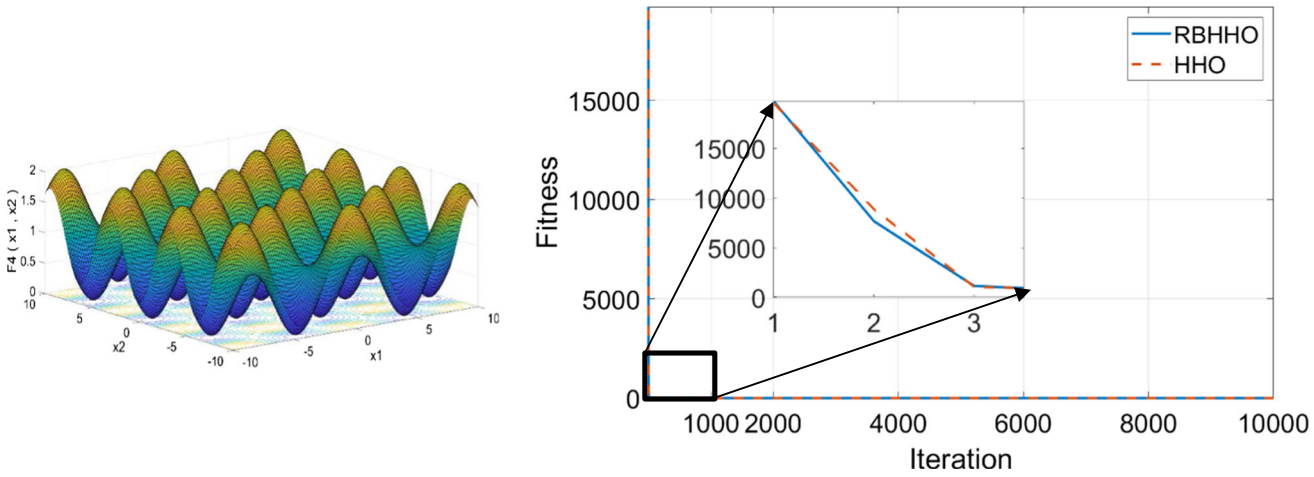


Fig. 10 The Griewank function with its convergence curve (1000–D)

Table 5 Results of Griewank function

Dim	Fitness	Algorithm	
		REHHO	HHO
1000–D	Best	0	0
	Median	0	0
	Worst	0	0
	Mean	0	0
	Std	0	0
5000–D	Best	0	0
	Median	0	0
	Worst	0	0
	Mean	0	0
	Std	0	0
10,000–D	Best	0	0
	Median	0	0
	Worst	0	0
	Mean	0	0
	Std	0	0

population success status is zero, then rule 2 will be triggered as explained in Sect. 3. In addition, REHHO reports the best mean value in the remaining functions from F14 to F20.

In conclusion, conducted analysis on CEC'2010 large-scale benchmark functions showed that the performances of REHHO have been improved considerably from the embedded rules.

Further analysis was conducted by investigating the graphical behavior of REHHO convergence during run time. Specifically, the base-10 logarithmic mean values of the fitness function from a total of 30 runs are computed and plotted in Fig. 13. It is shown that REHHO has faster convergence in all benchmark functions. This is due to the ability of REHHO to switch earlier to exploitation mode at the beginning of the search process, as mentioned previously.

4.2.1 Compared with other metaheuristic algorithms

In this section, the performance of REHHO has been compared with several well-known state-of-the-art optimization algorithms. These algorithms are particle swarm optimization (PSO), differential evolution (DE), BAT optimizer, Arithmetic Optimization Algorithm (AOA) [45], and Horse Herd Algorithm (HHA) [46]. It is worth mentioning that BAT, PSO, and DE are very famous population-based algorithms and are widely used in the literature as baseline comparison algorithms. The settings of these algorithms are given in Table 10. For all conducted algorithms, each experiment was repeated 30 times, and the population size was 30 agents with a maximum number of fitness evaluations 3×10^4.

The results of the conducted analysis are given in Table 11. As can be seen, REHHO reported the best mean value in most functions except for five of them, namely F5, F6, F8, and F10, where DE outperformed other algorithms. This is due to the benefits of crossover and mutation

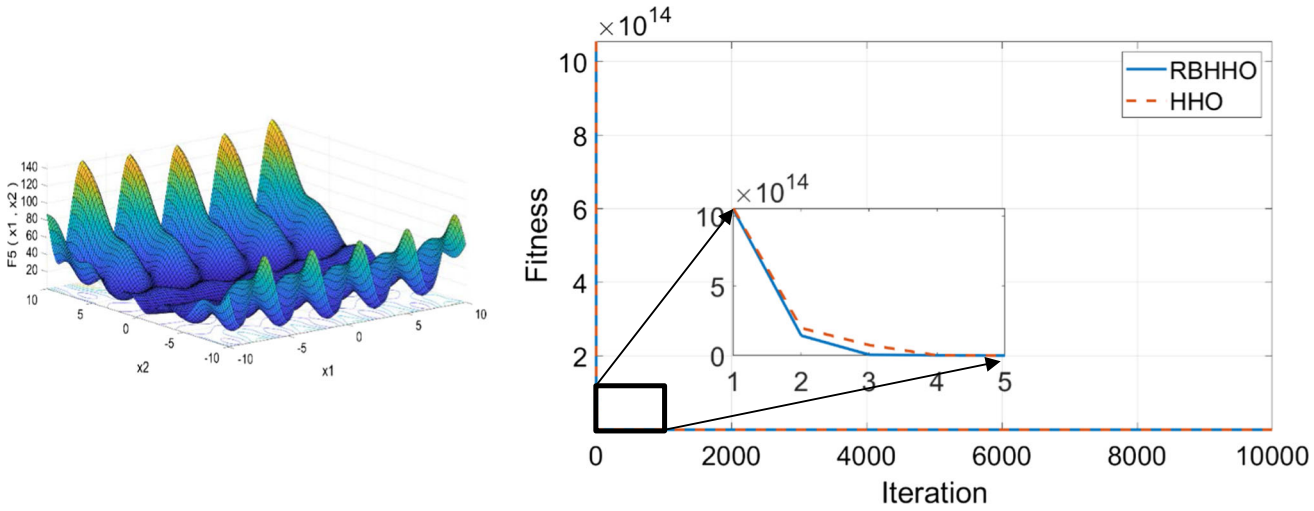


Fig. 11 The Penalized function with its convergence curve (1000–D)

Table 6 Results of Penalized function

Dim	Fitness	Algorithm	
		REHHO	HHO
1000–D	Best	7.4754e − 11	5.6363e − 10
	Median	2.0208e − 10	8.2071e − 10
	Worst	1.9952e − 09	6.4469e − 09
	Mean	**7.5735e − 10**	2.6104e − 09
	Std	1.0739e − 09	3.3250e − 09
5000–D	Best	1.8923e − 12	1.0878e − 10
	Median	4.1541e − 11	1.5618e − 10
	Worst	2.1488e − 10	6.7634e − 10
	Mean	**8.6105e − 11**	3.1377e − 10
	Std	1.1327e − 10	3.1489e − 10
10,000–D	Best	1.2802e − 10	2.5269e − 10
	Median	1.6376e − 10	8.4575e − 10
	Worst	5.9265e − 10	1.9220e − 09
	Mean	**2.5641e − 10**	1.0068e − 09
	Std	3.0082e − 09	8.4625e − 10

Bold values indicate the best mean value (i.e., average minimum)

operation used by the DE algorithm. Nevertheless, REHHO reports superior performances as compared with other recent algorithms, namely AOA and HHA. The worst results were archived by the BAT optimizer due to the lack of exploration operations used for handling large-scale problems.

4.2.2 Statistical analysis

In this section, the statistical t-test [47] was used to evaluate the outcome of the 1000-D CEC'2010 large-scale benchmark statistically. The t-test is used to determine if there is a significant difference between REHHO and other algorithms in terms of the reported mean value. To compute the significant difference using t- test, three values are required, which are the mean, the standard deviation, and the number of data (repeated times). From these computed values, the degrees of freedom and t-distribution value will be identified to generate the *p*-value.

To implement the t-test, the null hypothesis H_0 assumes that REHHO and other compared algorithms performed equally. However, the alternative hypothesis H_1 assumed that REHHO outperformed other algorithms. In this analysis, the *p-value* of the t-test was set at 0.05 (i.e., 95% confidence level), meaning that the alternative hypothesis H_1 would be accepted if the *p-value* was less than 0.05. The *p*-value results of the t-test are reported in Table 12, and it can be seen that most *p*-values were less than 0.05, which confirms that the proposed REHHO algorithm significantly outperforms other conducted algorithms in most of the functions.

4.2.3 Computational time analysis

The computational time of the proposed REHHO is compared against HHO on large-scale problems. This analysis is conducted to measure the overhead of the embedded

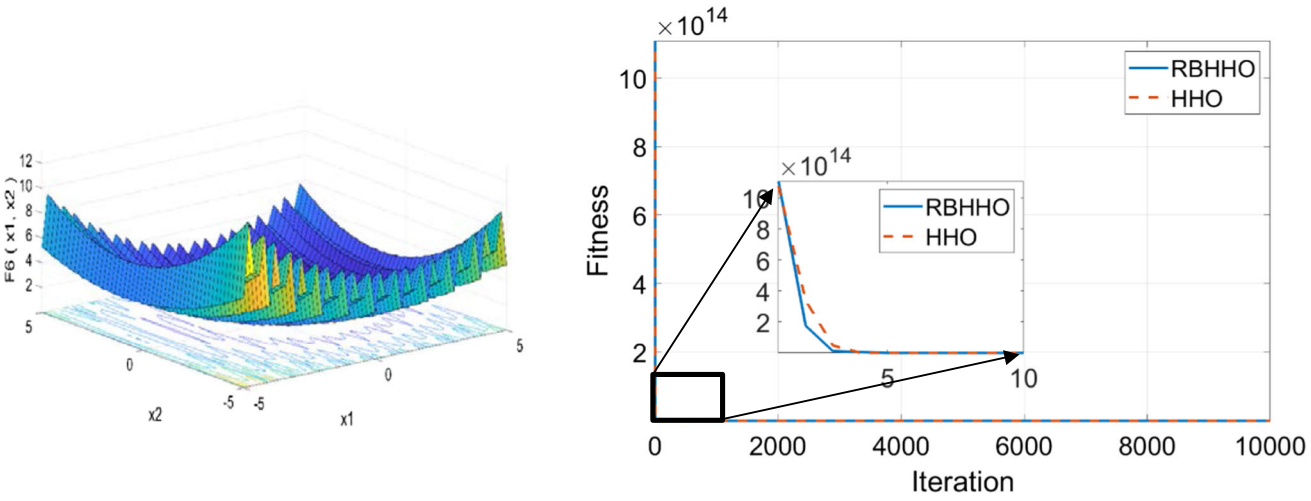


Fig. 12 The Penalized 2 function with its convergence curve (1000–D)

Table 7 Results of penalized 2 function

Dim	Fitness	Algorithm	
		REHHO	HHO
1000–D	Best	9.6019e − 08	2.9647e − 08
	Median	9.6562e − 08	4.0487e − 07
	Worst	2.0774e − 07	2.2211e − 06
	Mean	**1.3344e − 07**	8.8522e − 07
	Std	6.4348e − 08	1.1721e − 06
5000–D	Best	3.5014e − 08	3.0948e − 08
	Median	7.4315e − 07	5.2818e − 07
	Worst	2.0981e − 07	4.1433e − 06
	Mean	**1.0638e − 07**	1.5675e − 06
	Std	9.1702e − 07	2.2445e − 06
10,000–D	Best	2.4982e − 07	6.3753e − 06
	Median	9.9053e − 07	7.1744e − 06
	Worst	3.3143e − 06	2.6157e − 05
	Mean	**1.1461e − 06**	1.3236e − 05
	Std	1.8781e − 05	1.1198e − 05

Bold values indicate the best mean value (i.e., average minimum)

rules. The hardware and software specifications of the adopted PC are given in Table 13. The average computational time for 30 runs of the F1 function of large-scale problems is computed and reported in Table 14. As can be seen, REHHO required extra time due to the calculation of population status and agent location needed to fire the embedded rules. Nevertheless, the computational time overhead consumed by these rules is still affordable, which is 34 s only. This value represents around 10% of the total time needed by HHO.

4.3 NIR wavelength selection

This section evaluates the performances of REHHO in performing wavelength selection of the NIR spectrum of gasoline [48]. This case study contains 60 samples with wavelength (x-axis) of range from 900 to 1700 nm and intervals of 2 nm, which result in 401 channels/per sample. The values on the y-axis represent the amount of absorbed heat, as shown in Fig. 14. The dataset has been divided into 50% for training (30 samples) and 50% for testing.

To encode this problem, a 1D binary vector of size 401 is given in Fig. 15. As such, optimization algorithms were applied to find the most distinguished wavelengths. As can be seen in Fig. 15, when the value of the corresponding wavelength W is set to 1, then it will be selected; otherwise, it will be skipped.

Conducted optimization algorithms are guided by the accuracy of Partial Least Squares (PLS) regressor and complexity measure. The fitness function of this problem is formulated as follows.

$$\text{fitness} = -1*((0.9*\text{Accuracy}) - (0.1*\text{complexity})) \tag{23}$$

$$\text{Accuracy} = 1 - \frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{N}(y_i - \overline{y_i})^2}} \tag{24}$$

$$\text{Complexity} = \frac{\text{number of selected wavelengths}}{401} \tag{25}$$

where y_i is the actual value of absorbed heat, $\hat{y}_i$ is the predicted value, and $\overline{y_i}$ is the mean value of the training set.

Table 8 Description of 1000 − D CEC'2010 large − scale benchmark functions

Type	Function	Description	Dim	Range	f_{min}
Separable functions	$F_1(X) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} Z_i^2$	Shifted elliptic function	1000	[− 100, 100]	0
	$F_2(X) = \sum_{i=1}^{D} [Z_i^2 - 10\cos(2\pi Z_i) + 10]$	Shifted Rastrigin's function	1000	[− 5, 5]	0
	$F_3(X) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} Z_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi Z_i)\right) + 20 + e$	Shifted ackley's function	1000	[− 32, 32]	0
Single-group $\boldsymbol{m}-$ non-separable Functions	$F_4(X) = F_{rot\_elliptic}[Z(P_1 : P_m)] * 10^6 + F_{elliptic}[Z(P_{m+1} : P_D)]$	Single-group shifted and m-rotated elliptic function	1000	[− 100, 100]	0
	$F_5(X) = F_{rot\_rastrigin}[Z(P_1 : P_m)] * 10^6 + F_{rastrigin}[Z(P_{m+1} : P_D)]$	Single-group shifted and m-rotated rastrigin's function	1000	[− 5, 5]	0
	$F_6(X) = F_{rot\_ackley}[Z(P_1 : P_m)] * 10^6 + F_{ackley}[Z(P_{m+1} : P_D)]$	Single-group shifted and m-rotated ackley's function	1000	[− 32, 32]	0
	$F_7(X) = F_{schwefel}[Z(P_1 : P_m)] * 10^6 + F_{sphere}[Z(P_{m+1} : P_D)]$	Single-group shifted m-dimensional schwefel's	1000	[− 100, 100]	0
	$F_8(X) = F_{rosenbrock}[Z(P_1 : P_m)] * 10^6 + F_{sphere}[Z(P_{m+1} : P_D)]$	Single-group shifted m-dimensional rosenbrock's function	1000	[− 100, 100]	0
$\frac{\boldsymbol{D}}{\boldsymbol{2m}}$ group $\boldsymbol{m}$-non-separable functions	$F_9(X) = \sum_{k=1}^{\frac{D}{2m}} F_{rot\_elliptic}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right] + F_{elliptic}\left[z\left(P_{\frac{D}{2}+1} : P_D\right)\right]$	$\frac{D}{2m}$ group shifted and m-rotated elliptic function	1000	[− 100, 100]	0
	$F_{10}(X) = \sum_{k=1}^{\frac{D}{2m}} F_{rot\_rastrigin}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right] + F_{rastrigin}\left[z\left(P_{\frac{D}{2}+1} : P_D\right)\right]$	$\frac{D}{2m}$ group shifted and m-rotated rastrigin function	1000	[− 5, 5]	0
	$F_{11}(X) = \sum_{k=1}^{\frac{D}{2m}} F_{rot\_ackley}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right] + F_{ackley}\left[z\left(P_{\frac{D}{2}+1} : P_D\right)\right]$	$\frac{D}{2m}$ group shifted and m-rotated ackley's function	1000	[− 32, 32]	0
	$F_{12}(X) = \sum_{k=1}^{\frac{D}{2m}} F_{schwefel}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right] + F_{sphere}\left[z\left(P_{\frac{D}{2}+1} : P_D\right)\right]$	$\frac{D}{2m}$ group shifted m-rotated schwefel's	1000	[− 100, 100]	0
	$F_{13}(X) = \sum_{k=1}^{\frac{D}{2m}} F_{rosenbrock}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right] + F_{sphere}\left[z\left(P_{\frac{D}{2}+1} : P_D\right)\right]$	$\frac{D}{2m}$ group shifted m-rotated rosenbrock's function	1000	[− 100, 100]	0

Table 8 (continued)

Type	Function	Description	Dim	Range	f_{min}
$\frac{D}{m}$ group m-non-separable Functions	$F_{14}(X) = \sum_{k=1}^{\frac{D}{m}} F_{rot\_elliptic}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right]$	$\frac{D}{m}$ group shifted and m-rotated elliptic function	1000	[− 100, 100]	0
	$F_{15}(X) = \sum_{k=1}^{\frac{D}{m}} F_{rot\_rastrigin}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right]$	$\frac{D}{m}$ group shifted and m-rotated rastrigin function	1000	[− 5, 5]	0
	$F_{16}(X) = \sum_{k=1}^{\frac{D}{m}} F_{rot\_ackley}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right]$	$\frac{D}{m}$ group shifted and m-rotated ackley's function	1000	[− 32, 32]	0
	$F_{17}(X) = \sum_{k=1}^{\frac{D}{m}} F_{schwefel}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right]$	$\frac{D}{m}$ group shifted m-rotated schwefel	1000	[− 100, 100]	0
	$F_{18}(X) = \sum_{k=1}^{\frac{D}{m}} F_{rosenbrock}\left[z\left(P_{(k-1)*m+1} : P_{k*m}\right)\right]$	$\frac{D}{m}$ group shifted m-rotated rosenbrock's function	1000	[− 100,100]	0
Fully separable Functions	$F_{19}(X) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_i\right)^2$	Shifted schwefel's	1000	[− 100, 100]	0
	$F_{20}(X) = \sum_{i=1}^{D-1}\left[100\left(z_i^2 - z_{i+1}\right)^2 + (z_i - 1)^2\right]$	Shifted rosenbrock's function	1000	[− 100, 100]	0

Table 9 Results of 1000-D CEC'2010 large-scale functions

Function	Fitness	Algorithm	
		REHHO	HHO
F1	Best	2.6334e + 09	2.5870e + 09
	Median	2.8842e + 09	2.9525e + 09
	Worst	3.0811e + 09	3.1307e + 09
	Mean	**2.8583e + 09**	2.9077e + 09
	Std	1.3393e + 08	1.8042e + 08
F2	Best	1.5968e + 04	1.6194e + 04
	Median	1.6255e + 04	1.6394e + 04
	Worst	1.6685e + 04	1.6656e + 04
	Mean	**1.6285e + 04**	1.6405e + 04
	Std	221.9340	146.6359
F3	Best	20.2382	20.1865
	Median	20.5327	20.5663
	Worst	20.6923	20.7024
	Mean	**20.5073**	20.5474
	Std	0.1428	0.1591
F4	Best	1.2145e + 13	1.3675e + 13
	Median	1.8154e + 13	1.8249e + 13
	Worst	2.5972e + 13	3.1059e + 13
	Mean	**1.8575e + 13**	1.9648e + 13
	Std	3.7970e + 12	5.3968e + 12
F5	Best	4.1407e + 08	4.1267e + 08
	Median	4.5136e + 08	4.4175e + 08
	Worst	4.8347e + 08	4.9685e + 08
	Mean	**4.4886e + 08**	4.4936e + 08
	Std	2.5081e + 07	2.8098e + 07
F6	Best	1.9146e + 07	1.9100e + 07
	Median	1.9211e + 07	1.9226e + 07
	Worst	1.9276e + 07	1.9282e + 07
	Mean	**1.9212e + 07**	1.9215e + 07
	Std	3.9172e + 04	5.3649e + 04
F7	Best	1.3083e + 09	1.3848e + 09
	Median	1.7299e + 09	1.7041e + 09
	Worst	2.0005e + 09	2.6482e + 09
	Mean	**1.7102e + 09**	1.8329e + 09
	Std	2.4558e + 08	4.5454e + 08
F8	Best	4.6868e + 09	3.5725e + 09
	Median	6.2015e + 09	7.0823e + 09
	Worst	1.2757e + 10	1.3574e + 10
	Mean	**6.9842e + 09**	8.3367e + 09
	Std	2.5889e + 09	3.3609e + 09
F9	Best	4.3561e + 09	4.4583e + 09
	Median	4.5738e + 09	4.6868e + 09
	Worst	4.8799e + 09	5.1160e + 09
	Mean	**4.5915e + 09**	4.7577e + 09
	Std	1.6658e + 08	2.0995e + 08

Table 9 (continued)

Function	Fitness	Algorithm	
		REHHO	HHO
F10	Best	1.6334e + 04	1.6174e + 04
	Median	1.6526e + 04	1.6680e + 04
	Worst	1.6897e + 04	1.6753e + 04
	Mean	**1.6559e + 04**	1.6617e + 04
	Std	169.1596	177.6641
F11	Best	221.5103	220.8727
	Median	222.7064	222.9485
	Worst	224.8432	225.3188
	Mean	**222.8913**	223.1134
	Std	1.0573	1.2622
F12	Best	1.9385e + 06	1.9622e + 06
	Median	2.0562e + 06	2.0869e + 06
	Worst	2.1350e + 06	2.1904e + 06
	Mean	**2.0461e + 06**	2.0763e + 06
	Std	6.9452e + 04	8.4339e + 04
F13	Best	1.1077e + 08	1.0374e + 08
	Median	1.3137e + 08	1.3379e + 08
	Worst	1.6626e + 08	1.6147e + 08
	Mean	**1.3441e + 08**	1.3493e + 08
	Std	1.5571e + 07	1.9787e + 07
F14	Best	6.2729e + 09	6.0907e + 09
	Median	6.6919e + 09	6.8612e + 09
	Worst	8.0578e + 09	7.5727e + 09
	Mean	**6.7559e + 09**	6.9155e + 09
	Std	5.0437e + 08	4.8570e + 08
F15	Best	1.6416e + 04	1.6340e + 04
	Median	1.6534e + 04	1.6691e + 04
	Worst	1.6927e + 04	1.6892e + 04
	Mean	**1.6568e + 04**	1.6674e + 04
	Std	145.8575	145.5967
F16	Best	404.9208	407.5469
	Median	409.7903	409.6755
	Worst	410.9187	412.0824
	Mean	**409.1372**	409.8648
	Std	1.7926	1.4549
F17	Best	2.7638e + 06	2.8882e + 06
	Median	2.9555e + 06	3.0557e + 06
	Worst	3.1423e + 06	3.2906e + 06
	Mean	**2.9464e + 06**	3.0504e + 06
	Std	1.1326e + 05	1.4232e + 05
F18	Best	2.4614e + 09	2.5603e + 09
	Median	2.7451e + 09	2.7727e + 09
	Worst	3.1012e + 09	3.0197e + 09
	Mean	**2.7693e + 09**	2.8075e + 09
	Std	2.0060e + 08	1.4732e + 08
F19	Best	7.4615e + 06	8.8942e + 06
	Median	1.0164e + 07	1.0717e + 07
	Worst	1.3475e + 07	1.5390e + 07
	Mean	**1.0020e + 07**	1.1079e + 07
	Std	1.7817e + 06	1.8966e + 06
F20	Best	2.7733e + 09	2.5204e + 09
	Median	3.0091e + 09	3.0602e + 09
	Worst	3.2906e + 09	3.8450e + 09
	Mean	**3.0219e + 09**	3.1099e + 09
	Std	1.8120e + 08	3.8648e + 08

Bold values indicate the best mean value (i.e., average minimum)

For each algorithm, the mean fitness value, accuracy of testing set, number of selected channels are reported in Table 15. It can be seen that REHHO was able to reduce PLS complexity by using only 17 wavelengths. In terms of accuracy on the testing dataset, the best results were achieved by REHHO with 96.3%. Furthermore, the proposed REHHO was able to produce the best fitness value. The worst results have been reported by the BAT algorithm due to its lack of exploration ability as compared with other algorithms.

5 Conclusion

This work presents a novel REHHO algorithm that improves HHO by embedding several rules. The effectiveness of REHHO has been evaluated with a total of six standard high-dimensional functions ranging from 1000-D to 10,000-D, CEC'2010 large-scale benchmark, and the problem of NIR wavelength selection. Reported results indicated that REHHO was able to outperform HHO and other state-of-the-art optimization algorithms, including BAT, PSO, DE, AOA, and HHA. From the statistical analysis of the results, the t-test showed that REHHO significantly outperformed other algorithms with a 95% confidence level. As future work, REHHO could be applied for features selection problems and other real-world, large-scale problems.

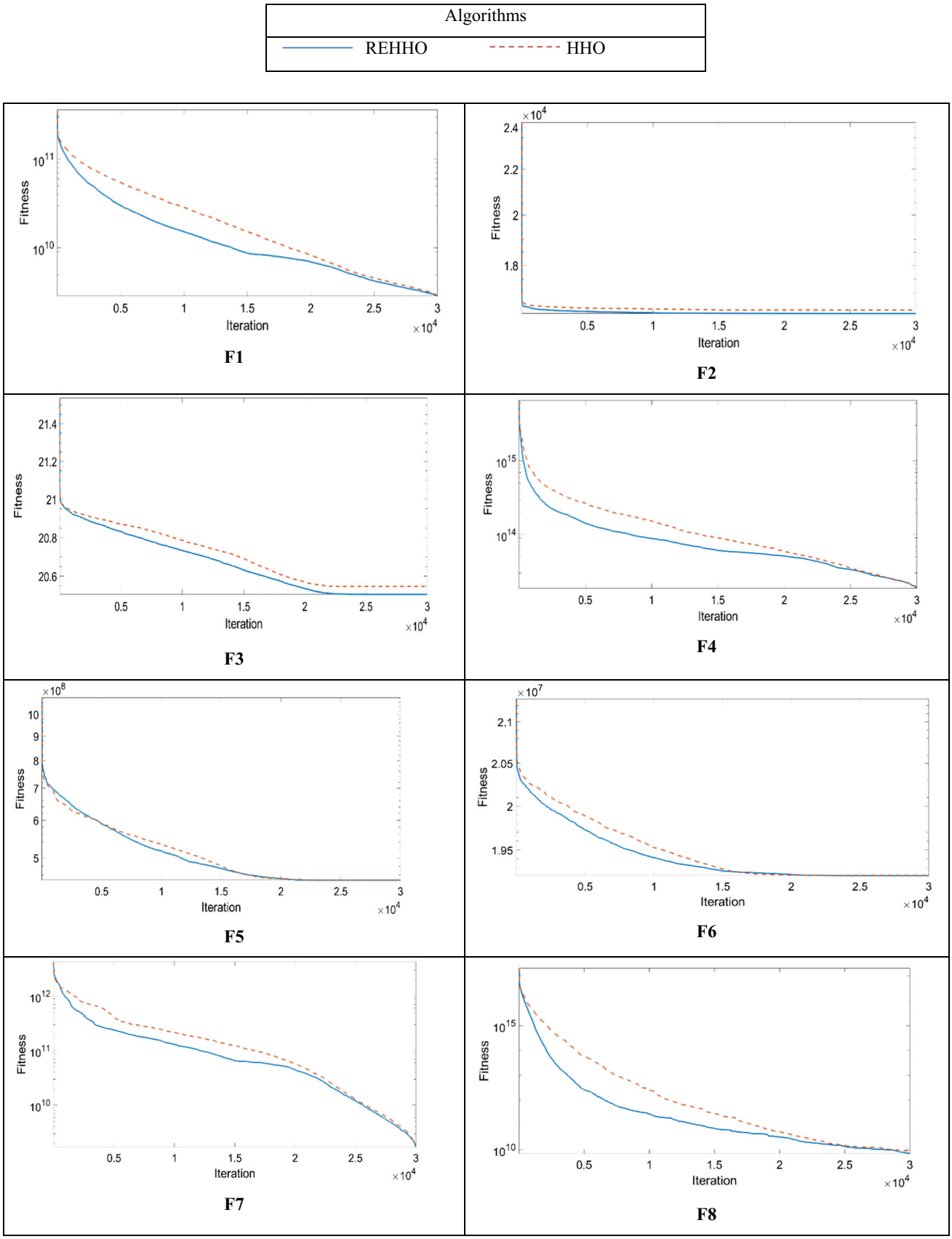


Fig. 13 The convergence curves for large-scale functions F1–F20

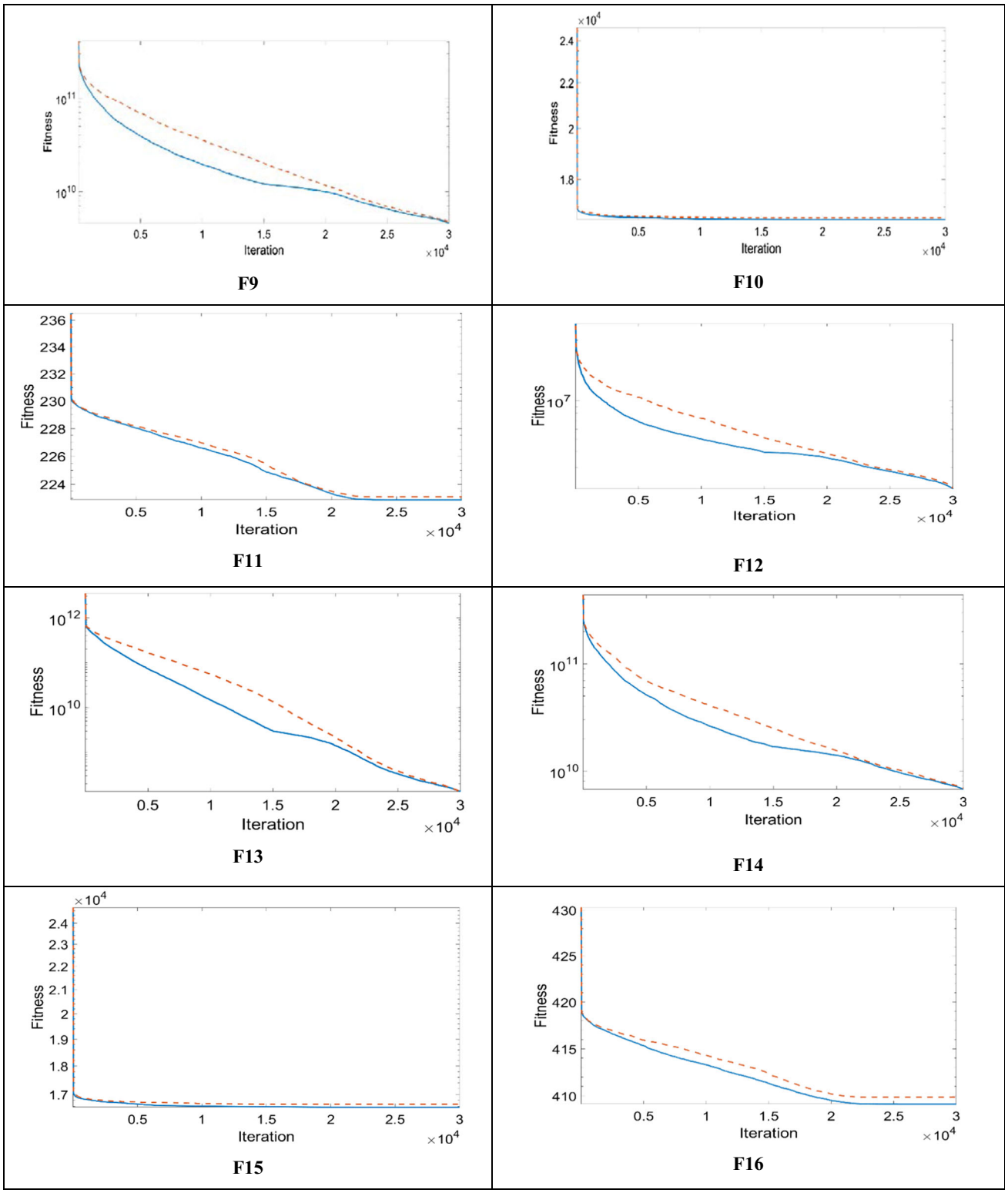


Fig. 13 continued

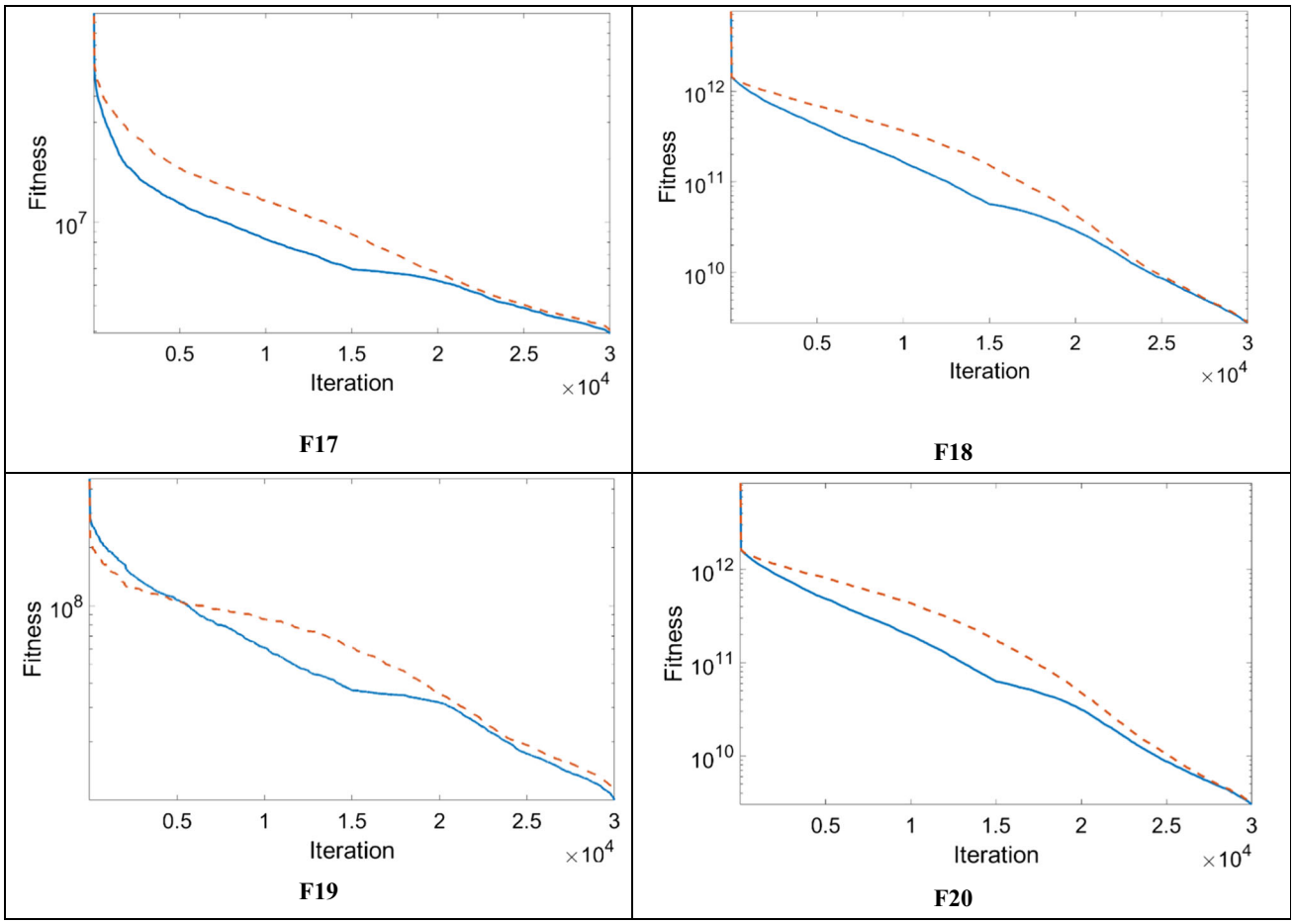


Fig. 13 continued

Table 10 Parameter settings

Method	Population	FEs	Parameters
BAT	30	$3*10^4$	Loudness = 0.5, Pulse rate = 0.5, Frequency minimum = 0, Frequency maximum = 2
PSO	30	$3*10^4$	c1 = 2.5 to 0.5, c2 = 0.5 to 2.5, w = 0.9 to 0.4
DE	30	$3*10^4$	Lower bound of scaling factor = 0.2, upper bound of scaling factor = 0.8, crossover probability = 0.2
AOA	30	$3*10^4$	MOP: 0.2–1, Alpha = 5, Mu = 0.499
HHA	30	$3 * 10^4$	hβ = 0.9, hγ = 0.5, sβ = 0.1, sγ = 0.2, iγ = 0.3, dα = 0.5, dβ = 0.2, dγ = 0.1, rδ = 0.1, and rγ = 0.05

Table 11 Results of 1000 − D CEC'2010 functions

Function	Fitness	Algorithm					
		REHHO	BAT	PSO	DE	AOA	HHA
F1	Best	2.6334e + 09	8.1920e + 11	7.2801e + 10	9.1436e + 09	1.7666e + 11	1.0932e + 11
	Median	2.8842e + 09	8.6848e + 11	8.1627e + 10	9.8334e + 09	1.8469e + 11	1.1305e + 11
	Worst	3.0811e + 09	8.8280e + 11	9.7007e + 10	1.0501e + 10	1.9057e + 11	1.2559e + 11
	Mean	**2.8583e + 09**	8.6279e + 11	8.3058e + 10	9.8575e + 09	1.8458e + 11	1.1615e + 11
	Std	1.3393e + 08	2.0195e + 10	8.1725e + 09	4.5750e + 08	5.7304e + 09	7.1356e + 0
F2	Best	1.5968e + 04	3.3933e + 04	1.8561e + 04	1.3561e + 04	1.6800e + 04	1.6718e + 04
	Median	1.6255e + 04	3.4126e + 04	1.9207e + 04	1.3724e + 04	1.6820e + 04	1.6781e + 04
	Worst	1.6685e + 04	3.4304e + 04	1.9749e + 04	1.3825e + 04	1.6840e + 04	1.7128e + 04
	Mean	1.6285e + 04	3.4128e + 04	1.9174e + 04	**1.3714e + 04**	1.6819e + 04	1.6847e + 04
	Std	221.9340	117.0192	413.7851	83.5222	16.8634	166.2090
F3	Best	20.2382	20.5806	21.2729	20.6231	20.9388	20.9921
	Median	20.5327	20.6414	21.3183	20.7270	20.9416	21.0046
	Worst	20.6923	20.6582	21.3342	20.9378	20.9454	21.0084
	Mean	**20.5073**	20.6323	21.3146	20.7673	20.9419	21.0031
	Std	0.1428	0.0265	0.0189	0.1058	0.0024	0.0065
F4	Best	1.2145e + 13	1.3747e + 16	2.2275e + 14	4.1787e + 14	8.8284e + 14	3.0634e + 14
	Median	1.8154e + 13	2.2254e + 16	2.6180e + 14	5.1197e + 14	1.0977e + 15	3.7229e + 14
	Worst	2.5972e + 13	2.7323e + 16	3.6314e + 14	6.0705e + 14	1.7835e + 15	4.0546e + 14
	Mean	**1.8575e + 13**	2.1821e + 16	2.7632e + 14	5.1684e + 14	1.2728e + 15	3.6432e + 14
	Std	3.7970e + 12	4.5174e + 15	4.7843e + 13	6.2056e + 13	4.4289e + 14	4.2287e + 13
F5	Best	4.1407e + 08	1.6805e + 09	3.9186e + 08	4.0647e + 08	4.7870e + 08	6.5631e + 08
	Median	4.5136e + 08	1.7357e + 09	4.2848e + 08	4.3649e + 08	6.0494e + 08	7.0156e + 08
	Worst	4.8347e + 08	1.7998e + 09	4.8385e + 08	4.4413e + 08	6.3414e + 08	7.1827e + 08
	Mean	4.4886e + 08	1.7385e + 09	4.3203e + 08	**4.3096e + 08**	5.8409e + 08	6.9519e + 08
	Std	2.5081e + 07	4.8769e + 07	2.3246e + 07	1.4678e + 07	6.4493e + 07	2.4010e + 07
F6	Best	1.9146e + 07	1.9982e + 07	1.2002e + 07	7.3976e + 04	1.9686e + 07	1.7861e + 07
	Median	1.9211e + 07	1.9982e + 07	1.3239e + 07	3.2774e + 05	1.9835e + 07	1.9947e + 07
	Worst	1.9276e + 07	2.0000e + 07	1.4555e + 07	1.4352e + 06	2.0073e + 07	2.0317e + 07
	Mean	1.9212e + 07	1.9987e + 07	1.3167e + 07	**4.9662e + 05**	1.9868e + 07	1.9248e + 07
	Std	3.9172e + 04	7.7142e + 03	7.2390e + 05	4.4267e + 05	1.7567e + 05	1.2673e + 06
F7	Best	1.3083e + 09	2.0432e + 13	4.6628e + 10	6.0392e + 10	5.9279e + 11	1.2764e + 11
	Median	1.7299e + 09	4.6735e + 13	6.2946e + 10	8.1327e + 10	1.2642e + 12	1.6919e + 11
	Worst	2.0005e + 09	5.1538e + 14	1.5931e + 11	1.0211e + 11	1.9517e + 12	2.5879e + 11
	Mean	**1.7102e + 09**	1.0400e + 14	7.3297e + 10	8.1724e + 10	1.3035e + 12	1.9109e + 11
	Std	2.4558e + 08	1.4957e + 14	3.2527e + 10	1.3181e + 10	5.5130e + 11	6.2565e + 10
F8	Best	4.6868e + 09	1.0082e + 18	3.0385e + 14	2.9923e + 08	2.6765e + 16	3.5186e + 15
	Median	6.2015e + 09	1.0712e + 18	4.6240e + 14	4.0148e + 08	3.0181e + 16	5.9064e + 15
	Worst	1.2757e + 10	1.1125e + 18	9.7152e + 14	5.0636e + 08	3.1786e + 16	8.3792e + 15
	Mean	6.9842e + 09	1.0662e + 18	5.6671e + 14	**3.9180e + 08**	2.9738e + 16	5.6005e + 15
	Std	2.5889e + 09	3.7550e + 16	2.5688e + 14	6.7384e + 07	1.9547e + 15	2.0058e + 15
F9	Best	4.3561e + 09	8.9819e + 11	6.6803e + 10	9.7943e + 10	2.0783e + 11	1.2854e + 11
	Median	4.5738e + 09	9.2339e + 11	7.5959e + 10	1.0200e + 11	2.2127e + 11	1.3227e + 11
	Worst	4.8799e + 09	9.3707e + 11	9.1954e + 10	1.1517e + 11	2.2824e + 11	1.4251e + 11
	Mean	**4.5915e + 09**	9.2298e + 11	7.6976e + 10	1.0320e + 11	2.2032e + 11	1.3331e + 11
	Std	1.6658e + 08	1.1524e + 10	8.8065e + 09	5.0416e + 09	8.2515e + 09	5.5935e + 09

Table 11 (continued)

Function	Fitness	Algorithm					
		REHHO	BAT	PSO	DE	AOA	HHA
F10	Best	1.6334e + 04	3.6158e + 04	1.8758e + 04	1.5455e + 04	1.6946e + 04	1.6843e + 04
	Median	1.6526e + 04	3.6548e + 04	1.9327e + 04	1.5655e + 04	1.7056e + 04	1.6944e + 04
	Worst	1.6897e + 04	3.6681e + 04	1.9866e + 04	1.5844e + 04	1.7207e + 04	1.7287e + 04
	Mean	1.6559e + 04	3.6479e + 04	1.9264e + 04	**1.5653e + 04**	1.7056e + 04	1.6998e + 04
	Std	169.1596	183.8412	432.9254	145.2213	101.6452	172.8607
F11	Best	221.5103	1.5455e + 04	223.7311	235.0548	228.7676	229.6113
	Median	222.7064	1.5655e + 04	227.7812	235.3412	228.8919	230.0050
	Worst	224.8432	1.5844e + 04	232.7328	235.6772	229.2603	230.3229
	Mean	**222.8913**	1.5653e + 04	228.0433	235.3281	228.9525	229.9827
	Std	1.0573	145.2213	2.6885	0.1939	0.1849	0.3204
F12	Best	1.9385e + 06	2.2378e + 09	8.0008e + 06	1.1730e + 07	1.1215e + 07	6.7499e + 06
	Median	2.0562e + 06	2.8304e + 09	9.0053e + 06	1.3104e + 07	1.4421e + 07	7.6562e + 06
	Worst	2.1350e + 06	4.1252e + 09	9.6380e + 06	1.4362e + 07	1.5800e + 07	8.1844e + 06
	Mean	**2.0461e + 06**	2.9619e + 09	8.9379e + 06	1.3093e + 07	1.3817e + 07	7.5167e + 06
	Std	6.9452e + 04	6.0028e + 08	6.0593e + 05	6.5046e + 05	1.8030e + 06	5.6543e + 05
F13	Best	1.1077e + 08	1.2709e + 13	4.3093e + 11	8.9269e + 10	6.6945e + 11	6.6277e + 11
	Median	1.3137e + 08	1.2883e + 13	5.6566e + 11	1.0461e + 11	6.7833e + 11	6.6784e + 11
	Worst	1.6626e + 08	1.2938e + 13	7.5955e + 11	1.1624e + 11	6.8313e + 11	6.8071e + 11
	Mean	**1.3441e + 08**	1.2866e + 13	5.6360e + 11	1.0477e + 11	6.7714e + 11	6.6928e + 11
	Std	1.5571e + 07	6.7440e + 10	9.8735e + 10	7.4351e + 09	5.2862e + 09	7.4645e + 09
F14	Best	6.2729e + 09	9.0032e + 11	5.8034e + 10	1.5575e + 11	2.3567e + 11	1.4172e + 11
	Median	6.6919e + 09	9.2654e + 11	7.0696e + 10	1.6093e + 11	2.3947e + 11	1.4433e + 11
	Worst	8.0578e + 09	9.3724e + 11	7.9735e + 10	1.6536e + 11	2.6200e + 11	1.6292e + 11
	Mean	**6.7559e + 09**	9.2284e + 11	7.0374e + 10	1.6057e + 11	2.4443e + 11	1.4773e + 11
	Std	5.0437e + 08	1.2867e + 10	6.6447e + 09	3.2319e + 09	1.0479e + 10	8.6053e + 09
F15	Best	1.6416e + 04	3.4679e + 04	1.8985e + 04	1.6483e + 04	1.6628e + 04	1.6793e + 04
	Median	1.6534e + 04	3.4896e + 04	1.9517e + 04	1.6919e + 04	1.6770e + 04	1.6952e + 04
	Worst	1.6927e + 04	3.5081e + 04	1.9770e + 04	1.7140e + 04	1.6945e + 04	1.7109e + 04
	Mean	**1.6568e + 04**	3.4896e + 04	1.9503e + 04	1.6870e + 04	1.6788e + 04	1.6962e + 04
	Std	145.8575	116.2722	217.1941	230.8942	115.2534	137.9735
F16	Best	404.9208	411.5244	420.8667	428.9731	416.3841	418.8719
	Median	409.7903	412.4078	423.6429	429.2859	416.8878	419.4299
	Worst	410.9187	413.3157	425.6089	429.5783	417.0327	419.5139
	Mean	**409.1372**	412.4006	423.4740	429.2628	416.7921	419.3333
	Std	1.7926	0.5727	1.6473	0.1800	0.2488	0.2649
F17	Best	2.7638e + 06	4.8625e + 09	1.3717e + 07	2.4597e + 07	3.2326e + 07	1.3599e + 07
	Median	2.9555e + 06	6.3958e + 09	1.5410e + 07	2.7451e + 07	4.1700e + 07	1.5993e + 07
	Worst	3.1423e + 06	7.7926e + 09	1.6907e + 07	3.0764e + 07	5.2378e + 07	1.6212e + 07
	Mean	**2.9464e + 06**	6.3036e + 09	1.5218e + 07	2.7800e + 07	4.0235e + 07	1.5448e + 07
	Std	1.1326e + 05	9.4376e + 08	9.9786e + 05	1.9342e + 06	8.1691e + 06	1.0856e + 06
F18	Best	2.4614e + 09	2.5509e + 13	2.6656e + 12	7.8896e + 11	1.4444e + 12	1.4367e + 12
	Median	2.7451e + 09	2.5680e + 13	2.9985e + 12	8.5376e + 11	1.4519e + 12	1.4437e + 12
	Worst	3.1012e + 09	2.5809e + 13	3.4511e + 12	9.0947e + 11	1.4572e + 12	1.4579e + 12
	Mean	**2.7693e + 09**	2.5679e + 13	3.0392e + 12	8.4958e + 11	1.4513e + 12 4.6140e + 09	1.4450e + 12
	Std	2.0060e + 08	1.1376e + 11	2.2922e + 11	3.9012e + 10		8.8728e + 09

Table 11 (continued)

Function	Fitness	Algorithm					
		REHHO	BAT	PSO	DE	AOA	HHA
F19	Best	7.4615e + 06	4.4343e + 10	2.5408e + 07	4.3615e + 07	4.5429e + 07	3.2616e + 07
	Median	1.0164e + 07	2.1524e + 11	3.0163e + 07	5.4726e + 07	5.7205e + 07	4.0733e + 07
	Worst	1.3475e + 07	1.0737e + 12	3.3165e + 07	6.0344e + 07	1.0451e + 08	4.2488e + 07
	Mean	**1.0020e + 07**	3.0652e + 11	3.0359e + 07	5.3491e + 07	6.9834e + 07	3.8587e + 07
	Std	1.7817e + 06	3.0170e + 11	2.2188e + 06	4.7828e + 06	2.5767e + 07	4.0426e + 06
F20	Best	2.7733e + 09	2.6682e + 13	2.6541e + 12	8.2017e + 11	1.6122e + 12	1.6243e + 12
	Median	3.0091e + 09	2.6791e + 13	3.4566e + 12	8.4314e + 11	1.6327e + 12	1.6272e + 12
	Worst	3.2906e + 09	2.6862e + 13	3.8441e + 12	9.2415e + 11	1.6350e + 12	1.6453e + 12
	Mean	**3.0219e + 09**	2.6773e + 13	3.3432e + 12	8.5322e + 11	1.6288e + 12	1.6318e + 12
	Std	1.8120e + 08	5.8015e + 10	3.6053e + 11	3.6249e + 10	9.4358e + 09	9.0083e + 09

Bold values indicate the best mean value (i.e., average minimum)

Table 12 The $p - values$ of statistical t − test

Function	HHO	REHHO	BAT	PSO	DE	AOA	HHA
F1	0.0049		0.0000	0.0000	0.0000	0.0000	0.0000
F2	0.0036		0.0000	0.0000	0.1725	0.0000	0.0000
F3	0.0015		0.0000	0.0000	0.0002	0.0000	0.0000
F4	0.0140		0.0000	0.0000	0.0000	0.0000	0.0000
F5	0.0000		0.0000	0.1370	**0.9669**	0.0000	0.0000
F6	0.0672		0.0000	0.0000	**0.9004**	0.0000	0.0000
F7	0.0021		0.0412	0.0000	0.0000	0.0000	0.0000
F8	0.0025		0.0000	0.0000	**0.3267**	0.0000	0.0000
F9	0.0004		0.0000	0.0000	0.0000	0.0000	0.0000
F10	0.0012		0.0000	0.0000	**0.4640**	0.0000	0.0000
F11	0.0048		0.0277	0.0000	0.0000	0.0000	0.0000
F12	0.0023		0.0000	0.0000	0.0000	0.0000	0.0000
F13	0.0008		0.0000	0.0000	0.0000	0.0000	0.0000
F14	0.0020		0.0000	0.0000	0.0000	0.0000	0.0000
F15	0.0012		0.0000	0.0000	0.0026	0.0000	0.0000
F16	0.0003		0.0000	0.0000	0.0000	0.0000	0.0000
F17	0.0008		0.0000	0.0000	0.0000	0.0000	0.0000
F18	0.0033		0.0000	0.0000	0.0000	0.0000	0.0000
F19	0.0021		0.0048	0.0000	0.0000	0.0000	0.0000
F20	0.0007		0.0000	0.0000	0.0000	0.0000	0.0000

Bold values indicate the best mean value (i.e., average minimum)

Table 13 The detailed settings of the PC

Item	Settings
CPU	i7 − 8700
Frequency	3.2 GHz
RAM	32 GB
Hard drive	512 GB SSD
Operating system	Windows 10
Language	MATLAB 2021a

Table 14 Computational time analysis

	HHO	REHHO
Time (second)	**324**	358

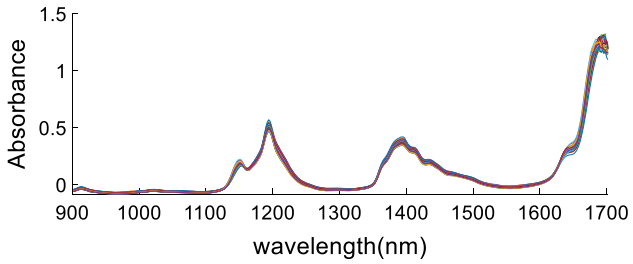


Fig. 14 NIR gasoline spectrum

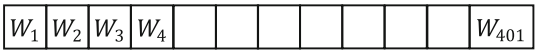


Fig. 15 Encoding scheme of NIR wavelength selection

Table 15 Results of NIR channels selection

Parameters/ algorithm	Number of selected wavelengths	Test accuracy (%)	Fitness
HHO	25	95.1	− 0.8895
REHHO	**17**	**96.3**	**− 0.8918**
BAT	169	92.5	− 0.8579
PSO	151	93.1	− 0.8623
DE	87	93.8	− 0.8601
AOA	55	94.5	− 0.8783
HHA	67	94.2	− 0.8721

Bold values indicate the best mean value (i.e., average minimum)

Acknowledgements This work is fully supported by Al-Imam Mohammad Ibn Saud Islamic University, Grant Scheme entitled "*Enhancement and Development of Smart Glasses System for Visually Impaired Persons by Using Intelligent System*", under Project Grant No. 18-11- 08-004.

Funding Al-Imam Mohammad Ibn Saud Islamic University, 18-11-08-004, Ali Sama.

Declarations

Conflict of interest Hussein Samma and Ali Salem Bin Sama declare that they have no conflict of interest.

References

1. Shi W, Chen W-N, Lin Y, Gu T, Kwong S, Zhang J (2017) An adaptive estimation of distribution algorithm for multipolicy insurance investment planning. IEEE Trans Evol Comput 23(1):1–14
2. Sun L, Lin L, Li H, Gen M (2019) Large scale flexible scheduling optimization by a distributed evolutionary algorithm. Comput Ind Eng 128:894–904
3. Min W, Liu J, Zhang S (2016) Network-regularized sparse logistic regression models for clinical risk prediction and biomarker discovery. IEEE/ACM Trans Comput Biol Bioinforma 15(3):944–953
4. Bellman R (1966) Dynamic programming. Science (80-) 153(3731):34–37
5. Ren Z, Liang Y, Wang M, Yang Y, Chen A (2021) An eigenspace divide-and-conquer approach for large-scale optimization. Appl Soft Comput 99:106911
6. Li D, Guo W, Lerch A, Li Y, Wang L, Wu Q (2021) An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization. Swarm Evol Comput 60:100789
7. Schoen F, Tigli L (2021) Efficient large scale global optimization through clustering-based population methods. Comput Oper Res 127:105165
8. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872
9. Abdullah JM, Ahmed T (2019) Fitness dependent optimizer: inspired by the bee swarming reproductive process. IEEE Access 7:43473–43486
10. Rahman CM, Rashid TA (2021) A new evolutionary algorithm: Learner performance based behavior algorithm. Egypt Inform J 22(2):213–223
11. Abdulhameed S, Rashid TA (2021) Child drawing development optimization algorithm based on child's cognitive development. Arab Sci Eng pp 1–15
12. Shamsaldin AS, Rashid TA, Al-Rashid Agha RA, Al-Salihi NK, Mohammadi M (2019) Donkey and smuggler optimization algorithm: a collaborative working approach to path finding. J Comput Des Eng 6(4):562–583
13. Rodriguez-Esparza E et al (2020) An efficient Harris hawks-inspired image segmentation method. Expert Syst Appl 155:113428
14. Hussain K, Neggaz N, Zhu W, Houssein EH (2021) An effcient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. Expert Syst Appl 114778
15. Mansoor M, Mirza AF, Ling Q (2020) Harris hawk optimization-based MPPT control for PV systems under partial shading conditions. J Clean Prod 274:122857
16. Essa FA, Abd Elaziz M, Elsheikh AH (2020) An enhanced productivity prediction model of active solar still using artificial neural network and Harris Hawks optimizer. Appl Therm Eng 170:115020
17. Yousri D, Babu TS, Fathy A (2020) Recent methodology based Harris Hawks optimizer for designing load frequency control incorporated in multi-interconnected renewable energy plants. Sustain. Energy Grids Networks 22:100352
18. Du P, Wang J, Hao Y, Niu T, Yang W (2020) A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2. 5 and PM10 forecasting. Appl Soft Comput 96:106620
19. Alamir MA (2021) An enhanced artificial neural network model using the Harris Hawks optimiser for predicting food liking in the presence of background noise. Appl Acoust 178:108022
20. Abd Elaziz M, Yousri D, Mirjalili S (2021) A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamics. Adv Eng Softw 154:102973
21. ElSayed SK, Elattar EE (2021) Hybrid Harris hawks optimization with sequential quadratic programming for optimal coordination of directional overcurrent relays incorporating distributed generation. Alexandria Eng J 60(2):2421–2433
22. Li C, Li J, Chen H, Heidari AA (2021) Memetic harris hawks optimization: developments and perspectives on project scheduling and QoS-aware web service composition. Expert Syst Appl 171:114529

23. Singh P, Prakash S (2020) Optimizing multiple ONUs placement in fiber-wireless (FiWi) access network using grasshopper and harris hawks optimization algorithms. Opt Fiber Technol 60:102357
24. Gölcük İ, Ozsoydan FB (2021) Quantum particles-enhanced multiple Harris Hawks swarms for dynamic optimization problems. Expert Syst Appl 167:114202
25. Hussain K, Neggaz N, Zhu W, Houssein EH (2021) An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. Expert Syst Appl 176:114778
26. Abdel-Basset M, Ding W, El-Shahat D (2021) A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. Artif Intell Rev 54(1):593–637
27. Abba SI et al (2021) Emerging Harris Hawks optimization based load demand forecasting and optimal sizing of stand-alone hybrid renewable energy systems—a case study of Kano and Abuja, Nigeria. Results Eng 12:100260
28. Ebrahim MA, Talat B, Saied EM (2021) Implementation of self-adaptive Harris Hawks optimization-based energy management scheme of fuel cell-based electric power system. Int J Hydrog Energy 46(29):15268–15287
29. Bandyopadhyay R, Basu A, Cuevas E, Sarkar R (2021) Harris Hawks optimisation with Simulated Annealing as a deep feature selection method for screening of COVID-19 CT-scans. Appl Soft Comput 111:107698
30. Suresh T, Brijet Z, Sheeba TB (2021) CMVHHO-DKMLC: a chaotic multi verse harris Hawks optimization (CMV-HHO) algorithm based deep kernel optimized machine learning classifier for medical diagnosis. Biomed Signal Process Control 70:103034
31. Mossa MA, Kamel OM, Sultan HM, Diab AAZ (2021) Parameter estimation of PEMFC model based on Harris Hawks' optimization and atom search optimization algorithms. Neural Comput Appl 33(11):5555–5570
32. Samma H, Lim CP, Saleh JM (2016) A new reinforcement learning-based memetic particle swarm optimizer. Appl Soft Comput 43:276–297
33. Abd Elaziz M, Heidari AA, Fujita H, Moayedi H (2020) A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems. Appl Soft Comput 95:106347
34. Wunnava A, Naik MK, Panda R, Jena B, Abraham A (2020) A differential evolutionary adaptive Harris hawks optimization for two dimensional practical Masi entropy-based multilevel image thresholding. J King Saud Univ Inf Sci
35. Yousri D, Mirjalili S, Machado JAT, Thanikanti SB, Fathy A (2021) Efficient fractional-order modified Harris hawks optimizer for proton exchange membrane fuel cell modeling. Eng Appl Artif Intell 100:104193
36. Chen H, Jiao S, Wang M, Heidari AA, Zhao X (2020) Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. J Clean Prod 244:118778
37. Liu Y et al (2020) Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models. Energy Convers Manag 223:113211
38. Qu C, He W, Peng X, Peng X (2020) Harris hawks optimization with information exchange. Appl Math Model 84:52–75
39. Li C, Li J, Chen H, Jin M, Ren H (2021) Enhanced Harris hawks optimization with multi-strategy for global optimization tasks. Expert Syst Appl 185:115499
40. Akdag O, Ates A, Yeroglu C (2021) Modification of Harris hawks optimization algorithm with random distribution functions for optimum power flow problem. Neural Comput Appl 33(6):1959–1985
41. Houssein EH, Neggaz N, Hosney ME, Mohamed WM, Hassaballah M (2021) Enhanced harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities. Neural Comput Appl 1–18
42. Krishna AB, Saxena S, Kamboj VK (2021) A novel statistical approach to numerical and multidisciplinary design optimization problems using pattern search inspired Harris hawks optimizer. Neural Comput Appl 33(12):7031–7072
43. Singh T (2020) A chaotic sequence-guided Harris hawks optimizer for data clustering. Neural Comput Appl 32:17789–17803
44. Tang K, Li X, Suganthan PN, Yang Z, Weise T (2010) Benchmark functions for the cec'2010 special session and competition on large-scale global optimization
45. Abualigah L, Diabat A, Mirjalili S, Abd-Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
46. MiarNaeimi F, Azizyan G, Rashki M (2021) Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems. Knowled.-Based Syst. 213:106711
47. Sheskin DJ (2003) Handbook of parametric and nonparametric statistical procedures, Chapman and Hall: CRC
48. Kalivas JH (1997) Two data sets of near infrared spectra. Chemom Intell Lab Syst 37(2):255–259