



A velocity-guided Harris hawks optimizer for function optimization and fault diagnosis of wind turbine

Wen Long^{1,2} · Jianjun Jiao³ · Ximing Liang⁴ · Ming Xu³ · Tiebin Wu⁵ · Mingzhu Tang⁶ · Shaohong Cai¹

Accepted: 4 July 2022

© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract

Harris hawks optimizer (HHO) is a relatively novel meta-heuristic approach that mimics the behavior of Harris hawk over the process of predated the rabbits. The simplicity and easy implementation of HHO have attracted extensive attention of many researchers. However, owing to its capability to balance between exploration and exploitation is weak, HHO suffers from low precision and premature convergence. To tackle these disadvantages, an improved HHO called VGHHO is proposed by embedding three modifications. Firstly, a novel modified position search equation in exploitation phase is designed by introducing velocity operator and inertia weight to guide the search process. Then, a nonlinear escaping energy parameter E based on cosine function is presented to achieve a good transition from exploration phase to exploitation phase. Thereafter, a refraction-opposition-based learning mechanism is introduced to generate the promising solutions and helps the swarm to flee from the local optimal solution. The performance of VGHHO is evaluated on 18 classic benchmarks, 30 latest benchmark tests from CEC2017, 21 benchmark feature selection problems, fault diagnosis problem of wind turbine and PV model parameter estimation problem, respectively. The simulation results indicate that VGHHO has higher solution quality and faster convergence speed than basic HHO and some well-known algorithms in the literature on most of the benchmark and real-world problems.

Keywords Harris hawks optimizer · Function optimization · Refraction-opposition learning · Wind turbine · Fault diagnosis

1 Introduction

Optimization can be defined as the process of choosing the best scheme from an available group of alternatives (Gupta et al. 2020a; Long et al. 2020a; Dhiman, 2021; Kumar and Dhiman 2021). By constructing an appropriate fitness function, many real-world applications in science research, management, and engineering can be formulated as function optimization problem (Long et al. 2020b, 2021a; Houssein et al. 2021c; Zhang et al. 2021).

✉ Shaohong Cai
gzcd_csh58@126.com

Extended author information available on the last page of the article

Many traditional gradient-based optimization methods have been developed to find solutions for optimization problems. However, in real-world applications, there is no guarantee that the fitness function is differentiable (Chatterjee 2021; Hassan et al. 2021; Vaishnav et al. 2021). The meta-heuristic optimization algorithms have some advantages such as gradient-free characteristics, easy implementation, and escape from local optima and have applied to tackle this type of problem successfully (Long et al. 2018a; Houssein et al. 2021a). Some of popular or recently proposed meta-heuristic optimization algorithms are particle swarm optimizer (PSO) (Kennedy and Eberhart 1995), differential evolution (DE) (Storn and Price 1997), polar bear optimization (PBO) (Polap and Wozniak 2017), cuckoo search (CS) (Gandomi et al. 2013), grey wolf optimizer (GWO) (Long et al. 2018b), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), spotted hyena optimizer (SHO) (Dhiman and Kumar 2017), sine cosine algorithm (SCA) (Mirjalili 2016), emperor penguin optimizer (EPO) (Dhiman and Kumar 2018), Harris hawks optimizer (HHO) (Heidari et al. 2019), seagull optimization algorithm (SOA) (Dhiman and Kumar 2019a), henry gas solubility optimization (HGSO) (Hashim et al. 2019), butterfly optimization algorithm (BOA) (Arora and Singh 2019), orientation search algorithm (OSA) (Dehghani et al. 2019), sooty tern optimization algorithm (STOA) (Dhiman and Kumar 2019b), marine predators algorithm (MPA) (Faramarzi et al. 2020), spring search algorithm (SSA) (Dehghani et al. 2020a), bald eagle search (BES) (Alsattar et al. 2020), darts game optimizer (DGO) (Dehghani et al. 2020b), Lévy flight distribution (LFD) (Houssein et al. 2020b), mayfly optimization algorithm (MOA) (Zervoudakis and Tsafarakis 2020), tunicate swarm algorithm (TSA) (Kaur et al. 2020), red fox optimization (RFO) (Polap and Wozniak 2021), slime mould algorithm (SMA) (Houssein et al. 2021b), chaos game optimization (CGO) (Talatahari and Azizi 2021), rat swarm optimizer (RSO) (Dhiman et al. 2021a), archimedes optimization algorithm (AOA) (Hashim et al. 2021), honey badger algorithm (HBA) (Hashim et al. 2022), and many others.

In this paper, we focused on the Harris hawks optimizer (HHO), which is firstly proposed by Heidari et al. (2019). HHO mimics the foraging behavior of hawks in nature. As a novel meta-heuristic algorithm, HHO can be easily implemented and has strong exploitation ability. The studies indicate that HHO has shown excellent performance on benchmark test optimization problems. Therefore, HHO has been widely utilized for dealing with real-world problems with satisfied results (Alabool et al. 2021). For instance, image thresholding (Elaziz et al. 2020; Wunnavu et al. 2020), photovoltaic models parameter extraction (Qais et al. 2020; Ridha et al. 2020), image segmentation (Rodríguez-Esparza et al. 2020), drug design and discovery (Houssein et al. 2020a), feature selection (Abdel-Basset et al. 2021), solar still productivity prediction (Essa et al. 2020), air pollution prediction (Du et al. 2020), PV array reconfiguration optimization (Yousri et al. 2020), data clustering (Singh 2020), neural network training (Ramalingam and Bakaran 2021), COVID-19 detection (Balaha et al. 2021), project scheduling and QoS-aware (Li et al. 2021), brain MRI segmentation (Bandyopadhyay et al. 2021b), slope stability prediction (Moayeddi et al. 2021), breast cancer detection (Kaur et al. 2021), Cardiomyopathy smart supervision (Ding et al. 2021), load frequency control (Abd Elaziz et al. 2021), chemical descriptors selection (Houssein et al. 2021d), vehicle suspension system optimization (Issa and Samn 2022), and many others.

Like other meta-heuristic optimization approaches, the conventional HHO still has some shortcomings such as unbalance of exploration and exploitation, poor solution quality, and easily fall into local optima, etc. Therefore, to mitigate these shortcomings, many HHO variants have been developed over the past two years. Some of them are summarized as follows. The escaping energy parameter E of HHO plays an important role in conversion from

exploration phase to the exploitation phase. Thus, investigating its escaping energy parameter E is one of research hot issues for the HHO algorithm. Several modified versions of the escaping energy parameter E have been suggested in the literature (Gupta et al. 2020b; Qu et al. 2020; Wunnavata et al. 2020; Yousri et al. 2020) to achieve a good conversion from exploration to exploitation. Although these HHO variants have performed well on low-dimensional benchmark problems, in some cases, especially on high-dimensional and/or complex multimodal problems, they may easily fall into local optima. In (Gupta et al. 2020b), the opposition-based learning (OBL) strategy was embedded into the basic HHO algorithm for escaping from the local optimal solution. The results indicated that the proposed approach obtains good performance on only benchmark test cases. Jiao et al. (2020) developed an enhanced HHO (EHHO) by introducing the orthogonal design (OD) operator and the general opposition learning mechanisms. In EHHO, the OD could improve the solution accuracy and the convergence performance, while the GOL could maintain the population diversity and the local search capability of HHO. Qu et al. (2020) put forward an improved HHO based on information exchange technique to solve numerical and engineering optimization problems with satisfied results. However, the results of the proposed method were only on benchmark test problems. In (Al-Betar et al. 2021), three different selection mechanisms (namely, tournament, proportional and linear rank-based approaches) were introduced into the basic HHO algorithm to improve its search performance. The experimental results on benchmark functions showed that the overall performance of HHO with tournament selection was better than other two selection strategies. In (Chen et al. 2020), a first powerful variant of HHO was proposed by combining the chaos, topological multi-population and differential evolution strategies to optimize the continuous functions. The comparison results indicated that the proposed technique had performed well on the selected benchmark tasks. Fan et al. (2020) developed a modified version of HHO by introducing quasi-reflection-based learning (QRBL) strategy. The proposed algorithm could effectively accelerate convergence and improve precision on benchmark test problems. In (Li et al. 2021), an enhanced HHO (called RLHHO) is proposed via incorporating three strategies (logarithmic spiral, opposition-based learning, and modified Rosenbrock method) for solving global optimization problems. The results revealed that RLHHO shows better performance than other compared algorithms on most problems. Arini et al. (2022) put forward an improved HHO with joint opposite selection (JOS) strategy for numerical optimization. In proposed approach, two opposition learning mechanisms such as selective leading opposition and dynamic opposite strategies are used to improve the performance of HHO. To utilize the advantages of different approaches, HHO were hybridized with other meta-heuristic algorithms such as salp swarm algorithm (SSA) (Elaziz et al. 2020), flower pollination algorithm (FPA) (Ridha et al. 2020), SCA (Kamboj et al. 2020; Hussain et al. 2021), multi-verse optimizer (MVO) (Ewees and Elaziz 2020), moth-flame optimization (MFO) (Elaziz et al. 2020), simulated annealing (SA) (Bandyopadhyay et al. 2021a), and so on. These hybrid variants obtain sufficiently satisfied performance, but fail to provide optimal values in some cases.

The above-mentioned variants have tried to enhance the overall optimization ability of the basic HHO by introducing some additional operators or mechanisms. However, no algorithm is perfect. From “No Free Lunch (NFL)” theorem (Wolpert and Macready 1997), there is no meta-heuristic approach best suited to solve all optimization problems. This theorem has made the area of intelligent optimization very active, which leads to improve existing algorithms and developing new approaches. Furthermore, the escaping energy parameter E of HHO based on a randomized policy cannot fully reflect the actual iterative search process, the transition ability from exploration phase to exploitation phase

is insufficient. At the same time, the conventional HHO is good at local exploitation while poor at global exploration, and it may easily fall into a local optima. Motivation of these considerations, this paper developed a novel HHO called velocity-guided Harris hawks optimizer (VGHHO) algorithm. More specifically, the primary contributions of this study are structured:

- (1) An improved variant of HHO (VGHHO) is proposed for solving function optimization, feature selection and fault diagnosis of wind turbine.
- (2) A velocity operator is embedded into the position search equation in exploitation stage of HHO that can guide the population search the potential region of solution space.
- (3) A modified escaping energy parameter based on cosine function is suggested in the HHO algorithm that can achieve a good transition from exploration to exploitation phases.
- (4) A refraction-opposition-based learning mechanism is introduced to enhance the diversity of VGHHO.
- (5) To investigate the comprehensive performance of VGHHO by using 18 classical benchmark functions, 30 latest benchmark functions from CEC2017, 21 benchmark feature selection problems, one practical wind turbine fault diagnosis problem, and PV model parameter estimation problem.

The remainder work of our study is arranged as follows. Section 2 briefly presents the conventional HHO. In Sect. 3, three modified strategies are explained and propose the framework of VGHHO. In Sect. 4, the feasibility of VGHHO is validated by using classical benchmark functions, and the comparisons are provided. In Sect. 5, the effectiveness of VGHHO is further verified on latest benchmark problems from CEC 2017. VGHHO is utilized for solving benchmark feature selection tests in Sect. 6. One practical fault diagnosis problem of wind turbine is solved by using VGHHO in Sect. 7. In Sect. 8, VGHHO is applied to solve the parameter estimation problem of PV model. Finally, Sect. 9 summarizes the conclusions and provides the future research directions.

2 The conventional HHO algorithm

In HHO, the hawks are considered the candidate individuals, while the rabbit denotes the best position found so far. Figure 1 shows the main stages of HHO.

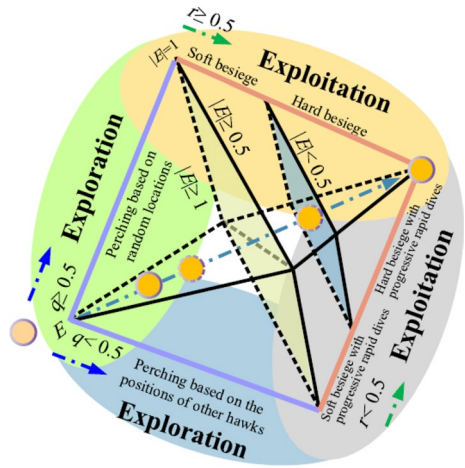
2.1 Exploration phase

In HHO, the exploration phase is performed by using two strategies:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \cdot |X_{rand}(t) - 2r_2 \cdot X(t)|, & q \geq 0.5 \\ (X_{rabbit}(t) - X_{mean}(t)) - r_3 \cdot (lb + r_4 \cdot (ub - lb)), & q < 0.5 \end{cases} \quad (1)$$

where X is the current position of hawk, t represents the number of iteration, X_{rand} indicates the randomly selected hawk from population, X_{rabbit} denotes the rabbit's position, r_1, r_2, r_3, r_4 , and q are random numbers, ub and lb are the left and right endpoints of the interval, and X_{mean} indicates the average position of hawks:

Fig. 1 Different stages of Harris hawks optimizer algorithm



$$X_{mean}(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (2)$$

where N indicates the swarm size.

2.2 Transition from exploration to exploitation

In the iterative process, the transition between exploration and exploitation is usually depended on the escaping energy coefficient (E), which is calculated by:

$$E = E_0 \times \left(2 - \frac{2t}{t_{max}} \right) \quad (3)$$

where E_0 denotes a random number range in $(-1, 1)$, and t_{max} represents the total iterative number.

2.3 Exploitation phase

After finding the target prey, hawks wait for a chance to attack the prey. However, the actual attack behavior is complicated; for example, the prey may be escaped from the enclosure. Therefore, four strategies are designed in exploitation phase to better mimic the attack characteristics of Harris hawks. Selecting the strategy is determined by both the escaping energy coefficient E and the random number r .

2.3.1 Hard besiege strategy

In HHO, when $r \geq 0.5$ and $|E| < 0.5$, the escaping energy is not enough and the prey has no chance to flee. So, the hawks will attack the prey via the hard besiege strategy as follows:

$$X(t + 1) = X_{rabbit}(t) - E \times |X_{rabbit}(t) - X(t)| \tag{4}$$

2.3.2 Soft besiege strategy

When $r \geq 0.5$ and $|E| \geq 0.5$, the escaping energy is enough but the prey has no chance to escape. So, the hawks will attack the prey by using the soft besiege way as:

$$X(t + 1) = (X_{rabbit}(t) - X(t)) - E \times |J \times X_{rabbit}(t) - X(t)| \tag{5}$$

where $J = 2(1 - r_5)$ indicates the jumping length of the prey and r_5 denotes a random number in $(0,1)$.

2.3.3 Soft besiege with progressive rapid dives

When $r < 0.5$ and $|E| \geq 0.5$, the escaping energy is very enough and the prey has chance to flee. In this phase, two steps are included. The first step is executed by the following equation:

$$Y = X_{rabbit}(t) - E \times |J \times X_{rabbit}(t) - X(t)| \tag{6}$$

After executing the first step, if the hawks' positions are not improved, the second step based on Lévy flight (LF) operator is executed by:

$$Z = Y + S \times LF(D) \tag{7}$$

where D represents the optimization problem's dimension, S denotes the random vector, and LF is the Lévy flight function as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \quad \sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta} \tag{8}$$

where u, v are the random constants of $LF, \beta = 1.5$.

In this phase, the positions of hawks are updated by the following equation:

$$X(t + 1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \tag{9}$$

2.3.4 Hard besiege with progressive rapid dives

When $r < 0.5$ and $|E| < 0.5$, the prey has a chance to flee but the energy after escaping is not enough. Hence, the positions of hawks are updated by:

$$X(t + 1) = \begin{cases} Y', & \text{if } F(Y') < F(X(t)) \\ Z', & \text{if } F(Z') < F(X(t)) \end{cases} \tag{10}$$

where Y' and Z' are calculated by the following equations:

$$Y' = X_{rabbit}(t) - E \times |J \times X_{rabbit}(t) - X_{mean}(t)| \tag{11}$$

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

Algorithm 1 introduces the step-wise explanation of the conventional HHO.

<p>Algorithm 1 Pseudo-code of the original HHO algorithm</p> <pre> 1: Initialize the parameters 2: Initialize the random population $X_i (i = 1, 2, \dots, N)$ 3: while ($t < t_{max}$) 4: Calculate the fitness value of each hawk 5: Find the best solution (X_{rabbit}) 6: for $i = 1$ to N 7: Calculate the escaping energy E using Eq. (3) 8: if $E \geq 1$ then 9: Execute exploration phase using Eq. (1) 10: else if $E < 1$ then 11: if $r \geq 0.5$ and $E < 0.5$ then 12: Execute hard besiege using Eq. (4) 13: else if $r \geq 0.5$ and $E \geq 0.5$ then 14: Execute soft besiege using Eq. (5) 15: else if $r < 0.5$ and $E \geq 0.5$ then 16: Execute soft besiege with progressive rapid dives using Eqs.(6)~(9) 17: else if $r < 0.5$ and $E < 0.5$ then 18: Execute hard besiege with progressive rapid dives using Eqs. (10)~(12) 19: end if 20: end if 21: end for 22: end while 23: Return X_{rabbit} </pre>
--

3 Proposed velocity-guided HHO algorithm

Similar to other meta-heuristic optimization methods, the conventional HHO cannot effectively explore the entire search space when solving complex optimization problems (Gupta et al. 2020b; Kamboj et al. 2020; Qu et al. 2020). Therefore, the objective of our work is to develop a new variant of HHO. It needs to be emphasized that our study does not change the framework of the conventional HHO algorithm, and improves HHO by embedding three strategies, i.e., velocity-guided position search equation, nonlinear escaping energy parameter, and refraction-opposition-based learning strategy.

3.1 Velocity-guided position search equation

The meta-heuristic algorithm is developed for achieving a good trade-off between exploration and exploitation over the iterative process. This trade-off is very important to the successful implementation of optimization method. The global exploration refers the capability to search for global optimum, while the local exploitation refers the capability to utilize the existing information to seek for better agents. The conventional HHO algorithm has shown unbalanced between global exploration and local exploitation on multimodal optimization problems (Kamboj et al. 2020). The main challenging issue of the original HHO

algorithm is that it may be trapped in local optima when handling multimodal optimization cases (Gupta et al. 2020b). The reason is the poor global exploration ability of HHO, while it is good at local exploitation capability. Moreover, as seen in Eqs. (4) and (5), the new candidate search agent is generated by conducting difference operation between the global best search solution (X_{rabbit}) and the current one. It may cause the algorithm to premature convergence. Therefore, for balancing between exploration and exploitation of HHO, modifying the position search equation in exploitation phase is one of the active research directions. Many HHO variants have been suggested to achieve this objective (Gupta et al. 2020b; Kamboj et al. 2020).

PSO is an efficient and effective meta-heuristic technique proposed by Kennedy and Eberhart (1995). Its idea is derived from the foraging behavior of birds. In PSO, each particle has its own position (X) and velocity (v). In the iterative process, the velocity and position of each particle are updated (Shi and Eberhart 1998):

$$v_i(t + 1) = w \times v_i(t) + c_1 \times r_1 \times (X_{pbest} - X_i(t)) + c_2 \times r_2 \times (X_{gbest} - X_i(t)) \quad (13)$$

$$X_i(t + 1) = X_i(t) + v_i(t + 1) \quad (14)$$

where v indicates the particles' velocity, w denotes the inertia weight, X_{pbest} represents the personal best position of particle, X denotes the particle's position, X_{gbest} represents the global best position, c_1 and c_2 are the coefficient factors, r_1 and r_2 are the random numbers.

Inspired by PSO, this paper designed a novel velocity-guided position search equation in exploitation phase and the detailed expressions are as follows.

In hard besiege strategy, the position is updated by

$$v(t + 1) = w \times v(t) + c_3 \times r_3 \times (X_{pbest} - X(t)) + X_{rabbit}(t) + E \times |X_{rabbit} - X(t)| \quad (15)$$

$$X(t + 1) = X(t) + v(t + 1) \quad (16)$$

In soft besiege strategy, the position is updated by

$$v(t + 1) = w \times v(t) + c_4 \times r_4 \times (X_{pbest} - X(t)) + (X_{rabbit}(t) - X(t)) + E \times |J \times X_{rabbit} - X(t)| \quad (17)$$

$$X(t + 1) = X(t) + v(t + 1) \quad (18)$$

where v denotes the velocity of each hawk, X_{pbest} is the personal best position of each hawk, c_3 and c_4 are the memory factors, r_3 and r_4 are the random numbers in $[0, 1]$, w represents the inertia weight and is calculated by:

$$w(t) = w_{initial} - (w_{initial} - w_{end}) \times \left| \frac{t}{1 + \sqrt{1 + t^2}} \right| \quad (19)$$

where $w_{initial}$ and w_{end} are respectively the initial and end values of w .

The first part on the right side of Eqs. (15) and (17) denotes the dynamical flight velocity of each hawk, which provides the necessary motivation for hawks to search throughout the solution space. Similar to PSO, the second term of Eqs. (15) and (17) is called as "cognitive" component, and represents the personal thinking of each hawk, which guides the hawk to move toward its own historical best position. Compared with the position search Eqs. (4) and (5) in the conventional HHO algorithm, the proposed position search Eqs. (15)

and (17) have three different features: (1) The velocity term is added for enhancing the global search capability of the position search equation; (2) The hawk learns not only from its own information but also from the knowledge of the other hawks in the population; and (3) The inertia weight w is introduced for dynamically balancing the optimization performance of HHO.

3.2 Nonlinear escaping energy parameter

The search efficiency of meta-heuristic algorithm depends on how well it achieves a good transition from exploration to exploitation over the optimization process. In the original HHO algorithm, we observed that the escaping energy coefficient E plays a crucial role in transiting between exploration and exploitation. A large value of the escaping energy parameter E (≥ 1) is in favor of the exploration capability, while a small value (< 1) is helpful to exploitation ability. Therefore, it is quite important to select the suitable values of the escaping energy parameter E for HHO. However, the escaping energy parameter E values of the conventional HHO are random, and the range of this randomness decreases from 2 to 0 over the iterative process. This escaping energy coefficient E has proved to be effective for some problems, but it is invalid in other cases (Gupta et al. 2020b; Qu et al. 2020). Due to the iterative search of HHO is highly nonlinear and quite complicated, the linearly decrease transition rule of E cannot truly reflect the actual optimization process. Thus, a potential research interest is to investigate the new transition parameter E rules in the HHO for achieving a good transition from global exploration to local exploitation. Many decrease nonlinearly strategies of the escaping energy parameter E have been suggested to achieve this goal (Gupta et al. 2020b; Qu et al. 2020; Yousri et al. 2020).

Different from previous proposed nonlinearly decrease strategies of E , this paper proposes a novel increase nonlinearly scheme of the escaping energy parameter E . The reasons are explained as follows. On the one hand, the population of hawks has a good diversity in the early phase of iteration search. The good diversity means that HHO has a powerful capability to explore throughout the search space. The main goal of this phase is to accelerate convergence (i.e., $E < 1$). On the other hand, in the later phase of the iterative search, HHO may be converged at a certain point in the search space, which the loss of population diversity. Maintaining the population diversity and escaping the local optimum are main purpose of this phase (i.e., $E \geq 1$). Therefore, the calculated formulation of the proposed parameter E is

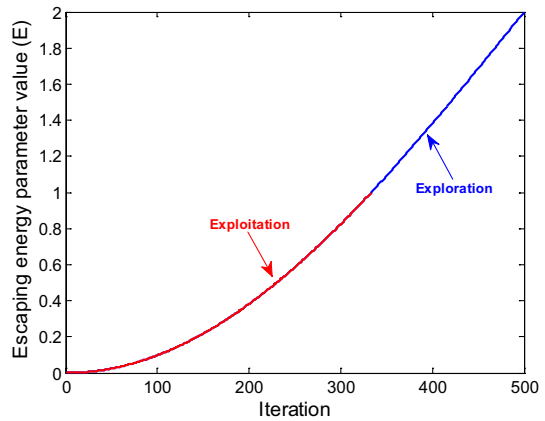
$$E(t) = E_{\max} - (E_{\max} - E_{\min}) \cdot \cos\left(\frac{t}{t_{\max}} \times \frac{\pi}{2}\right) \quad (20)$$

where E_{\max} and E_{\min} are respectively the max and min values of E .

Compared with the decrease linearly strategy of the escaping energy parameter E , the increase nonlinearly strategy described in Eq. (20) is to take a longer time for exploitation as compared to exploration. Figure 2 shows the graph of the proposed nonlinear escaping energy parameter E over the course of iteration process.

From Fig. 2 and Eq. (20), the values of the proposed nonlinear escaping energy parameter E are small ($E < 1$) in the early and middle phases of iteration, which demonstrates that it is concentrated on the local exploitation phase for a long time (about 67% of the total iteration numbers) as compared to the global exploration phase. Figure 2 also indicates that the proposed nonlinear escaping energy parameter E is large in the later phase, which is focused on exploration only for about 33% iterations.

Fig. 2 Proposed escaping energy parameter E



3.3 Refraction-opposition-based learning strategy

For meta-heuristic optimization algorithms, in the later stage of search, the other individuals in the population are attracted by the current best individual obtained so far, and gather towards it, thereby resulting in the loss of the population diversity and easing to fall into the local optima. This is the inherent shortcoming of meta-heuristic optimization algorithms. That is to say, enhancing the population diversity of meta-heuristic algorithms in the later stage of optimization is very important. To overcome this shortcoming, some additional strategies such as mutation operation (Gupta et al. 2020c), Lévy flight (Chawla and Duhan 2018), opposition-based learning (OBL) (Rahnamayan et al. 2008) and pinhole-imaging-based learning (Long et al. 2021b) are introduced in the meta-heuristic algorithms.

The OBL strategy is a candidate technique to effectively improve the optimization ability of meta-heuristic algorithm. However, premature convergence may be occurred in the later phase of algorithms. Thus, this paper improves the OBL strategy based on the refraction theory and proposes a novel refraction-opposition learning (ROL) mechanism for the global best solution. The detailed implementation process is provided in Fig. 3.

In Fig. 3, according to the principle of light refraction, the incidence light slants from medium 1 into medium 2 and its direction will be changed. The refraction angle is smaller than the incidence angle. Based on the Snell's Law (Griffiths 1998), the following mathematical formula is obtained by:

$$n = \frac{\sin \theta_1}{\sin \theta_2} = \frac{((a + b)/2 - X_{rabbit})/h}{(X'_{rabbit} - (a + b)/2)/h'} \quad (21)$$

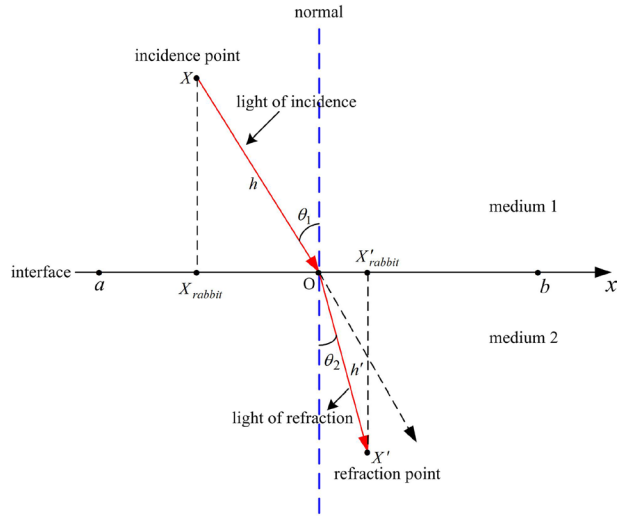
where n represents the refraction index, θ_1 and θ_2 are the incidence angle and refraction angle, a and b are the left and right endpoints of interval, X_{rabbit} is the global best individual, X'_{rabbit} is called as opposite individual of X_{rabbit} , h and h' are the distance of XO and $X'O$, respectively.

Let $k = h/h'$, Eq. (21) is modified as

$$kn = ((a + b)/2 - X_{rabbit}) / (X'_{rabbit} - (a + b)/2) \quad (22)$$

According to Eq. (22), the refraction learning opposition solution X'_{rabbit} is computed as

Fig. 3 Proposed refraction-opposition-based learning process



$$X'_{rabbit} = (a + b)/2 + (a + b)/2kn - X_{rabbit}/kn \tag{23}$$

When $k = 1$ and $n = 1$, Eq. (23) is reduced to

$$X'_{rabbit} = (a + b) - X_{rabbit} \tag{24}$$

where Eq. (24) is the mathematical formula of OBL in (Rahnamayan et al. 2008). That is to say, OBL [Eq. (24)] is a special case of the ROL strategy [Eq. (23)].

The Eq. (23) can be generalized to D -dimensional space:

$$X'_{rabbit,j} = (a_j + b_j)/2 + (a_j + b_j)/2kn - X_{rabbit,j}/kn \tag{25}$$

where a_j and b_j represent the left and right bounds of j th dimensional variable, $X_{rabbit,j}$ and $X'_{rabbit,j}$ are the j th dimension of X_{rabbit} and X'_{rabbit} . Algorithm 2 provides the steps of the ROL strategy on X_{rabbit} .

Algorithm 2 Refraction-opposition-based learning

```

1: Generate a random number  $r$  in  $[0,1]$ 
2: if  $r < 0.5$ 
3:   Generate an opposite solution  $X'_{rabbit}$  by using Eq. (25)
4:   Calculate the fitness value of  $X'_{rabbit}$ 
5:   if  $f(X'_{rabbit}) < f(X_{rabbit})$ 
6:      $X_{rabbit} = X'_{rabbit}$ 
7:   else
8:      $X_{rabbit} = X_{rabbit}$ 
9:   end if
10: endif

```

In summary, the flow chart of VGHHO algorithm is provided in Fig. 4.

3.4 Computational complexity analysis

The worst time complexity of VGHHO is calculate according to big- O notation using its pseudo codes. The step-wise description of the obtained complexity of VGHHO is as follows:

The population initialization of VGHHO requires $O(N \times D)$ time, where N is the population size and D is the dimension of the problem.

Calculate the fitness value of each hawk requires $O(N)$ time.

Selection of rabbit (best solution obtained so far) requires $O(N)$ time.

Position update mechanism of each hawk in VGHHO requires $O(N \times D)$ time.

Greedy selection technique in VGHHO requires $O(N)$ time.

Refraction-opposition-based learning strategy requires $O(1 \times D)$ time.

In summary, the total computational time of VGHHO is $O(N \times D \times t_{\max})$ for t_{\max} iterations.

4 Experiments on classical benchmark functions

4.1 Classical benchmark test functions

18 classical benchmark functions are applied for experiments. Table 1 lists the detailed characteristics of these functions. These functions are unimodal (f_1 - f_8) and multimodal (f_9 - f_{18}) functions. The unimodal problem has only one global best value, which are used for testing exploitation ability of metaheuristic approaches. Conversely, multimodal functions are usually utilized to investigate exploration ability of metaheuristic since it has many local optimal solutions (Long et al. 2018a). In Table 1, f_{\min} represents the theoretical optimum value.

Fig. 4 The flow chart of the proposed VGHHO algorithm

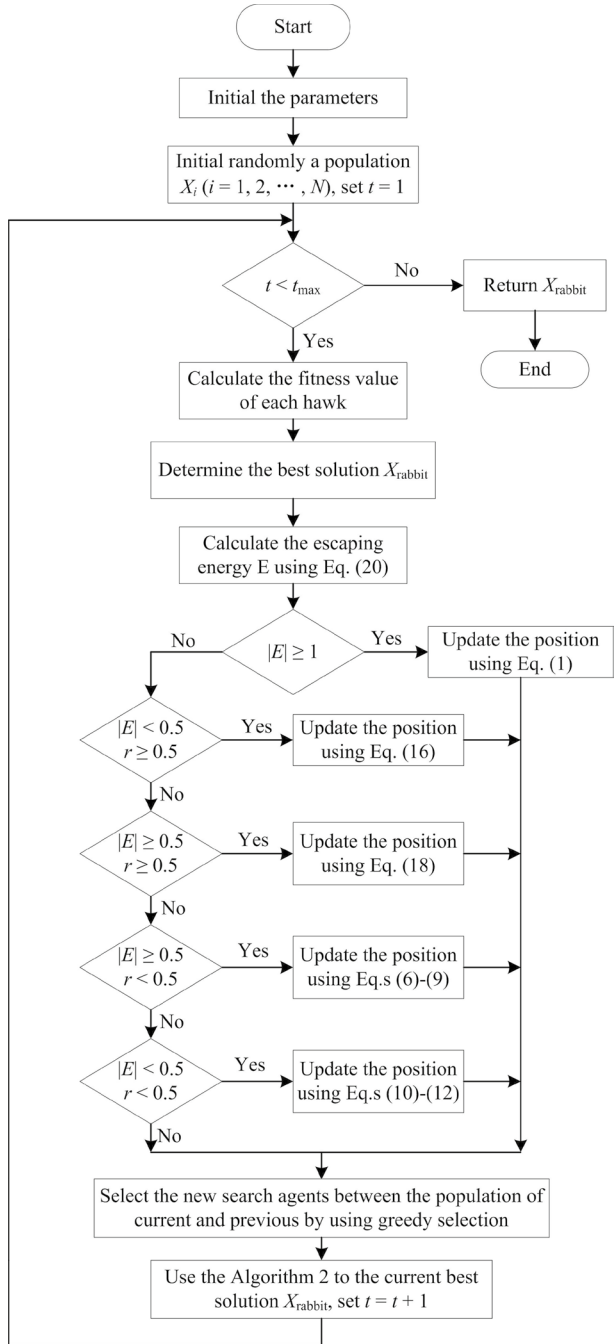


Table 1 The 18 classical benchmark test functions

Function equation	Domain	f_{\min}
$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0
$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	0
$f_3(x) = \max_i \{ x_i , 1 \leq x_i \leq D \}$	$[-100, 100]^D$	0
$f_4(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	0
$f_5(x) = \sum_{i=1}^D i x_i^4 + \text{random} [0, 1)$	$[-1.28, 1.28]^D$	0
$f_6(x) = \sum_{i=1}^D i x_i^2$	$[-10, 10]^D$	0
$f_7(x) = \sum_{i=1}^D x_i ^{(i+1)}$	$[-1, 1]^D$	0
$f_8(x) = \sum_{i=1}^D (10^6)^{(i-1)/(D-1)} x_i^2$	$[-100, 100]^D$	0
$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	0
$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32, 32]^D$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$	0
$f_{12}(x) = \sum_{i=1}^D x_i \cdot \sin(x_i) + 0.1 \cdot x_i $	$[-10, 10]^D$	0
$f_{13}(x) = \sin^2(\pi x_1) + \sum_{i=1}^{D-1} [x_i^2 \cdot (1 + 10 \sin^2(\pi x_{i+1})) + (x_i - 1)^2 \cdot \sin^2(2\pi x_i)]$	$[-10, 10]^D$	0
$f_{14}(x) = 1 - \cos \left(2\pi \sqrt{\sum_{i=1}^D x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$	$[-100, 100]^D$	0
$f_{15}(x) = 0.1 \left(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2 (1 + \sin^2(2\pi x_D)) \right)$	$[-5, 5]^D$	0
$f_{16}(x) = \sum_{i=1}^D (0.2x_i^2 + 0.1x_i^2 \cdot \sin(2x_i))$	$[-10, 10]^D$	0
$f_{17}(x) = \sum_{i=1}^{D-1} (x_i^2 + 2x_{i+1}^2)^{0.25} \cdot \left((\sin 50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1 \right)$	$[-10, 10]^D$	0
$f_{18}(x) = \sum_{i=1}^D x_i^6 \cdot \left(2 + \sin \frac{1}{x_i} \right)$	$[-1, 1]^D$	0

4.2 Comparison of VGHHO with other approaches on low-dimensional problems

The search capability of VGHHO is compared with other six meta-heuristic algorithms, i.e., BOA (Arora and Singh 2019), SOA (Dhiman and Kumar 2019a, b), HHO (Heidari et al. 2019), adaptive guided differential evolution (AGDE) (Mohamed et al. 2019), exploration-enhanced GWO (EEGWO) (Long et al. 2018a), and improved SCA (ISCA) (Long et al. 2019). BOA, SOA and HHO are the tradition meta-heuristic optimization techniques, while AGDE, EEGWO and ISCA are the state-of-the-art meta-heuristic algorithms. The population size and the total iterative numbers of VGHHO and other six optimization techniques are respectively fixed to 30 and 500 for ensuring fair of comparison. In VGHHO, $c_3 = c_4 = 2$, $w_{\text{initial}} = 1$, $w_{\text{end}} = 0$, $E_{\text{max}} = 2$, $E_{\text{min}} = 0$, $k = 5$, $n = 5$. In this experiment, the dimensions of functions in Table 1 are set to 30. The source codes of all approaches are implemented by MATLAB R2014a software. In order to reduce errors, each algorithm is independently run 30 trials for each function. Table 2 summaries the values of the average (Mean) and standard deviation (Std) of seven algorithms for 18 functions. In addition, the Friedman ranking test values based on ‘‘Mean’’ and ‘‘Std’’ results are also presented in Table 2. The best value of each function is highlighted in bold in Table 2.

Table 2 Comparisons of VGHHO and other six approaches for 18 classical problems with 30D in Table 1

Function	Index	BOA	SOA	HHO	AGDE	EEGW	ISCA	VGHHO
f_1	Mean	2.64E-11	4.17E-13	6.59E-102	4.87E-03	0	0	0
	Std	3.10E-12	4.36E-13	2.94E-102	3.63E-03	0	0	0
	Ranking	6	5	4	7	1	1	1
f_2	Mean	8.65E-09	1.30E-08	9.20E-50	1.09E-02	5.42E-240	8.99E-211	0
	Std	4.62E-09	1.14E-08	2.06E-49	5.10E-03	0	0	0
	Ranking	5	6	4	7	2	3	1
f_3	Mean	1.27E-08	3.44E-03	1.46E-48	8.73E+00	8.80E-228	1.36E-207	0
	Std	2.22E-09	4.09E-03	3.26E-48	8.09E-01	0	0	0
	Ranking	5	6	4	7	2	3	1
f_4	Mean	2.90E+01	2.81E+01	7.15E-02	6.19E+01	2.89E+01	2.89E+01	4.40E-03
	Std	2.58E-02	6.22E-01	6.67E-02	5.65E+01	2.61E-02	1.81E-02	4.43E-03
	Ranking	6	3	2	7	4	4	1
f_5	Mean	1.23E-03	2.08E-03	2.27E-04	7.60E-02	2.56E-05	4.24E-05	5.17E-06
	Std	5.41E-04	1.55E-03	1.66E-04	4.53E-02	3.85E-05	5.55E-05	1.15E-05
	Ranking	5	6	4	7	2	3	1
f_6	Mean	2.65E-11	8.30E-13	1.87E-107	3.55E-04	0	0	0
	Std	3.25E-12	6.74E-13	2.88E-107	1.46E-04	0	0	0
	Ranking	6	5	4	7	1	1	1
f_7	Mean	2.87E-13	1.41E-47	2.12E-129	3.10E-22	0	0	0
	Std	2.19E-13	3.06E-47	4.63E-129	1.38E-22	0	0	0
	Ranking	7	5	4	6	1	1	1
f_8	Mean	2.79E-11	3.71E-09	7.36E-96	8.33E+00	0	0	0
	Std	3.05E-12	5.66E-09	1.64E-95	6.30E+00	0	0	0
	Ranking	5	6	4	7	1	1	1
f_9	Mean	1.20E+02	2.96E+00	0	4.26E+01	0	0	0
	Std	1.11E+02	4.62E+00	0	3.95E+00	0	0	0
	Ranking	7	5	1	6	1	1	1
f_{10}	Mean	1.23E-08	2.00E+01	8.88E-16	1.53E-02	8.88E-16	8.88E-16	8.88E-16
	Std	1.40E-09	6.20E-04	0	1.83E-03	0	0	0
	Ranking	5	7	1	6	1	1	1

Table 2 (continued)

Function	Index	BOA	SOA	HHO	AGDE	EEGWO	ISCA	VGHHO
f_{11}	Mean	1.55E-11	2.61E-02	0	1.42E-02	0	0	0
	Std	1.05E-11	3.64E-02	0	9.85E-03	0	0	0
	Ranking	5	7	1	6	1	1	1
f_{12}	Mean	1.68E-09	4.24E-04	1.71E-57	2.57E-02	8.44E-239	2.03E-210	0
	Std	1.02E-09	7.29E-04	2.59E-57	3.38E-03	0	0	0
	Ranking	5	6	4	7	2	3	1
f_{13}	Mean	4.20E-12	1.31E-13	3.17E-99	3.30E-01	0	0	0
	Std	4.36E-12	1.19E-13	1.42E-98	5.71E-01	0	0	0
	Ranking	6	5	4	7	1	1	1
f_{14}	Mean	3.29E-01	1.42E-01	8.18E-49	1.10E+00	5.39E-63	0	0
	Std	4.26E-02	8.55E-02	1.82E-48	2.00E-01	4.52E-63	0	0
	Ranking	6	5	4	7	3	1	1
f_{15}	Mean	2.38E-12	3.47E-16	2.34E-105	2.71E-06	0	0	0
	Std	8.81E-13	4.68E-16	3.92E-105	1.39E-06	0	0	0
	Ranking	6	5	4	7	1	1	1
f_{16}	Mean	2.08E-11	1.30E-13	3.60E-100	9.69E-06	0	0	0
	Std	4.44E-12	2.81E-13	8.05E-100	7.84E-06	0	0	0
	Ranking	6	5	4	7	1	1	1
f_{17}	Mean	5.01E-05	1.46E-02	2.28E-27	4.96E+00	0	0	0
	Std	8.06E-05	5.10E-03	7.11E-27	8.73E-01	0	0	0
	Ranking	5	6	4	7	1	1	1
f_{18}	Mean	5.27E-10	3.53E-37	0	9.73E-16	0	0	0
	Std	4.76E-10	7.31E-37	0	8.57E-16	0	0	0
	Ranking	7	5	1	6	1	1	1
Average ranking		5.72	5.44	3.22	6.72	1.50	1.61	1.00
Total ranking		6	5	4	7	2	3	1

The best value of each function is highlighted in bold in the table

From Table 2, VGHHO gets the theoretical optima (0) for all the other problems except for f_4 , f_5 , and f_{10} . Compared with BOA, SOA and AGDE algorithms, VGHHO shows excellent performance on all the optimization tasks. With respect to HHO, VGHHO finds better values on fourteen cases. For f_9 – f_{11} , and f_{18} , the same values are

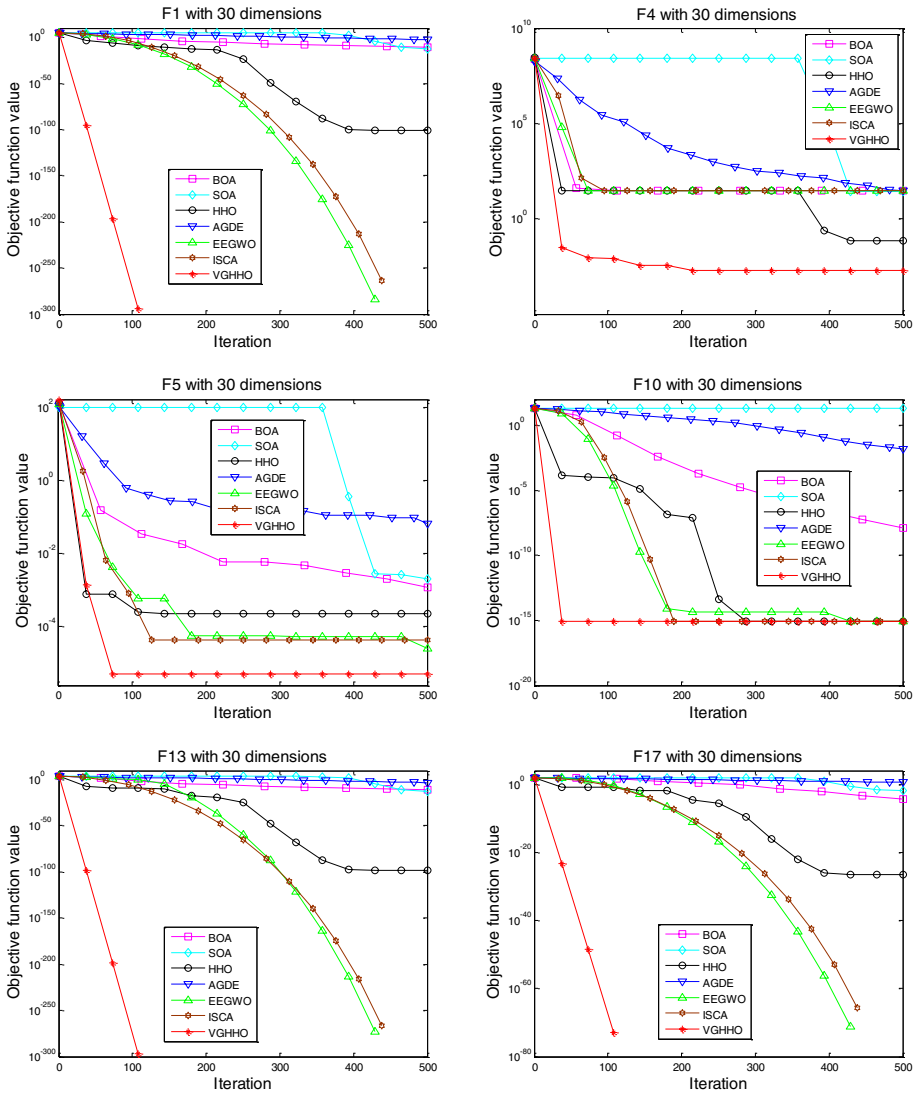


Fig. 5 The iterative curves of seven approaches for six representative 30D functions

obtained by two approaches. Compared to the EEGWO algorithm, VGHHO obtains excellent and same performance on twelve and six problems (i.e., f_2 - f_5 , f_{12} , and f_{14}), respectively. VGHHO is superior to ISCA on thirteen benchmark functions. In addition, two algorithms obtain similar values on five functions (i.e., f_2 - f_5 , and f_{12}). Regarding to the average Friedman ranking test results in Table 2, VGHHO obtains the first rank, followed by EEGWO, ISCA, HHO, SOA, BOA, and AGDE. To intuitively show the convergence performance, Fig. 5 plots the iterative curves of VGHHO and other

six techniques on six representative optimization cases with $30D$. As seen from Fig. 5, VGHHO obtains faster convergence speed than other approaches.

4.3 Scalability test

Furthermore, the VGHHO is used for dealing with the higher dimensions of the 18 classical benchmark tasks in Table 1 (i.e., $D = 100$ and 1000) to further investigate its scalability. For all algorithms, the same parameter settings are used as in Sect. 4.2. The mean and std results of VGHHO and other six algorithms on 18 problems with 100 and 1000 dimensions are outlined in Tables 3 and 4.

From Tables 3 and 4, VGHHO shows very excellent scalability for the search dimensions on most problems in Table 1. In other words, the comprehensive performance of VGHHO does not seriously deteriorate. It must be emphasized that a function with 1000 dimensions is very challenging for HHO. The reason is that it does not use the specific search strategies customized to deal with high-dimensional optimization problems. Compared with BOA and SOA, VGHHO obtains better performance on all the functions with 100 and 1000 dimensions. VGHHO provides better results than the conventional HHO algorithm on fourteen functions with high dimensionality. For most high dimensional functions, EEGWO, ISCA and VGHHO obtain similar results. From the average results of Friedman ranking test in Tables 3 and 4, the first rank is obtained by VGHHO, followed by EEGWO, ISCA, HHO, BOA, SOA, and AGDE. Furthermore, Figs. 6 and 7 plot the iterative curves of seven algorithms for six representative $100D$ and $1000D$ problems.

From Figs. 6 and 7, VGHHO shows very faster convergence speed than other techniques for six representative problems with high dimensionality. It can be seen from the above comparison results that VGHHO is a highly competitive meta-heuristic algorithm for solving high-dimensional optimization problems.

4.4 Statistical test analysis

In this section, Wilcoxon rank sum test with a significance level of 5% based on the average values of 30 times trials is used to investigate the difference between VGHHO and BOA, SOA, HHO, AGDE, EEGWO, and ISCA on 18 benchmark functions with 30, 100, and 1000 dimensions. Tables 5 shows the Wilcoxon rank sum test results between VGHHO and other six approaches. The “ p -value” is the significance determines whether the statistical hypothesis should be rejected.

The results in Table 5 indicate that VGHHO gets larger “ $R+$ ” results than “ $R-$ ” results on different cases. Furthermore, the p -values of VGHHO versus BOA, SOA, HHO, and AGDE on the classical benchmark problems with $30D$, $100D$, and $1000D$ are less than 0.05. That is to say, the performance difference of VGHHO and BOA, SOA, HHO, and AGDE is quite obvious.

4.5 Comparison of CPU runtime

In this subsection, the CPU runtime results of the basic HHO and the proposed VGHHO are introduced to investigate the computational complexity analysis and comparisons of two algorithms. In this experiment, 18 classical test functions in Table 1 are used. The

Table 3 Comparisons of VGHHO and six algorithms on 18 classical benchmark problems with 100 dimensions in Table 1

Function	Index	BOA	SOA	HHO	AGDE	EEGWO	ISCA	VGHHO
f_1	Mean	2.81E-11	2.88E-05	1.05E-96	3.67E+02	0	0	0
	Std	8.26E-13	1.61E-05	1.91E-96	7.32E+01	0	0	0
	Ranking	5	6	4	7	1	1	1
f_2	Mean	1.16E+47	7.31E-05	2.47E-48	1.58E+01	1.95E-227	2.68E-208	0
	Std	1.85E+47	5.90E-05	3.76E-48	1.17E+00	0	0	0
	Ranking	7	5	4	6	2	3	1
f_3	Mean	1.46E-08	6.88E+01	8.44E-48	5.88E+01	4.36E-220	2.13E-204	0
	Std	1.40E-09	2.19E+01	6.62E-48	1.45E+00	0	0	0
	Ranking	5	7	4	6	2	3	1
f_4	Mean	9.89E+01	9.81E+01	3.15E-01	7.65E+04	9.89E+01	9.90E+01	5.39E-02
	Std	2.95E-02	4.10E-01	4.41E-01	1.99E+04	2.78E-02	1.19E-02	8.93E-02
	Ranking	4	3	2	7	4	6	1
f_5	Mean	1.82E-03	1.35E-02	7.20E-04	9.83E-01	3.72E-05	7.46E-05	2.00E-05
	Std	1.88E-03	1.30E-02	5.92E-04	3.01E-01	3.26E-05	1.59E-04	2.86E-05
	Ranking	5	6	4	7	2	3	1
f_6	Mean	2.78E-11	2.24E-05	2.24E-100	1.60E+02	0	0	0
	Std	3.99E-12	1.73E-05	5.13E-100	3.62E+01	0	0	0
	Ranking	5	6	4	7	1	1	1
f_7	Mean	3.91E-13	4.14E-06	2.02E-124	2.40E-12	0	0	0
	Std	4.73E-13	5.90E-06	3.25E-124	9.92E-12	0	0	0
	Ranking	5	7	4	6	1	1	1
f_8	Mean	3.08E-11	1.69E-02	3.91E-93	6.59E+05	0	0	0
	Std	2.80E-12	3.79E-02	3.10E-93	1.21E+05	0	0	0
	Ranking	5	6	4	7	1	1	1
f_9	Mean	1.68E+02	5.08E+00	0	5.94E+02	0	0	0
	Std	3.75E+02	5.53E+00	0	1.18E+01	0	0	0
	Ranking	6	5	1	7	1	1	1
f_{10}	Mean	1.27E-08	2.00E+01	8.88E-16	4.29E+00	8.88E-16	8.88E-16	8.88E-16
	Std	1.68E-09	4.72E-04	0	1.71E-01	0	0	0
	Ranking	5	7	1	6	1	1	1

Table 3 (continued)

Function	Index	BOA	SOA	HHO	AGDE	EEGWO	ISCA	VGHHO
f_{11}	Mean	2.95E-11	2.53E-02	0	4.22E+00	0	0	0
	Std	1.65E-11	2.84E-02	0	1.19E+00	0	0	0
	Ranking	5	6	1	7	1	1	1
f_{12}	Mean	2.44E-09	1.11E-03	2.52E-49	3.71E+01	1.28E-228	2.14E-208	0
	Std	2.06E-09	8.77E-05	2.06E-49	5.48E+00	0	0	0
	Ranking	5	6	4	7	2	3	1
f_{13}	Mean	3.14E-11	8.95E+03	1.34E-94	2.53E+02	0	0	0
	Std	1.50E-11	1.32E+02	2.22E-94	6.51E+01	0	0	0
	Ranking	5	7	4	6	1	1	1
f_{14}	Mean	3.34E-01	3.40E-01	1.52E-47	8.37E+00	8.15E-52	0	0
	Std	4.55E-02	5.48E-02	1.06E-47	1.31E+00	1.62E-51	0	0
	Ranking	5	6	4	7	3	1	1
f_{15}	Mean	7.01E-12	9.76E-09	5.50E-99	4.16E-01	0	0	0
	Std	5.36E-12	8.24E-09	7.71E-99	5.14E-02	0	0	0
	Ranking	5	6	4	7	1	1	1
f_{16}	Mean	2.39E-11	5.85E-08	1.23E-98	2.38E+00	0	0	0
	Std	1.99E-12	5.72E-08	1.17E-98	1.11E+00	0	0	0
	Ranking	5	6	4	7	1	1	1
f_{17}	Mean	3.14E-06	4.19E-01	2.11E-25	1.17E+02	0	0	0
	Std	1.50E-06	2.27E-01	1.32E-25	4.02E+00	0	0	0
	Ranking	5	6	4	7	1	1	1
f_{18}	Mean	1.60E-10	5.20E-20	0	2.27E-04	0	0	0
	Std	1.63E-10	3.79E-20	0	2.67E-04	0	0	0
	Ranking	6	5	1	7	1	1	1
Average ranking		5.17	5.89	3.22	6.72	1.50	1.72	1.00
Total ranking		5	6	4	7	2	3	1

The best value of each function is highlighted in bold in the table

dimensions of these functions are set 30, 100, and 1000. Table 6 lists the average CPU runtime (in seconds) results of the basic HHO and VGHHO.

As can be seen in Table 6 that the CPU runtime of the basic HHO is less than VGHHO on all of the functions except for f_4 with $D = 1000$. This shows that the introduction of three operators (i.e., velocity-guided position search equation, nonlinear escaping energy parameter and refraction-opposition-based learning strategy) in HHO will increase the computational cost. However, this increased computational cost is

Table 4 Comparisons of VGHHO and six approaches for 18 classical benchmark problems with 1000 dimensions in Table 1

Function	Index	BOA	SOA	HHO	AGDE	EEGWO	ISCA	VGHHO
f_1	Mean	3.15E-11	8.02E-01	7.12E-95	5.17E+05	0	0	0
	Std	3.20E-12	2.80E-01	6.82E-95	4.78E+04	0	0	0
	Ranking	5	6	4	7	1	1	1
f_2	Mean	NA	1.62E-02	2.12E-47	NA	0	1.26E-207	0
	Std	NA	4.43E-03	4.73E-47	NA	0	0	0
	Ranking	6	5	4	6	1	3	1
f_3	Mean	1.50E-08	9.97E+01	8.92E-48	9.51E+01	2.55E-214	1.46E-203	0
	Std	8.56E-10	1.00E-01	6.99E-48	1.97E-01	0	0	0
	Ranking	5	7	4	6	2	3	1
f_4	Mean	9.99E+02	1.21E+04	8.34E+00	8.03E+08	9.99E+02	9.99E+02	9.39E-01
	Std	2.87E-02	1.04E+04	7.02E+00	3.51E+07	4.99E-02	1.66E-02	6.05E-01
	Ranking	3	6	2	7	3	3	1
f_5	Mean	2.33E-03	6.72E-01	1.23E-03	1.13E+04	9.13E-05	9.99E-05	4.80E-05
	Std	2.44E-04	2.45E-01	1.58E-03	2.39E+03	6.66E-05	1.91E-04	3.96E-05
	Ranking	5	6	4	7	2	3	1
f_6	Mean	3.22E-11	5.23E+00	2.56E-93	2.42E+06	0	0	0
	Std	3.07E-12	1.29E+00	4.37E-93	2.30E+05	0	0	0
	Ranking	5	6	4	7	1	1	1
f_7	Mean	7.50E-13	3.24E+00	4.85E-124	1.60E-06	0	0	0
	Std	6.08E-13	6.38E-01	5.33E-124	5.71E-07	0	0	0
	Ranking	5	7	4	6	1	1	1
f_8	Mean	3.56E-11	4.27E+04	5.97E-90	7.75E+09	0	0	0
	Std	2.44E-12	4.30E+04	6.41E-90	7.38E+08	0	0	0
	Ranking	5	6	4	7	1	1	1
f_9	Mean	1.89E+03	5.04E+01	0	1.15E+04	0	0	0
	Std	4.21E+03	7.96E+01	0	5.16E+02	0	0	0
	Ranking	6	5	1	7	1	1	1
f_{10}	Mean	1.28E-08	2.00E+01	8.88E-16	1.72E+01	8.88E-16	8.88E-16	8.88E-16
	Std	7.25E-10	4.46E-05	0	2.81E-01	0	0	0
	Ranking	5	7	1	6	1	1	1

Table 4 (continued)

Function	Index	BOA	SOA	HHO	AGDE	EEGWO	ISCA	VGHHO
f_{11}	Mean	3.12E-11	2.01E-01	0	4.22E+00	0	0	0
	Std	4.10E-12	1.63E-01	0	1.19E+00	0	0	0
	Ranking	5	6	1	7	1	1	1
f_{12}	Mean	4.90E-09	1.27E-01	2.89E-48	1.34E+03	2.61E-218	8.06E-208	0
	Std	2.17E-09	2.19E-01	2.37E-48	4.61E+01	0	0	0
	Ranking	5	6	4	7	2	3	1
f_{13}	Mean	3.59E-11	9.96E+04	5.73E-94	4.57E+04	0	0	0
	Std	5.94E-12	6.95E+01	4.60E-94	7.07E+03	0	0	0
	Ranking	5	7	4	6	1	1	1
f_{14}	Mean	3.60E-01	8.80E-01	3.53E-46	8.43E+01	3.15E-47	1.22E-58	0
	Std	4.30E-02	1.30E-01	3.27E-46	2.92E+00	6.52E-47	2.46E-58	0
	Ranking	5	6	4	7	3	2	1
f_{15}	Mean	1.97E-11	2.18E-04	2.24E-97	2.24E+02	0	0	0
	Std	1.31E-11	2.49E-04	1.96E-97	2.48E+01	0	0	0
	Ranking	5	6	4	7	1	1	1
f_{16}	Mean	2.88E-11	9.94E-03	7.85E-96	9.80E+02	0	0	0
	Std	2.25E-12	5.93E-03	6.53E-96	7.76E+01	0	0	0
	Ranking	5	6	4	7	1	1	1
f_{17}	Mean	5.28E-06	3.80E+03	3.72E-24	2.59E+03	0	0	0
	Std	6.03E-07	1.37E+00	4.10E-24	7.81E+01	0	0	0
	Ranking	5	7	4	6	1	1	1
f_{18}	Mean	6.87E-14	4.80E-03	3.89E-291	5.49E+00	0	0	0
	Std	3.02E-14	5.59E-03	0	1.05E+00	0	0	0
	Ranking	5	6	4	7	1	1	1
Average ranking	5.00	6.17	3.39	6.67	1.39	1.61	1.00	
Total ranking	5	6	4	7	2	3	1	

NA represents no available solution

The best value of each function is highlighted in bold in the table

acceptable. In addition, it should be pointed out that the optimization performance of VGHHO is significantly better than that of basic HHO.

4.6 Comparison with existing studies

In this section, VGHHO is also compared with three HHO variants such as Leader HHO (LHHO) (Naik et al. 2021), HHO with joint opposite selection (HHO-JOS) (Arini et al.

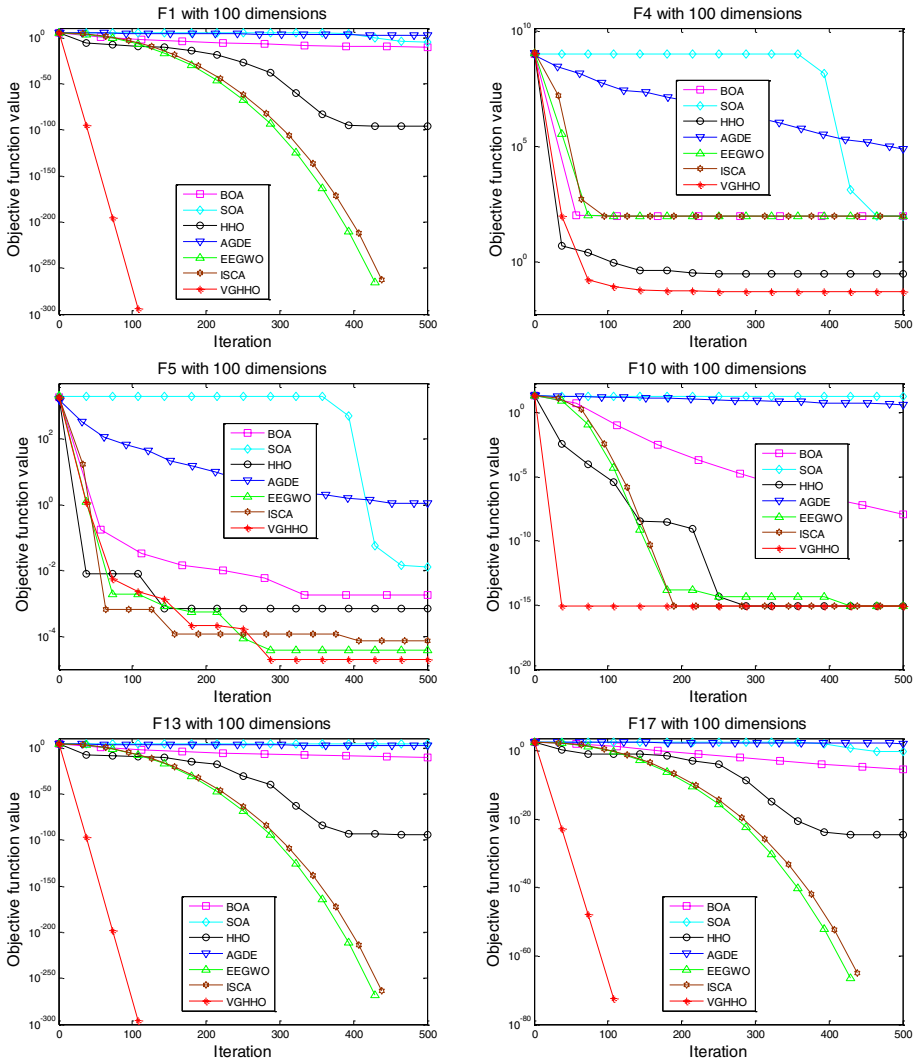


Fig. 6 The iterative curves of seven approaches for six representative 100D functions

2022) and modified HHO (m-HHO) (Gupta et al. 2020b). In this experiment, 18 benchmark test functions in Table 1 are used. The dimensions of all functions are set 30. The population size and the total iterative numbers of VGHHO and LHHO, HHO-JOS, m-HHO are respectively fixed to 30 and 500 for ensuring fair of comparison. Table 7 lists the “Mean” and “Std” results of four HHO variants on 18 classical benchmark functions with 30 dimensions. In addition, the Friedman ranking test results of four algorithms are also shown in Table 7.

From Table 7, compared with LHHO algorithm, VGHHO obtains better and similar results on fourteen and four functions (i.e., f_9 – f_{11} , f_{18}), respectively. The comprehensive performance of VGHHO is superior to HHO-JOS on thirteen benchmark functions. Furthermore, the similar results are obtained by two algorithms on five functions (i.e., f_7 ,

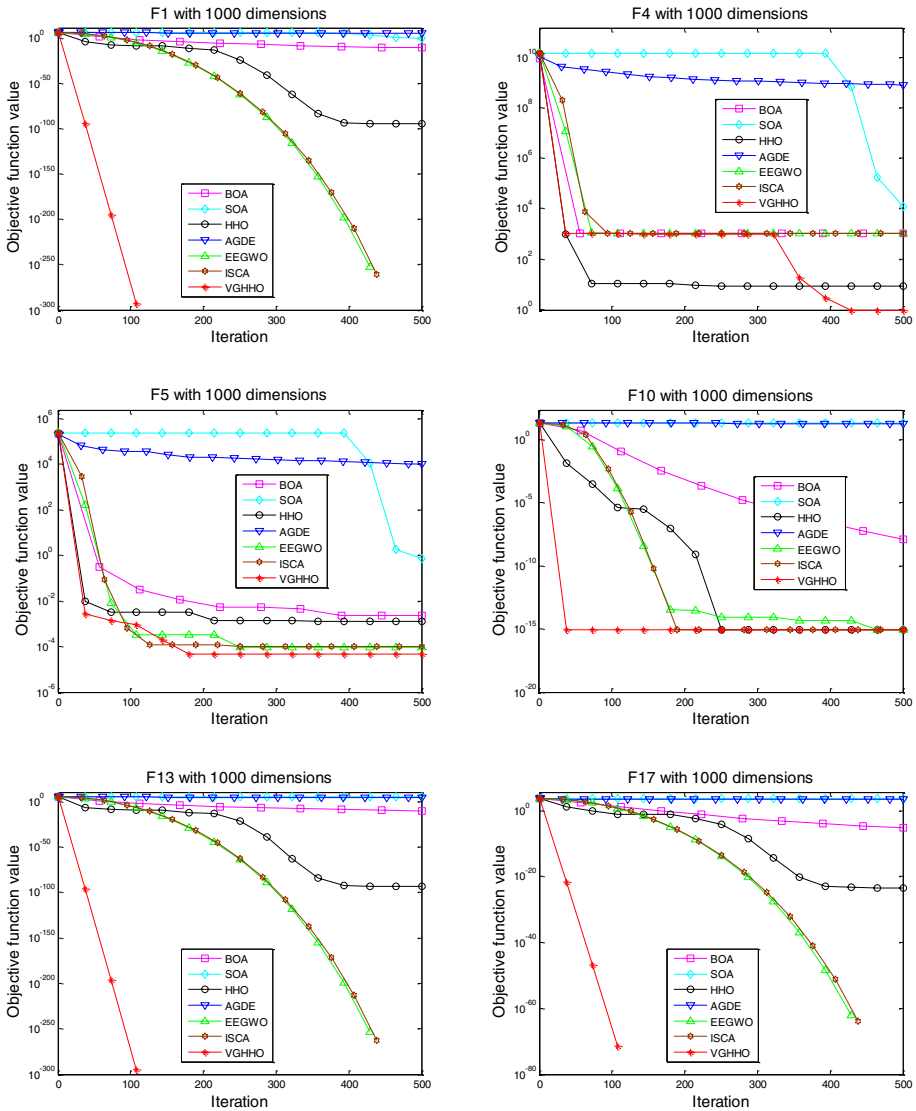


Fig. 7 The iterative curves of seven approaches for six representative 1000D functions

f_9 – f_{11} , f_{18}). VGHHO and mHHO achieve the similar optimization performance on all of functions except for two functions. For f_4 and f_5 , the better values are obtained by VGHHO.

4.7 Parameters sensitivity analysis

In the proposed VGHHO algorithm, there are eight parameters such as c_3 , c_4 , $w_{initial}$, w_{end} , E_{max} , E_{min} , k and n . Similar to PSO, the values of c_3 , c_4 , $w_{initial}$, w_{end} are set to 2, 2, 1, and 0, respectively. At the same time, it can be seen that these four parameters are not sensitive

Table 5 Wilcoxon rank sum test results between VGHHO and other six algorithms

Dimension	Algorithm	Better	Equal	Worst	R+	R-	p-value	$\alpha=0.05$
D=30	VGHHO versus BOA	18	0	0	171	0	1.1614E-05	Yes
	VGHHO versus SOA	18	0	0	171	0	6.2991E-06	Yes
	VGHHO versus HHO	14	4	0	166	5	0.0022	Yes
	VGHHO versus AGDE	18	0	0	171	0	5.4783E-07	Yes
	VGHHO versus EEGWO	6	12	0	132	39	0.2216	No
	VGHHO versus ISCA	5	13	0	125.5	45.5	0.3386	No
D=100	VGHHO versus BOA	18	0	0	171	0	9.9822E-06	Yes
	VGHHO versus SOA	18	0	0	171	0	1.7630E-06	Yes
	VGHHO versus HHO	14	4	0	166	5	0.0022	Yes
	VGHHO versus AGDE	18	0	0	171	0	1.9314E-07	Yes
	VGHHO versus EEGWO	6	12	0	132	39	0.2216	No
	VGHHO versus ISCA	5	13	0	125.5	45.5	0.3386	No
D=1000	VGHHO versus BOA	18	0	0	171	0	1.1614E-05	Yes
	VGHHO versus SOA	18	0	0	171	0	5.4783E-07	Yes
	VGHHO versus HHO	15	3	0	168	3	0.0010	Yes
	VGHHO versus AGDE	18	0	0	171	0	1.6174E-07	Yes
	VGHHO versus EEGWO	5	13	0	125.5	45.5	0.3386	No
	VGHHO versus ISCA	6	12	0	132	39	0.2216	No

Table 6 Comparisons of the average CPU runtime (in seconds) for two algorithms on 18 classical benchmark functions

Function	D=30		D=100		D=1000	
	HHO	VGHHO	HHO	VGHHO	HHO	VGHHO
f_1	1.0451	1.2615	1.1395	1.4812	1.7977	3.4857
f_2	0.9950	1.2931	1.0226	1.4901	1.8860	3.6052
f_3	1.0394	1.3604	1.1186	1.5773	2.2271	3.4668
f_4	1.2772	1.2776	1.4510	1.4655	3.1194	2.8505
f_5	1.0876	1.5038	1.5567	2.0870	7.6704	8.5438
f_6	1.0112	1.3339	1.0953	1.5364	1.9348	3.7386
f_7	1.1420	1.5690	1.4996	2.1386	5.7890	8.3051
f_8	1.1014	1.5692	1.4959	2.1844	7.0387	10.265
f_9	1.1616	1.4301	1.3188	1.6639	3.1826	4.2151
f_{10}	1.2473	1.5421	1.3563	1.7880	3.2324	4.1397
f_{11}	1.2148	1.4596	1.3398	1.7005	3.3207	4.2764
f_{12}	1.0722	1.3912	1.1648	1.5633	2.0695	3.7206
f_{13}	1.1760	1.5239	1.3851	1.8166	2.4075	4.8079
f_{14}	1.1374	1.4903	1.2937	1.7207	2.0801	3.8852
f_{15}	1.1023	1.4430	1.2356	1.6858	2.2886	4.8273
f_{16}	1.1698	1.4916	1.2758	1.7256	2.2983	4.8760
f_{17}	1.4360	1.9296	2.2013	2.7898	12.837	14.776
f_{18}	1.3162	1.7422	2.1196	2.7616	12.777	16.458

Table 7 Comparison results of four HHO variants on 18 classical benchmark functions with 30D

Function	LHHO		HHO-JOS		mHHO		VGHHO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	3.23E-149	5.60E-149	2.64E-261	0	0	0	0	0
f_2	1.37E-78	1.49E-78	7.26E-137	1.62E-136	0	0	0	0
f_3	5.26E-73	9.11E-73	6.57E-122	1.46E-121	0	0	0	0
f_4	7.23E-03	1.10E-02	4.82E-03	5.55E-03	7.68E-02	1.60E-01	4.40E-03	4.43E-03
f_5	1.28E-04	1.74E-04	1.21E-04	1.27E-04	7.47E-05	7.17E-05	5.17E-06	1.15E-05
f_6	3.64E-151	6.31E-151	6.58E-261	0	0	0	0	0
f_7	1.01E-201	0	0	0	0	0	0	0
f_8	1.30E-140	2.25E-140	2.22E-247	0	0	0	0	0
f_9	0	0	0	0	0	0	0	0
f_{10}	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0
f_{11}	0	0	0	0	0	0	0	0
f_{12}	3.24E-79	4.75E-79	8.99E-138	1.90E-137	0	0	0	0
f_{13}	4.15E-156	5.12E-156	8.80E-244	0	0	0	0	0
f_{14}	1.47E-72	2.53E-72	3.92E-130	8.76E-130	0	0	0	0
f_{15}	1.06E-151	1.80E-151	7.36E-265	0	0	0	0	0
f_{16}	1.26E-154	2.18E-154	6.66E-237	0	0	0	0	0
f_{17}	1.27E-39	1.15E-39	8.88E-74	1.97E-73	0	0	0	0
f_{18}	0	0	0	0	0	0	0	0
Average ranking	3.28		2.39		1.22		1.00	
Total ranking	4		3		2		1	

The best value of each function is highlighted in bold in the table

to affect the performance of the VGHHO algorithm by conducting many trials. The values of $E_{\max}=2$ and $E_{\min}=0$ are similar to the base HHO for a fair comparison. Furthermore, in VGHHO, k and n are the two critical parameters which help the population of algorithm to escape from the local optima. Therefore, in this subsection, a series of experiments are conducted to investigate the sensitivity of the parameters k and n . We manipulate the values of k and n while keeping the other parameters fixed. Table 8 lists the experimental results of different k and n values on 18 classical benchmark functions with $D=30$. The results related to $k=2.0$ and $n=2.0$ are also reported, along with those of new values in Table 8.

From Table 8, the comprehensive convergence accuracy of VGHHO with $k=2.0$ and $n=2.0$ is better than that of other values. Furthermore, we conducted several experiments VGHHO with the larger k ($k>2.0$), n ($n>2.0$) and compared with $k=2.0$, $n=2.0$. The comparison results showed that they exhibited similar performance on average. Therefore, considering all of the k and n values analyzed, it concluded that the setting of $k=2.0$ and $n=2.0$ for VGHHO is an appropriate choice.

Table 8 Comparison results of VGHHO using different k and n values on 18 functions with 30D

Function	$k=0.1, n=0.1$		$k=0.5, n=0.5$		$k=1.0, n=1.0$		$k=1.5, n=1.5$		$k=2.0, n=2.0$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	1.07E-92	2.40E-92	4.90E-185	0	1.91E-298	0	0	0	0	0
f_2	3.89E-55	8.28E-55	2.32E-98	2.14E-98	2.19E-142	3.21E-142	2.79E-235	0	0	0
f_3	1.43E-51	2.79E-51	8.55E-99	1.31E-98	6.82E-138	7.25E-138	5.88E-230	0	0	0
f_4	6.62E-01	3.97E-01	7.32E-02	1.12E-01	2.30E-02	4.28E-02	1.71E-02	2.17E-02	4.40E-03	4.43E-03
f_5	2.35E-03	1.83E-03	3.73E-04	7.04E-04	3.51E-04	2.53E-04	5.00E-05	4.61E-05	5.17E-06	1.15E-05
f_6	1.17E-98	2.61E-98	2.96E-196	0	3.30E-293	0	0	0	0	0
f_7	9.85E-135	1.78E-134	3.71E-262	0	0	0	0	0	0	0
f_8	7.44E-97	1.66E-96	3.94E-185	0	2.59E-288	0	0	0	0	0
f_9	0	0	0	0	0	0	0	0	0	0
f_{10}	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0
f_{11}	0	0	0	0	0	0	0	0	0	0
f_{12}	8.98E-58	2.01E-57	1.08E-101	9.61E-100	2.20E-149	7.15E-150	2.66E-237	0	0	0
f_{13}	1.10E-95	2.47E-95	4.00E-180	0	6.52E-288	0	0	0	0	0
f_{14}	3.64E-36	8.14E-36	2.37E-93	3.75E-93	3.72E-144	3.89E-144	0	0	0	0
f_{15}	1.12E-112	2.20E-112	2.48E-197	0	3.21E-292	0	0	0	0	0
f_{16}	5.10E-106	1.14E-105	6.92E-194	0	8.59E-285	0	0	0	0	0
f_{17}	2.06E-29	3.16E-29	2.79E-50	5.44E-50	3.36E-72	4.94E-72	0	0	0	0
f_{18}	9.13E-238	0	0	0	0	0	0	0	0	0

The best value of each function is highlighted in bold in the table

Table 9 Results of VGHHO and other seven optimization techniques for CEC 2017 problems

Problem	Index	BOA	SOA	HHO	HHO-JOS	EEGWO	AGDE	ISCA	VGHHO
F01	Mean	6.64E+10	8.34E+09	4.16E+07	1.38E+07	5.03E+09	1.42E-14	5.55E+09	5.25E+06
	Stddev	9.56E+09	1.72E+09	5.71E+06	9.46E+05	2.10E+08	1.00E-14	2.93E+08	1.86E+06
F02	Mean	-	-	-	-	-	-	-	-
	Stddev	-	-	-	-	-	-	-	-
F03	Mean	7.78E+04	3.47E+04	5.01E+04	5.46E+03	6.28E+04	5.68E-14	6.05E+04	5.76E+02
	Stddev	6.05E+03	9.16E+03	1.34E+04	2.66E+03	5.66E+03	0	7.37E+03	2.73E+01
F04	Mean	2.38E+04	3.17E+02	1.85E+02	1.15E+02	1.43E+04	1.17E+01	1.86E+04	1.04E+02
	Stddev	8.02E+03	1.09E+02	6.59E+01	1.38E+01	8.90E+01	2.62E+01	4.59E+03	1.95E+01
F05	Mean	4.23E+02	1.44E+02	2.60E+02	2.15E+02	3.96E+02	5.78E+01	4.15E+02	1.78E+02
	Stddev	2.11E+01	2.06E+01	3.83E+01	1.42E+01	3.66E+01	1.23E+01	1.86E+01	4.83E+01
F06	Mean	9.15E+01	3.34E+01	6.41E+01	4.11E+01	8.19E+01	1.14E-13	8.38E+01	4.92E+01
	Stddev	4.78E+00	9.36E+00	5.88E+00	2.22E+00	7.05E+00	0	3.31E+00	3.53E+00
F07	Mean	7.50E+02	3.71E+02	6.15E+02	5.49E+02	6.11E+02	1.03E+02	5.92E+02	3.94E+02
	Stddev	5.03E+01	3.36E+01	6.02E+01	1.01E+02	3.82E+01	5.19E+00	6.72E+01	3.46E+01
F08	Mean	3.41E+02	1.40E+02	1.80E+02	1.49E+02	3.24E+02	7.04E+01	3.39E+02	1.33E+02
	Stddev	2.38E+01	2.82E+01	2.38E+01	9.25E+00	1.53E+01	7.36E+00	1.08E+01	2.59E+01
F09	Mean	1.02E+04	3.54E+03	6.98E+03	5.36E+03	9.94E+03	0	8.46E+03	4.53E+03
	Stddev	1.28E+03	1.19E+03	1.26E+03	4.60E+02	7.95E+02	0	8.34E+02	1.31E+03
F10	Mean	7.78E+03	4.39E+03	5.70E+03	3.28E+03	6.58E+03	3.55E+03	7.25E+03	3.34E+03
	Stddev	4.41E+02	8.02E+02	9.21E+02	2.18E+02	8.82E+02	3.56E+02	3.62E+02	9.67E+02
F11	Mean	9.81E+03	3.82E+02	2.34E+02	1.50E+02	7.69E+03	2.75E+01	5.93E+03	9.48E+01
	Stddev	2.88E+03	1.21E+02	3.83E+01	4.79E+01	2.73E+03	2.61E+01	1.35E+03	7.65E+01
F12	Mean	1.60E+10	3.51E+08	5.11E+07	9.31E+06	1.55E+09	6.57E+03	1.37E+09	3.38E+06
	Stddev	4.38E+09	1.68E+08	5.67E+07	2.39E+06	3.18E+08	5.79E+03	3.06E+08	2.32E+06

Table 9 (continued)

Problem	Index	BOA	SOA	HHO	HHO-JOS	EEGWO	AGIDE	ISCA	VGHHO
F13	Mean	1.66E+10	1.11E+08	9.29E+05	3.34E+05	3.59E+09	3.74E+01	5.96E+09	1.79E+05
	Stddev	4.85E+09	3.30E+07	2.20E+05	7.83E+04	5.46E+08	1.69E+01	5.42E+09	1.96E+05
F14	Mean	2.62E+07	6.91E+04	1.67E+06	5.66E+04	3.29E+05	2.74E+01	3.35E+05	3.38E+04
	Stddev	2.81E+07	2.97E+04	1.37E+06	6.01E+04	2.36E+05	1.61E+01	2.53E+05	1.68E+03
F15	Mean	8.93E+08	2.41E+05	1.45E+05	2.09E+04	7.07E+07	2.85E+01	7.63E+07	1.67E+04
	Stddev	4.45E+08	2.66E+05	1.42E+05	9.56E+03	3.99E+07	3.06E+01	3.49E+07	3.82E+04
F16	Mean	5.08E+03	9.14E+02	2.12E+03	1.77E+03	4.88E+03	4.67E+02	4.58E+03	1.15E+03
	Stddev	1.37E+03	2.44E+02	3.96E+02	2.48E+02	7.79E+02	1.16E+02	8.42E+02	2.96E+02
F17	Mean	7.39E+03	3.83E+02	1.13E+03	9.08E+02	3.56E+03	6.49E+01	2.14E+03	5.11E+02
	Stddev	4.38E+03	2.20E+02	2.33E+02	6.88E+01	4.95E+02	1.04E+01	3.32E+02	2.27E+02
F18	Mean	2.51E+08	6.63E+05	4.04E+06	7.63E+05	1.16E+07	2.37E+03	3.53E+07	1.59E+05
	Stddev	1.56E+08	1.96E+06	3.51E+06	4.12E+05	1.81E+04	2.16E+03	2.10E+07	2.38E+05
F19	Mean	1.96E+09	5.23E+06	1.52E+06	3.42E+05	3.08E+08	9.78E+00	5.63E+08	2.17E+05
	Stddev	8.40E+08	6.72E+06	8.67E+05	1.41E+05	1.64E+08	5.11E+00	2.50E+08	1.04E+05
F20	Mean	1.10E+03	4.87E+02	5.95E+02	8.60E+02	1.04E+03	1.15E+02	7.98E+02	5.53E+02
	Stddev	1.20E+02	1.90E+02	1.59E+02	2.20E+02	1.62E+02	6.16E+01	1.30E+02	1.92E+02
F21	Mean	6.52E+02	3.66E+02	4.65E+02	4.06E+02	5.41E+02	2.68E+02	6.21E+02	2.93E+02
	Stddev	2.21E+01	3.36E+01	2.19E+01	2.32E+01	2.98E+01	9.09E+00	4.73E+01	7.62E+01
F22	Mean	6.59E+03	4.51E+03	5.52E+03	1.20E+02	5.24E+03	1.00E+02	7.09E+03	2.03E+03
	Stddev	1.12E+03	7.01E+02	7.73E+02	6.13E+01	1.16E+03	0.00E+00	2.62E+02	5.08E+02
F23	Mean	1.33E+03	4.95E+02	9.83E+02	6.47E+02	1.12E+03	4.10E+02	1.40E+03	6.12E+02
	Stddev	1.69E+02	2.30E+01	2.22E+02	7.46E+01	2.53E+02	1.04E+01	9.97E+01	7.96E+01
F24	Mean	1.50E+03	5.81E+02	1.12E+03	9.68E+02	1.46E+03	4.79E+02	1.38E+03	5.38E+02
	Stddev	9.01E+01	2.89E+01	1.69E+02	2.68E+01	1.90E+02	1.42E+01	2.04E+02	9.58E+01

Table 9 (continued)

Problem	Index	BOA	SOA	HHO	HHO-JOS	EEGWO	AGDE	ISCA	VGHHO
F25	Mean	4.62E+03	6.11E+02	4.50E+02	4.16E+02	2.96E+03	3.87E+02	3.20E+03	3.85E+02
	St.dev	6.79E+02	8.78E+01	2.02E+01	2.31E+01	4.21E+02	3.01E+01	7.26E+02	2.01E+01
F26	Mean	1.01E+04	2.57E+03	5.29E+03	3.07E+02	9.38E+03	1.58E+03	9.46E+03	1.36E+03
	St.dev	1.05E+03	2.39E+02	6.36E+02	2.16E+01	7.07E+02	8.94E+01	5.29E+02	5.42E+02
F27	Mean	1.73E+03	5.63E+02	8.43E+02	5.79E+02	2.68E+03	4.97E+02	2.12E+03	5.23E+02
	St.dev	3.77E+02	3.05E+01	1.35E+02	3.20E+01	2.99E+02	1.16E+01	2.24E+02	3.27E+01
F28	Mean	5.92E+03	2.35E+03	5.15E+02	4.55E+02	3.24E+03	3.41E+02	3.65E+03	4.46E+02
	St.dev	3.83E+02	1.51E+03	3.82E+01	2.58E+01	6.71E+02	5.66E+01	8.85E+02	3.81E+01
F29	Mean	1.18E+04	1.34E+03	1.99E+03	1.33E+03	4.42E+03	5.10E+02	4.56E+03	1.11E+03
	St.dev	1.29E+04	2.28E+02	6.22E+02	5.34E+02	4.93E+02	2.34E+01	5.40E+02	2.98E+02
F30	Mean	2.32E+09	1.21E+07	1.24E+07	7.67E+05	1.98E+09	5.11E+03	1.44E+09	1.04E+06
	St.dev	1.27E+09	8.65E+06	9.62E+06	3.03E+05	8.34E+08	2.97E+03	7.58E+08	7.29E+05

The best value of each function is highlighted in bold in the table

Fig. 8 Friedman test ranks of seven approaches for CEC 2017 problems

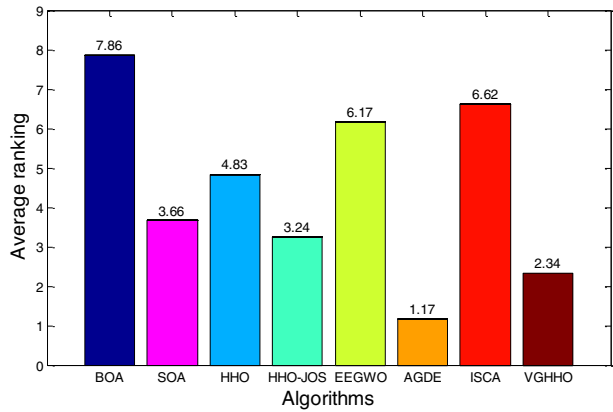


Table 10 Wilcoxon’s rank sum test values are obtained by VGHHO versus other seven approaches

Algorithm	Better	Equal	Worst	<i>R</i> +	<i>R</i> −	<i>p</i> -value	$\alpha=0.05$
VGHHO vs. BOA	29	0	0	435	0	0.0054	Yes
VGHHO vs. SOA	21	0	8	373	62	0.4368	No
VGHHO vs. HHO	29	0	0	435	0	0.2078	No
VGHHO vs. HHO-JOS	24	0	5	359	76	0.7795	No
VGHHO vs. EEGWO	29	0	0	435	0	0.0103	Yes
VGHHO vs. AGDE	3	0	26	26	409	1.08E-04	Yes
VGHHO vs. ISCA	29	0	0	435	0	0.0075	Yes

5 Experiments on latest benchmark problems for CEC 2017

The overall performance of VGHHO is further evaluated on 30 latest CEC 2017 benchmark problems, which are more complicated than the eighteen classical functions from Table 1. These problems are classified as four types, namely, unimodal (F01–F03), multimodal (F04–F10), hybrid (F11–F20), and composite cases (F21–F30) to investigate different search ability of algorithm (Awad et al. 2016). In this experiment, the dimensions of each problem are set to 30. The experimental results of VGHHO are compared with BOA, SOA, HHO, HHO-JOS, EEGWO, AGDE and ISCA, respectively. The terminate condition for seven algorithms is the same fitness evaluation maximum number (i.e., $10^4 \times D$, *D* denotes the problems’ dimension) to maintain fairness. The error value [$f(x) - f(x_0)$] of each algorithm is calculated over 30 independent trials on each problem. The $f(x)$ is a best value obtained by each method, while $f(x_0)$ is the theoretical best value on each problem. The comparison results of seven algorithms are listed in Table 9. Due to its unstable behavior, F02 has been deleted from test set and its result is not reported.

The optimization results of VGHHO in Table 9 are much better than BOA, HHO, EEGWO and ISCA on all the problems. Compared with SOA, VGHHO finds the better error values on 21 benchmark test problems. However, the better results are obtained by SOA on other eight problems (namely, F05–F07, F09, F16, F17, F20, and F23). With respect to the HHO-JOS algorithm, VGHHO gets the better and worse error values on

Table 11 The detailed information of 21 UCI datasets

Number	Dataset	Number of features	Number of samples
1	Breastcancer	9	699
2	BreastEW	30	569
3	Clean1	166	476
4	Clean2	166	6598
5	CongressEW	16	435
6	Exactly	13	1000
7	Exactly2	13	1000
8	HeartEW	13	270
9	IonosphereEW	34	351
10	KrvskpEW	36	3196
11	Lymphography	18	148
12	M-of-n	13	1000
13	PenglungEW	325	73
14	Semeion	265	1593
15	SonarEW	60	208
16	SpectEW	22	267
17	Tic-tac-toe	9	958
18	Vote	16	300
19	WaveformEW	40	5000
20	WineEW	13	178
21	Zoo	16	101

24 and five problems (i.e., F6, F10, F22, F26 and F30), respectively. AGDE is a state-of-the-art meta-heuristic algorithm for CEC 2017 benchmark test suit. AGDE obtains better results than VGHHO on 26 problems. However, the better values of three problems (i.e., F10, F25 and F26) are obtained by VGHHO algorithm. According to the non-parametric statistical Friedman ranking test results, Fig. 8 plots the column chart of the average ranking results of seven algorithms on 30 benchmark problems from CEC 2017. From Fig. 8, AGDE achieves the first rank, followed by VGHHO, HHO-JOS, SOA, HHO, EEGWO, ISCA and BOA.

Additionally, the Wilcoxon's rank sum statistical test based on "Mean" values in Table 9 is also used to investigate the difference between VGHHO and other seven optimization algorithms. Table 10 provides the statistical performance of VGHHO and other seven algorithms. From Table 10, VGHHO obtains higher "R+" than "R-" results on all of cases except for VGHHO versus AGDE. The p -values of VGHHO versus BOA, EEGWO, AGDE and ISCA are less than 0.05.

6 VGHHO for benchmark feature selection problems

In this section, the feasibility of VGHHO is further verified by dealing with feature selection (FS) problems. In fact, FS is a typical combinatorial optimization problem and its solution space is represented by binary values (Neggaz et al. 2020; Dhiman et al. 2021b;

Table 12 The average classification rates are obtained by seven algorithms on twenty-one selected datasets

Dataset	BOA	SOA	HHO	m-HHO	EEGW	ISCA	VGHHO
Breastcancer	0.9696	0.9710	0.9736	0.9760	0.9668	0.9768	0.9822
BreastEW	0.9572	0.9750	0.9587	0.9617	0.9483	0.9519	0.9736
Clean1	0.8938	0.9363	0.8421	0.9333	0.8683	0.8875	0.9366
Clean2	0.9685	0.9783	0.9722	0.9740	0.9669	0.9626	0.9759
CongressEW	0.9720	0.9720	0.9693	0.9770	0.9627	0.9604	0.9846
Exactly	0.7818	0.9061	0.7167	0.7117	0.7596	0.6909	0.9178
Exactly2	0.7646	0.7606	0.7433	0.7617	0.7606	0.7576	0.7697
HeartEW	0.8165	0.8277	0.8184	0.8272	0.8090	0.8352	0.8642
IonosphereEW	0.8986	0.9478	0.9190	0.9190	0.8898	0.9275	0.9333
KrvskpEW	0.9149	0.9813	0.9515	0.9588	0.8789	0.9408	0.9760
Lymphography	0.8611	0.9097	0.7701	0.9080	0.8681	0.8750	0.9136
M-of-n	0.8242	0.9586	0.8550	0.9067	0.8020	0.8858	0.9200
PenglungEW	0.9028	0.9583	0.9286	0.9286	0.8750	0.9167	0.9583
Semeion	0.9708	0.9853	0.9811	0.9895	0.9695	0.9803	0.9860
SonarEW	0.9069	0.9461	0.9187	0.9593	0.8922	0.8529	0.9293
SpectEW	0.8446	0.8598	0.8930	0.8868	0.8485	0.8788	0.8977
Tic-tac-toe	0.7384	0.7700	0.7801	0.7853	0.7584	0.7690	0.8083
Vote	0.9562	0.9596	1.0000	1.0000	0.9360	0.9528	1.0000
WaveformEW	0.7400	0.7998	0.7440	0.7703	0.7384	0.7331	0.7739
WineEW	0.9712	0.9827	0.9333	1.0000	0.9712	0.9712	1.0000
Zoo	0.9596	0.9596	0.9500	0.9833	0.9596	0.8990	0.9866
Average ranking	5.17	2.60	4.57	2.83	6.07	5.24	1.52
Total ranking	5	2	4	3	7	6	1

The best value of each function is highlighted in bold in the table

Hussain et al. 2021). However, VGHHO is a continuous version optimization technique which needs to transform it from continuous space into binary one when solving FS problems. One of the easiest handle ways is to introduce a transfer function (Tubishat et al. 2020). The biggest characteristic of this way is not to change the framework of VGHHO. In this paper, the following S-shaped transfer function is used:

$$T(x) = \frac{1}{1 + e^{-\tau x}} \quad (26)$$

where τ is a constant number.

In our experiment, twenty-one feature selection benchmark datasets from UCI are used. These datasets have been widely used to verify the optimization performance of meta-heuristic algorithms and their detailed information are shown in Table 11.

For twenty-one datasets, the wrapper technique and k -Nearest Neighbors (KNN) classifier ($k=3$) are combined to use. The number of samples of each dataset is randomly classified as two groups, namely, 80% is utilized for training while 20% is used for testing. In this experiment, the same population scale ($N=10$) and maximum iterative number ($t_{\max}=100$) for all the approaches are used to obtain a fair comparison. Each algorithm runs 30 independently trials on each dataset. Table 12 provides the average classification accuracy for

Table 13 The average feature numbers are obtained by seven algorithms on twenty-one selected datasets

Dataset	BOA	SOA	HHO	m-HHO	EEGWO	ISCA	VGHHO
Breastcancer	5.0000	3.3333	5.6667	4.0000	5.3333	5.3333	3.6667
BreastEW	14.667	9.6667	16.000	12.000	16.000	16.000	12.000
Clean1	73.667	36.000	76.667	69.000	79.667	81.667	63.000
Clean2	68.667	44.333	89.667	80.667	88.667	85.333	39.333
CongressEW	4.6667	2.0000	8.3333	3.3333	7.3333	2.0000	3.0000
Exactly	5.0000	5.6667	9.3333	4.3333	3.3333	1.0000	1.0000
Exactly2	1.0000	2.3333	5.3333	3.0000	1.0000	1.0000	1.0000
HeartEW	7.3333	4.3333	8.6667	6.0000	8.3333	3.6667	4.3333
IonosphereEW	11.333	4.6667	17.333	4.3333	16.000	3.6667	3.3333
KrvskpEW	19.333	14.333	30.333	29.333	20.667	15.667	14.667
Lymphography	7.6667	4.3333	11.333	6.0000	10.333	5.6667	4.0000
M-of-n	8.0000	6.0000	13.000	8.3333	7.3333	5.6667	7.6667
PenglungEW	55.667	14.333	157.00	37.000	158.00	141.33	22.000
Semeion	112.67	49.333	132.67	111.33	123.00	133.33	102.67
SonarEW	25.667	13.667	29.000	23.333	28.000	27.333	24.667
SpectEW	10.667	4.3333	14.000	8.3333	9.6667	7.3333	7.0000
Tic-tac-toe	6.0000	4.6667	6.6667	6.6667	6.3333	4.3333	4.3333
Vote	6.3333	3.3333	7.3333	1.3333	7.0000	2.3333	2.0000
WaveformEW	23.333	18.667	34.667	21.000	20.000	8.0000	18.333
WineEW	6.3333	5.0000	4.3333	5.6667	8.0000	4.0000	4.0000
Zoo	7.0000	4.6667	7.0000	7.6667	8.0000	4.3333	4.0000
Average ranking	4.38	2.38	6.43	4.05	5.38	3.24	2.00
Total ranking	5	2	7	4	6	3	1

The best value of each function is highlighted in bold in the table

seven algorithms on twenty-one datasets. The average feature numbers of seven methods on each dataset are listed in Table 13. Furthermore, the non-parametric statistical Friedman test results are also provided in Tables 12 and 13.

From Table 12, compared with BOA, EEGWO, and ISCA, VGHHO gets better classification accuracy on all the datasets. VGHHO obtains better classification accuracy than SOA on thirteen and one (i.e., PenglungEW) datasets, respectively. However, the better results are obtained by SOA on BreastEW, Clean2, IonosphereEW, KrvskpEW, M-of-n, SonarEW, and WaveformEW datasets, respectively. With respect to the basic HHO algorithm, VGHHO achieves better performance on all the datasets except for Vote. For Vote dataset, two algorithms find similar classification accuracy. Compared to the m-HHO algorithm, VGHHO provides better and similar classification rate for seventeen and two (i.e., Vote and WineEW) datasets. However, the better results are found by OBL-HHO on Semeion and SonarEW datasets. In addition, according to the Friedman test results in Table 12, the ranking order is VGHHO, SOA, m-HHO, HHO, BOA, ISCA, and EEGWO.

From Table 13, the numbers of the selected features of VGHHO are less than BOA on all the datasets except for Exactly2. Compared with SOA, VGHHO selects less and more feature numbers on eleven and nine datasets, respectively. For M-of-n dataset, two algorithms find the equal numbers of features. The numbers of the selected features of HHO are

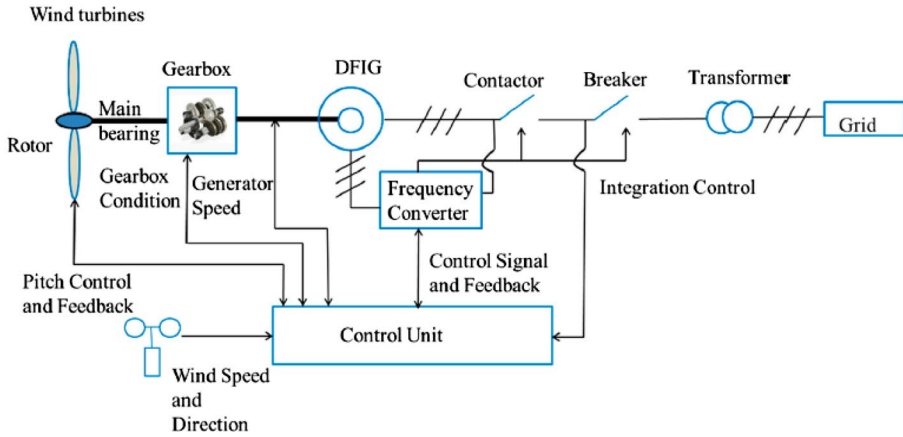


Fig. 9 The components connection framework of a typical wind turbine

Table 14 The feature and sample numbers of two fault datasets

Dataset	Faculty type	Number of features	Number of samples
Dataset-1	Variable pitch system super capacitor voltage low fault	106	2883
Dataset-2	Variable pitch paddle 3 super capacitor voltage low fault	110	3585

Table 15 The average classification accuracy of seven algorithms on two fault datasets

Dataset	BOA	SOA	HHO	OBL-HHO	EEGWO	ISCA	VGHHO
Dataset-1	0.9968	0.9988	0.9993	1.0000	0.9979	1.0000	1.0000
Dataset-2	0.9944	0.9974	0.9970	0.9944	0.9972	0.9972	1.0000
Average ranking	6.75	3.50	4.50	4.25	4.75	2.75	1.50
Total ranking	7	3	5	4	6	2	1

The best value of each function is highlighted in bold in the table

more than VGHHO on all the datasets. VGHHO selects the numbers of features less than m-HHO on all the datasets except for Sonar and Vote. With respect to EEGWO, VGHHO obtains more feature numbers on nineteen datasets. The numbers of the selected features of VGHHO are less and more than ISCA on fifteen and four datasets, respectively. Additionally, VGHHO obtains the first rank based on Friedman test.

Table 16 The average feature numbers are obtained by seven algorithms on two fault datasets

Dataset	BOA	SOA	HHO	OBL-HHO	EEGWO	ISCA	VGHHO
Dataset-1	53.333	46.000	12.667	8.0000	34.333	23.000	6.3333
Dataset-2	54.000	47.000	19.667	3.3333	31.667	19.000	2.6667
Average ranking	7.00	6.00	3.50	2.5	5.00	3.50	1.00
Total ranking	7	6	3	2	5	3	1

The best value of each function is highlighted in bold in the table

7 VGHHO for fault diagnosis of wind turbine

Although VGHHO has shown excellent performance on benchmark problems, it is necessary for investigating its effectiveness in real-world problems. Therefore, in this section, a practical fault diagnosis problem of wind turbine is used to verify the effectiveness of VGHHO. Wind turbine is a kind of clean energy, which has been widely used. Figure 9 plots the framework diagram of connection between components of wind turbine.

Pitch control system is an important parts of wind turbine. Its internal structure is complex. When it operates in extremely harsh environment, it is likely to cause its failure (Tang et al. 2020a). The fault of variable pitch system directly affects the power operation efficiency of wind turbine (Tang et al. 2020b). Therefore, the research on fault diagnosis of variable pitch system plays an important role in reducing the operating cost of wind turbine and improving the power generation (Cho et al. 2018). One year monitoring and data acquisition (SACDA) data set of a wind farm in East China is the experimental data of fault data of variable pitch system. Table 14 lists the fault feature and sample numbers of two datasets.

The wrapper method with KNN classifier and VGHHO on feature selection is used on two fault datasets. Each dataset is randomly divided into two parts, namely, 80% is the training set while 20% is the testing one. The performance of VGHHO is compared against BOA, SOA, HHO, OBL-HHO, EEGWO and ISCA. For all algorithms, the swarm scale is 10 and the maximum of iterative numbers is 10. Each algorithm runs 30 independently trials on each dataset. Table 15 provides the average classification accuracy of seven algorithms on two datasets. The average feature numbers of seven algorithms on two datasets is shown in Table 16. In addition, the non-parametric statistical Friedman test results on seven algorithms are also provided in Tables 15 and 16.

As seem in Tables 15, the average classification accuracy of VGHHO is better than BOA, SOA, HHO, and EEGWO on Dataset-1. Compared with OBL-HHO and ISCA, VGHHO obtains similar results on Dataset-1. For Dataset-2, the result of VGHHO is better than other six algorithms. From the statistical Friedman test results, VGHHO achieves the first rank. From Table 16, the numbers of the selected features of VGHHO are less than other six algorithms on Dataset-1. For Dataset-2, VGHHO achieves better results than BOA, SOA, HHO, OBL-HHO, EEGWO, and ISCA. With respect to the non-parametric statistical Friedman test results in Table 16, VGHHO obtains the first rank, followed by OBL-HHO, HHO, ISCA, EEGWO, SOA, and BOA.

Fig. 10 The equivalent circuit structure of SD model

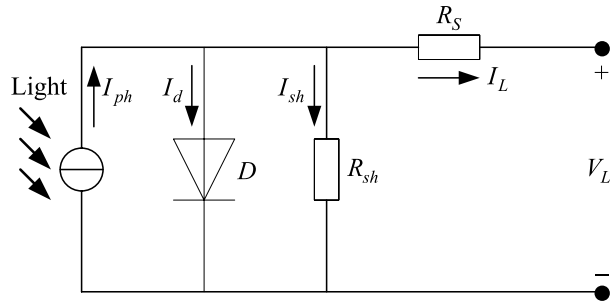


Table 17 The best estimated parameters and their corresponding RMSE values of various algorithms

Algorithm	I_{ph} (A)	I_{sd} (μ A)	R_s (Ω)	R_{sh} (Ω)	n	RMSE
CLPSO (Liang et al. 2006)	0.7608	0.34302	0.0361	54.1965	1.4873	9.9633E-04
DE-BBO (Gong et al. 2010)	0.7605	0.32477	0.0364	55.2627	1.4817	9.9922E-04
BSA (Civicioglu 2013)	0.7609	0.37749	0.0358	56.5266	1.4970	1.0398E-03
GOTLBO (Chen et al. 2016)	0.7608	0.32970	0.0363	53.3664	1.4833	9.8856E-04
IBSA (Nama et al. 2017)	0.7607	0.35502	0.0361	58.2012	1.4907	1.0092E-03
GWCS (Long et al. 2020a)	0.760773	0.32192	0.03639	53.6320	1.4808	9.8607E-04
EABOA (Long et al. 2021a)	0.760771077	0.322929	0.036379593	53.76600144	1.481153457	9.8602E-04
HHO (Heidari et al. 2019)	0.7599465	0.358115	0.0373477	82.48671	1.4912567	2.4122E-03
VGHHO	0.7607549	0.324388	0.0363521	53.94424	1.4816135	9.8628E-04

8 VGHHO for parameter estimation of photovoltaic model

In recent years, low-carbon technology has developed rapidly around the world. Solar energy is considered one of the most promising renewable energy resources due to its abundance, cleanliness, and pollution-free. Photovoltaic (PV) power generation systems can convert solar energy into electrical energy. As the main component of the PV power generation system, accurately estimate the parameters of PV cells is a great significant to model the PV systems. Parameters with low accuracy will not only cause large errors, but may even lead to the failure of the maximum power point tracking (MPPT). Therefore, establishing a reliable mathematical model based on the measure data that describe the nonlinear characteristics of solar cells and accurately estimating its parameters can provide a guarantee for the design and application of solar cell fault diagnosis and MPPT control.

In generally, single diode (SD) is one of the widely used models in solar PV power generation system. The equivalent circuit structure of SD model is shown in Fig. 10.

In Fig. 10, based on Shockley equation, the output current (I_L) of SD model is calculated as follows (Long et al. 2020a, 2021a):

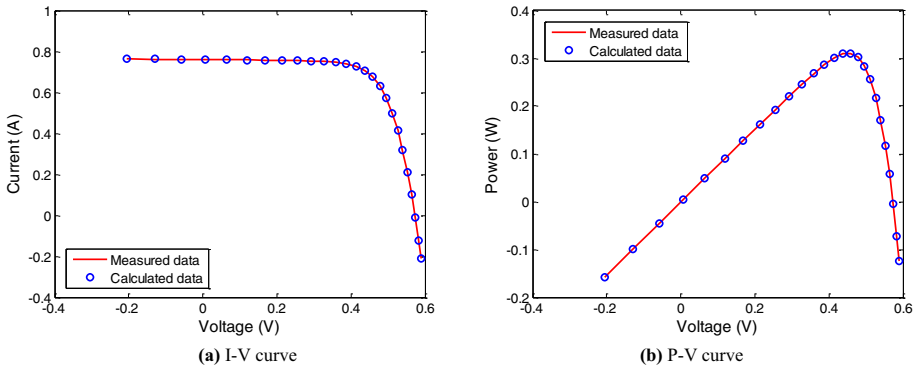


Fig. 11 The calculated values obtained by VGHHO and the measured values for SD model

$$I_L = I_{ph} - I_{sd} \cdot \left[\exp \left(\frac{q \cdot (V_L + R_S \cdot I_L)}{n \cdot k \cdot T} \right) - 1 \right] - \frac{V_L + R_S \cdot I_L}{R_{sh}} \tag{27}$$

where I_{ph} is the photo-generated current, I_{sd} represents the reverse saturation current, R_S denotes the series resistance, q is the electron charge ($= 1.60217646$), k represents the Boltzmann constant (1.38×10^{-23}), V_L denotes the output voltage, R_{sh} is the shunt resistance, n represents the ideality factor, and T denotes the cell temperature in Kelvin. From Eq. (27), for SD model, five parameters (i.e., I_{ph} , I_{sd} , R_S , R_{sh} , and n) are required to be estimated based on the measured I_L and V_L data.

Over the past twenty years, many methods have been proposed to estimate the unknown parameters of SD model. Among them, meta-heuristic optimization algorithm is the most popular parameters estimation method of SD model (Long et al. 2020a, 2021a). In this paper, VGHHO is used to estimate the unknown parameters of SD model. The range of five parameters is set as follows: $0 \leq I_{ph}$, $I_{sd} \leq 1$, $0 \leq R_S \leq 0.5$, $0 \leq R_{sh} \leq 100$, and $1 \leq n \leq 2$. The measured current–voltage (I-V) data are acquired from Easwarakhanthan et al. (1986). VGHHO is also compared with the standard HHO and other seven algorithms. All algorithms have the same maximum number of fitness evaluations 50,000. The best estimated parameter and their corresponding RMSE values of various methods are listed in Table 17.

From Table 17, the best parameters and RMSE value of SD model are obtained by EABOA. Compared with CLPSO, DE-BBO, BSA, GOTLBO, IBSA, and HHO algorithms, VGHHO obtains better RMSE value for SD model. However, the RMSE value of VGHHO is worse than that of GWOCS and EABOA. Furthermore, based on the best estimated parameters, the fitting curves of the calculated data obtained by VGHHO and the measured data is shown in Fig. 11. As can be seen from Fig. 11, the calculated data obtained by VGHHO are in very good agreement with the measured data for SD model.

9 Conclusions

The main purpose of this study was to develop an improved version of HHO (i.e., VGHHO) by introducing three modified strategies to overcome the drawbacks of HHO. The velocity and inertia weight were added into the position search equation in exploitation phase for guiding search direction of algorithm. Thus, the convergence speed and solution precision were improved. To obtain a good transition from exploration to exploitation, a nonlinear escaping energy coefficient E based on cosine function was proposed. The refraction-opposition-based learning mechanism was introduced to enhance the population diversity and avoid premature convergence.

To investigate the effectiveness of VGHHO, several experiments were conducted. In the first experiment, eighteen classical benchmark functions with different scales were selected to evaluate the performance of VGHHO and compared it with other algorithms. The experimental results indicated that VGHHO had higher precision and better scalability than other algorithms on most classical benchmark functions. In the second experiment, 30 latest benchmark functions from CEC 2017 were tested. Simulations indicated that VGHHO obtained better performance than other algorithms on complex test functions. In the third experiment, we utilized twenty-one benchmark feature selection problems from UCI to further test the optimization ability of VGHHO. The test investigation showed that the VGHHO gave satisfactory results in term of classification rate. Finally, a practical fault diagnosis problem of wind turbine and a parameter estimation problem of PV model were used to verify the performance of VGHHO in solving the real-world applications. The comprehensive results indicated that VGHHO was a feasible and promising technique for wind turbine fault diagnosis and PV model parameter estimation. These experiments confirmed that VGHHO performed better competitiveness than other selected algorithms on benchmark functions, benchmark feature selection problems and real-world problems.

The main limitation and disadvantage of VGHHO was that different parameters need to be set for different optimization problems. Furthermore, for complex optimization problems such as CEC 2017 benchmark functions and feature selection problems of UCI complex datasets, the solution results of VGHHO are not very satisfied.

In the future, VGHHO will be used for solving more complex optimization problems and real-world applications, especially in the fields of constrained, multi-objective, and combinational optimization problems, signal processing, pattern recognition, and automatic control as well as data mining. Moreover, the velocity-guided strategy can also be added into other meta-heuristic algorithms and further investigates its effectiveness and feasibility.

Acknowledgements This work was partly supported by the National Natural Science Foundation of China (61463009,62173050), Science and Technology Foundation of Guizhou Province, China ([2020]1Y012), Innovation Groups Project of Education Department of Guizhou Province, China (KY[2021]015), Guizhou Key Laboratory of Big Data Statistics Analysis (BDSA20200101 and BDSA20190106), Key Projects of Education Department of Hunan Province (19A254), and Natural Science Foundation of Hunan Province (2020JJ4382).

References

Abdel-Basset M, Ding W, El-Shahat D (2021) A hybrid Harris hawks optimization algorithm with simulated annealing for feature selection. *Artif Intell Rev* 54:593–637

- Alabool HM, Alarabiat D, Abualigah L, Heidari AA (2021) Harris hawks optimization: a comprehensive review of recent variants and applications. *Neural Comput Appl* 33:8939–8980
- Al-Betar MA, Awadallah MA, Heidari AA, Chen H, Al-khraisat H, Li C (2021) Survival exploration strategies for Harris hawks optimizer. *Expert Syst Appl* 168:114243
- Alsattar HA, Zaidan AA, Zaidan BB (2020) Novel meta-heuristic bald eagle search optimization algorithm. *Artif Intell Rev* 53:2237–2264
- Arini FY, Chiewchanwattana S, Soomlek C, Sunat K (2022) Joint Opposite Selection (JOS): A premiere joint of selective leading opposition and dynamic opposite enhanced Harris' hawks optimization for solving single-objective problems. *Expert Syst Appl* 188:116001. <https://doi.org/10.1016/j.eswa.2021.116001>
- Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23:715–734
- Awad NH, Ali MZ, Liang JJ, Qu B, Suganthan PN (2016) Problem definitions and evaluation criteria for the CEC 2017 Special Session and competition on objective bound constrained real-parameter numerical optimization, Technical Report, Nanyang Technological University Singapore
- Balaha HM, El-Gendy EM, Saafan MM (2021) CovH2SD: a COVID-19 detection approach based on Harris hawks optimization and stacked deep learning. *Expert Syst Appl* 186:115805
- Bandyopadhyay R, Basu A, Cuevas E, Sarkar R (2021a) Harris hawks optimization with simulated annealing as a deep feature selection method for screening of COVID-19 CT-scans. *Appl Soft Comput* 111:107698
- Bandyopadhyay R, Kundu R, Oliva D, Sarkar R (2021b) Segmentation of brain MRI using an altruistic Harris hawks optimization algorithm. *Knowl Based Syst* 232:107468
- Chatterjee I (2021) Artificial intelligence and patentability: review and discussions. *Int J Modern Res* 1:15–21
- Chawla M, Duhan M (2018) Lévy flights in metaheuristics optimization algorithms—a review. *Appl Artif Intell* 32:802–821
- Chen X, Yu K, Du W, Zhao W, Liu G (2016) Parameters identification of solar cell models using generalized oppositional teaching learning based optimization. *Energy* 99:170–180. <https://doi.org/10.1016/j.energy.2016.01.052>
- Chen H, Herdari AA, Chen H, Wang M, Pan Z, Gandomi AH (2020) Multi-population differential evolution-assisted Harris hawks optimization: framework and case studies. *Future Gene Comput Syst* 111:175–198
- Cho S, Gao Z, Moan T (2018) Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines. *Renew Energy* 120:306–321
- Civicioglu P (2013) Backtracking Search Optimization Algorithm for numerical optimization problems. *Appl Math Comput* 219(15): 8121–8144. <https://doi.org/10.1016/j.amc.2013.02.017>
- Dehghani M, Montazeri Z, Malik OP, Dhiman G, Chahar V (2019) BOSA: binary orientation search algorithm. *Int J Innov Technol Explor Eng* 9:5306–5310
- Dehghani M, Montazeri Z, Dhiman G, Malik OP (2020a) A spring search algorithm applied to engineering optimization. *Appl Sci* 10:6173
- Dehghani M, Montazeri Z, Givi H, Guerrero JM (2020b) Darts game optimizer: a new optimization technique based on darts game. *Int J Intell Eng Syst* 13:286–294
- Dhiman G (2021) ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Eng Comput* 37:323–353
- Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70
- Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl Based Syst* 159:20–50
- Dhiman G, Kumar V (2019a) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl Based Syst* 165:169–196
- Dhiman G, Kumar V (2019b) STOA: a bio-inspired based optimization algorithm for industrial engineering problems. *Eng Appl Artif Intell* 82:148–174
- Dhiman G, Garg M, Nagar A, Kumar V, Dehghani M (2021a) A novel algorithm for global optimization: Rat swarm optimizer. *J Ambient Intell Human Comput* 12:8457–8482
- Dhiman G, Oliva D, Kaur A, Singh KK, Vimal S, Sharma A, Cengiz K (2021b) BEPO: a novel binary emperor penguin optimizer for automatic feature selection. *Knowl Based Syst* 211:106560
- Ding WP, Abdel-Basset M, Eldrandaly KA, Abdel-Fatah L, De Albuquerque VHC (2021) Smart supervision of cardiomyopathy based on fuzzy Harris hawks optimizer and wearable sensing data optimization: a new model. *IEEE Trans Cybern* 51:4944–4958


- 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 PM_{2.5} and PM₁₀ forecasting. *Appl Soft Comput* 96:106620
- Easwarakhanthan T, Bottin J, Bouhouch I, Boutric C (1986) Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers. *Int J Sol Energy* 4:1–12
- Elaziz MA, 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* 96:106347
- Elaziz MA, Yousri D, Mirjalili S (2021) A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamic. *Adv Eng Softw* 154:102973
- Essa FA, Elaziz MA, Elsheikh AH (2020) An enhanced productivity prediction model of active solar still using artificial neural network and Harris hawks optimizer. *Appl Ther Eng* 170:115020
- Ewees AA, Elaziz MA (2020) Performance analysis of chaotic multi-verse Harris hawks optimization: a case study on solving engineering problems. *Eng Appl Artif Intell* 88:103370
- Fan Q, Chen Z, Xia Z (2020) A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems. *Soft Comput* 24:14825–14843
- Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. *Expert Syst Appl* 152:113377
- Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29:17–35
- Gong W, Cai Z, Ling CX (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15(4):645–665. <https://doi.org/10.1007/s00500-010-0591-1>
- Griffiths DJ (1998) Introduction to electrodynamics. Prentice Hall of India, New Delhi
- Gupta S, Deep K, Engelbrecht AP (2020a) A memory guided sine cosine algorithm for global optimization. *Eng Appl Artif Intell* 93:103718
- Gupta S, Deep K, Heidari AA, Moayedi H, Wang M (2020b) Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis. *Expert Syst Appl* 158:113510
- Gupta S, Deep K, Mirjalili S, Kim JH (2020c) A modified sine cosine algorithm with novel transition parameter and mutation operator for global optimization. *Expert Syst Appl* 154:113395
- Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. *Future Gene Comput Syst* 101:646–667
- Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl Intell* 51:1531–1551
- Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W (2022) Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. *Math Comput Simul* 192:84–110
- Hassan MH, Houssein EH, Mahdy MA, Kamel S (2021) An improved manta ray foraging optimizer for cost-effective emission dispatch problems. *Eng Appl Artif Intell* 100(2021):104155
- Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
- Houssein EH, Hosney ME, Oliva D, Mohamed WM, Hassaballah M (2020a) A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery. *Comput Chem Eng* 133:106656
- Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M (2020b) Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization. *Eng Appl Artif Intell* 94:103731
- Houssein EH, Gad AG, Wazery YM, Suganthan PN (2021a) Task scheduling in cloud computing based on meta-heuristic: review, taxonomy, open challenges, and future trends. *Swarm Evol Comput* 62:100841
- Houssein EH, Mahdy MA, Blondin MJ, Shebl D, Mohamed WM (2021b) Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst Appl* 174:114689
- Houssein EH, Mahdy MA, Eldin MG, Shebl D, Mohamed WM, Abdel-Aty M (2021c) Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm. *J Adv Res* 29:147–157
- Houssein EH, Neggaz N, Hosney ME, Mohamed WM, Hassaballah M (2021d) Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities. *Neural Comput Appl* 33:13601–13618
- 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
- Issa M, Samn A (2022) Passive vehicle suspension system optimization using Harris hawk optimization algorithm. *Math Comput Simul* 191:328–345
- Jiao S, Chong G, Huang C, Hu H, Wang M, Heidari AA, Chen H, Zhao X (2020) Orthogonally adapted Harris hawks optimization for parameter estimation of photovoltaic models. *Energy* 203:117804

- Kamboj VK, Nandi A, Bhadoria A, Sehgal S (2020) An intensify Harris hawks optimizer for numerical and engineering optimization problems. *Appl Soft Comput* 89:106018
- Kaur S, Awasthi LK, Sangal AL, Dhiman G (2020) Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. *Eng Appl Artif Intell* 90:103541
- Kaur N, Kaur L, Cheema SS (2021) An enhanced version of Harris hawks optimization by dimension learning-based hunting for breast cancer detection. *Sci Rep* 11:21933
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*. pp 1942–1948
- Kumar R, Dhiman G (2021) A comparative study of fuzzy optimization through fuzzy number. *Int J Modern Res* 1:1–14
- Li C, Li J, Chen H, Heidari AA, Zhao X (2021) Memetic Harris hawks optimization: developments and perspectives on project scheduling and QoS-aware web service composition. *Expert Syst Appl* 171:114529
- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295. <https://doi.org/10.1109/TEVC.2005.857610>
- Long W, Jiao J, Liang X, Tang M (2018a) An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Eng Appl Artif Intell* 68:63–80
- Long W, Jiao J, Liang X, Tang M (2018b) Inspired grey wolf optimizer for solving large-scale function optimization problems. *Appl Math Model* 60:112–126
- Long W, Wu T, Liang X, Xu S (2019) Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert Syst Appl* 123:108–126
- Long W, Cai S, Jiao J, Xu M, Wu T (2020a) A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. *Energy Convers Manage* 203:112243
- Long W, Wu T, Jiao J, Tang M, Xu M (2020b) Refraction-learning-based whale optimization algorithm for high-dimensional problems and parameter estimation of PV model. *Eng Appl Artif Intell* 89:103457
- Long W, Jiao J, Liang X, Wu T, Xu M, Cai S (2021a) Pinhole-imaging-based learning butterfly optimization algorithm for global optimization and feature selection. *Appl Soft Comput* 103:107146
- Long W, Wu T, Xu M, Tang M, Cai S (2021b) Parameters identification of photovoltaic models by using an enhanced adaptive butterfly optimization algorithm. *Energy* 229:120750
- Mirjalili S (2016) SCA: A sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Moayedi H, Osouli A, Nguyen H, Rashid ASA (2021) A novel Harris hawks optimization and k-fold cross-validation predicting slope stability. *Eng Comput* 37:369–379
- Mohamed AW, Mohamed AK (2019) Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *Int J Mach Learn Cybern* 10:253–277
- Naik MK, Panda R, Wunnavu A, Jena B, Abraham A (2021) A leader Harris hawks optimization for 2-D Masi entropy-based multilevel imaging thresholding. *Multimed Tools Appl* 80:35543–35583
- Nama S, Saha AK, Ghosh S (2017) Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c- Φ backfill. *Appl Soft Comput* 52:885–897. <https://doi.org/10.1016/j.asoc.2016.09.037>
- Neggaz N, Houssein EH, Hussain K (2020) An efficient henry gas solubility optimization for feature selection. *Expert Syst Appl* 152:113364
- Polap D, Woźniak M (2017) Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* 9:203
- Polap D, Woźniak M (2021) Red fox optimization algorithm. *Expert Syst Appl* 166:114107
- Qais MH, Hasanien HM, Alghuwainem S (2020) Parameters extraction of three-diode photovoltaic model using computation and Harris hawks optimization. *Energy* 195:117040
- Qu C, He W, Peng X, Peng X (2020) Harris hawks optimization with information exchange. *Appl Math Model* 84:52–75
- Rahnamayan S, Tizhoosh HR, Salama MMA (2008) Opposition-based differential evolution. *IEEE Trans Evol Comput* 12:64–79
- Ramalingam S, Bakaran K (2021) An efficient data prediction model using hybrid Harris hawk optimization with random forest algorithm in wireless sensor network. *J Intell Fuzzy Syst* 40:5171–5195
- Ridha HM, Heidari AA, Wang M, Chen H (2020) Boosted mutation-based Harris hawks optimizer for parameters identification of single-diode solar cell models. *Energy Convers Manage* 209:112660

- Rodríguez-Esparza E, Zanella-Calzada LA, Oliva D, Heidari AA, Zaldivar D, Pérez-Cisneros M, Foong LK (2020) An efficient Harris hawks-inspired image segmentation method. *Expert Syst Appl* 155:113428
- Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*. pp 69–73
- Singh T (2020) A chaotic sequence-guided Harris hawks optimizer for data clustering. *Neural Comput Appl* 32:17789–17803
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
- Talatahari S, Azizi M (2021) Chaos game optimization: a novel metaheuristic algorithm. *Artif Intell Rev* 54:917–1004
- Tang M, Hu J, Kuang Z, Wu H, Zhao Q, Peng S (2020a) Fault detection of the wind turbine variable pitch system based on large margin distribution machine optimized by the state transition algorithm. *Math Prob Eng* 2020:9718345
- Tang M, Zhao Q, Ding SX, Wu H, Li L, Long W, Huang B (2020b) An improved lightGBM algorithm for online fault detection of wind turbine gearboxes. *Energies* 13:807
- Tubishat M, Idris N, Shuib L, Abushariah MAM, Mirjalili S (2020) Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst Appl* 145:113122
- Vaishnav PK, Sharma S, Sharma P (2021) Analytical review analysis for screening COVID-19. *Int J Modern Res* 1:22–29
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82
- Wunnav A, Naik MK, Panda R, Jena B, Abraham A (2020) An adaptive Harris hawks optimization technique for two dimensional grey gradient based multilevel image thresholding. *Appl Soft Comput* 95:106526
- Yousri D, Allam D, Eteiba MB (2020) Optimal photovoltaic array reconfiguration for alleviating the partial shading influence based on a modified Harris hawks optimizer. *Energy Convers Manage* 206:112470
- Zervoudakis K, Tsafarakis S (2020) A mayfly optimization algorithm. *Comput Ind Eng* 145:106559
- Zhang H, Xie J, Zong B (2021) Bi-objective particle swarm optimization algorithm for the search and track tasks in the distributed multiple-input and multiple-output radar. *Appl Soft Comput* 101:107000

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Wen Long^{1,2} · Jianjun Jiao³ · Ximing Liang⁴ · Ming Xu³ · Tiebin Wu⁵ · Mingzhu Tang⁶ · Shaohong Cai¹ 

¹ Guizhou Key Laboratory of Big Data Statistical, Guizhou University of Finance and Economics, Guiyang 550025, China

² Guizhou Key Laboratory of Economics System Simulation, Guizhou University of Finance and Economics, Guiyang 550025, China

³ School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang 550025, China

⁴ School of Science, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

⁵ School of Energy and Electrical Engineering, Hunan University of Humanities Science and Technology, Loudi 417000, China

⁶ School of Energy Power and Engineering, Changsha University of Science and Technology, Changsha 410114, China