Neural Computing and Applications
https://doi.org/10.1007/s00521-024-09603-3

ORIGINAL ARTICLE

IHHO: an improved Harris Hawks optimization algorithm for solving engineering problems

Dalia T. Akl[1] · Mahmoud M. Saafan[1] · Amira Y. Haikal[1] · Eman M. El-Gendy[1]

Received: 14 October 2023 / Accepted: 6 February 2024

Abstract

Harris Hawks optimization (HHO) algorithm was a powerful metaheuristic algorithm for solving complex problems. However, HHO could easily fall within the local minimum. In this paper, we proposed an improved Harris Hawks optimization (IHHO) algorithm for solving different engineering tasks. The proposed algorithm focused on random location-based habitats during the exploration phase and on strategies 1, 3, and 4 during the exploitation phase. The proposed modified Harris hawks in the wild would change their perch strategy and chasing pattern according to updates in both the exploration and exploitation phases. To avoid being stuck in a local solution, random values were generated using logarithms and exponentials to explore new regions more quickly and locations. To evaluate the performance of the proposed algorithm, IHHO was compared to other five recent algorithms [grey wolf optimization, BAT algorithm, teaching–learning-based optimization, moth-flame optimization, and whale optimization algorithm] as well as three other modifications of HHO (BHHO, LogHHO, and MHHO). These optimizers had been applied to different benchmarks, namely standard benchmarks, CEC2017, CEC2019, CEC2020, and other 52 standard benchmark functions. Moreover, six classical real-world engineering problems were tested against the IHHO to prove the efficiency of the proposed algorithm. The numerical results showed the superiority of the proposed algorithm IHHO against other algorithms, which was proved visually using different convergence curves. Friedman's mean rank statistical test was also inducted to calculate the rank of IHHO against other algorithms. The results of the Friedman test indicated that the proposed algorithm was ranked first as compared to the other algorithms as well as three other modifications of HHO.

Keywords Harris Hawks optimization · Improved algorithm · Benchmarks · Engineering problems · Metaheuristics

1 Introduction

Optimization solved nonlinear, complex, real-time issues. The metaheuristic algorithm became an intrinsic feature of all optimization processes [1]. Metaheuristics had been grown as a solution for real-world optimization problems in the last two decades. A method of optimization was defining the system's objective function, variables, limits, system properties, and optimal solution [2]. Trajectory-based and population-based approaches were two types of metaheuristic optimization. These two groups exhibit a wide range of probable answers employed in each iterative stage [3].

Randomness marked stochastic search and optimization algorithms. Two types of stochastic algorithms were heuristic and metaheuristic [4]. Heuristics are problem-dependent, and numerous heuristics can be created [5]. A metaheuristic method, on the other hand, makes few assumptions about the problem and can combine various heuristics and generate candidate solutions [6]. Meta-heuristics solve intractable optimization problems. Since

✉ Eman M. El-Gendy
eman_elgendy@mans.edu.eg

Dalia T. Akl
daliatarek@mans.edu.eg

Mahmoud M. Saafan
saafan2007@mans.edu.eg

Amira Y. Haikal
amirayh@mans.edu.eg

[1] Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

Published online: 20 April 2024

the initial metaheuristic was proposed, several new algorithms have now been created [7].

Metaheuristic optimization algorithms' computational efficiency is dependent on striking the right balance between exploration and exploitation [8]. Combining the terms Meta and Heuristic produces the term "metaheuristics" [9]. Metaheuristic algorithms are often characterized as just a master strategy that guides and modifies other heuristics to get answers beyond those typically generated in the pursuit of local optimality [10]. Optimizing metaheuristic algorithms has become such an active area of research and one of the most well-known high-level procedures for generating, selecting, or locating heuristics that optimize solutions as well as provide a better objective function for a real-world optimization problem [11] [12].

Every metaheuristic technique must achieve a reasonable balance between exploration and utilization of the search space for optimal performance [13]. Metaheuristics have gained appeal over exact methods for addressing optimization issues due to the ease and resilience of the answers they give in a variety of sectors, such as engineering, business, transportation, as well as the social sciences [14] [15]. A metaheuristic is an algorithm meant to tackle a wide variety of difficult optimization problems without requiring extensive problem-specific adaptation. The prefix "meta" indicates that these heuristics are "higher level" than problem-specific heuristics. Metaheuristics are often used to solve unsolvable situations [16].

For mathematical optimization problems with more than one objective for which no single solution exists, multi-objective optimization is used [17]. Stochastic optimization methods, such as metaheuristics, use mechanisms inspired by nature to solve optimization problems [18]. Metaheuristic algorithms are very good optimization techniques that have been utilized to solve a wide variety of optimization issues. Metaheuristics can be viewed as a global algorithmic framework utilized to solve multiple optimization problems with minimal modification [19], 20.

Metaheuristics outperform simple heuristics. Metaheuristic algorithms use randomization and local search. In most situations, complexity analysis, performance assessment, and metaheuristic parameter adjustment were ignored when metaheuristics were used to address optimization problems [21], 22. Single-solution metaheuristics focus on a single starting solution, whereas population-based metaheuristics focus on a large number of possible solutions [18]. It is generally more effective to use the standard metaheuristics when dealing with typical research challenges [23].

Genetic algorithms [24], evolutionary methods [25], and cultural algorithms [26] are a few of the most well-known types of algorithms. Particular to this topic is the metaheuristics for optimizing the foraging of particles as well as those for bees, insects, and bacteria [27]. The basic flow diagram for population-based metaheuristic optimization is shown in Fig. 1.

Structural control issues that require a fast convergence of the best solution sometimes benefit from metaheuristic methods. These heuristic techniques are called "metaheuristics" since they are based on a real-world phenomenon [28]. High-level problem-solving techniques can be applied regardless of the nature of the challenge. Metaheuristic procedures have an advantage over traditional methods since they can establish a unique starting point, convexity, continuity, and differentiability [29]. As illustrated in Fig. 2, metaheuristic algorithms can be categorized into a variety of subclasses based on the theory they are derived from [30]. Evolutionary, physical, and bio-inspired algorithms-based metaheuristics are the three main kinds of metaheuristics. The first technique is based on evolution, while the second technique is based on physical phenomena. On the other hand, swarm intelligence-based

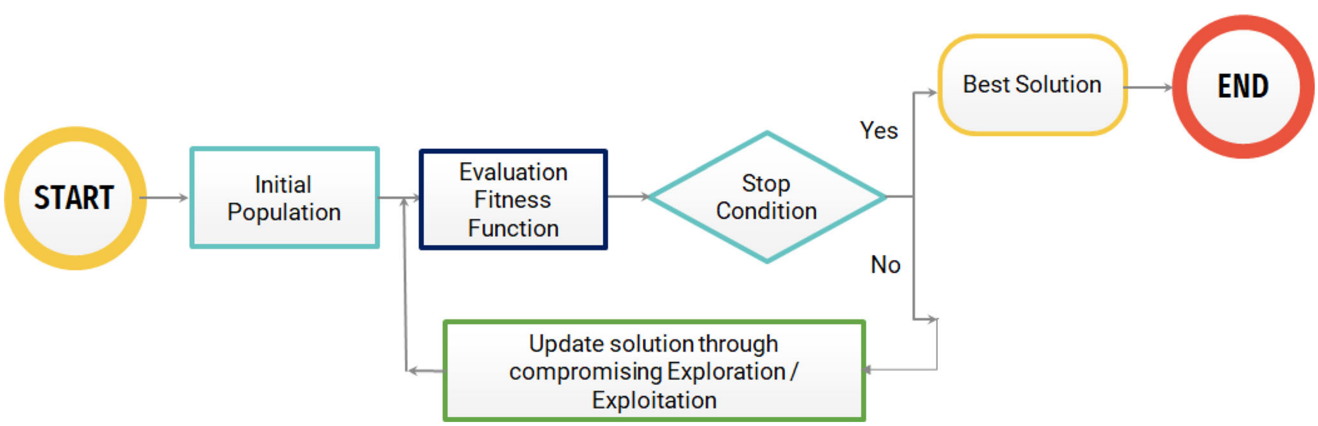


Fig. 1 Schematic flow diagram for population-based metaheuristic optimization methods

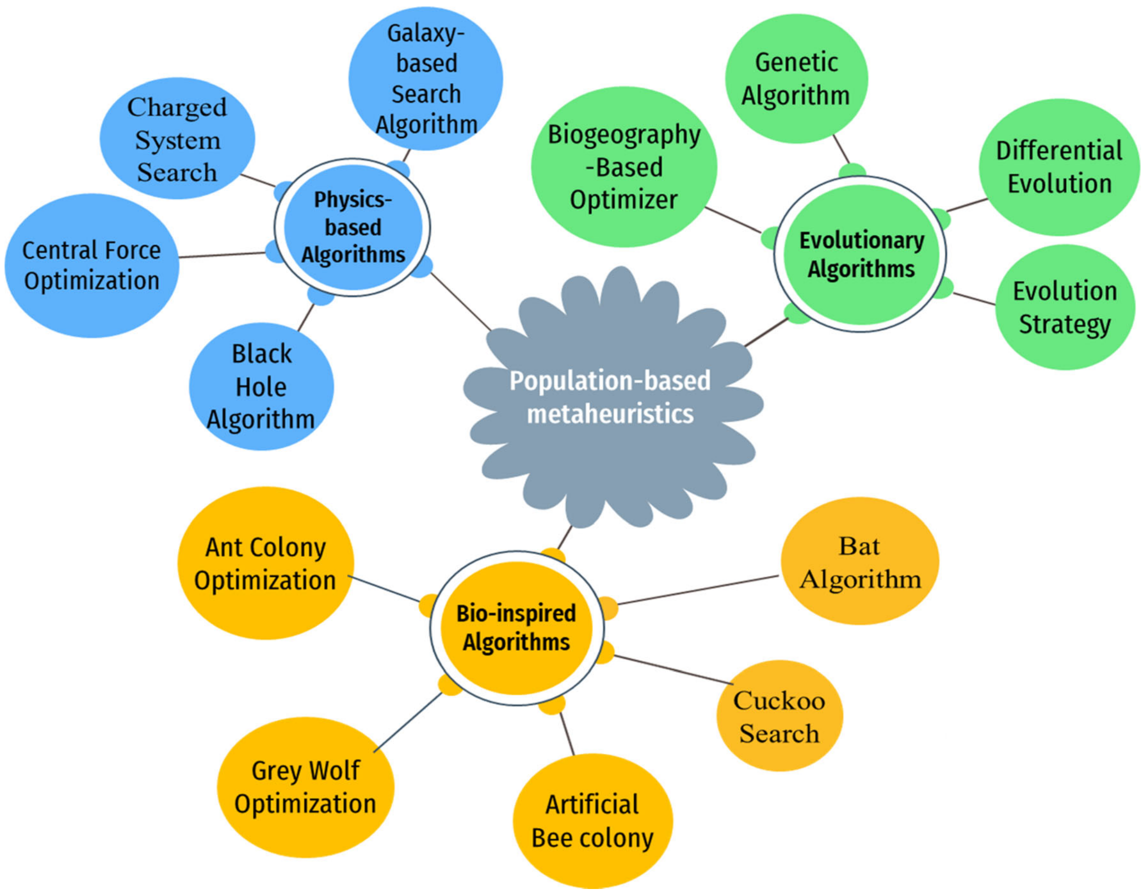


Fig. 2 Formalization of metaheuristic methods

metaheuristics mimic social species' collective activity[31]. Single-solution algorithms may trap local optima, preventing us from finding global optimum, because they generate just one solution for a particular problem [32]. Population-based algorithms can escape local optima [33]. These algorithms are classed by their theoretical roots. Adaptive algorithms take their cues from natural selection, mutation, and recombination in nature[29]. These algorithms pick the best candidate based on population survival [34].

One of the popular swarm-based metaheuristic optimization algorithms is called the Harris Hawks optimization (HHO) algorithm [14]. HHO is inspired by the feeding method that Harris Hawks use to search for and attack the prey. This algorithm has powerful exploration and exploitation capabilities that make it a suitable choice for solving complex problems [35]. However, HHO can sometimes fail to balance between local exploitation and global exploration. Therefore, improvements in both the exploration and exploitation phases are necessary to find additional ideal locations.

In this paper, an improved HHO (IHHO) that improves upon HHO by fixing its flaws was presented. We have compared our proposed algorithm to the original HHO and to other modifications of HHO algorithm implementations, namely BHHO [36], MHHO [37], and logHHO [38]. Also, we have compared IHHO to other recent algorithms, namely grey wolf optimization (GWO) [39], BAT algorithm [40], teaching–learning-based optimization (TLBO)[41], moth-flame optimization (MFO) [42], and whale optimization algorithm (WOA)[43]. GWO has simpler principles and fewer parameters. However, it suffers from poor convergence speed and limited solution accuracy [44]. BAT algorithm has rapid convergence but suffers from poor exploration [19]. Although the TLBO operation has no required parameters, the algorithm's drawbacks include its time-consuming iterations [41]. MFO has few configuration parameters, but early convergence is its biggest downside [42]. WOA exhibits slow convergence as a drawback, while it boasts the advantage

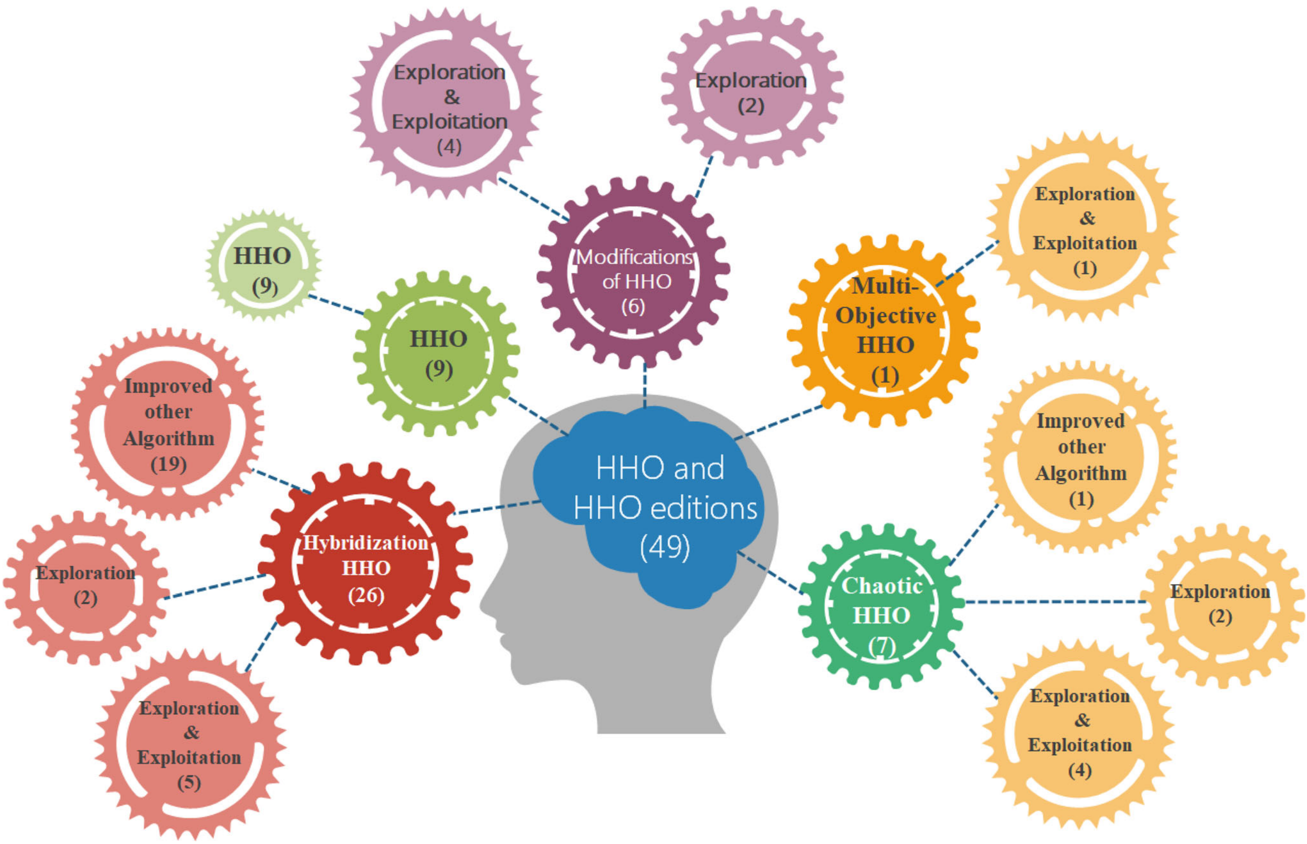


Fig. 3 HHO and HHO editions

of straightforward operation [43]. Mainly, this paper contributed in the following ways:

- Enhancing exploration ability by using the advantages of logarithms and exponentials.
- Enhancing exploitation ability by using the concept of traveling distance rate.
- Testing and comparing IHHO with other modifications of HHO and other algorithms like GWO, BAT, TLBO, WOA, and MFO on 23 standard test functions, CEC2017, CEC2019, and CEC2020.
- Testing and comparing IHHO with other modifications of HHO and other algorithms like GWO, BAT, TLBO, WOA, and MFO on six classical real-world engineering problems.
- Testing and comparing IHHO with other modifications of HHO and other algorithms like GWO, BAT, TLBO, WOA, and MFO on standard benchmark functions consists of 14 variable-dimension unimodal, 5 fixed-dimension unimodal, 20 multimodal fixed-dimension, and 13 multimodal fixed-dimension benchmark functions.
- Using Friedman mean rank statistical test to calculate the rank of IHHO against other algorithms.

The remainder of this study is structured as follows. The second section provides a literature review regarding HHO and its modifications. In Sect. 3, the mathematical concept and computing techniques of the HHO algorithm are explained. In Sect. 4, the concepts behind our proposed IHHO are provided. In Sect. 5, the simulation results of IHHO as compared to other algorithms on several benchmarks are discussed. Finally, Sect. 6 includes the conclusion of the work and future works.

2 Related work

Harris hawks are one of nature's most intelligent birds. These birds know how to lead a group and work together to find a certain rabbit. Various forms of assault and evasion take place during this stage [45]. A Harris hawk hunts with support from family members in the same stable [46]. This desert predator can follow, surround, and flush its prey [47]. This hawk's chase habits may differ based on its environment and prey. A Harris hawk switches activity and works with other predators to confuse prey [48]. HHO is based on Harris hawks' cooperative hunting and escaping. Multiple Harris hawks strike from different angles to

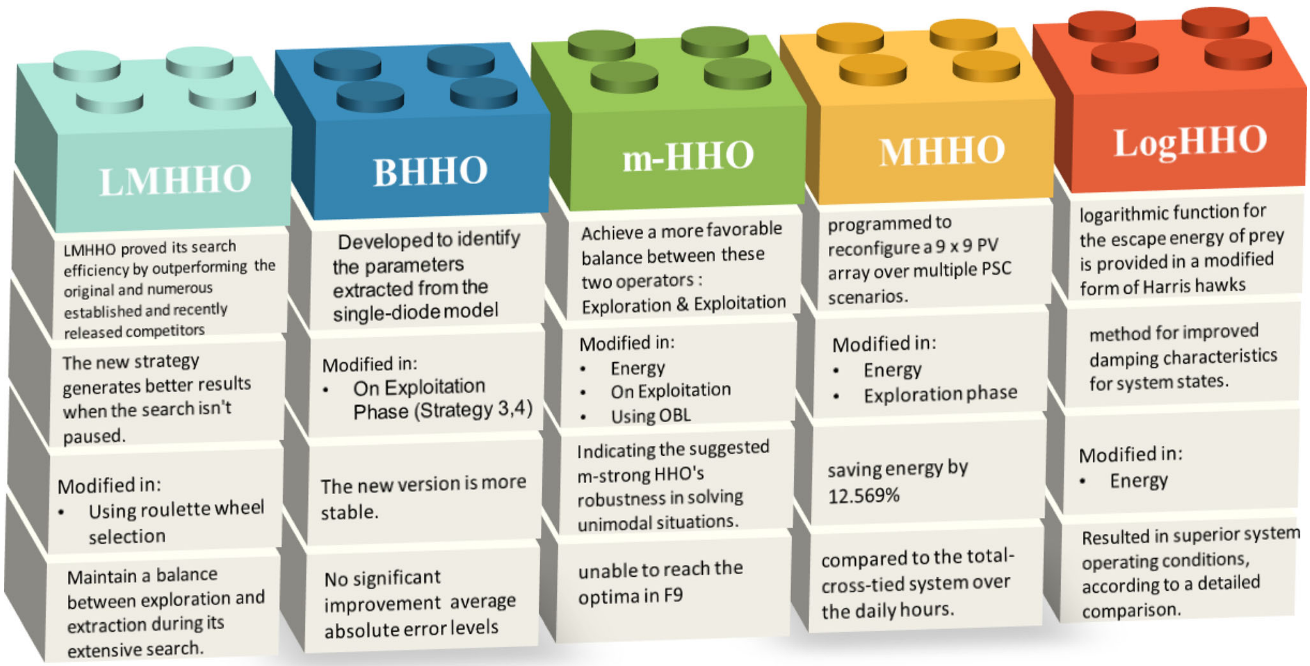


Fig. 4 Specifics of some HHO editions

converge on an escaping rabbit and use varied hunting strategies [49].

Harris Hawks optimization (HHO) is based on hawk hunting. With fewer modification parameters and a significant optimization effect, it's competitive with similar methods [50]. Many academics prefer HHO's global search efficiency as a population-based metaheuristic algorithm [51]. HHO keeps a flock of hawks. The literature shows that HHO has solved challenging optimization problems in several fields [52].

According to Alabool et al., 2021, Fig. 3 shows that HHO's adaptability has been enhanced, and/or additional algorithms have been merged to increase the quality of optimization solutions. There were 49 different HHO variants studied in this review, including six different modifications, 26 distinct hybridizations, one multiobjective version 2, a binarization variant, and 7 chaotic variants [14]. This form of HHO demonstrates HHO's capacity to develop answers to a wide range of optimization issues.

2.1 Modifications of HHO

According to Ridha et al. [36], the exploration and exploitation stages of the boosted mutation-based Harris Hawks optimization (BHHO) algorithm are built using two adaptive approaches. Random steps inspired by the flower pollination algorithm (FPA) with levy flight (LF) ideal notion are employed in the first adaptive approach. Also, the second adaptive technique examines an ideal mutation vector using differential evolution (DE) influenced by the 2-Opt method. A wide range of meteorological conditions has been used to test the proposed BHHO algorithm's performance [53]. The new version has a lot more sturdiness. An increase in diversification is the starting point of its search patterns, but it quickly moves on to exploitation. Average absolute error levels for the proposed BHHO approach were no better than those for other methods [54].

According to Gupta et al. [56], differential fast dives and opposition-based learning are all part of typical HHO. As a result of these techniques, HHO's search efficiency is increased and problems like stagnation at poor options and premature convergence are resolved. modified HHO (mHHO) was the reworked HHO. Standard HHO can't achieve the balance between these two operators that are needed to improve exploration and extraction quality [55]. In issue F7, the mHHO solution surpasses traditional HHO and obtains its optimal value. mHHO can't get to the optima in F9, but it's more accurate than regular HHO in terms of searching [56].

According to Yousri et al. [58], using a new technique dubbed the modified Harris Hawks optimizer, the switching matrix can be reconfigured in such a way as to maximize array generated power, according to the findings in the research [57]. Different stages of the investigation apply the basic Harris Hawk's optimizer for the same problem. To top it all off, the enhanced Harris Hawks optimizer generates the optimal solar array arrangement in less than a single second [58].

According to Devarapalli et al. [59], the HHO algorithm was put out as a potential solution. Damping devices were evaluated with HHO, GWO, and MFO algorithms under various loads. To find the best control parameters, 500 iterations were used for each method to undertake a variety of system analyses. Eigenvalue analysis and system performance characteristics under perturbation were compared among the various optimization methodologies. According to a detailed comparison, the logarithmic function of the modified HHO resulted in superior system operating conditions [38].

According to Hussain et al. [52], long-term memory HHO (LMHHO) could be better investigated. When the search isn't halted at a predetermined level, the new method produces superior results. It was important for LMHHO to maintain a balance between exploration and extraction during its extensive search [59]. On numerical optimization challenges with low- and high-dimensional landscapes, LMHHO proved its search efficiency by outperforming the original and numerous established and recently released competitors [60]. Figure 4 presents summary of the modifications to HHO.

2.2 Hybridization of HHO

Fan et al., 2020, claimed that HHO tends to be stuck in limited variety, local optimums, and imbalanced exploitation abilities. As a way to increase the performance of HHO, a new quasi-reflected Harris Hawk's algorithm (QRHHO) was presented, which incorporated the HHO method and the QRBL together. After introducing the QRBL mechanism to promote population diversity, QRBL was added to each population update to raise the convergence rate. The QRBL mechanism is primarily used in the starting phase and each population update phase of the proposed technique. The experimental results revealed that the QRHHO algorithm performs better than the standard HHO, two versions of HHO, and six other swarm-based intelligent algorithms [61].

According to Kamboj et al. [63], the sine–cosine algorithm was used to speed up HHO's global search stage and keep it out of the local search space in the hybrid Harris Hawks sine–cosine algorithm (hHHO-SCA). The proposed optimizer has been evaluated on engineering design problems that are nonlinear, nonconvex, and heavily constrained. When compared to HHO, the suggested hybrid hHHO-SCA algorithm outperforms it, proving that the metaheuristic optimization algorithm derived from natural phenomena is useful for a wide range of optimization applications, including those involving multidisciplinary engineering design [62].

Du et al., [64], proposed a strategy for achieving many objectives for air pollution prediction. The HHO technique has been created to fine-tune the parameters of a machine learning (ELM) model to achieve high forecast accuracy and stability. MOHHO, a novel multiobjective method, is being developed to optimize the ELM model's parameters for predicting air pollution concentrations with high accuracy and stability. The sequence of pollutant concentrations can be predicted using an ELM model that has been tuned. In terms of prediction accuracy, the MOHHO-based hybrid model could outperform existing multiobjective algorithms [63].

According to Dhawale et al. 2020, the hybrid metaheuristic optimization approach is created by combining Harris Hawks' and Improved Grey wolf optimization techniques. A more effective method of transitioning from exploration to extraction is expected to be discovered. HHHO-IGWO, on the other hand, allows for a greater range of possibilities when it comes to solving optimization problems. The unique optimization technique provides improved phase exploitation, faster convergence, and improved optimization accuracy. The hHHO-IGWO algorithm outperforms the previous strategy when it comes to convergence, according to researchers [64].

In the study by Zhong et al. [66], the authors provided a first-order reliability method based on Harris Hawks optimization (HHO-FORM) for high-dimensional reliability analysis. HHO-FORM reliability index is a constrained optimization problem solution by FORM theory. After that, a penalty function is used to account for the limits. FORM's high-dimensional dependability issues can be resolved more quickly with HHO-simplicity. Using HHO-FORM, high-dimensional dependability challenges can be tackled with precision and efficiency [65].

Sehabeldeen et al. [67], proposed a new hybrid ANFIS-HHO model, which integrates a modified form of an adaptive neuro-fuzzy inference engine (ANFIS) and HHO, that can be used to forecast weld joint UTS. HHO's search for optimal ANFIS parameter values turned out the optimal operating conditions for the FSW process. Using the ANFIS-HHO method, it is possible to accurately estimate the FSW parameters [66].

2.3 Other versions of HHO

Chen et al. [68] proposed a multipopulation differential evolution-assisted Harris Hawks optimization algorithm (CMDHHO) as an improved HHO. HHO's incapacity to balance exploration and extraction could lead to a local optimum. The first strong variant of HHO integrates chaotic strategy, topological multipopulation strategy, and differential evolution (DE) strategy to fix its flaws. A chaos mechanism is added to the original algorithm to improve

HHO's exploitation. Three strategies were used to enhance HHO (chaos, multipopulation, and DE). The multipopulation strategy includes DNS, SRS, and PDS. Simulations demonstrated that the proposed technique avoided a local optimum and increased the basic HHO's convergence speed. CMDHHO is superior to HHO and can be used to tackle discrete problems [67].

While searching for prey cooperatively, HHO might be unable to maintain exploration and development capacities of information sharing. Qu et al. [69], proposed an improved HHO algorithm with information exchange (IEHHO), Harris Hawks that can trade and share information while hunting. This enhanced accuracy and convergence speed. Nonlinear escape energy factors with chaotic disturbance could balance the algorithm's exploration and exploitation. Exploring the shared region allowed Harris Hawks to exchange information. The IEHHO approach took longer to run but yielded better solutions and had a greater convergence rate. Parameters were IEHHO's biggest limitation. Ideal parameters couldn't be guaranteed despite testing. IEHHO is a random search methodology; experimental results might be affected by environment and constraint handling [68].

Ewees et al. [70] proposed a chaotic multiverse Harris Hawks optimization (CMVHHO), a modified form of multiverse optimizer (MVO) based on chaos theory and HHO. The suggested method uses chaotic maps to get the optimal MVO parameters. Local searches help MVO maximize search space. The original MVO was modified using HHO. Chaos maps were used to determine appropriate MVO settings, while HHO increased the MVO's searchability. The recommended strategy for global optimal solutions showed great convergence and statistical analysis. The proposed CMVHHO solved four engineering problems better than the current methods. When compared to PSO and GA, CMVHHO consumed more CPU time, although it is still faster than using the simple MVO algorithm [69].

Menesy et al., 2019, proposed chaotic HHO (CHHO) to precisely estimate the operational parameters of the proton exchange membrane fuel cell (PEMFC), which models and duplicates its electrical performance. By preventing HHO from being stuck in local optima, the CHHO was intended to improve on the normal HHO's search capabilities while still maintaining its functionality. CHHO outperformed current metaheuristic optimization techniques. In all case scenarios, the CHHO predicted the optimal PEMFC stack characteristics precisely [70].

It is clear from the discussion above that there is a lot of literature in this field. The Harris Hawks optimization (HHO) method is a sophisticated metaheuristic algorithm for addressing complicated problems [11]. However, HHO can easily fall within the local minimum. Because of the

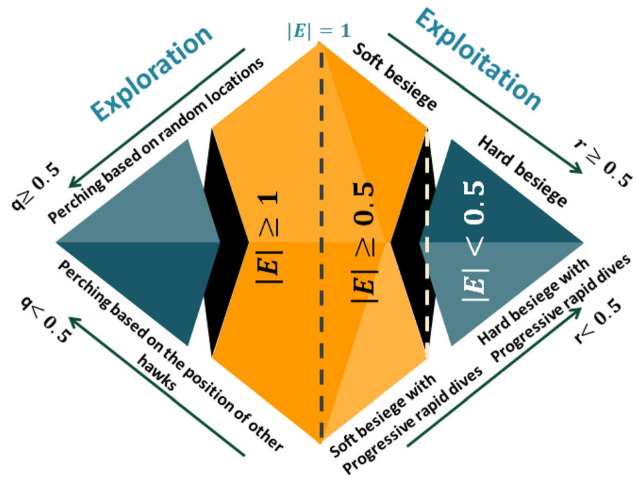


Fig. 5 Explains the HHO's major stages

adjustments to each phase, the other modifications of HHO make it impossible to achieve a healthy balance between exploration and exploitation [39]. Improved Harris Hawks optimization (IHHO) algorithm is capable of handling various engineering jobs and is capable of reaching a good equilibrium between exploitation and exploration as a result of phase-by-phase adjustments [59].

3 Harris Hawk's optimization (HHO)

The Harris Hawk Optimizer, proposed by Heidari et al., is an optimization technique inspired by the natural world Harris Hawks [47]. HHO is an innovative swarm-based optimizer that is both quick and effective. It possesses several exploratory and exploitative methods that are deceptively straightforward yet very efficient, and it has a dynamic structure for addressing continuous problems [71]. This method is a population-based metaheuristic that mathematically mimics natural events. In addition, the coordinated manner in which Harris's Hawks pursue their prey and surprise it served as inspiration for this algorithm's design [72]. The victim is startled because multiple hawks are attempting to attack it from different directions at the same time [73]. Since HHO is an inhabitant's optimization technique that does not use gradients, it can be used for any optimization issue as long as it is formulated correctly. Figure 5 depicts all of the phases of HHO, each of which will be discussed within the next subsections.

3.1 Exploration phase

During this stage of the HHO, Harris' hawks perch at random in various areas and wait to locate a victim characterized by two tactics that take into account the fact that the hawks would wait at various places to identify the prey. Following is a mathematical model Eq. (1). of the search strategies utilized for the investigation of the search space:

$$X(t+1) = \begin{cases} X_{\mathrm{rand}}(t) - r_1|X_{\mathrm{rand}}(t) - 2r_2X(t)| & q \geq 0.5 \\ (X_{\mathrm{rabbit}}(t) - X_m(t)) - r_3(\mathrm{LB} + r_4(\mathrm{BU} - \mathrm{LB}))q & q < 0.5 \end{cases} \quad (1)$$

Here $X(t+1)$ seems to be the coordinates of hawks within the next iteration,$X_{rabbit}(t)$ is the location for rabbit, $X(t)$ is just the current position vector for hawks, and r_1, r_2, r_3, r_4, and q are randomized values within (0,1) updated in each iteration. LB and UB display variable upper and lower boundaries, $X_{\mathrm{rand}}(t)$ is chosen at the random hawk, and X_m is the population's average position. Equation (2) gives the average hawk posture.

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

where N is the count of Hawks and $X_i(t)$ is its position at iteration t.

3.2 The transformation from exploration to exploitation

Exploration and exploitation are two different stages of the HHO algorithm. It is possible to model the energy required to make this shift using Eq. (3):

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

E represents the prey's fleeing energy, T is just the maximum number of rounds, and E_0 randomly fluctuates within the interval (−1, 1) which is modeled using Eq. (4).

$$E_0 = 2 * \mathrm{rand} - 1 \quad (4)$$

3.3 Exploitation phase

The hawks use the surprise pounce to strike the rabbit in their transition from the exploring stage to the exploitation phase. Additionally, the rabbit makes a desperate bid to flee this perilous scenario. The hawks are required to use a variety of chasing techniques. Harris' hawks use a variety of evasive tactics to get their prey. The attacking stage can be described using four well-considered techniques.

The r represents the probability of the rabbit evading the hawks. When it is greater than or equal to 0.5, the rabbit is unable to evade the hawks who employ either a soft besiege or a hard besiege. When the rabbit's health is less than 0.5, hard besieges with progressive rapid dives or soft besieges with progressive rapid dives are the techniques employed by the hawks.

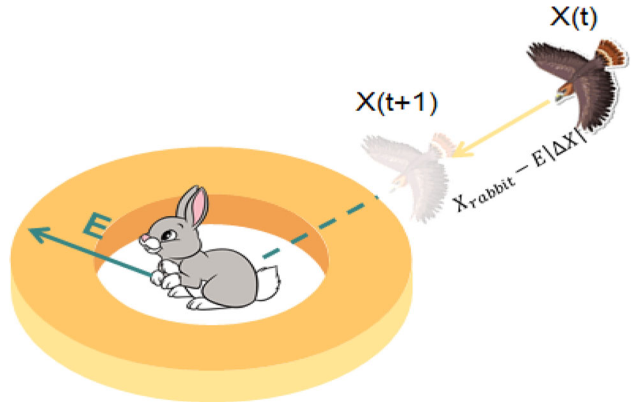


Fig. 6 Vectors for a hard-besieged situation

3.3.1 Soft besiege

When r and $|E| \geq 0.5$, the rabbit can elude the hawks because it has a great deal of endurance. As a result, the Harris hawks gently encircle the rabbit before performing the surprise pounce. This behavior is modeled by using Eq. (5) and Eq. (6):

$$X(t+1) = \Delta X(t) - E|JX_{\mathrm{rabbit}}(t) - X(t)| \quad (5)$$

where $J = 2 * (1 - r_5)$

$$\Delta X(t) = X_{\mathrm{rabbit}}(t) - X(t) \quad (6)$$

$\Delta X(t)$ specifies the difference between the rabbit's position vector as well as its current position in iteration t. The r 5 parameter is a random number between 0 and 1, and J reflects the rabbit's randomized leap strength as it flees.

3.3.2 Hard besiege

When $r \geq 0.5$ and $|E| < 0.5$, the rabbit can evade the hawks, but he has quite a bit of energy. To pull off the final surprise pounce, the Harris hawks barely have time to encircle their prey. This behavior is modeled by using Eq. (7):

$$\Delta X(t) = X_{\mathrm{rabbit}}(t) - E|\Delta X(\mathrm{t})| \quad (7)$$

Figure 6 shows a simple illustration of this process with one hawk.

3.3.3 Soft besiege with progressive rapid dives

When $r < 0.5$ and$|E| \geq 0.5$, the rabbit is still alive, but it has lost its capacity to run away. This flying mimics the rabbit's zigzag motion and the erratic dives made by hawks around a fleeing rabbit. Using the Levy flight, this behavior is modeled by using Eq. (8):

$$Y = X_{\mathrm{rabbit}}(t) - E|JX_{\mathrm{rabbit}}(t) - X(t)| \quad (8)$$

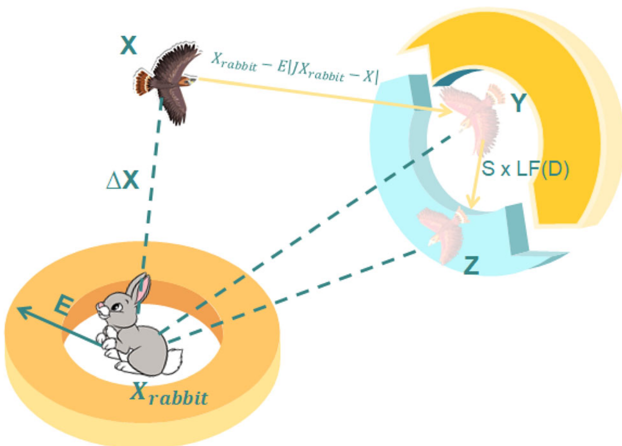


Fig. 7 An illustration of general vectors in the scenario of a hard besiege with progressive quick dives

Then, they compare the likely outcome of such a movement to the prior dive to determine whether or not it was a good dive. They will dive according to the following rule by Eq. (9):

$$Z = Y + S \times \mathrm{LF}(D) \tag{9}$$

where D denotes the problem dimension, S denotes a random vector of size $1 \times D$, and LF denotes the levy flight function, which is computed using Eq. (10):

$$\mathrm{LF}(x) = 0.01 \times \frac{\mathrm{u} \times \sigma}{|\mathrm{v}|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \tag{10}$$

where u, v are random values within (0,1), and β is a constant set to 1.5 by default.

Thus, Eq. (11) can be used to determine the ultimate approach for updating the positions of hawks throughout the gentle besiege phase:

$$X(t+1) = \begin{cases} Y \ if \ F(\mathrm{Y}) < F(\mathrm{X(t)}) \\ Z \ if \ F(\mathrm{Z}) < F(\mathrm{X(t)}) \end{cases} \tag{11}$$

Figure 7 shows an example of this process applied to a single hawk.

3.3.4 Hard besiege with progressive rapid dives

When r and $|E| < 0.5$, the rabbit has exhausted its reserves of energy and is unable to flee. Hawks make an effort to decrease the distance between their usual locations. This behavior is modeled by using Eq. (12):

$$X(t+1) = \begin{cases} Y \ if \ F(\mathrm{Y}) < F(\mathrm{X(t)}) \\ Z \ if \ F(\mathrm{Z}) < F(\mathrm{X(t)}) \end{cases} \tag{12}$$

where Y and Z are calculated according to new principles in Eq. (13) and Eq. (14). Figure 8 shows a simple illustration of this phase in action.

$$Y = X_{\mathrm{rabbit}}(t) - E|JX_{\mathrm{rabbit}}(t) - X_m(t)| \tag{13}$$

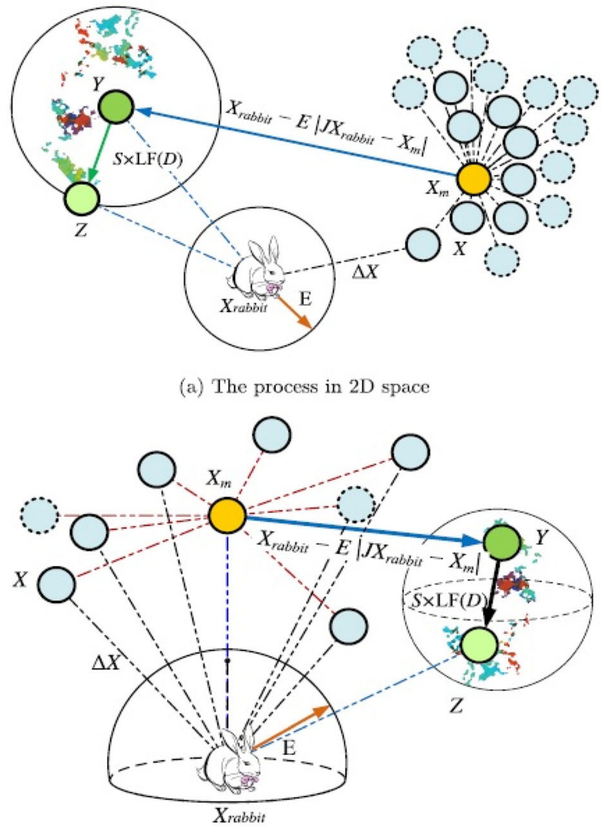
$$Z = Y + S \times LF(D) \tag{14}$$

where $X_m(t)$ is found by solving Eq. (2).

Algorithm 1 presents the details of how the traditional HHO works.

Algorithm 1 Pseudo-code of HHO algorithm

Inputs: The population size $\boldsymbol{N}$ and maximum number of iterations $\boldsymbol{T}$
Outputs: The location of rabbit and its fitness value
Initialize the random population $\boldsymbol{X_i}(\boldsymbol{i} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{N})$
while (stopping condition is not met) **do**
  Calculate the fitness values of hawks
  Set $\boldsymbol{X_{rabbit}}$ as the location of rabbit (best location)
  for (each hawk ($\boldsymbol{X_i}$)) **do**
   Update the initial energy $\boldsymbol{E_0}$ and jump strength $\boldsymbol{J}$
   $\boldsymbol{E_0} = \mathbf{2} * \boldsymbol{rand} - \mathbf{1}, \quad \boldsymbol{J} = \mathbf{2}(\mathbf{1} - \boldsymbol{rand})$
   Update $\boldsymbol{E}$ using Eq. (3)
   if ($|\boldsymbol{E}| \geq \mathbf{1}$) **then** ▷ Exploration phase
    Update the location vector using Eq. (1)
   if ($|\boldsymbol{E}| < \mathbf{1}$) **then** ▷ Exploitation phase
    if ($\boldsymbol{r} \geq \mathbf{0.5}$ and $|\boldsymbol{E}| \geq \mathbf{0.5}$) **then** ▷ Soft besiege
     Update the location vector using Eq. (5)
    else if ($\boldsymbol{r} \geq \mathbf{0.5}$ and $|\boldsymbol{E}| < \mathbf{0.5}$) **then** ▷ Hard besiege
     Update the location vector using Eq. (7)
    else if ($\boldsymbol{r} < \mathbf{0.5}$ and $|\boldsymbol{E}| \geq \mathbf{0.5}$) **then** ▷ Soft besiege with progressive rapid dives
     Update the location vector using Eq. (11)
    else if ($\boldsymbol{r} < \mathbf{0.5}$ and $|\boldsymbol{E}| < \mathbf{0.5}$) **then** ▷ Hard besiege with progressive rapid dives
     Update the location vector using Eq. (12)
Return $\boldsymbol{X_{rabbit}}$



(a) The process in 2D space

(b) The process in 3D space

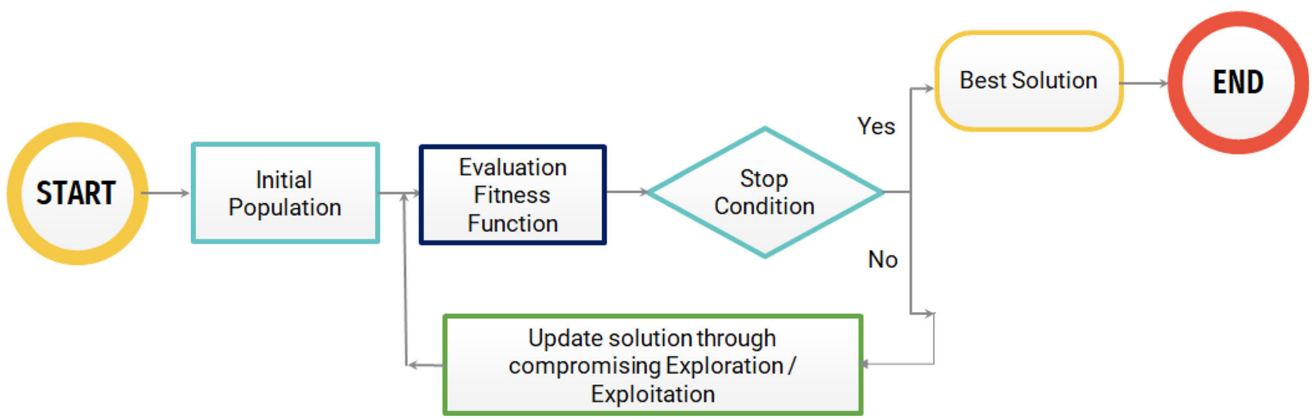


Fig. 8 An illustration of general vectors in the scenario of a hard besiege with progressive quick dives in 2D and 3D space [47]

The accompanying flowchart is in Fig. 9 which outlines the operation of HHO.

4 Proposed modification of HHO

Even though the original HHO has a special weight compared to other common approaches, while working on the local exploitation of feasible solutions, it may not reach an optimal set of scales between locally accurate exploitation and worldwide exploratory search because some of the used strategies are simple and can quickly converge, which might lead it to skip several of the optimal regions. Therefore, improvements in both the exploration and exploitation phases are necessary to find additional ideal locations and optimize the exploitation approach. Therefore, the following can serve as updates to the exploration phase (perching based on random locations) and exploitation strategies 1, 3, and 4:

4.1 Updating exploration phase (perching based on random locations):

To find prey, Harris' hawks use one of two methods during the exploration phase: perching at random in various areas and waiting for the right moment. When solving for the first line of Eq. (1) in the standard HHO, the hawks typically use the prey's location and that of any other household members. Setting each phase of the global search to be completely random is unlikely to happen. Consequently, it still might be unable to achieve a perfect balance between very specific local searches and broad, exploratory ones. In conventional HHO, the first line of Eq. (1) is dependent on the range of values for the variable from 0 to 1. There are two ways to find a hawk's prey: either immediately or after a lengthy search.

To avoid becoming stuck in a rut and settling for sub-optimal solutions, we exploit the features of logarithms and exponentials to generate random values in previously unexplored regions [74]. It can be seen that the exponential function of the current iteration concerning the upper bound of the iteration problem helps to increase the exploration rate in the early stages of the iteration. To avoid being stuck in a local solution, we generated random values using logarithms and exponentials to explore new regions more quickly and locations we had not yet reached.

$$X(t+1) = \begin{cases} X_m(t)*0.5-((X_{\text{rand}}(t)+r_1*X(t))+(X_{\text{rand}}(t)+r_1*X(t))) & q \ge 0.5 \\ (X_{\text{rabbit}}(t)-X_m(t))-r_3(LB+r_4(UB-LB)) & q < 0.5 \end{cases} \quad (15)$$

$$r_1 = 1 - log\left(1+e^{ub}\right) \quad (16)$$

Here $X(t+1)$ seems to be the coordinates of hawks within the next iteration, $X_{rabbit}(t)$ is the location for rabbit, $X(t)$ is just the current position vector for hawks, r_3, r_4, and q are randomized values within (0,1) updated in each iteration, LB, and UB display variable upper and lower boundaries, $X_{rand}(t)$ is chosen at the random hawk, and $X_m(t)$ is the population's average position.

4.2 Updating exploitation (strategy 1):

At this stage, the Harris hawks have barely cornered their victim before they pounce. The prey may be wary, but this stage ignores the rabbit's unnatural jumps to freedom that can occur. Since this stage contains local drops that cause local falls, we modified it by providing the probability of random jumps of prey that might occur, allowing us to quickly reach an optimal solution. The following equation ***Eq.*** 17 is an updated version of ***Eq.*** 7:

$$X(t+1) = 0.01*E*X_{\text{rabbit}}(t) - 0.1 * |J_1 * X_{\text{rabbit}}(t) - X_m(t)| \quad (17)$$

where $J_1 = (2*rand-1)$

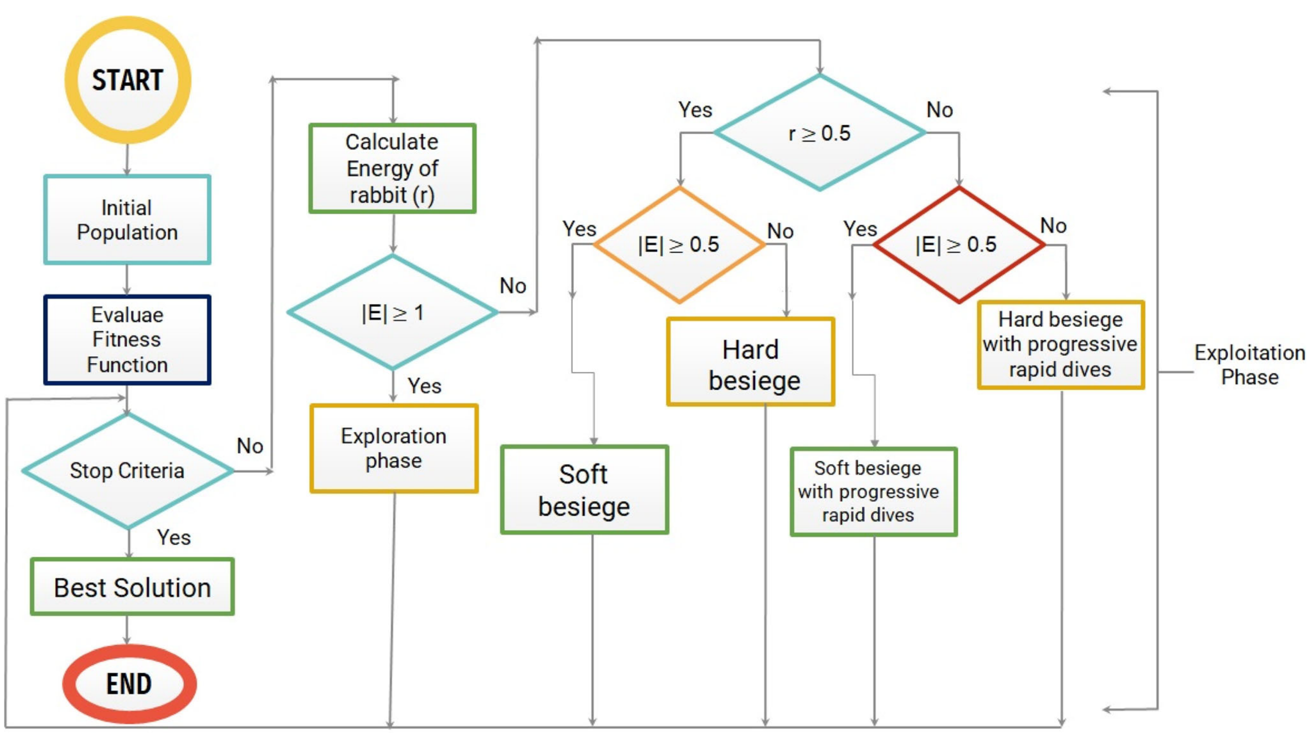


Fig. 9 HHO flowchart

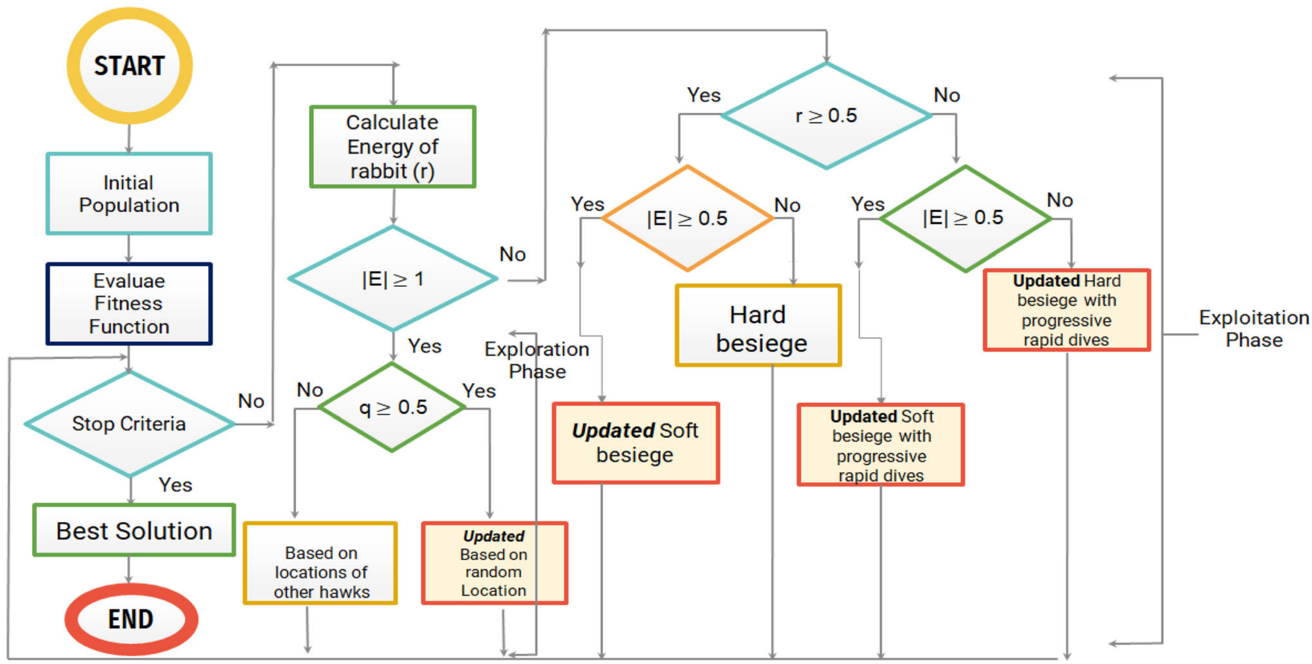


Fig. 10 IHHO flowchart

X_m is the population's average position. The *rand* parameter is a random number between 0 and 1, and J reflects the rabbit's randomized leap strength as it flees.

4.3 Updating exploitation (strategy 3):

To swiftly increase the diversity of options for the existing population, Strategy 3 is replaced by steps inspired by the average distance traveled [75], which helps to (1) discover new solutions, (2) prevent slipping into the local solution as a result of the larger diversity of results, and (3) reach the ideal solution in a faster way as a result of the faster increase in the existing population. It can be expressed by the following equation ***Eq.*** 18:

$$G_1 = i * \left(1 - \frac{t^{\frac{1}{i}}}{T^{\frac{1}{i}}}\right) \tag{18}$$

where i is a constant that dictates the exploitation accuracy as iterations go. That is, when i is higher, exploitation and local search occur faster and with more precision. The following equations Eq. (19), Eq. (20) is an updated version of Eq. 8 and Eq. 9:

$$\mathrm{Y} = \frac{\mathrm{X_{rabbit}(t)}}{\mathrm{i}} - \mathrm{rand} * \mathrm{E}|\mathrm{G_1} * \mathrm{JX}_{rabbit}(\mathrm{t}) - \mathrm{X(t)}| \tag{19}$$

$$Z = X_{\mathrm{rabbit}}(t) - \mathrm{rand} * E|G_1 * JX_{\mathrm{rabbit}}(t) - X(t)| - 1.5 * \frac{G_2}{\mathrm{Levy(dim)}} \tag{20}$$

where G_2 considers i in Eq. 18 to be a constant set to 6.

$$X(t+1) = \begin{cases} Y \ if \ F(Y) < F(X(t)) \\ Z \ if \ F(Z) < F(X(t)) \end{cases} \tag{21}$$

4.4 Updating exploitation (strategy 4)

Hawks make an effort to shorten the distance between their typical roosting and feeding grounds. To swiftly enhance the diversity of possibilities for the current population, Strategy 4 is replaced with steps inspired by the average distance traveled [61]. Because there is a greater variety of results, it is easier to (1) discover new solutions and (3)

Table 1 Parameter Setting

Algorithm	Setting
HHO	$\beta = 1.5$
IHHO	$\beta = 1.5$
BAT	$\alpha = 0.5, \gamma = 0.5, \rho = 0.001$
GWO	$r_1 = \mathrm{rand}, r_2 = \mathrm{rand}$
WOA	$r_1 = \mathrm{rand}, r_2 = \mathrm{rand}, c = 2 * r_2$
TLBO	$TF = \mathrm{rand}$
MFO	$b = 1$
MHHO	$\beta = 1.5, q = \mathrm{rand}$
LogHHO	$\beta = 1.5, q = \mathrm{rand}$
BHHO	$\beta = 1.5, q = \mathrm{rand}, F = 1.2 * \mathrm{rand} - 1$

locate the optimum answer faster because the present population is growing at a faster rate. The modeling of this behavior utilizes the Eq. (20):

$$X(t+1) = \begin{cases} Y \text{ if } F(\mathrm{Y}) < F(\mathrm{X(t)}) \\ Z \text{ if } F(\mathrm{Z}) < F(\mathrm{X(t)}) \end{cases} \tag{22}$$

where Y and Z are calculated according to new principles in Eq. (23) and Eq. (24)

$$Y = \frac{X_{\mathrm{rabbit}}(t)}{i} - \mathrm{rand} * E|G_1 * JX_{\mathrm{rabbit}}(t) - Xm(t)| \tag{23}$$

$$Z = X_{\mathrm{rabbit}}(t) - \mathrm{rand} * E|G_1 * JX_{\mathrm{rabbit}}(t) - Xm(t)| - 1.5 * \frac{G_2}{\mathrm{Levy(dim)}} \tag{24}$$

Algorithm 2 gives explains the details of Improved HHO (IHHO).

Algorithm 2 Pseudo-code of modified HHO algorithm (IHHO)

Inputs: The population size $\boldsymbol{N}$ and maximum number of iterations $\boldsymbol{T}$
Outputs: The location of rabbit and its fitness value
Initialize the random population $\boldsymbol{X_i}(\boldsymbol{i} = \mathbf{1, 2, \ldots, N})$
while (stopping condition is not met) **do**
  Calculate the fitness values of hawks
  Set $\boldsymbol{X_{rabbit}}$ as the location of rabbit (best location)
  for (each hawk ($\boldsymbol{X_i}$)) **do**
    Update the initial energy $\boldsymbol{E_0}$ and jump strength $\boldsymbol{J}$ ▷
$$\boldsymbol{E_0 = 2 * rand - 1}, \quad \boldsymbol{J = 2(1 - rand)}$$
    Update $\boldsymbol{E}$ using Eq. (13)
    if ($|\boldsymbol{E}| \geq \mathbf{1}$) **then** ▷ Exploration phase
      Update the location vector using Eq. (1)
    if ($|\boldsymbol{E}| < \mathbf{1}$) **then** ▷ Exploitation phase
      if ($\boldsymbol{r} \geq \mathbf{0.5}$ and $|\boldsymbol{E}| \geq \mathbf{0.5}$) **then** ▷ Soft besiege
        Update the location vector using Eq. (15)
      else if ($\boldsymbol{r} \geq \mathbf{0.5}$ and $|\boldsymbol{E}| < \mathbf{0.5}$) **then** ▷ Hard besiege
        Update the location vector using Eq. (7)
      else if ($\boldsymbol{r} < \mathbf{0.5}$ and $|\boldsymbol{E}| \geq \mathbf{0.5}$) **then** ▷ Soft besiege with progressive rapid dives
        Update the location vector using Eq. (19)
      else if ($\boldsymbol{r} < \mathbf{0.5}$ and $|\boldsymbol{E}| < \mathbf{0.5}$) **then** ▷ Hard besiege with progressive rapid dives
        Update the location vector using Eq. (20)
Return $\boldsymbol{X_{rabbit}}$

The accompanying flowchart is in Fig. 10 which outlines the operation of IHHO.

5 Experimental results and analysis

This section presents the experimental results of applying IHHO on various benchmarks and testing its performance against other algorithms. To ensure that all algorithms are given a fair chance, all algorithms have been applied for 30 separate runs and 500 iterations. All algorithms have the same population size of 30. The parameters of the different algorithms are given in Table 1.

The Friedman mean rank statistical test was used. Friedman recommended using a rank-based statistic to overcome the implicit assumption of normalcy in the analysis of variance[76]. The independence of several experiments leading to ranks is evaluated using Friedman's test. In practical use, the hypothesis testing can be obtained using an asymptotic analytical approximation valid for big N or large k, or from published tables containing accurate values for small k and N[77]. This assertion applies to both approximate tests of significance for comparing all $(^{k}{}_{2})$ = $k(k-1)/2$ pairs of treatments and tests for comparing $k-1$ treatments with a single control. The utility of asymptotic tests is dependent on their ability to approximate the exact sample distribution of the discrete rank sum difference statistic [78].

Table 2 Description of unimodal benchmark functions of CEC2005

Function	Range	F min
F1(x) = $\sum_{i=1}^{dim} x_i^2$	[−100, 100]	0
F2(x)=$\sum_{i=1}^{dim}\lvert x_i\rvert + \prod_{i=1}^{dim}\lvert x_i\rvert$	[−10, 10]	0
F3(x) = $\sum_{i=1}^{dim}\left(\sum_{n-1}^{i} x_n\right)^2$	[−100, 100]	0
F4(x) = $max_i\{\lvert x_i\rvert, 1 \le i \le dim\}$	[−100, 100]	0
F5(x) = $\sum_{i=1}^{dim-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	[−30, 30]	0
F6(x) = $\sum_{i=1}^{dim}\left([x_i+0.5]\right)^2$	[−100, 100]	0
F7(x) = $\sum_{i=1}^{dim} ix_i^4 + random[0, 1)$	[−1.28, 1.28]	0

Table 3 Description of High-dimensional multimodal benchmark functions of CEC2005

Function	Range	F min
F8(x) = $\sum_{i=1}^{dim} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	[−500, 500]	−418.983
F9(x) = $\sum_{i=1}^{dim}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	[−5.12, 5.12]	0
F10(x) = $-20exp\left(-0.2\sqrt{\frac{1}{dim}\sum_{i=1}^{dim} x_i^2}\right) - exp\left(\frac{1}{dim}\sum_{i=1}^{dim}\cos(2\pi x_i)\right) + 20 + e$	[−32, 32]	0
F11(x) = $\frac{1}{4000}\sum_{i=1}^{dim} x_i^2 - \prod_{i=1}^{dim} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	−600, 600]	0
F12(x) = $\frac{\pi}{dim}\left\{10\sin(\pi y_1) + \sum_{i=1}^{dim-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right] + (y_{dim}-1)^2\right\} + \sum_{i=1}^{dim} u(x_i, 10, 100, 4)$	[−50, 50]	0
$y_i = 1 + \frac{x_i+1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(y_i-1)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i-a)^m x_i < -a \end{cases}$		
F13(x)=$0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{dim}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right] + (x_{dim}-1)^2\left[1+\sin^2(3\pi x_{dim})\right]\right\} + \sum_{i=1}^{dim} u(x_i, 5, 100, 4)$	[−50, 50]	0

The precise null distribution improves significance testing in the comparison of Friedman rank sums and serves as a foundation for assessing theoretical approximations of the genuine distribution. For both many-one and all-pairs comparisons, the simple normal approximation matches the precise results the best among the large-sample approximation techniques [78]. For major occurrences farther into the distribution's tail, there may be a big discrepancy between the approximation p-values that are exact and normal. These kinds of events specifically happen when there are a lot of groups (k) and few blocks (n). Application of the normal approximation raises the likelihood of a Type-II error in a multiple testing setting with "large k and small n," leading to mistaken acceptance of the null hypothesis of "no difference."[79]

5.1 CEC 2005 benchmark functions

The proposed algorithm IHHO is examined by utilizing a well-studied collection of various benchmark functions taken from the IEEE CEC 2005 competition [80]. Unimodal (UM), High-dimensional multimodal (MM), and fixed-dimension multimodal are the three basic types of benchmark landscapes that are represented in this set. These are the UM operations (F1–F7), the MM Procedures (F8–F13), and functions of several dimensions that are fixed (F14–F23). Tables 2, 3, 4 demonstrate the mathematical formalism and features of Unimodal, High-dimensional multimodal, and fixed-dimension multimodal problems, respectively.

Table 4 Description of fixed-dimension multimodal benchmark functions of CEC2005

Function	Range	F min
$F14(x) = \left(\frac{1}{500} + \sum_{i=1}^{25} \frac{1}{i+\sum_{j=1}^{2}(x_j - a_{ji})^6}\right)^{-1}$	[−65.536, 65.536]	1
$F15(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	[−5, 5]	0.0003
$F16(x)= 4\,x_1^2 - 2.1\,x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4\,x_2^2 + 4\,x_2^4$	[−5, 5]	−1.0316
$F17(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	lb = [−5,0] ub = [10, 15]	0.398
$F18(x) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] *$ $[30 + (2x_1 - 3x_2)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)]$	[−5, 5]	3
$F19(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	[1, 3]	−3.86
$F20(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	[0,1]	−3.32
$F21(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	[0,10]	−10.1532
$F22(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	[0,10]	−10.4028
$F23(x) - \sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	[0,10]	−10.5363

5.1.1 IHHO vs HHO

The proposed IHHO has shown to be effective not just for lower-dimensional problems, but also for higher-dimensional jobs, and these benchmarks have been used in earlier studies to highlight the effect of dimension on the quality of solutions. In this experiment, original HHO and IHHO are utilized to solve the 30-dimensional, scaled versions of the UM, MM, and fixed-dimension multimodal F1-F23 multimodal test cases. For each metric, we record and compare the average AVG, STD, minimum, and maximum of the achieved results. The outcomes of IHHO against HHO to F1-F23 issues are shown in Table 5.

As compared to HHO, IHHO achieves vastly improved results attesting to the optimizer's higher performance. Convergence curves shown in Fig. 11 allow us to visually compare the convergence rates of IHHO and HHO algorithms. These diagrams depict typical objective values achieved by algorithms at various stages of their iterative process. Convergence plots depict iterations on the horizontal axis and objective function values on the vertical axis. The graphs make it clear that the suggested IHHO has a faster convergence rate than the original HHO method.

5.1.2 IHHO vs other algorithms

In this part, we compare the proposed IHHO to various optimization algorithms, namely **GWO** [39], **BAT** [40], **MFO** [42], **TLBO** [41], and **WOA** [43]. The outcomes are compared along the 30th dimension and from F14 to F23 with a fixed dimension. Table 6 shows the obtained experimental results for scalable problems. The table shows the average and standard deviation of the objective function. Figure 12 also presents the findings about the convergence of the various techniques.

As evidence of the proposed optimizer's superior performance, IHHO generates significantly better outcomes than competing methods. In particular, the theoretical best can be attained by IHHO for F1–F4. The purpose of unimodal benchmark problems is to evaluate a system's potential for exploitation. Using F1-F7 as experimental tests showed that the suggested algorithm is quite good at local searches.

It is clear from these tables and curves that the IHHO has solved the majority of the difficulties to their satisfaction. The IHHO's reliability is demonstrated by the low standard value observed for the vast majority of the problems. Because of this, the suggested IHHO has been proven to have higher accuracy in finding the global optimum by extensive experimental and statistical research.

Table 5 Comparison of results of IHHO and HHO on standard benchmark functions

Benchmark	optimizer	Mean	STD	MIN	MAX
F1	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	3.22E−96	1.72E−95	3.10E−119	3.47E−89
F2	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	1.83E−50	8.95E−50	9.40E−61	3.15E−49
F3	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	1.67E−74	2.94E−71	6.49E−103	6.795e−74
F4	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	1.32E−47	7.24E−47	2.8476e−56	9.9167e−48
F5	IHHO	**0.0023**	**0.0098289**	**1.63E−11**	**0.074396**
	HHO	0.010235	0.01594	9.54E−06	0.10659
F6	IHHO	**6.33E−06**	**1.12E−05**	**3.5965e−09**	**1.84E−04**
	HHO	1.92E−04	2.20E−04	6.3899e−07	9.51E−04
F7	IHHO	**5.84E−05**	**4.24E−05**	**5.10E−06**	**4.71E−04**
	HHO	1.25E−04	1.10E−04	4.48E−06	4.21E−04
F8	IHHO	**−12,569.3028**	**0.2026**	**−12,569.486**	**−12,568.705**
	HHO	−1.26E + 04	39.8386	−12,569.487	−12,334.059
F9	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
F10	IHHO	**4.44E−16**	**0.00E + 00**	**4.44E−16**	**4.44E−16**
	HHO	**4.44E−16**	**0.00E + 00**	**4.44E−16**	**4.44E−16**
F11	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
F12	IHHO	**2.2386e−07**	**2.03E−06**	**3.20E−10**	**7.72E−06**
	HHO	1.03E−05	1.36E−05	1.56E−09	5.99E−05
F13	IHHO	**7.15E−06**	**2.47E−05**	**1.47E−10**	**2.17E−04**
	HHO	1.25E−04	1.30E−04	1.54E−07	3.62E−04
F14	IHHO	**9.98E−01**	**1.90E−11**	**0.998**	**0.998**
	HHO	1.43E + 00	9.94E−01	0.998	5.9288
F15	IHHO	**3.20E−04**	**2.01E−05**	**3.08E−04**	**3.66E−04**
	HHO	3.41E−04	3.74E−05	3.08E−04	1.52E−03
F16	IHHO	**−1.0316**	**1.50E−09**	**−1.0316**	**−1.0316**
	HHO	**−1.0316**	**1.50E−09**	**−1.0316**	**−1.0316**
F17	IHHO	**3.98E−01**	1.90E−05	**3.98E−01**	**3.98E−01**
	HHO	**3.98E−01**	**1.85E−05**	**3.98E−01**	**3.98E−01**
F18	IHHO	**3.00E + 00**	2.11E−04	**3.00E + 00**	**3.00E + 00**
	HHO	**3.00E + 00**	**1.94E−06**	**3.00E + 00**	**3.00E + 00**
F19	IHHO	**−3.86E + 00**	6.94E−03	**−3.86E + 00**	−3.8121
	HHO	**−3.86E + 00**	**4.51E−03**	**−3.86E + 00**	**−3.8428**
F20	IHHO	**−3.24E + 00**	**6.45E−02**	**−3.31E + 00**	**−3.08E + 00**
	HHO	−3.06E + 00	1.86E−01	**−3.31E + 00**	−2.82E + 00
F21	IHHO	**−1.01E + 01**	**1.22E−02**	**−1.02E + 01**	**1.01E + 01**
	HHO	−5.53E + 00	1.46E + 00	−9.60E + 00	−5.02E + 00
F22	IHHO	**−1.04E + 01**	**9.92E−03**	**−1.04E + 01**	**−1.03E + 01**
	HHO	−5.15E + 00	1.14E + 00	−1.03E + 01	−5.05E + 00
F23	IHHO	**−1.05E + 01**	**0.0079963**	**−10.5363**	**−10.5074**
	HHO	−5.0064	0.6313	−5.1284	−2.3934

Best results are highlighted in bold

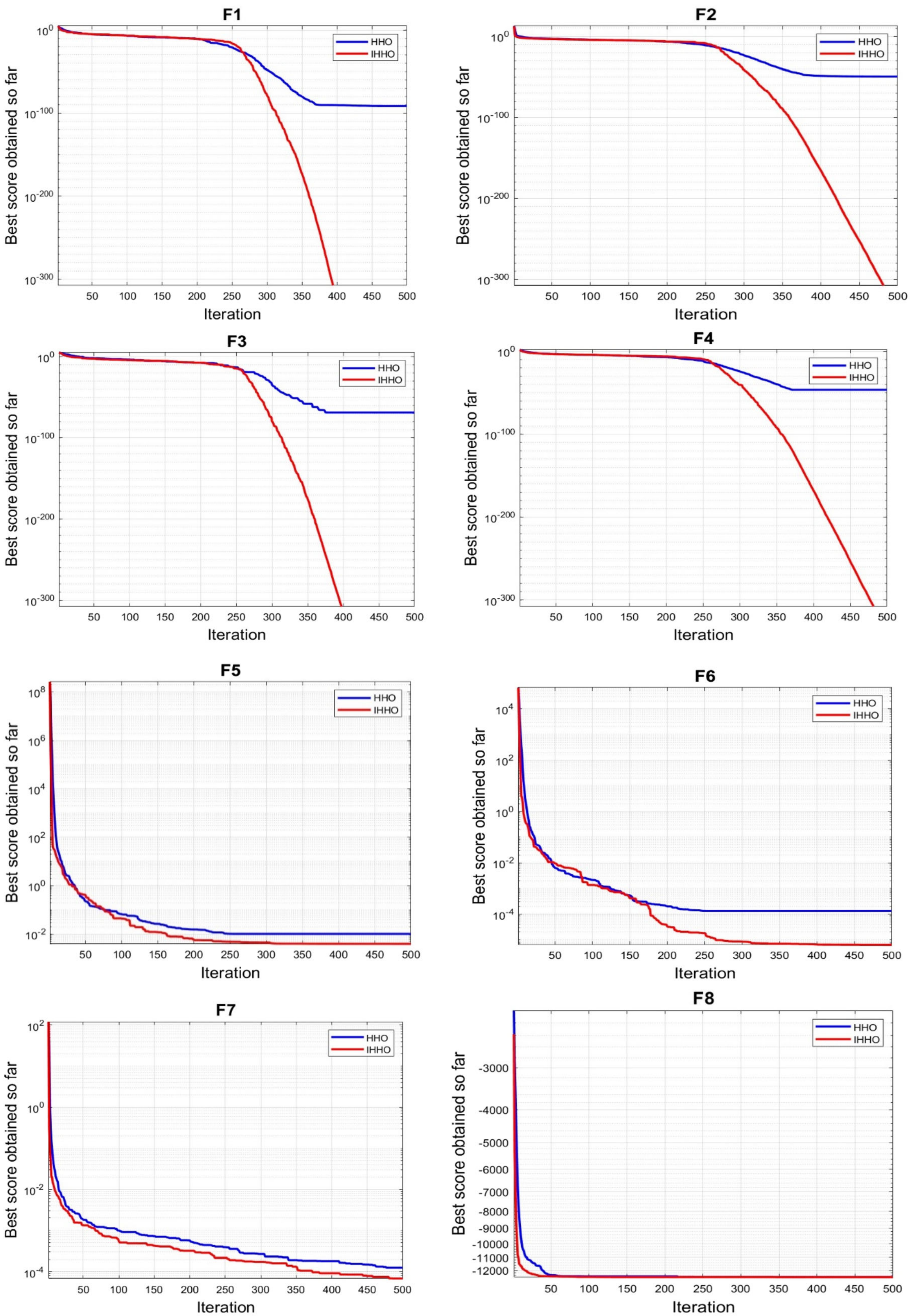


Fig. 11 Convergence curves for IHHO vs HHO on standard benchmark functions

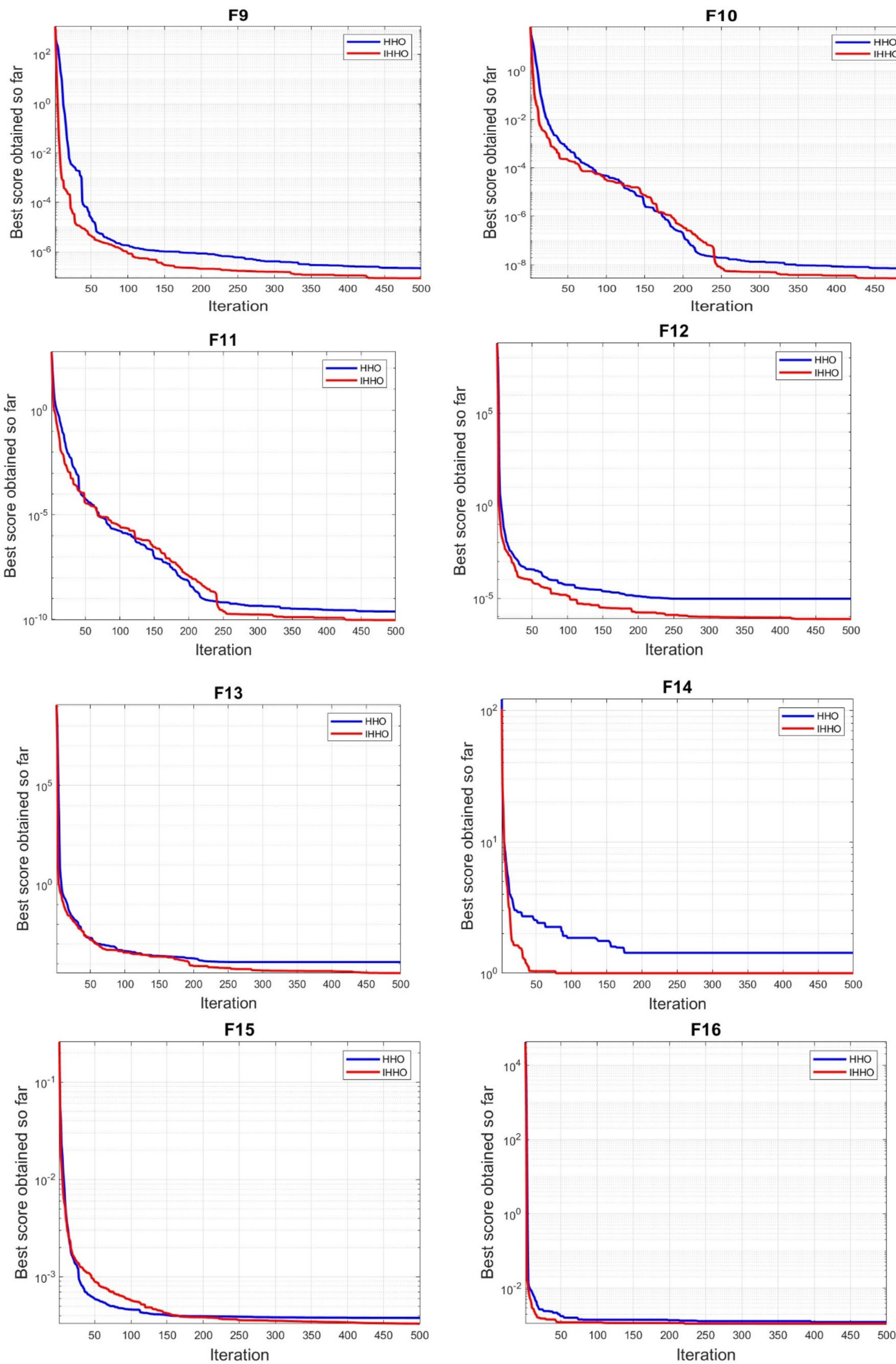


Fig. 11 continued

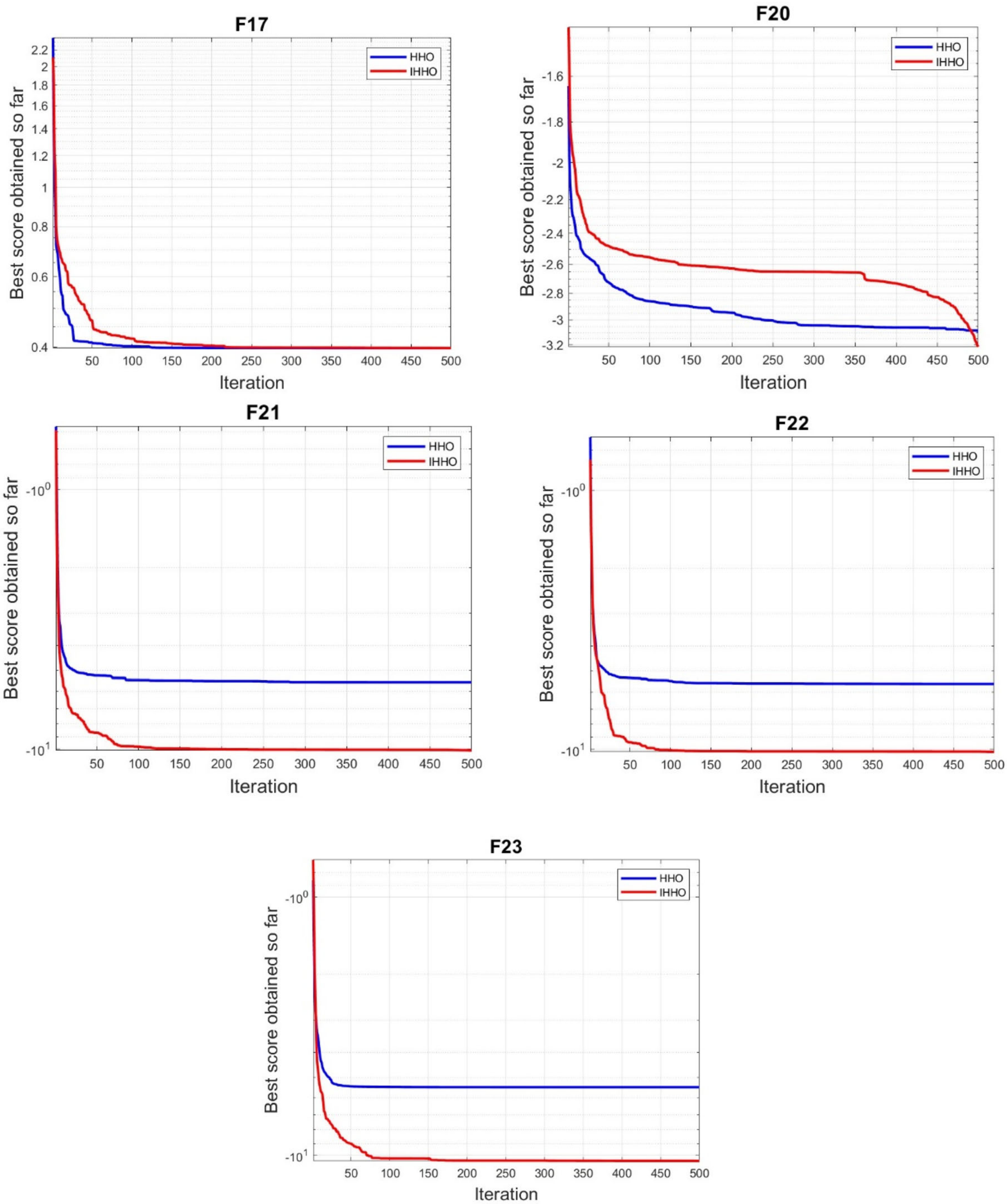


Fig. 11 continued

It is possible to visually compare the convergence rates of IHHO vs other algorithms by inspecting the curves depicted in Fig. 12. The diagrams show the usual objective values attained by algorithms at different iterations. The convergence plot displays the value of the goal function on the vertical axis and the number of iterations on the horizontal axis. The figures demonstrate that the proposed IHHO has a higher rate of convergence than other recent algorithms.

The purpose of multimodal functions is to assess a system's exploratory prowess; IHHO achieves the best mean values (except F19) as well as the best standard

Table 6 Comparison of results between IHHO and other algorithms on standard benchmark functions

Benchmark		IHHO	HHO	GWO[39]	BAT[81]	MFO[42]	TLBO[41]	WOA[43]
F1	AVG	**0.00E + 00**	3.22E−96	6.13E−28	3.61E + 04	2.34E + 03	1.74E−89	4.21E−72
	STD	**0.00E + 00**	1.72E−95	6.10E−28	7.02E + 03	6.26E + 03	1.78E−89	2.17E−71
	Rank	**1**	2	5	7	6	3	4
F2	AVG	**0.00E + 00**	1.83E−50	1.26E−16	1.28E + 07	31.4937	4.00E−45	2.03E−51
	STD	**0.00E + 00**	8.95E−50	1.05E−16	6.17E + 07	20.2128	2.62E−45	8.29E−51
	Rank	**1**	3	5	7	6	4	2
F3	AVG	**0.00E + 00**	1.67E−74	1.28E−05	8.61E + 04	1.93E + 04	7.19E−18	4.40E + 04
	STD	**0.00E + 00**	2.94E−71	3.00E−05	3.09E + 04	1.17E + 04	1.88E−17	1.57E + 04
	Rank	**1**	2	4	7	5	3	6
F4	AVG	**0.00E + 00**	1.32E−47	9.13E−07	71.6218	68.7883	1.37E−36	48.1917
	STD	**0.00E + 00**	7.24E−47	1.24E−06	8.79	8.9973	9.11E−37	30.7822
	Rank	**1**	2	4	7	6	3	5
F5	AVG	**0.0023**	0.010235	27.0242	1.06E + 08	1.01E + 04	25.3678	27.9931
	STD	**0.0098289**	0.01594	0.7532	5.32E + 07	2.73E + 04	0.4444	0.4596
	Rank	**1**	2	4	7	6	3	5
F6	AVG	**6.33E−06**	1.92E−04	0.7853	3.75E + 04	2.05E + 03	1.90E−04	0.337
	STD	**1.12E−05**	2.20E−04	0.3742	1.03E + 04	4.87E + 03	9.45E−04	0.2367
	Rank	**1**	3	5	7	6	2	4
F7	AVG	**5.84E−05**	1.25E−04	0.0021	77.8913	2.7092	0.0011	0.004
	STD	**4.24E−05**	1.10E−04	0.0014	33.2022	5.5266	5.23E−04	0.0064
	Rank	**1**	2	4	7	6	3	5
F8	AVG	**−12,569.302**	−1.26E + 04	−5.92E + 03	−5.81E + 03	−8.54E + 03	−7.51E + 03	−1.06E + 04
	STD	**0.2026**	39.8386	951.8653	4.28E + 03	833.415	1.02E + 03	1.87E + 03
	Rank	2	**1**	6	7	4	5	3
F9	AVG	**0.00E + 00**	**0.00E + 00**	3.4078	378.3514	154.0734	14.301	**0.00E + 00**
	STD	**0.00E + 00**	**0.00E + 00**	3.9093	58.3831	33.0526	4.9273	**0.00E + 00**
	Rank	**1**	**1**	4	7	6	5	**1**
F10	AVG	**4.44E−16**	**4.44E−16**	1.03E−13	19.6481	15.6966	6.01E−15	4.94E−15
	STD	**0.00E + 00**	**0.00E + 00**	2.10E−14	0.8417	6.6761	1.79E−15	2.63E−15
	Rank	**1**	**1**	5	7	6	4	3
F11	AVG	**0.00E + 00**	**0.00E + 00**	0.0018	347.3149	28.3382	**0.00E + 00**	0.0123
	STD	**0.00E + 00**	**0.00E + 00**	0.006	74.2876	47.8696	**0.00E + 00**	0.0474
	Rank	**1**	**1**	4	7	6	**1**	5
F12	AVG	**2.2386e−07**	1.03E−05	0.0416	1.34E + 08	8.53E + 06	0.0035	0.0186
	STD	**2.03E−06**	1.36E−05	0.0228	8.52E + 07	4.67E + 07	0.0189	0.0118
	Rank	**1**	2	5	7	6	3	4
F13	AVG	**7.15E−06**	1.25E−04	0.6901	3.16E + 08	189.5869	0.0594	0.4933
	STD	**2.47E−05**	1.30E−04	0.3017	1.50E + 08	718.4921	0.0697	0.2477
	Rank	**1**	2	5	7	6	3	4
F14	AVG	**9.98E−01**	1.43E + 00	3.4192	20.3851	3.1341	–	3.3146
	STD	**1.90E−11**	9.94E−01	3.4648	34.6933	2.5415	–	3.8762
	Rank	**1**	2	5	6	3	–	4
F15	AVG	**3.20E−04**	3.41E−04	0.0049	0.1045	0.0017	0.007	6.58E−04
	STD	**2.01E−05**	3.74E−05	0.0119	0.2543	0.0036	0.0202	3.29E−04
	Rank	**1**	2	5	7	4	6	3
F16	AVG	**−1.0316**	**−1.0316**	**−1.0316**	−0.52	**−1.0316**	−0.8177	**−1.0316**
	STD	1.50E−09	1.50E−09	2.52E−08	0.6137	6.78E−16	0.2707	**8.13E−10**
	Rank	**1**	**1**	**1**	7	1	6	**1**

Table 6 (continued)

Benchmark		IHHO	HHO	GWO[39]	BAT[81]	MFO[42]	TLBO[41]	WOA[43]
F17	AVG	**3.98E−01**	**3.98E−01**	0.3979	0.9501	0.3979	–	0.3979
	STD	1.90E−05	1.85E−05	8.66E−05	0.8328	**0**	–	1.49E−05
	Rank	4	4	**1**	6	**1**	–	**1**
F18	AVG	**3.00E + 00**	**3.00E + 00**	**3**	33.2527	**3**	132.3356	3.0001
	STD	2.11E−04	1.94E−06	4.48E−05	32.6853	**1.79E−15**	159.3076	3.58E−04
	Rank	**1**	**1**	**1**	6	**1**	7	5
F19	AVG	**−3.86E + 00**	**−3.86E + 00**	−3.8615	−3.6039	−3.8628	–	−3.8574
	STD	6.94E−03	4.51E−03	0.0024	0.2486	**2.71E−15**	–	0.0097
	Rank	3	3	2	6	**1**	–	5
F20	AVG	**−3.24E + 00**	−3.06E + 00	−3.2236	−1.9483	−3.2194	–	−3.2152
	STD	6.45E−02	1.86E−01	0.1085	0.5673	**0**	–	0.2145
	Rank	**1**	5	2	6	3	–	4
F21	AVG	**−1.01E + 01**	−5.53E + 00	−8.8873	−0.9957	−7.1271	–	−8.4063
	STD	**1.22E−02**	1.46E + 00	2.3685	0.8525	3.1953	–	2.6596
	Rank	**1**	5	2	6	4	–	3
F22	AVG	**−1.04E + 01**	−5.15E + 00	−10.4011	−0.9663	−7.6474	–	−7.4581
	STD	**9.92E−03**	1.14E + 00	9.11E−04	0.467	3.4917	–	3.0958
	Rank	2	5	**1**	6	3	–	4
F23	AVG	**−1.05E + 01**	−5.0064	**−1.05E + 01**	−1.329	−7.4001	–	−7.4382
	STD	0.0079963	0.6313	**8.85E−04**	0.6039	3.6882	–	3.4276
	Rank	**1**	5	**1**	6	4	–	3
Percentage		**1.3043**	**2.4783**	**3.5217**	**6.6522**	**4.3478**	**–**	**3.6522**
Total Rank		**1**	**2**	**3**	**6**	**5**	**–**	**4**

Best results are highlighted in bold

deviations (except F17, F20, and F23). The results of the F8-F23 test demonstrate that IHHO has superb exploration ability. As seen from Table 6, the total rank is 1, making the IHHO algorithm the best of all algorithms.

In conclusion, the IHHO algorithm is more efficient than the other examined algorithms for unimodal functions; this is because the random walk method effectively increases the algorithm's capacity to jump out of the local optimum.

5.1.3 IHHO vs other modifications of HHO

In this part, the proposed IHHO to other modifications of HHO was compared, namely **BHHO**[36]**, MHHO**[37], and **LogHHO**[38]. The outcomes are compared along the 30th dimension and from F14 to F23 with a fixed dimension. Experimental results addressing scalability issues are presented in Table 7. A summary of the objective function's mean and standard deviation is displayed below. Results on the convergence of the various methods are also shown in Fig. 13.

When compared to other modifications made to HHO, the mean values achieved by IHHO are superior. For all functions except F20, IHHO has the highest average values in accordance with BHHO, especially for (F1, F2, F3, F4). When compared to MHHO and LogHHO, the average values for IHHO are higher in every respect (F1–F4 and F20–F23). Except for F5 and F15, the median values achieved by IHHO are the highest. The results of the F8-F23 tests confirm IHHO's exceptional global capability. Table 7 shows that overall the IHHO algorithm ranks first, making it the best of its kind.

Since the random walk method effectively increases the algorithm's ability to jump from the local optimum, it can be concluded that the IHHO algorithm is more efficient than other modification algorithms analyzed for unimodal functions.

5.2 CEC 2017 benchmark functions

One of the most challenging evaluation tools is called CEC2017 [82], and it consists of thirty test jobs in a variety

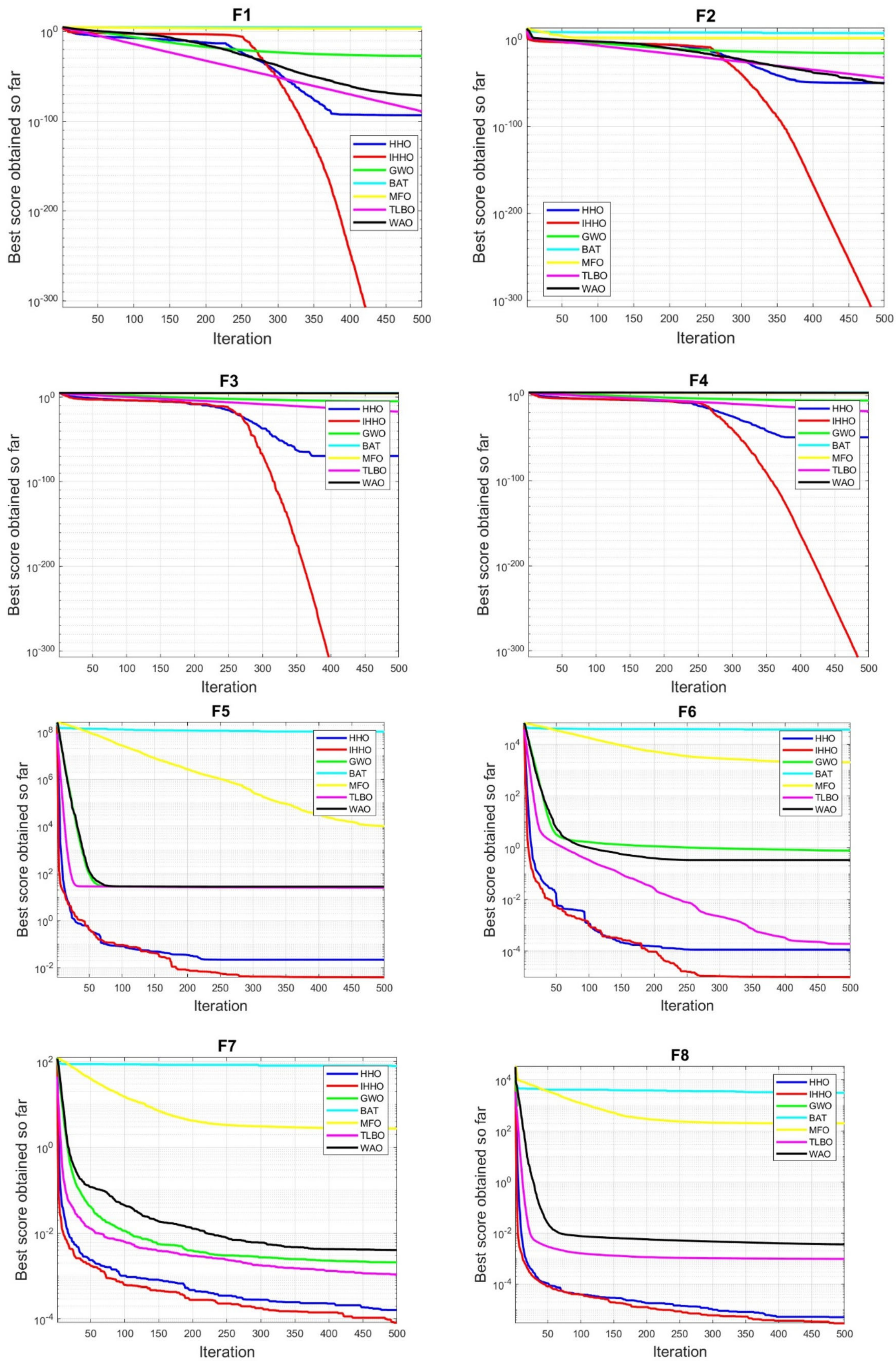


Fig. 12 Convergence curves for IHHO vs other algorithms on standard benchmark functions

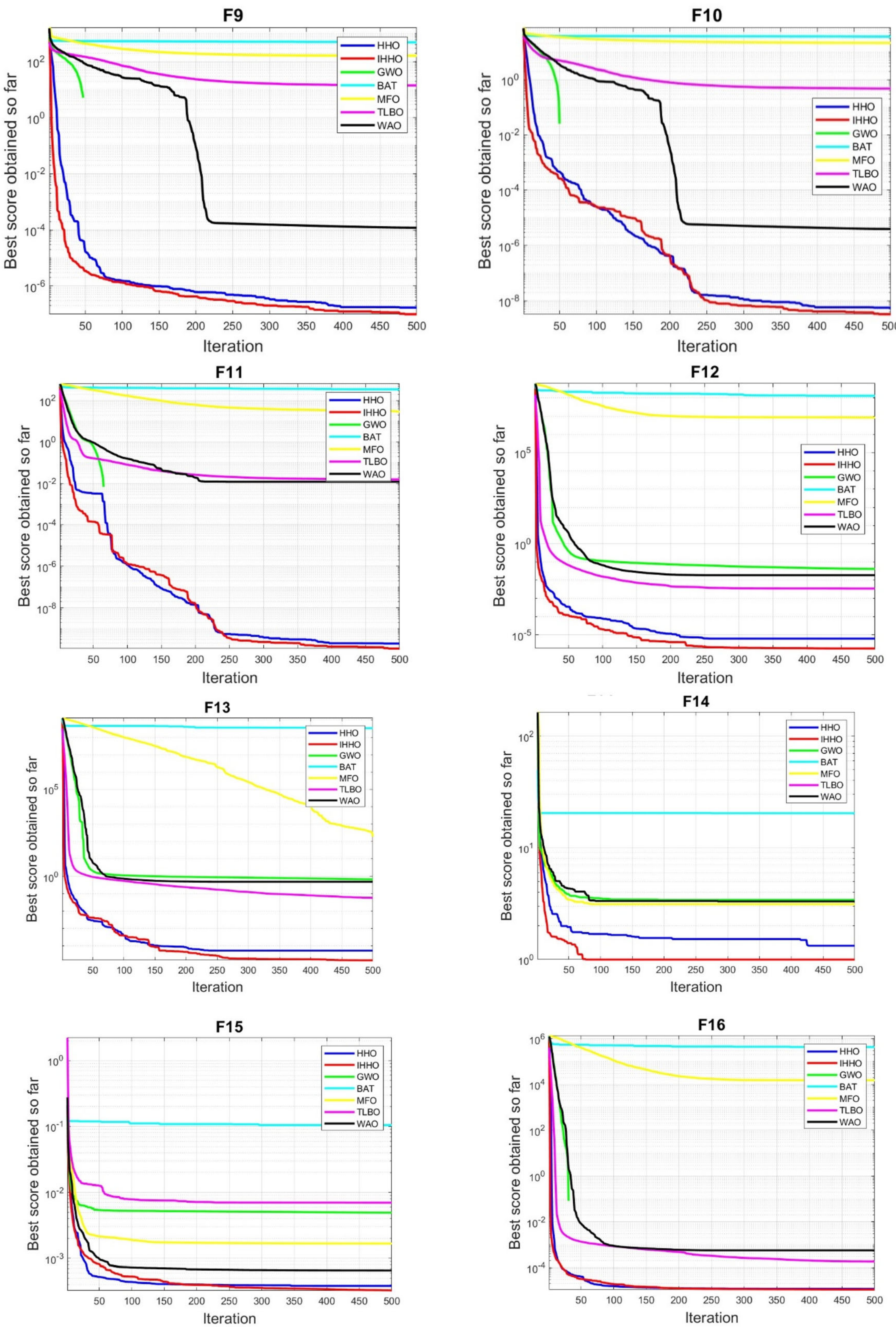


Fig. 12 continued

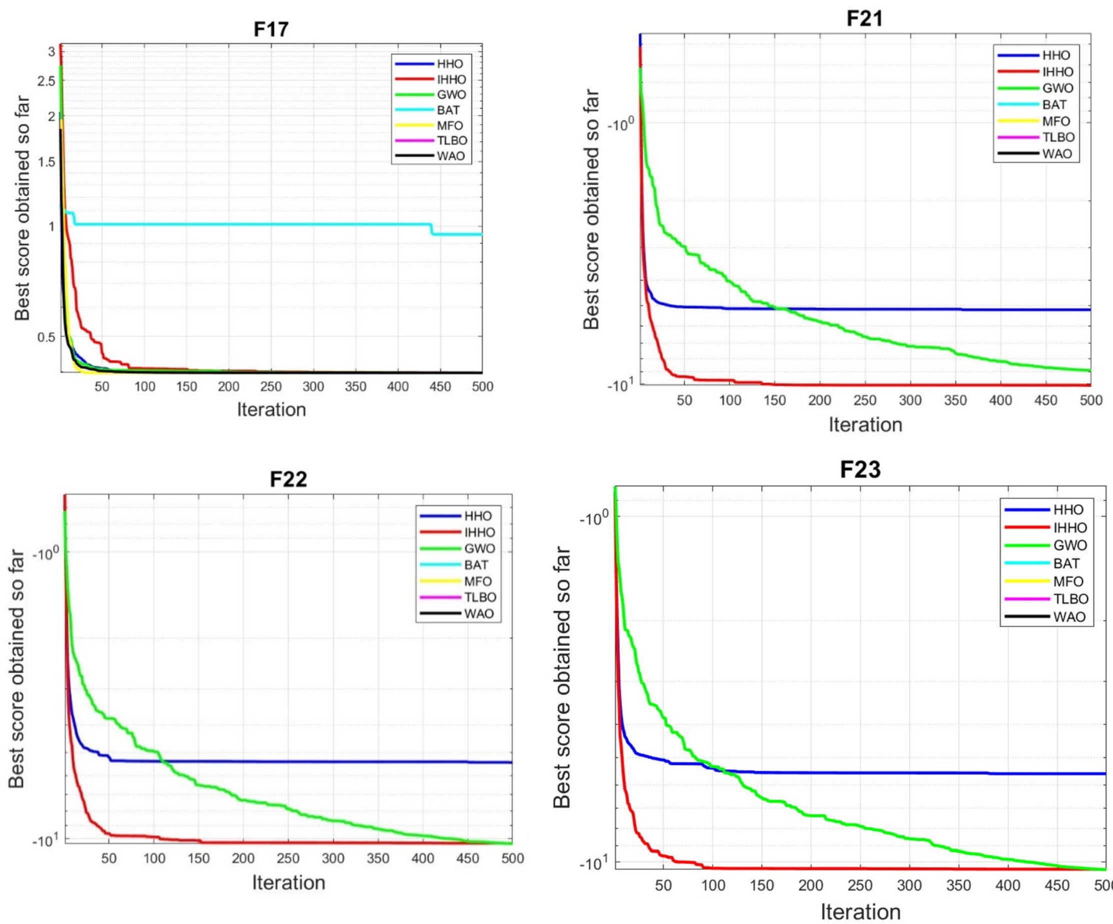


Fig. 12 continued

of formats as shown in Table 8, namely single-modal **(F1),** multimodal **(F3–F9),** hybrid **(F10–F19),** or combination **(F20–F30)**. In this investigation, we set the dimensions to 10.

5.2.1 IHHO vs HHO

The HHO approach is not very useful for solving difficult issues such as CEC 2017, but it does exceptionally well on routine, low-dimensional projects. In agreement with the first HHO, the newly proposed IHHO has proven to be effective in finding solutions to challenging challenges.

In Table 9, IHHO to HHO after putting it through its paces on the CEC2017 test functions (except F2, which is unstable) was compared. This was done to determine which of the two was superior.

As can be seen in Table 9, the suggested algorithm IHHO fared well in all functions with the original HHO. According to the tests, IHHO outperforms the original HHO algorithm in every used metric. Generally speaking, IHHO achieves the greatest median values, except for F8, F21, and F24. IHHO algorithm outperforms other analyzed modification methods for unimodal functions due to its increased capacity to escape the local optimum using the random walk approach.

Visually contrasting the convergence of the original HHO algorithm is done by examining the curves in Fig. 14. Typical objective values reached by algorithms throughout multiple iterations are depicted in these figures. On the vertical axis of the convergence plot is the value of the objective function, and on the horizontal axis is the number of iterations. As can be seen in the figures, the suggested IHHO has a faster speed of convergence than that of the original HHO.

5.2.2 IHHO vs other algorithms

Here, we evaluate the proposed IHHO in comparison to other popular optimization methods, namely the BAT algorithm [40], MFO [42], TLBO [41], and WOA [43].

Table 7 Comparison of results of IHHO vs other modifications of HHO on standard benchmark functions

Benchmark		IHHO	HHO	BHHO[36]	MHHO[37]	LogHHO[38]
F1	AVG	**0.00E + 00**	3.22E−96	7.85E−60	1.00E−244	1.34E−17
	STD	**0.00E + 00**	1.72E−95	8.71E−59	**0.00E + 00**	1.12E−16
	Rank	**1**	3	4	2	5
F2	AVG	**0.00E + 00**	1.83E−50	1.07E−29	3.07E−129	1.81E−26
	STD	**0.00E + 00**	8.95E−50	8.27E−29	4.20E−127	7.53E−25
	Rank	**1**	3	4	2	5
F3	AVG	**0.00E + 00**	1.67E−74	7.02E−40	1.32E−196	2.71E + 04
	STD	**0.00E + 00**	2.94E−71	3.43E−39	**0.00E + 00**	9.30E + 03
	Rank	**1**	3	4	2	5
F4	AVG	**0.00E + 00**	1.32E−47	5.51E−32	3.76E−123	14.4832
	STD	**0.00E + 00**	7.24E−47	4.65E−32	5.37E−120	16.8466
	Rank	**1**	3	4	2	5
F5	AVG	**0.0023**	0.010235	0.0063	27.5121	28.8904
	STD	**0.0098289**	0.01594	0.0084	0.5056	0.0531
	Rank	**1**	3	2	4	5
F6	AVG	**6.33E−06**	1.92E−04	2.58E−05	0.2962	4.1674
	STD	**1.12E−05**	2.20E−04	8.26E−05	0.2276	0.8661
	Rank	**1**	3	2	4	5
F7	AVG	**5.84E−05**	1.25E−04	0.00025828	0.00013326	0.05209
	STD	**4.24E−05**	1.10E−04	2.98E−04	2.70E−04	0.0385
	Rank	**1**	2	4	3	5
F8	AVG	**−12,569.302**	−1.26E + 04	−1.25E + 04	−1.05E + 04	−5.54E + 03
	STD	**0.2026**	39.8386	158.6328	2.10E + 03	951.9147
	Rank	**2**	1	3	4	5
F9	AVG	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	1.24E + 02
	STD	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	103.8591
	Rank	**1**	**1**	**1**	**1**	5
F10	AVG	**4.44E−16**	**4.44E−16**	**4.44E−16**	**4.44E−16**	2.62E−10
	STD	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	7.83E−10
	Rank	**1**	**1**	**1**	**1**	5
F11	AVG	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	6.04E−02
	STD	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	0.1202
	Rank	**1**	**1**	**1**	**1**	5
F12	AVG	**2.2386e−07**	1.03E−05	3.38E−06	1.30E−02	3.40E−01
	STD	**2.03E−06**	1.36E−05	3.63E−06	0.0058	2.2784
	Rank	**1**	3	2	4	5
F13	AVG	**7.15E−06**	1.25E−04	2.53E−05	2.82E−01	1.57E + 00
	STD	**2.47E−05**	1.30E−04	4.41E−05	0.1789	0.3978
	Rank	**1**	3	2	4	5
F14	AVG	**9.98E−01**	1.43E + 00	**9.98E−01**	5.08E + 00	6.82E + 00
	STD	1.90E−11	9.94E−01	**1.15E−13**	3.406	5.5571
	Rank	**1**	3	**1**	4	5
F15	AVG	3.20E−04	3.41E−04	3.52E−04	5.71E−04	4.07E−03
	STD	**2.01E−05**	3.74E−05	1.66E−04	1.40E−04	0.006
	Rank	**1**	2	3	4	5
F16	AVG	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**
	STD	1.50E−09	1.50E−09	1.58E−11	6.17E−13	1.61E−06
	Rank	**1**	**1**	**1**	**1**	**1**

Table 7 (continued)

Benchmark		IHHO	HHO	BHHO[36]	MHHO[37]	LogHHO[38]
F17	AVG	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**
	STD	1.90E−05	1.85E−05	4.46E−08	4.10E−07	1.20E−04
	Rank	**1**	**1**	**1**	**1**	**1**
F18	AVG	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**
	STD	2.11E−04	1.94E−06	1.15E−06	3.33E−07	0.0037
	Rank	**1**	**1**	**1**	**1**	**1**
F19	AVG	**−3.86E + 00**	**−3.86E + 00**	**−3.86E + 00**	**−3.86E + 00**	**−3.86E + 00**
	STD	**6.94E−03**	4.51E−03	0.0022	0.0012	0.0039
	Rank	**1**	**1**	**1**	**1**	**1**
F20	AVG	−3.24E + 00	−3.06E + 00	**−3.2778**	−3.26E + 00	−3.07E + 00
	STD	**6.45E−02**	1.86E−01	0.0664	0.0657	0.1252
	Rank	3	5	**1**	2	4
F21	AVG	**−1.01E + 01**	−5.53E + 00	−9.98E + 00	−8.03E + 00	−6.43E + 00
	STD	**1.22E−02**	1.46E + 00	1.21E−04	2.5693	2.3338
	Rank	**1**	5	2	3	4
F22	AVG	**−1.04E + 01**	−5.15E + 00	**−1.04E + 01**	−8.25E + 00	−6.00E + 00
	STD	**9.92E−03**	1.14E + 00	7.38E−05	3.0059	2.8613
	Rank	**1**	5	**1**	3	4
F23	AVG	**−1.05E + 01**	−5.0064	**−1.05E + 01**	−7.38E + 00	−5.66E + 00
	STD	0.0079963	0.6313	1.24E−04	3.2505	3.1372
	Rank	**1**	5	**1**	3	4
Percentage		**1.1304**	**2.5652**	**2.0435**	**2.4783**	**4.1304**
Total Rank		**1**	**4**	**2**	**3**	**5**

Best results are highlighted in bold

Experimental results addressing scalability issues are presented in Table 10, which presents a summary of the objective function's mean and standard deviation. Results on the convergence of the various methods are also shown in Fig. 15.

The average values attained by using IHHO are higher than those obtained using other recent algorithms. According to the TLBO and BAT algorithms, IHHO typically has the greatest average values across the board for all frequencies. IHHO has higher average values than the WOA algorithm does across the board, but especially for (F1, F3, F9, F12, F14, F19, F30). In every case, the average values for IHHO are greater than those for the MFO method, particularly for (F1 to F10, F19, F26, F30). Except for F18 and F27, IHHO consistently produces the highest average results. Table 10 shows that the performance of the proposed IHHO algorithm outperforms the performance of the original HHO method as well as all other algorithms and the rank for all equations is 1.

The IHHO technique outperforms other analyzed modification methods for unimodal functions due to its increased capacity to escape the local optimum using the random walk approach. Because the random walk approach effectively boosts the algorithm's capacity to jump out of the local optimum, the IHHO algorithm outperforms the other algorithms studied for unimodal functions.

5.2.3 IHHO vs other modifications of HHO

Here, the proposed IHHO to other HHO variants such as BHHO [54], MHHO[37], and LogHHO [38]. Table 11 displays experimental results that address scalability concerns in terms of the mean and standard deviation of the objective function evaluated. Figure 16 also displays the convergence results for the different approaches.

Using CEC2017 functions, we compared the suggested change IHHO to other modifications of the original HHO and found that IHHO provided much better performance. The IHHO algorithm is superior to the other examined variations for functions because of random walk technique significantly improves the algorithm's ability to escape the local optimum. Table 11 shows that the performance of the IHHO algorithm outperforms the performance of the original HHO method as well as all other HHO modification algorithms and the rank for all equations is 1.

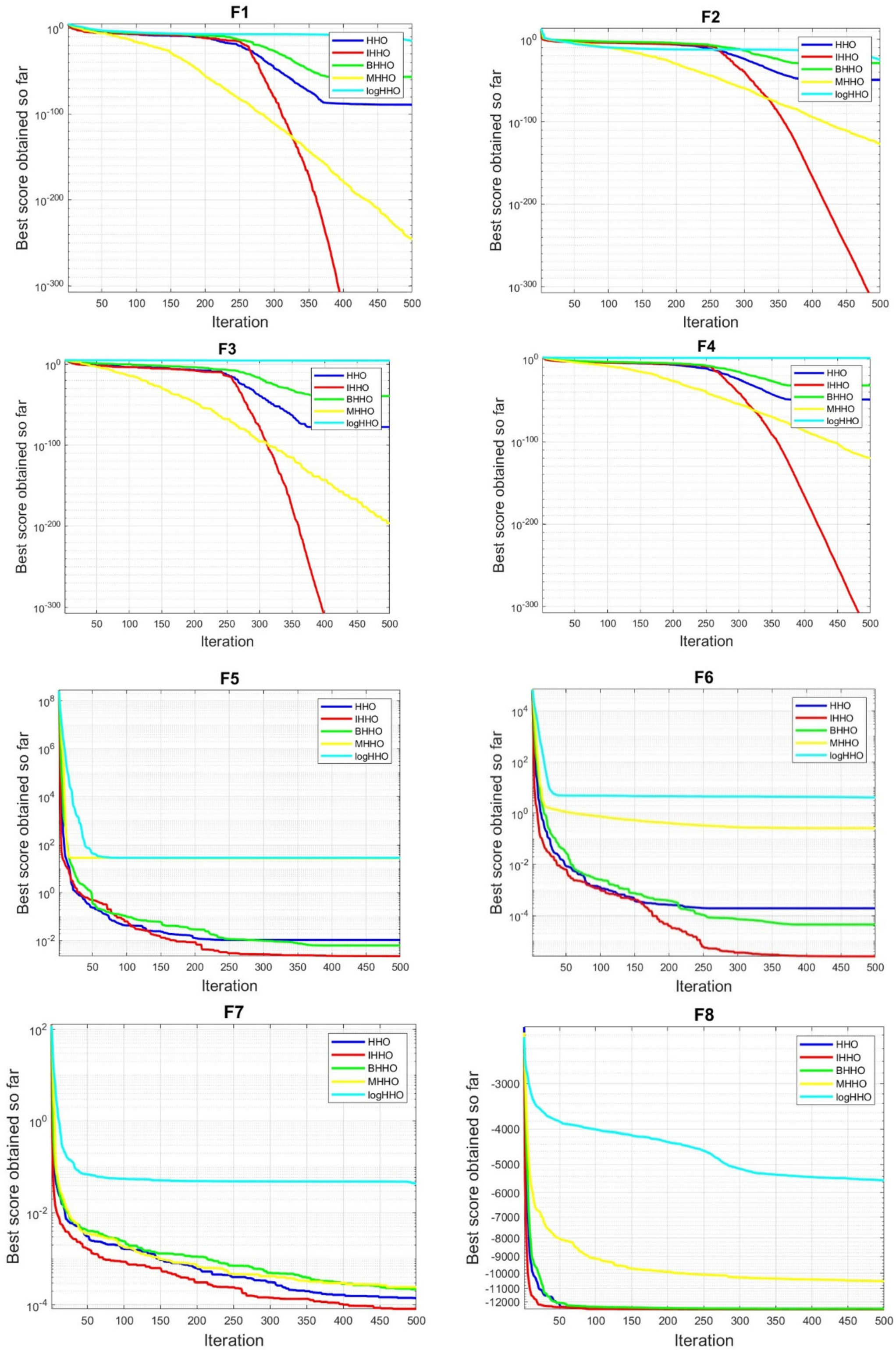


Fig. 13 Convergence curves for IHHO vs HHO modifications on standard benchmark functions

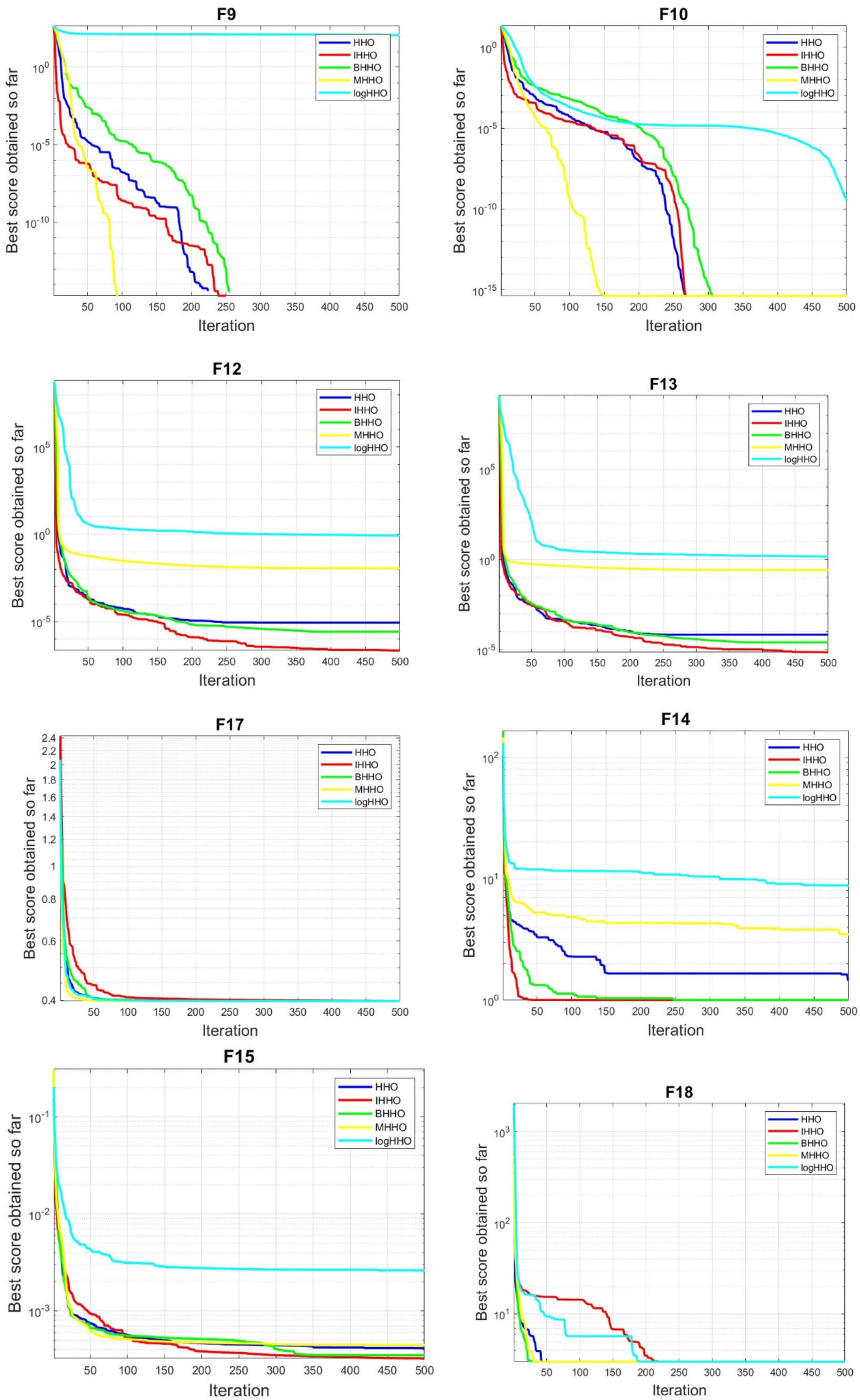


Fig. 13 continued

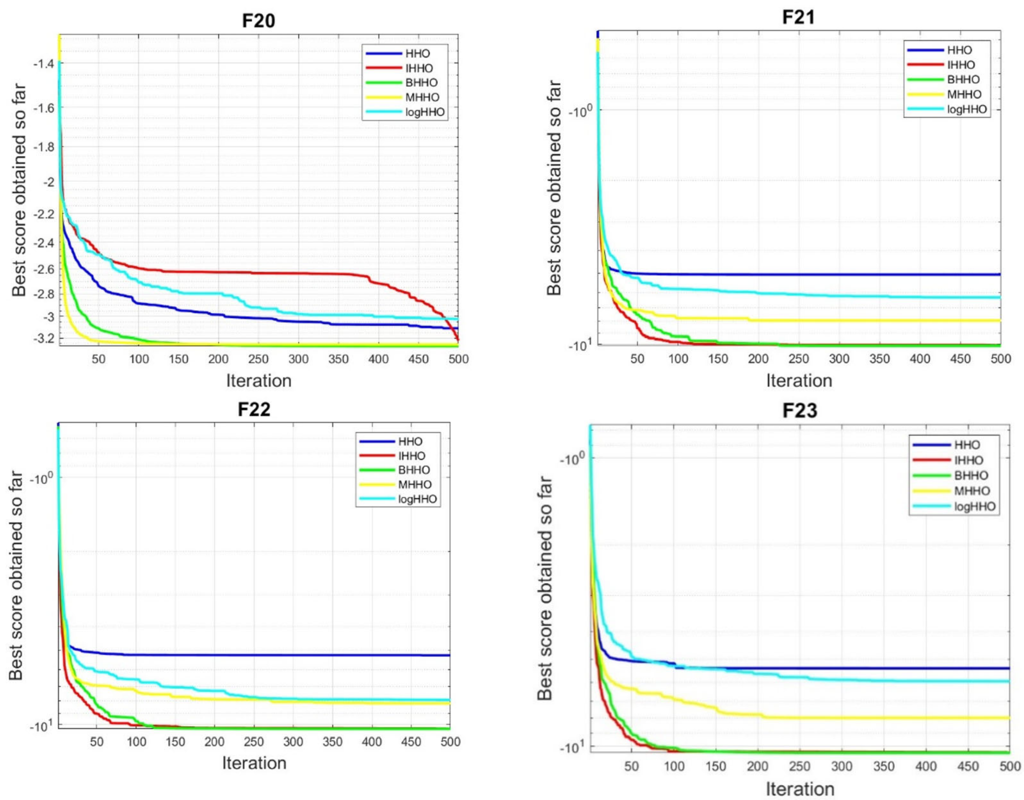


Fig. 13 continued

Table 8 Description of CEC 2017

Benchmark	Name	Dim	Range	F min
Unimodal functions				
F1	Shifted and Rotated Bent Cigar Function	10	[−100, 100]	100
Multimodal functions				
F3	Shifted and Rotated Rosenbrock's Function	10	[−100, 100]	300
F4	Shifted and Rotated Rastrigin's Function	10	[−100, 100]	400
F5	Shifted and Rotated Expanded Scaffer's F6 Function	10	[−100, 100]	500
F6	Shifted and Rotated Lunacek BiRastriginFunction	10	[−100, 100]	600
F7	Shifted and Rotated Non-Continuous Rastrigin's Function	10	[−100, 100]	700
F8	Shifted and Rotated Levy Function	10	[−100, 100]	800
F9	Shifted and Rotated Schwefel's Function	10	[−100, 100]	900
Hybrid Functions (N is the basic number of functions)				
F10	Hybrid Function 1 (N = 3)	10	[−100, 100]	1000
F11	Hybrid Function 2 (N = 3)	10	[−100, 100]	1100
F12	Hybrid Function 3 (N = 3)	10	[−100, 100]	1200
F13	Hybrid Function 4 (N = 4)	10	[−100, 100]	1300
F14	Hybrid Function 5 (N = 4)	10	[−100, 100]	1400
F15	Hybrid Function 6 (N = 4)	10	[−100, 100]	1500
F16	Hybrid Function 6 (N = 5)	10	[−100, 100]	1600
F17	Hybrid Function 6 (N = 5)	10	[−100, 100]	1700
F18	Hybrid Function 6 (N = 5)	10	[−100, 100]	1800

Table 8 (continued)

Benchmark	Name	Dim	Range	F min
F19	Hybrid Function 6 (N = 6)	10	[−100, 100]	1900
Composite Functions (N is the basic number of functions)				
F20	Composite Function 1 (N = 3)	10	[−100, 100]	2000
F21	Composite Function 2 (N = 3)	10	[−100, 100]	2100
F22	Composite Function 3 (N = 4)	10	[−100, 100]	2200
F23	Composite Function 3 (N = 4)	10	[−100, 100]	2300
F24	Composite Function 5 (N = 5)	10	[−100, 100]	2400
F25	Composite Function 6 (N = 5)	10	[−100, 100]	2500
F26	Composite Function 7 (N = 6)	10	[−100, 100]	2600
F27	Composite Function 8 (N = 6)	10	[−100, 100]	2700
F28	Composite Function 9 (N = 6)	10	[−100, 100]	2800
F29	Composite Function 10 (N = 3)	10	[−100, 100]	2900
F30	Composite Function 11 (N = 3)	10	[−100, 100]	3000

Table 9 Comparison of results of IHHO and HHO on CEC2017

Benchmark	optimizer	Mean	STD	MIN	MAX
F1	IHHO	**2.17E + 04**	**8666.9596**	5703.7119	57,055.175
	HHO	9.91E + 06	6.56E + 06	3.46E + 05	5.15E + 07
F3	IHHO	**300.2742**	**0.18863**	300.083	301.4204
	HHO	790.2067	498.1986	304.5662	1688.0762
F4	IHHO	**405.4989**	**12.3172**	400.0072	480.1062
	HHO	441.6495	41.2061	400.6048	598.8978
F5	IHHO	**523.0827**	**9.7255**	503.0001	550.7509
	HHO	560.9182	19.6982	522.2605	594.5018
F6	IHHO	**601.2425**	**1.6446**	600.1019	621.2993
	HHO	641.3173	11.3266	618.6942	655.219
F7	IHHO	**740.3267**	**13.5653**	719.1461	785.6197
	HHO	803.3728	15.7372	733.6704	827.7796
F8	IHHO	**822.0687**	8.89E + 00	8.08E + 02	8.47E + 02
	HHO	832.37	**6.4979**	810.0621	850.1502
F9	IHHO	**911.342**	**30.507**	900.0069	1224.0933
	HHO	1.52E + 03	214.4839	995.9883	2083.4515
F10	IHHO	**1.59E + 03**	**264.6654**	1255.325	2089.5942
	HHO	2.08E + 03	3.09E + 02	1376.439	2.52E + 03
F11	IHHO	**1123.0802**	**1.28E + 01**	1.11E + 03	1.31E + 03
	HHO	1198.231	89.4682	1132.6825	1417.3129
F12	IHHO	**1.35E + 06**	**1,377,342.073**	8339.2102	8,639,874.3521
	HHO	3.45E + 06	3.67E + 06	9.54E + 04	1.38E + 07
F13	IHHO	**1.16E + 04**	**9346.6084**	2718.654	40,856.3071
	HHO	1.37E + 04	1.03E + 04	2.17E + 03	8.92E + 04
F14	IHHO	**1.6116e + 03**	**350.1143**	1454.1619	2255.0692
	HHO	2.28E + 03	1.08E + 03	1464.2827	5.94E + 03
F15	IHHO	**4.93E + 03**	**1959.5614**	1742.1252	9149.3695
	HHO	8.89E + 03	3.18E + 03	2552.419	11,565.6309

Table 9 (continued)

Benchmark	optimizer	Mean	STD	MIN	MAX
F16	IHHO	**1.7763e + 03**	**147.1462**	1601.6159	2076.1548
	HHO	1.92E + 03	1.64E + 02	1635.5939	2204.3433
F17	IHHO	**1.76E + 03**	**18.3225**	1728.0332	1810.0079
	HHO	1.80E + 03	7.52E + 01	1738.1283	2038.5601
F18	IHHO	**1.6275e + 04**	**10,835.5077**	2581.9049	54,986.4211
	HHO	1.7680e + 04	1.36E + 04	2.10E + 03	4.13E + 04
F19	IHHO	**7.19E + 03**	**6647.6979**	1971.5397	28,369.1424
	HHO	3.59E + 04	6.38E + 04	4404.9096	57,366.6541
F20	IHHO	**2.07E + 03**	**49.3032**	2025.3629	2260.5939
	HHO	2.20E + 03	8.87E + 01	2080.0135	2446.0105
F21	IHHO	**2.28E + 03**	60.2607	2200.0087	2351.5616
	HHO	2352.2351	**30.8292**	2207.2868	2385.9732
F22	IHHO	**2.29E + 03**	**24.7819**	2225.9647	2998.756
	HHO	2376.921	237.0443	2274.969	3132.1365
F23	IHHO	**2622.469**	**10.5649**	2606.0671	2665.1997
	HHO	2681.7155	31.1696	2621.561	2785.2163
F24	IHHO	**2719.675**	92.5488	2500.3054	2789.4829
	HHO	2822.6602	**78.5028**	2501.3983	2944.7632
F25	IHHO	**2928.3439**	**23.1189**	2897.752	2970.8463
	HHO	2.93E + 03	6.88E + 01	2898.9264	2954.9231
F26	IHHO	**2891.4091**	**250.8454**	2602.3498	3968.6475
	HHO	3690.8329	566.4344	2612.9373	4499.7978
F27	IHHO	**3.10E + 03**	**8.18E + 00**	3090.7815	3182.0987
	HHO	3.18E + 03	4.66E + 01	3099.3257	3328.3845
F28	IHHO	**3.36E + 03**	**9.25E + 01**	3100.333	3736.1829
	HHO	3.44E + 03	1.40E + 02	3103.5738	3749.371
F29	IHHO	**3.24E + 03**	**57.3682**	3143.2242	3410.3241
	HHO	3.33E + 03	8.88E + 01	3189.2023	3553.63
F30	IHHO	**2.45E + 05**	**4.07E + 05**	7.40E + 03	3.28E + 06
	HHO	2.05E + 06	3.01E + 06	7.36E + 04	9.39E + 06

Best results are highlighted in bold

Figure 16 shows typical objective values attained by algorithms over the course of numerous iterations, allowing for a visual comparison of the convergence rates. A convergence plot displays the value of the goal function along the vertical axis and the number of repetitions along the horizontal axis. The results clearly show that the proposed IHHO has a higher rate of convergence compared to the baseline HHO method and various modifications.

5.3 CEC 2019 benchmark functions

The benchmark test functions from the IEEE Congress on Evolutionary Computation serve as the basis for IHHO's evaluation (CEC-C06, 2019 Competition) [83]. Functions for 2019 Benchmark Tests in the CEC-C06 Standard As a secondary measure, IHHO is applied to 10 state-of-the-art CEC benchmark test functions.

Unlike CEC01-CEC03, CEC04-CEC10 undergoes both translation and rotation. There is, however, scalability across the board for the test procedures. Although CEC04–CEC10 have the same dimensions as a 10-dimensional constrained optimization problem in the [-100, 100] border range, CEC01–CEC03 have different dimensions, as indicated in Table 12.

5.3.1 IHHO vs HHO

In Table 13, a comparison of the results of IHHO vs HHO after putting it through its paces on the CEC2019 test functions is presented. This was done so that we could

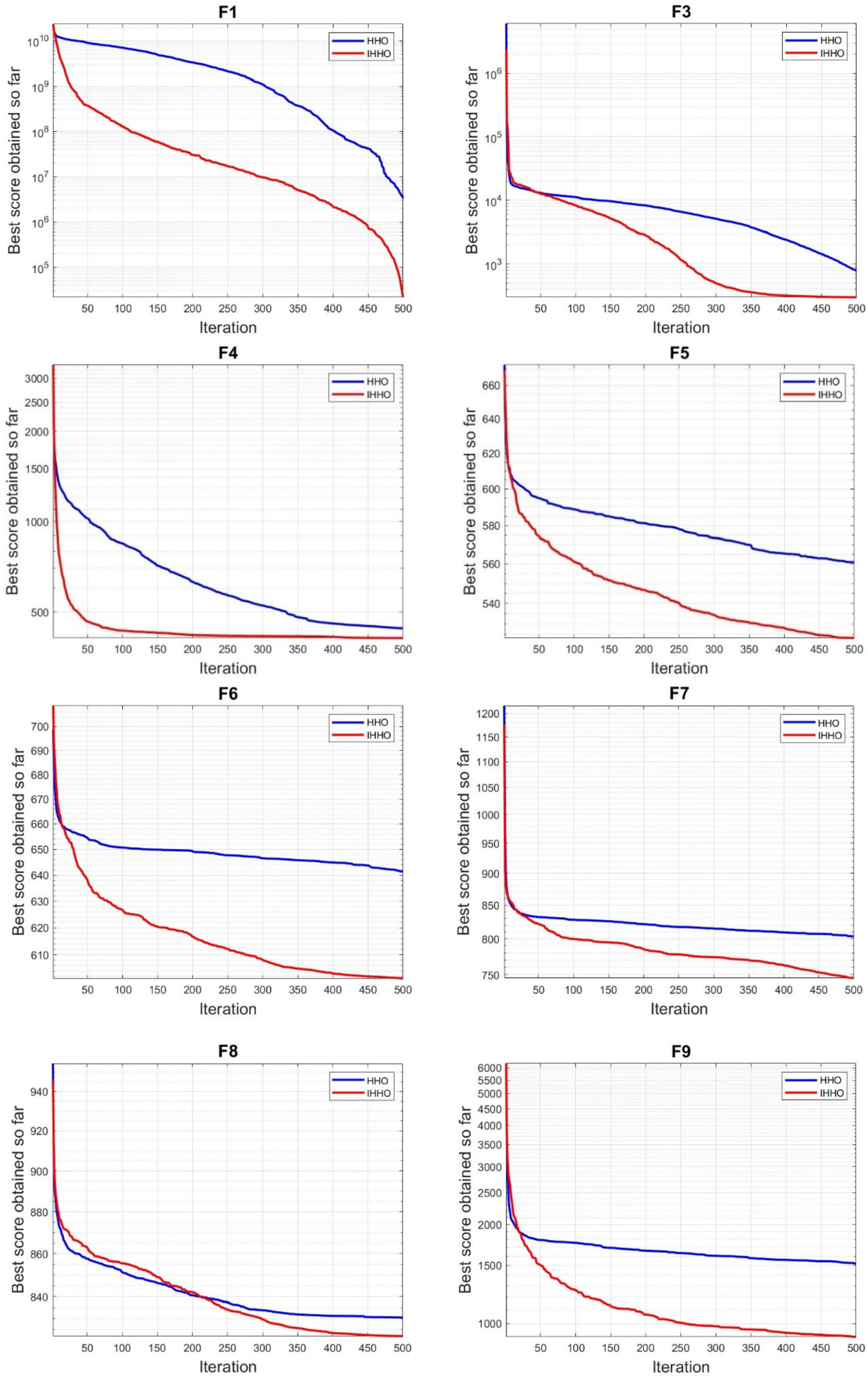


Fig. 14 Convergence curves for IHHO vs HHO on CEC2017

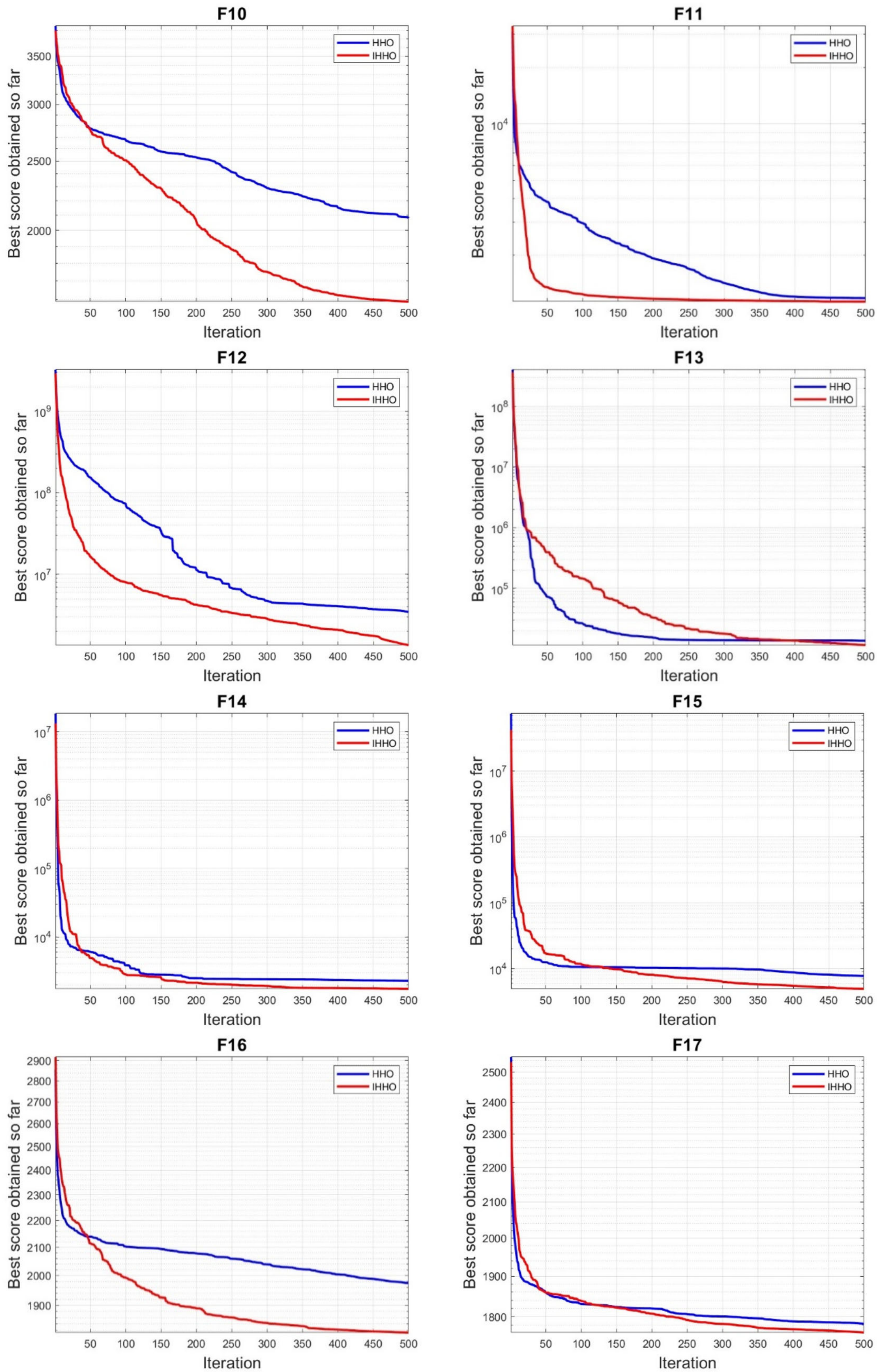


Fig. 14 continued

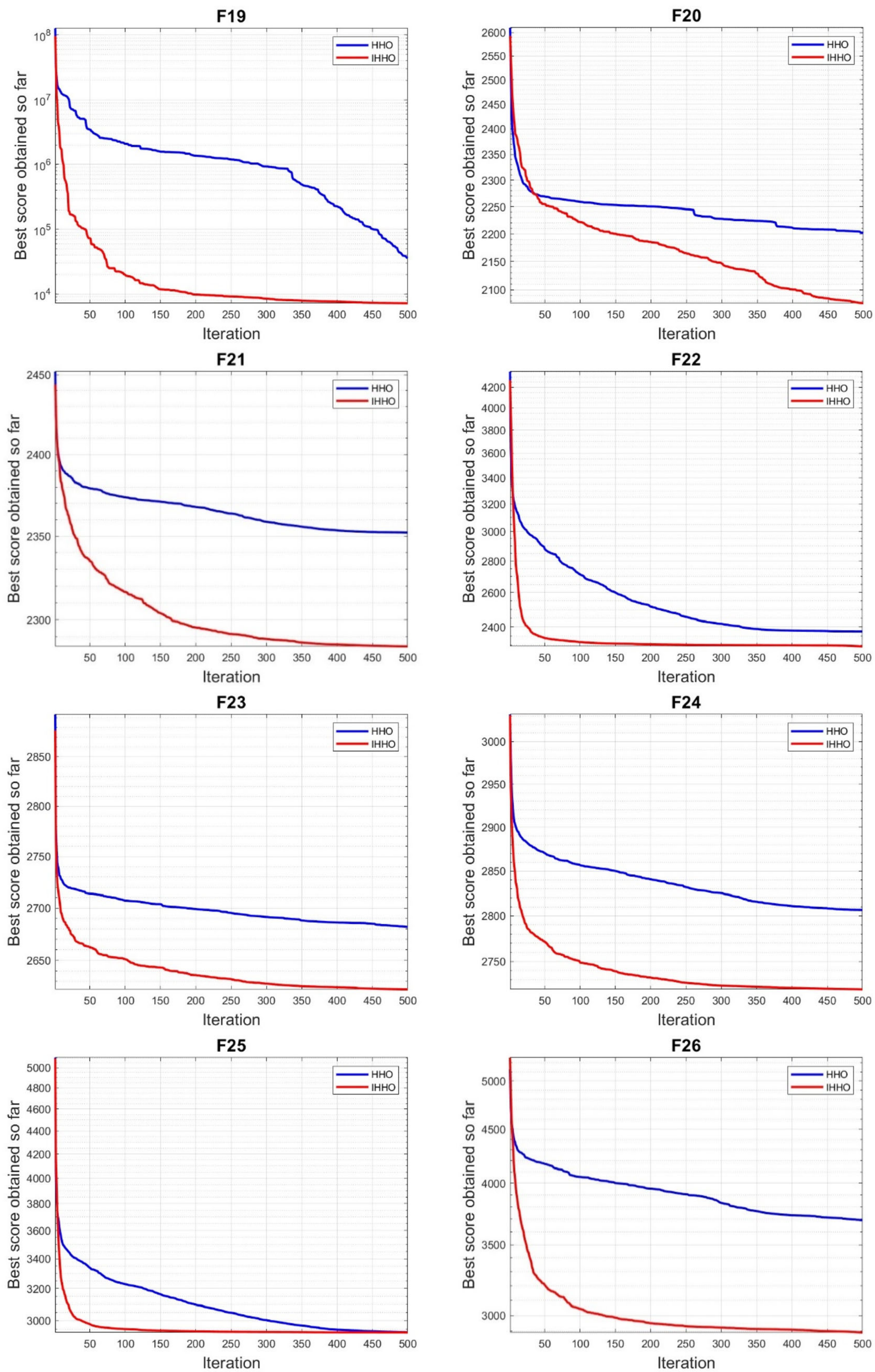


Fig. 14 continued

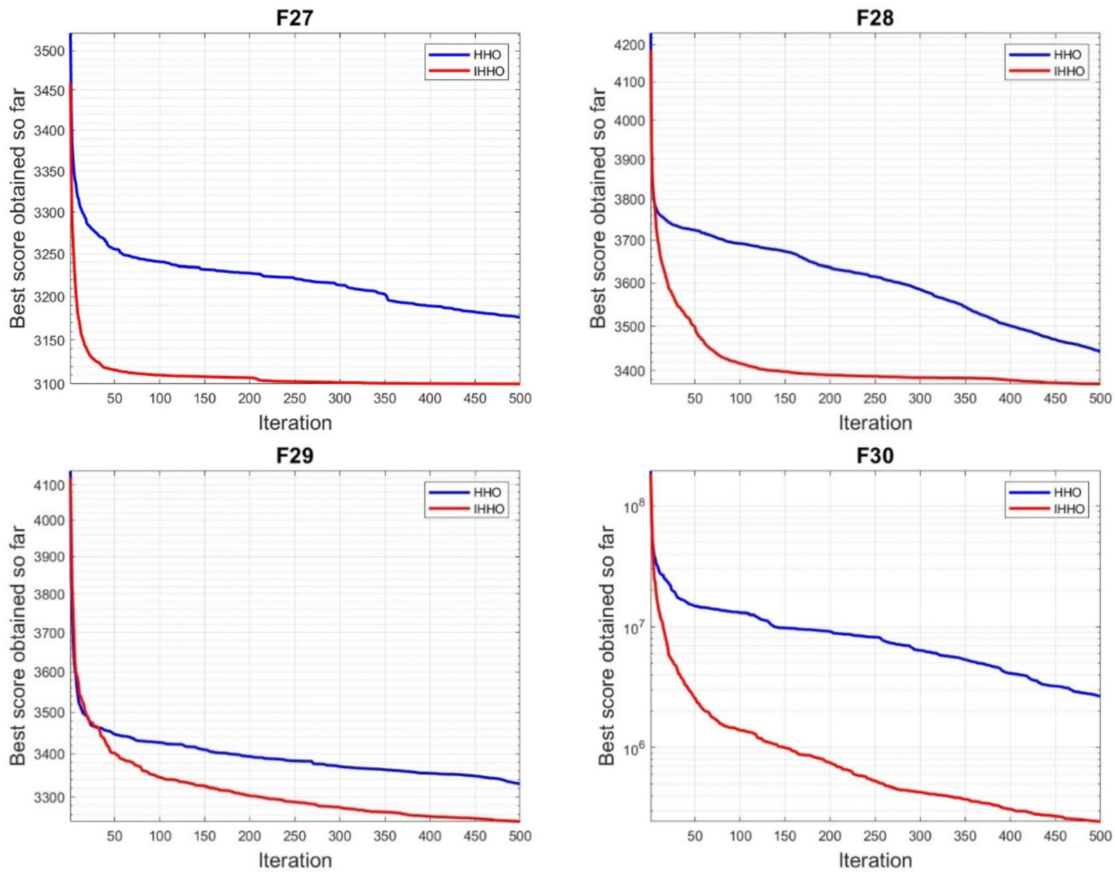


Fig. 14 continued

Table 10 Comparison of results between IHHO and other algorithms on CEC2017

Benchmark		IHHO	HHO	BAT[40]	MFO[42]	TLBO[41]	WOA[43]
F1	AVG	**2.17E + 04**	9.91E + 06	1.97E + 10	2.03E + 08	5.14E + 09	7.45E + 07
	STD	**8666.9596**	6.56E + 06	7.85E + 09	4.69E + 08	3.85E + 09	7.89E + 07
	Rank	**1**	2	6	4	5	3
F3	AVG	**300.2742**	790.2067	7.63E + 04	1.10E + 04	6.58E + 04	6.21E + 03
	STD	**0.18863**	498.1986	3.03E + 04	1.42E + 04	1.25E + 04	4.32E + 03
	Rank	**1**	2	6	4	5	3
F4	AVG	**405.4989**	441.6495	2.19E + 03	420.3863	1.40E + 03	457.3243
	STD	**12.3172**	41.2061	1.02E + 03	30.7835	994.2903	57.1373
	Rank	**1**	3	6	2	5	4
F5	AVG	**523.0827**	560.9182	631.5225	531.9467	706.7606	562.1442
	STD	**9.7255**	19.6982	28.0893	12.3566	50.1425	18.2611
	Rank	**1**	3	5	2	6	4
F6	AVG	**601.2425**	641.3173	685.5707	604.9002	641.6572	639.4219
	STD	**1.6446**	11.3266	22.1028	7.252	9.4312	14.5207
	Rank	**1**	4	6	2	5	3

Table 10 (continued)

Benchmark		IHHO	HHO	BAT[40]	MFO[42]	TLBO[41]	WOA[43]
F7	AVG	**740.3267**	803.3728	1.06E + 03	743.3572	1.10E + 03	778.9864
	STD	**13.5653**	15.7372	100.0475	14.3218	70.0049	27.5069
	Rank	**1**	4	5	2	6	3
F8	AVG	**822.0687**	832.37	920.1498	832.2549	965.7717	845.293
	STD	8.89E + 00	**6.4979**	21.3481	11.9416	34.9736	17.2473
	Rank	**1**	3	5	2	6	4
F9	AVG	**911.342**	1.52E + 03	4.44E + 03	1.18E + 03	6.07E + 03	1.84E + 03
	STD	**30.507**	214.4839	1.44E + 03	334.0652	1.48E + 03	865.9342
	Rank	**1**	3	5	2	6	4
F10	AVG	**1.59E + 03**	2.08E + 03	3.38E + 03	1.96E + 03	7.94E + 03	2.28E + 03
	STD	**264.6654**	3.09E + 02	376.9642	340.7119	1.03E + 03	344.1797
	Rank	**1**	3	5	2	6	4
F11	AVG	**1123.0802**	1198.231	2.09E + 04	1.29E + 03	1.77E + 03	1.22E + 03
	STD	**1.28E + 01**	89.4682	2.46E + 04	340.1241	779.766	61.4991
	Rank	**1**	2	6	4	5	3
F12	AVG	**1.35E + 06**	3.45E + 06	1.12E + 09	2.54E + 06	8.26E + 07	4.35E + 06
	STD	**1,377,342.073**	3.67E + 06	8.52E + 08	4.31E + 06	1.20E + 08	4.41E + 06
	Rank	**1**	3	6	2	5	4
F13	AVG	**1.16E + 04**	1.37E + 04	1.46E + 08	1.35E + 04	1.35E + 06	1.68E + 04
	STD	**9346.6084**	1.03E + 04	1.75E + 08	1.16E + 04	3.92E + 06	1.16E + 04
	Rank	**1**	3	6	2	5	4
F14	AVG	**1.6116e + 03**	2.28E + 03	8.07E + 03	5.29E + 03	4.83E + 04	2.75E + 03
	STD	**350.1143**	1.08E + 03	8.65E + 03	6.04E + 03	5.84E + 04	1.35E + 03
	Rank	**1**	2	5	4	6	3
F15	AVG	**4.93E + 03**	8.89E + 03	7.33E + 04	1.05E + 04	1.24E + 04	1.33E + 04
	STD	**1959.5614**	3.18E + 03	9.42E + 04	1.27E + 04	1.33E + 04	8.42E + 03
	Rank	**1**	2	6	3	4	5
F16	AVG	**1.7763e + 03**	1.92E + 03	2.58E + 03	1.83E + 03	2.97E + 03	1.92E + 03
	STD	147.1462	1.64E + 02	176.5344	**138.7651**	427.2746	136.0335
	Rank	**1**	3	5	2	6	3
F17	AVG	**1.76E + 03**	1.80E + 03	2.24E + 03	1.78E + 03	2.39E + 03	1.84E + 03
	STD	**18.3225**	7.52E + 01	183.3624	57.0341	304.3592	81.4459
	Rank	**1**	3	5	2	6	4
F18	AVG	**1.6275e + 04**	1.7680e + 04	3.63E + 08	2.23E + 04	1.22E + 06	1.84E + 04
	STD	10,835.5077	1.36E + 04	5.44E + 08	1.58E + 04	2.40E + 06	**9.53E + 03**
	Rank	**1**	2	6	4	5	3
F19	AVG	**7.19E + 03**	3.59E + 04	1.76E + 07	1.91E + 04	1.36E + 04	4.18E + 04
	STD	**6647.6979**	6.38E + 04	7.63E + 07	1.89E + 04	1.67E + 04	6.30E + 04
	Rank	**1**	4	6	3	2	5
F20	AVG	**2.07E + 03**	2.20E + 03	2.46E + 03	2.10E + 03	2.57E + 03	2.19E + 03
	STD	**49.3032**	8.87E + 01	121.9582	60.9747	185.4204	67.5468
	Rank	**1**	4	5	2	6	3
F21	AVG	**2.28E + 03**	2352.2351	2.42E + 03	2.31E + 03	2.48E + 03	2.33E + 03
	STD	60.2607	30.8292	50.8977	46.0485	**45.8557**	51.7961
	Rank	**1**	4	5	2	6	3
F22	AVG	**2.29E + 03**	2376.921	3.98E + 03	2.31E + 03	4.58E + 03	2.40E + 03
	STD	24.7819	237.0443	652.9891	**14.2038**	1.85E + 03	304.6141
	Rank	**1**	3	5	2	6	4

Table 10 (continued)

Benchmark		IHHO	HHO	BAT[40]	MFO[42]	TLBO[41]	WOA[43]
F23	AVG	**2622.469**	2681.7155	2.87E + 03	2.63E + 03	2.99E + 03	2.65E + 03
	STD	**10.5649**	31.1696	65.3471	10.9218	72.5619	18.752
	Rank	**1**	4	5	2	6	3
F24	AVG	**2719.675**	2822.6602	3.02E + 03	2.76E + 03	3.15E + 03	2.78E + 03
	STD	92.5488	78.5028	112.3605	**50.1124**	94.752	62.1067
	Rank	**1**	4	5	2	6	3
F25	AVG	**2928.3439**	2.93E + 03	4.27E + 03	2.93E + 03	3.14E + 03	2.98E + 03
	STD	23.1189	6.88E + 01	678.1578	**21.6507**	89.0578	39.0756
	Rank	**1**	2	6	2	5	4
F26	AVG	**2891.4091**	3690.8329	4.94E + 03	3.08E + 03	7.18E + 03	3.70E + 03
	STD	250.8454	566.4344	541.4107	**231.8064**	921.5414	595.1693
	Rank	**1**	3	5	2	6	4
F27	AVG	**3.10E + 03**	3.18E + 03	3.35E + 03	**3.10E + 03**	3.39E + 03	3.15E + 03
	STD	8.18E + 00	4.66E + 01	121.1479	**4.5736**	95.9356	48.5217
	Rank	**1**	4	5	**1**	6	3
F28	AVG	**3.36E + 03**	3.44E + 03	3.82E + 03	3.37E + 03	3.75E + 03	3.43E + 03
	STD	**9.25E + 01**	1.40E + 02	307.4426	95.193	283.7452	171.5057
	Rank	**1**	4	6	2	5	3
F29	AVG	**3.24E + 03**	3.33E + 03	3.81E + 03	3.2542e + 03	4.50E + 03	3.36E + 03
	STD	**57.3682**	8.88E + 01	253.5827	65.0163	362.6095	94.5697
	Rank	**1**	3	5	2	6	4
F30	AVG	**2.45E + 05**	2.05E + 06	6.07E + 07	9.07E + 05	1.37E + 06	1.70E + 06
	STD	**4.07E + 05**	3.01E + 06	5.95E + 07	7.62E + 05	5.82E + 06	1.90E + 06
	Rank	**1**	5	6	2	3	4
Percentage		**1**	3.137931	5.448276	2.37931	5.344827586	3.586206897
Total Rank		**1**	3	6	2	5	4

Best results are highlighted in bold

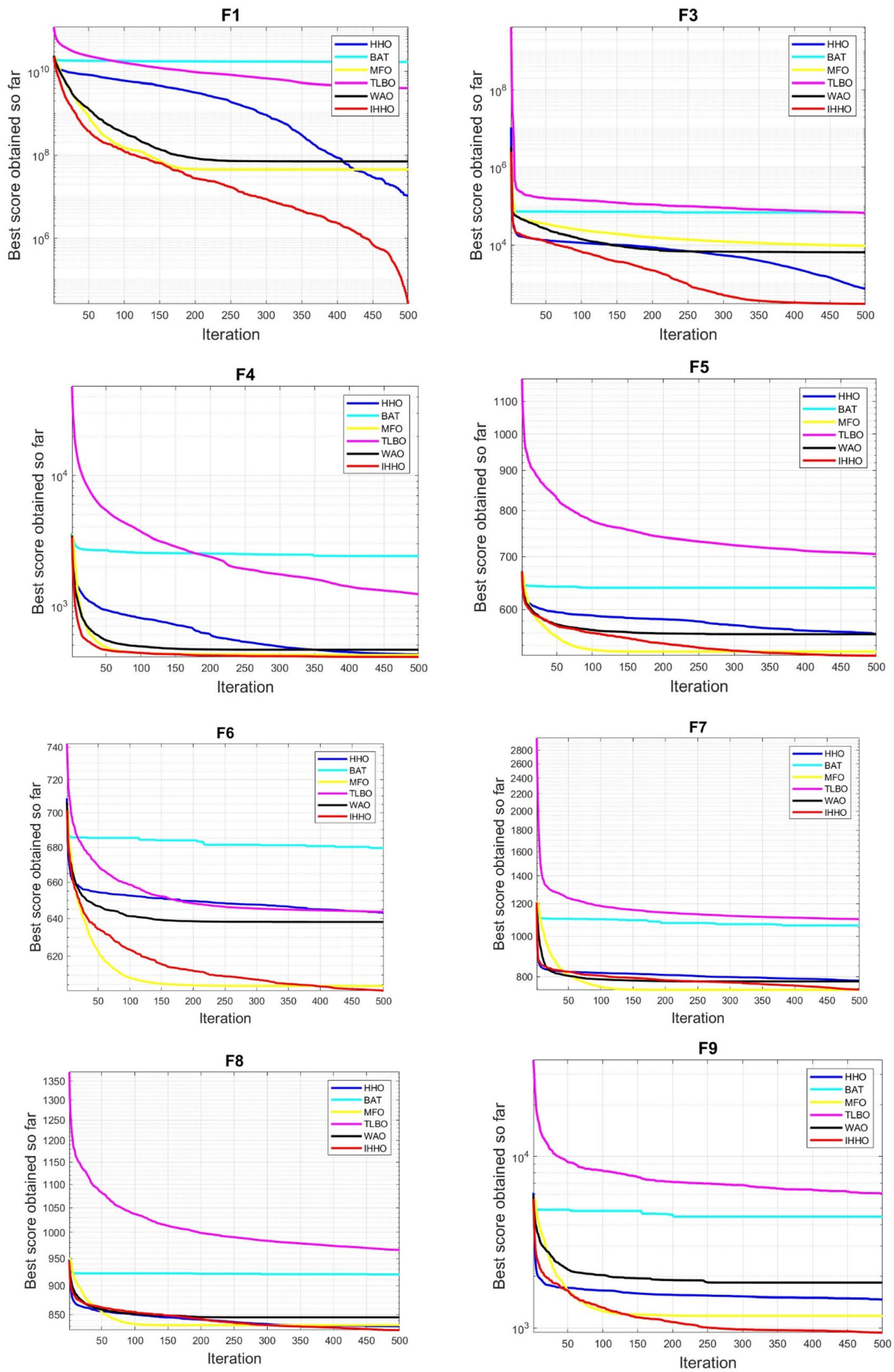


Fig. 15 Convergence curves for IHHO vs other algorithms on CEC2017

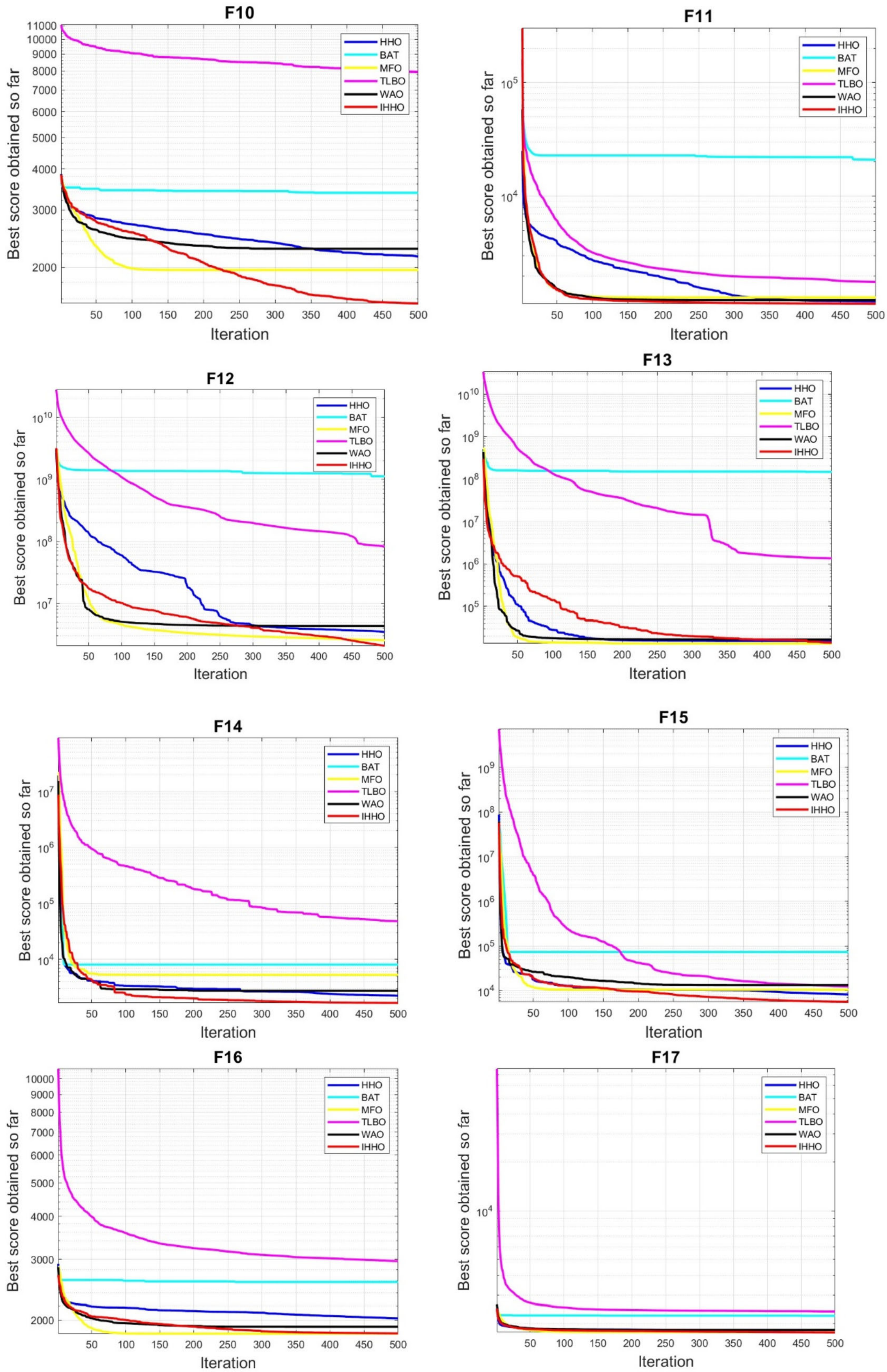


Fig. 15 continued

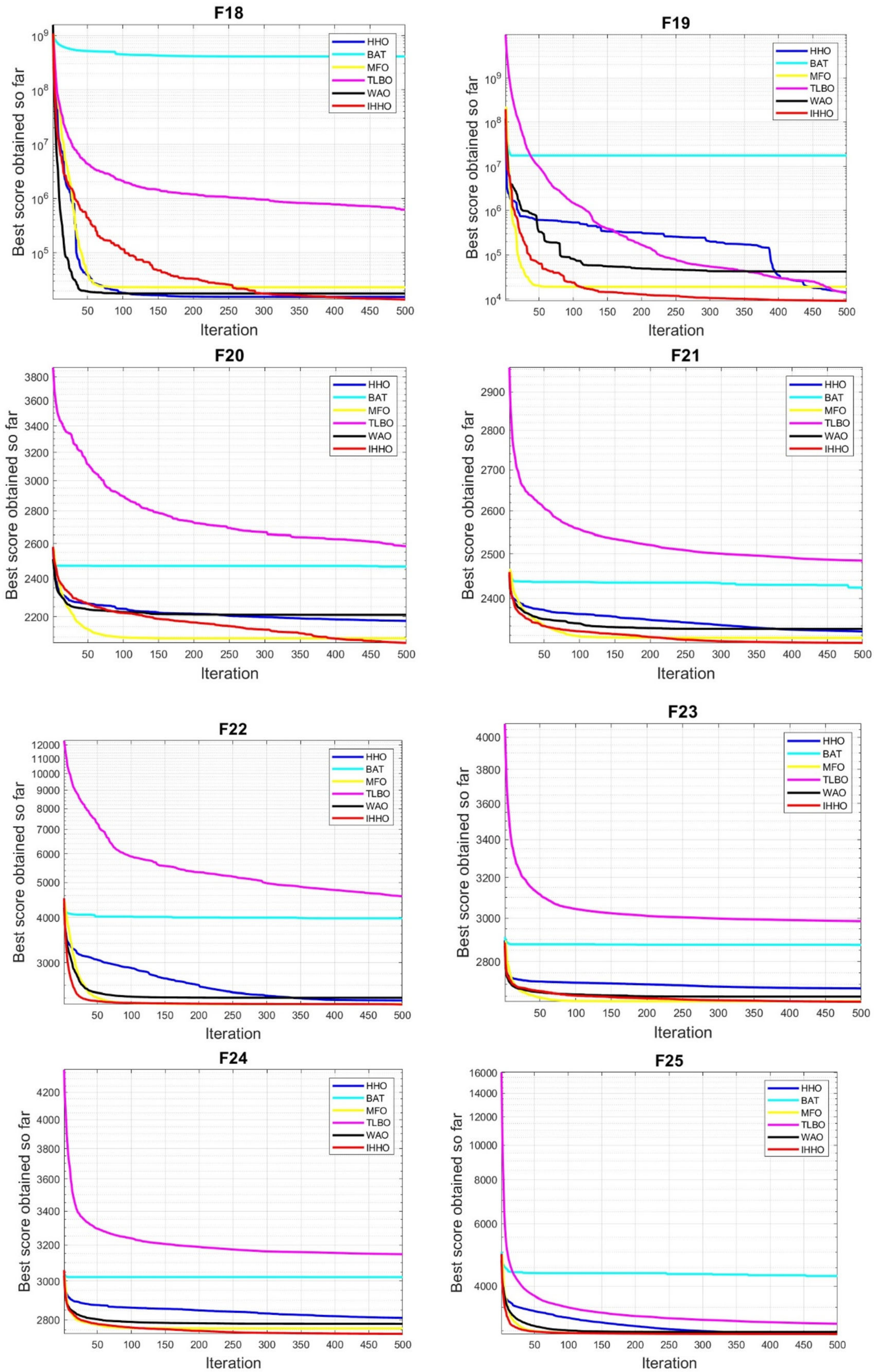


Fig. 15 continued

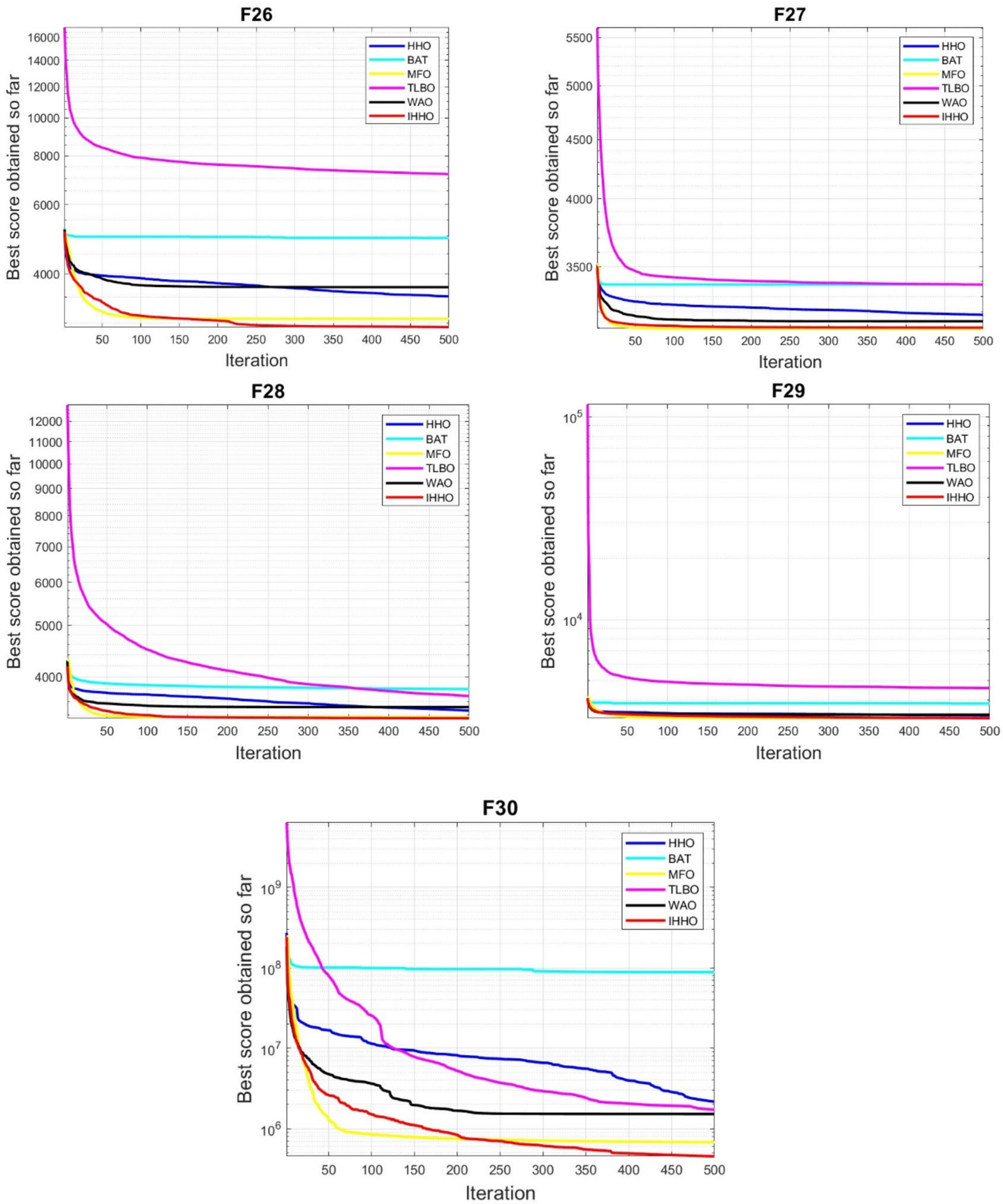


Fig. 15 continued

determine which one was superior. CEC01-CEC03 has varied dimensions, however, CEC04-CEC10 had a 10-dimensional constrained optimization problem within the [–100, 100] border range.

According to Table 13, the performance of IHHO is superior to that of the original HHO, with the exception of CEC02. IHHO outperforms other analyzed approaches for modifying functions due to its higher capacity to escape the local optimum utilizing the random walk approach. This is the primary reason for IHHO's superior performance.

Viewing the curves in Fig. 17 allows for a comparison of the convergence periods of IHHO and the original HHO

Table 11 Comparison of results of IHHO vs other modifications of HHO on CEC2017

Benchmark		IHHO	HHO	BHHO[36]	MHHO[37]	LogHHO[38]
F1	AVG	2.17E + 04	9.91E + 06	3.77E + 06	1.47E + 06	5.52E + 06
	STD	**8666.9596**	6.56E + 06	4.12E + 06	8.42E + 05	7.26E + 06
	Rank	**1**	5	3	2	4
F3	AVG	**300.2742**	790.2067	1.19E + 03	398.1691	5.93E + 03
	STD	**0.18863**	498.1986	770.3361	104.8285	6.10E + 03
	Rank	**1**	3	4	2	5
F4	AVG	**405.4989**	441.6495	436.7338	416.015	430.7354
	STD	**12.3172**	41.2061	44.7437	21.0623	55.0594
	Rank	**1**	5	4	2	3
F5	AVG	**523.0827**	560.9182	548.4437	548.8574	564.0366
	STD	**9.7255**	19.6982	14.4186	13.9218	23.9574
	Rank	**1**	4	2	3	5
F6	AVG	**601.2425**	641.3173	628.4876	631.5325	649.3339
	STD	**1.6446**	11.3266	13.7503	10.9249	16.4377
	Rank	**1**	4	2	3	5
F7	AVG	**740.3267**	803.3728	776.8391	784.2181	801.2626
	STD	**13.5653**	15.7372	23.8562	21.698	31.341
	Rank	**1**	5	2	3	4
F8	AVG	**822.0687**	832.37	831.3131	830.8505	847.8072
	STD	8.89E + 00	6.4979	7.9076	11.4977	17.132
	Rank	**1**	4	3	2	5
F9	AVG	**911.342**	1.52E + 03	1206.649	1312.3297	1808.6
	STD	**30.507**	214.4839	157.5356	258.1827	573.6312
	Rank	**1**	4	2	3	5
F10	AVG	**1.59E + 03**	2.08E + 03	2.03E + 03	2.04E + 03	2.29E + 03
	STD	264.6654	3.09E + 02	**226.4783**	371.8428	444.9148
	Rank	**1**	4	2	3	5
F11	AVG	**1123.0802**	1198.231	1.18E + 03	1.17E + 03	1.21E + 03
	STD	**1.28E + 01**	89.4682	45.4478	43.8321	66.5747
	Rank	**1**	4	3	2	5
F12	AVG	**1.35E + 06**	3.45E + 06	3.74E + 06	4.87E + 06	4.61E + 06
	STD	**1,377,342.073**	3.67E + 06	5.30E + 06	3.66E + 06	6.07E + 06
	Rank	**1**	2	3	5	4
F13	AVG	**1.16E + 04**	1.37E + 04	1.72E + 04	1.84E + 04	2.42E + 04
	STD	**9346.6084**	1.03E + 04	8.10E + 03	1.13E + 04	1.51E + 04
	Rank	1	2	3	4	5
F14	AVG	**1.6116e + 03**	2.28E + 03	1.67E + 03	2.83E + 03	5.06E + 03
	STD	350.1143	**1.08E + 03**	150.0148	1.54E + 03	3.51E + 03
	Rank	**1**	3	2	4	5
F15	AVG	**4.93E + 03**	8.89E + 03	5.36E + 03	6.69E + 03	1.94E + 04
	STD	**1959.5614**	3.18E + 03	2.41E + 03	2.71E + 03	1.30E + 04
	Rank	**1**	4	2	3	5
F16	AVG	**1.7763e + 03**	1.92E + 03	1.88E + 03	1.94E + 03	2.09E + 03
	STD	**147.1462**	1.64E + 02	126.3299	128.9934	205.4413
	Rank	**1**	3	2	4	5
F17	AVG	**1.76E + 03**	1.80E + 03	1.78E + 03	1.80E + 03	1.88E + 03
	STD	**18.3225**	7.52E + 01	25.6485	61.2233	118.0672
	Rank	**1**	3	2	3	5

Table 11 (continued)

Benchmark		IHHO	HHO	BHHO[36]	MHHO[37]	LogHHO[38]
F18	AVG	**1.6275e + 04**	1.7680e + 04	2.1607e + 04	2.09E + 04	1.7347e + 04
	STD	10,835.5077	1.36E + 04	1.25E + 04	**1.05E + 04**	1.16E + 04
	Rank	**1**	3	5	4	2
F19	AVG	**7.19E + 03**	3.59E + 04	1.24E + 04	1.17E + 04	4.63E + 04
	STD	**6647.6979**	6.38E + 04	9.54E + 03	1.50E + 04	1.27E + 05
	Rank	**1**	4	3	2	5
F20	AVG	**2.07E + 03**	2.20E + 03	2.15E + 03	2.16E + 03	2.28E + 03
	STD	**49.3032**	8.87E + 01	57.2151	70.9044	101.669
	Rank	**1**	4	2	3	5
F21	AVG	**2.28E + 03**	2352.2351	2.30E + 03	2.34E + 03	2.35E + 03
	STD	60.2607	30.8292	61.2391	**38.6362**	46.3345
	Rank	**1**	5	2	3	4
F22	AVG	**2.29E + 03**	2376.921	2308.454	2359.7196	2463.2328
	STD	**24.7819**	237.0443	5.0388	11.2163	470.9754
	Rank	**1**	4	2	3	5
F23	AVG	**2622.469**	2681.7155	2644.4426	2665.5052	2699.1157
	STD	**10.5649**	31.1696	21.0086	20.9853	42.8784
	Rank	**1**	4	2	3	5
F24	AVG	**2719.675**	2822.6602	2765.9147	2798.9617	2867.0919
	STD	92.5488	78.5028	92.1919	74.8073	**58.9812**
	Rank	**1**	4	2	3	5
F25	AVG	**2928.3439**	2.93E + 03	2.94E + 03	2.93E + 03	2.94E + 03
	STD	23.1189	6.88E + 01	25.3672	**16.7946**	32.0362
	Rank	**1**	2	4	2	4
F26	AVG	**2891.4091**	3690.8329	3166.3901	3310.234	3918.0487
	STD	**250.8454**	566.4344	326.8225	623.6603	642.3783
	Rank	**1**	4	2	3	5
F27	AVG	**3.10E + 03**	3.18E + 03	3.13E + 03	3.14E + 03	3.20E + 03
	STD	**8.18E + 00**	4.66E + 01	12.949	31.8968	60.0345
	Rank	**1**	4	2	3	5
F28	AVG	**3.36E + 03**	3.44E + 03	**3.36E + 03**	3.42E + 03	3.40E + 03
	STD	**9.25E + 01**	1.40E + 02	105.204	144.1123	167.4323
	Rank	**1**	5	**1**	4	3
F29	AVG	**3.24E + 03**	3.33E + 03	3.29E + 03	3.33E + 03	3.42E + 03
	STD	57.3682	8.88E + 01	**52.3279**	75.0874	126.0629
	Rank	**1**	3	2	3	5
F30	AVG	**2.45E + 05**	2.05E + 06	6.16E + 05	1.10E + 06	4.09E + 06
	STD	**4.07E + 05**	3.01E + 06	102E + 06	1.39E + 06	1.05E + 07
	Rank	**1**	4	2	3	5
Percentage		1	3.75862069	2.482758621	3	4.586206897
Total Rank		1	4	2	3	5

Best results are highlighted in bold

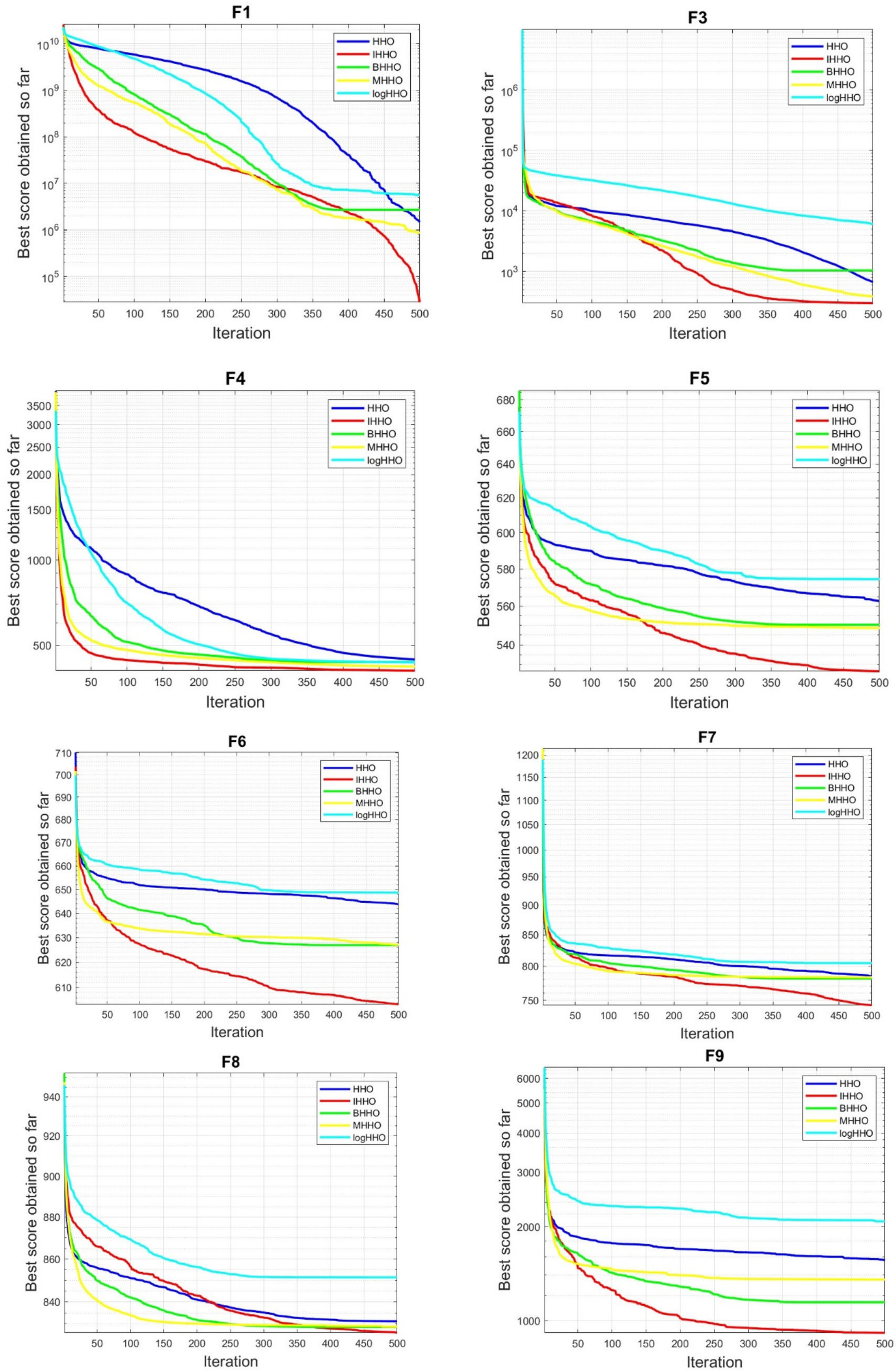


Fig. 16 Convergence curves for IHHO vs HHO modifications on CEC2017

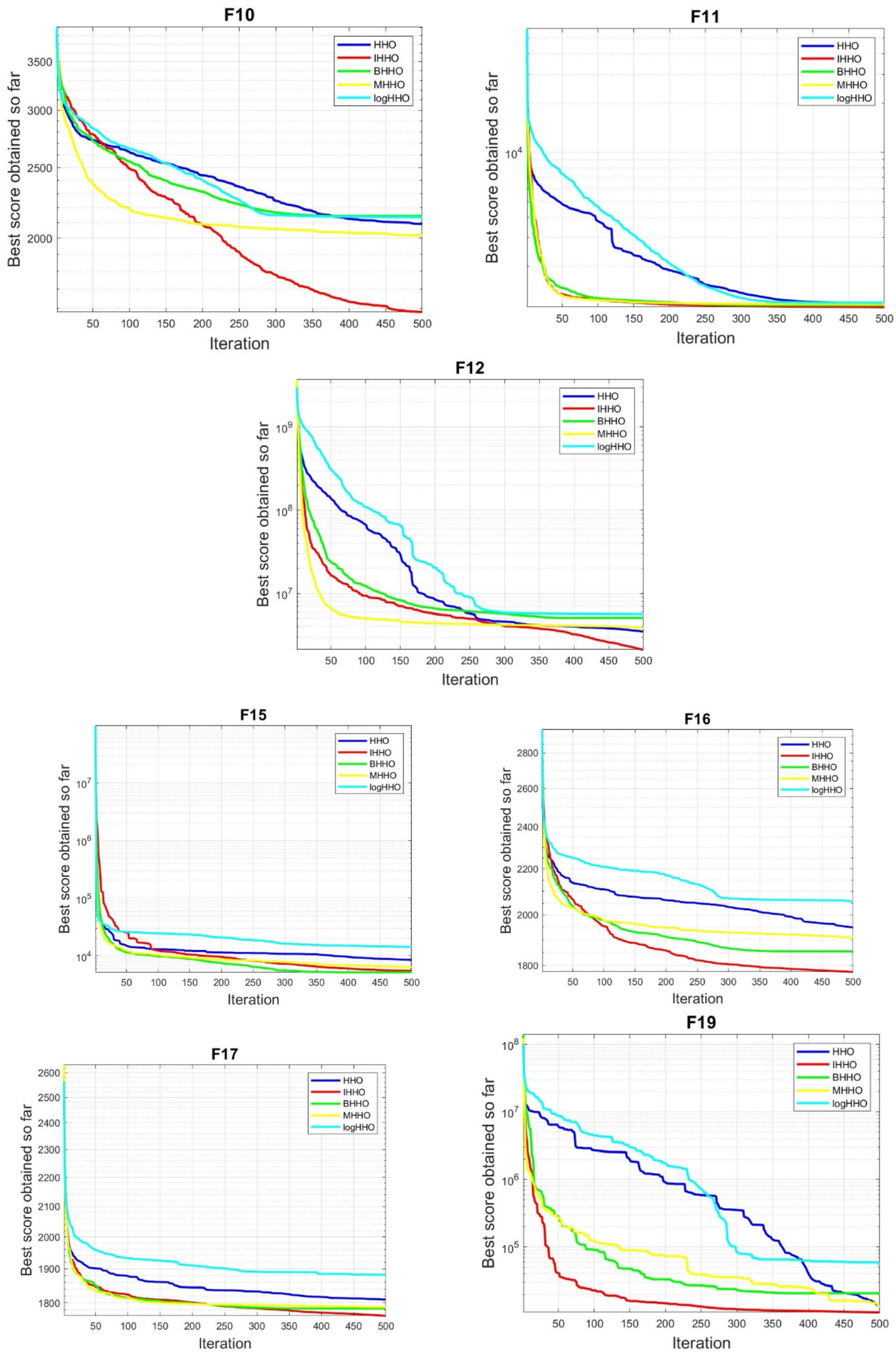


Fig. 16 continued

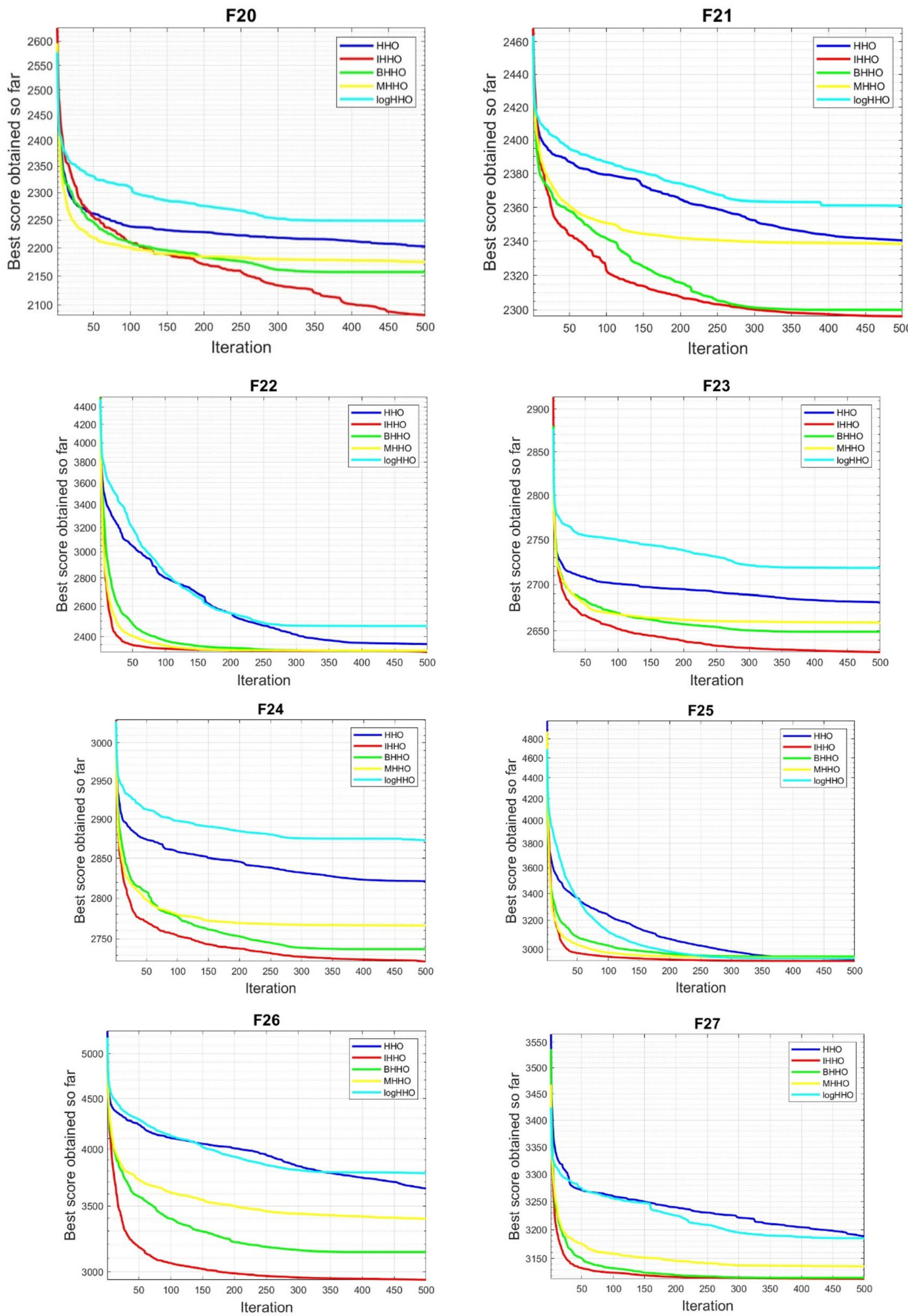


Fig. 16 continued

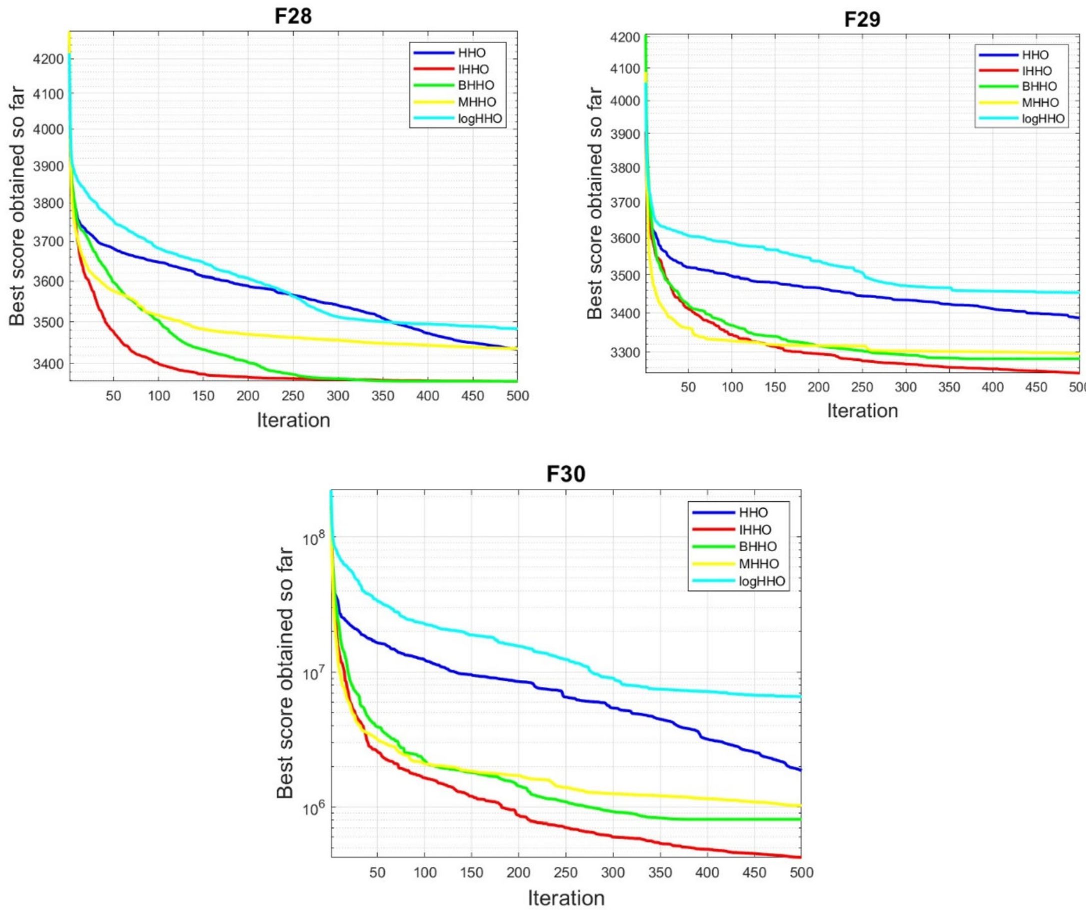


Fig. 16 continued

method to be carried out visually. The figures present an illustration of the typical objective values that are attained by algorithms after going through some iterations. It is clear from looking at the figures that the proposed IHHO method converges at a point more quickly than the conventional HHO approach does.

5.3.2 IHHO vs other algorithms

Our comparison of IHHO to other algorithms, namely GWO [39], BAT [40], MFO [42], TLBO [41], and WOA [43], can be found in Table 14 after putting it through the paces on the test functions for the CEC2019 examination. This was done so that we could decide which algorithm was the better one. For CEC04-CEC10, a 10-dimensional optimal solution is restricted within the [− 100, 100] border range. Since CEC01-CEC03 has different dimensions evaluated this function 30 times.

According to Table 14, the performance of the IHHO algorithm is better than that of all other algorithms, with the exception of the CEC02 method. IHHO outperforms BAT by a significant margin when the two are compared to each other. When compared to MFO, IHHO performs better, except for CEC06. IHHO outperforms TLBO since the latter is unable to solve CEC2019 functions ranging from CEC04 to CEC10 which the former can solve. IHHO performs better than GWO and WOA in all functions with the exception of CEC02. IHHO has an overall rank of 1.

A visual comparison of the convergence periods of the IHHO method can be carried out using the convergence

Table 12 The description of CEC2019

Benchmark	Name	Dim	Range	*F* min
CEC01	STORN'S CHEBYSHEV POLYNOMIAL FITTING PROBLEM	9	[−8192, 8192]	1
CEC02	INVERSE HILBERT MATRIX PROBLEM	16	[−16384, 16384]	1
CEC03	LENNARD–JONES MINIMUM ENERGY CLUSTER	18	[−4, 4]	1
CEC04	RASTRIGIN'S FUNCTION	10	[−100, 100]	1
CEC05	GRIEWANGK'S FUNCTION	10	[−100, 100]	1
CEC06	WEIERSTRASS FUNCTION	10	[−100, 100]	1
CEC07	MODIFIED SCHWEFEL'S FUNCTION	10	[−100, 100]	1
CEC08	EXPANDED SCHAFFER'S F6 FUNCTION	10	[−100, 100]	1
CEC09	HAPPY CAT FUNCTION	10	[−100, 100]	1
CEC10	ACKLEY FUNCTION	10	[−100, 100]	1

Table 13 Comparison of results of IHHO and HHO on CEC2019

Benchmark	optimizer	Mean	STD	MIN	MAX
F1	IHHO	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	HHO	5.36E + 04	5.51E + 03	43,436.1761	69,580.1764
F2	IHHO	3.04E + 02	4.63E + 02	18.5521	2112.115
	HHO	**1.74E + 01**	**6.64E−03**	17.3466	17.3821
F3	IHHO	**12.7024**	**2.60E−06**	12.7024	12.7024
	HHO	**12.7024**	1.07E−05	12.7024	12.7024
F4	IHHO	**31.5259**	**9.0423**	21.0913	58.6274
	HHO	207.4399	78.8245	70.1553	551.5347
F5	IHHO	**1.1784**	**0.11771**	1.0562	1.4325
	HHO	2.6186	0.56213	1.5114	4.0057
F6	IHHO	**8.5407**	1.2269	5.5651	10.8712
	HHO	9.5065	**0.97188**	6.6634	11.0721
F7	IHHO	**278.1647**	**116.2791**	−12.5279	697.0265
	HHO	401.4808	199.1003	−26.7632	883.2117
F8	IHHO	**5.3407**	0.66282	4.2631	6.1681
	HHO	5.7866	**0.55709**	4.3016	7.1031
F9	IHHO	**2.493**	**5.05E−02**	2.38E + 00	3.01E + 00
	HHO	3.44E + 00	4.83E−01	2.59E + 00	4.17E + 00
F10	IHHO	**18.7551**	2.03E + 01	1.74E + 00	2.02E + 01
	HHO	**0.0333**	1.10E−01	2.00E + 01	2.04E + 01

Best results are highlighted in bold

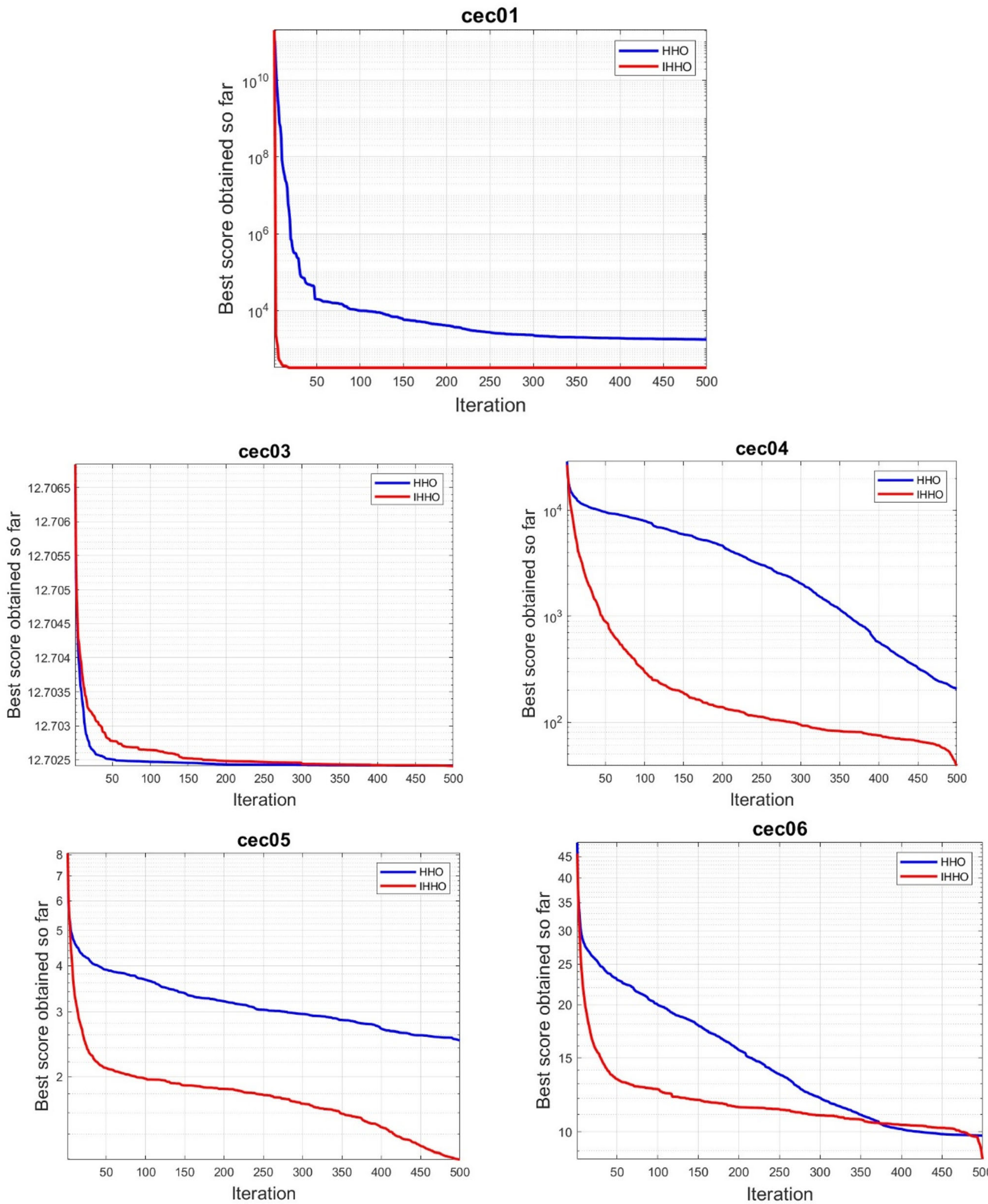


Fig. 17 Convergence curves for IHHO vs HHO on CEC2019

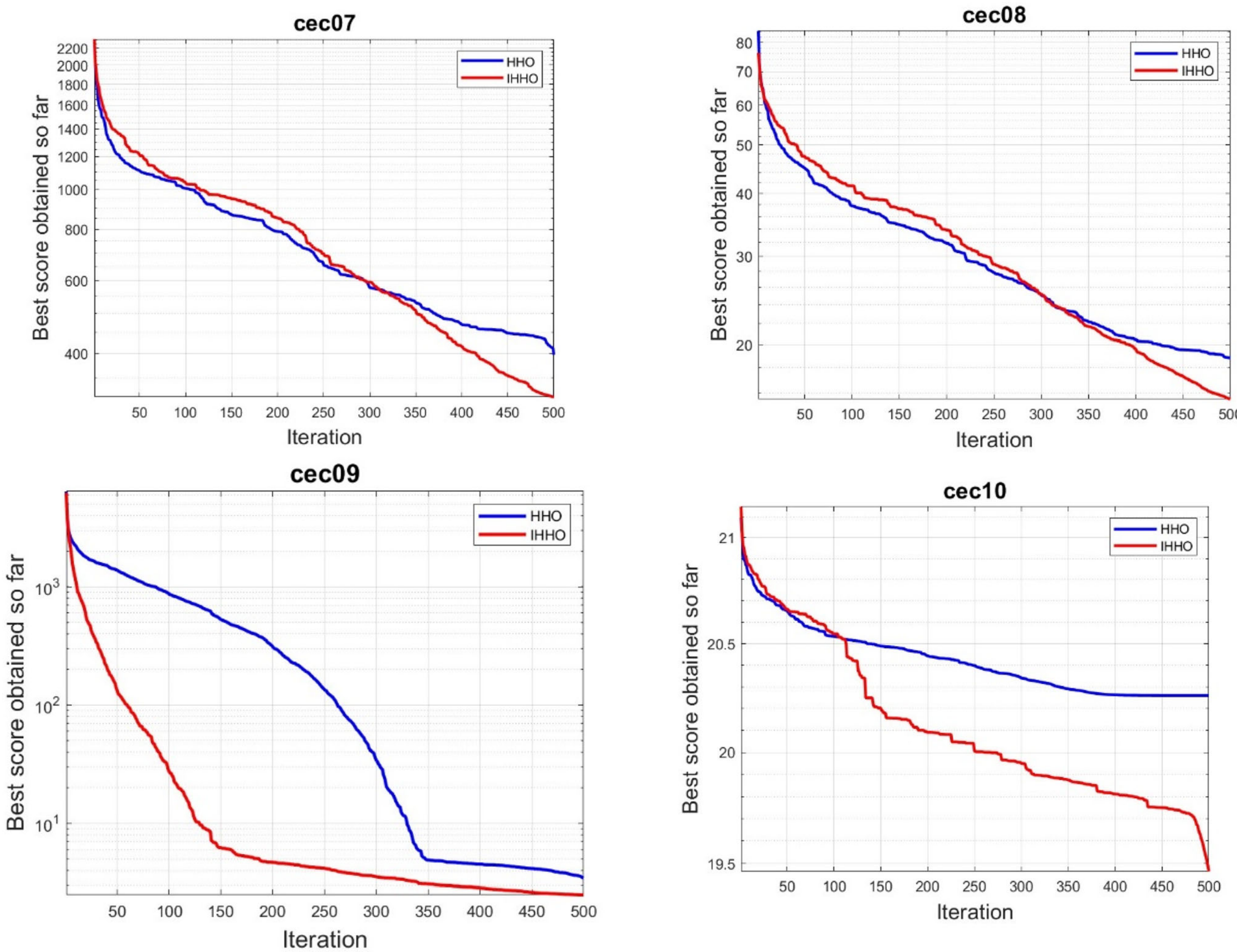


Fig. 17 continued

Table 14 Comparison of results between IHHO and other algorithms on CEC2019

Benchmark		IHHO	HHO	GWO[39]	BAT[81]	MFO[42]	TLBO[41]	WOA[43]
F1	AVG	**0.00E + 00**	5.36E + 04	1.83E + 08	4.11E + 12	1.93E + 10	1.13E + 08	3.20E + 10
	STD	**0.00E + 00**	5.51E + 03	3.49E + 08	3.39E + 12	2.74E + 10	2.13E + 08	4.44E + 10
	Rank	**1**	2	4	7	5	3	6
F2	AVG	3.04E + 02	1.74E + 01	**17.3439**	1.31E + 04	**17.3429**	17.3429	17.3493
	STD	4.63E + 02	6.64E−03	3.69E−04	4.14E + 03	7.23E−15	**6.76E−15**	0.0056
	Rank	6	5	3	7	**1**	**1**	4
F3	AVG	**12.7024**	**12.7024**	**12.7024**	12.7066	12.7025	**12.7024**	**12.7024**
	STD	**2.60E−06**	1.07E−05	1.89E−04	0.001	2.63E−04	4.2862e−10	1.37E−06
	Rank	**1**	**1**	**1**	7	6	1	1
F4	AVG	**31.5259**	207.4399	109.3664	2.47E + 04	84.1661	**NAN**	344.9699
	STD	**9.0423**	78.8245	257.2913	8.93E + 03	134.4453		180.3267
	Rank	**1**	4	3	6	2		5
F5	AVG	**1.1784**	2.6186	1.3805	6.5935	1.2273	**NAN**	1.8245
	STD	**0.11771**	0.56213	0.2312	1.7679	0.1902		0.3299
	Rank	**1**	5	3	6	2		4
F6	AVG	8.5407	9.5065	11.2022	13.5474	**6.387**	**NAN**	9.496
	STD	1.2269	0.97188	**0.9005**	1.0495	2.2168		1.1407
	Rank	2	4	5	6	**1**		3
F7	AVG	**278.1647**	401.4808	423.922	1.70E + 03	467.3279	**NAN**	539.6716
	STD	**116.2791**	199.1003	299.3608	290.6627	296.3268		251.1866
	Rank	**1**	2	3	6	4		5
F8	AVG	**5.3407**	5.7866	5.4677	7.6263	5.5632	**NAN**	6.1317
	STD Rank	0.66282	**0.55709**	1.0985	**0.5511**	0.5941		0.5627
		1	4	2	6	3		5
F9	AVG	**2.493**	3.44E + 00	4.5825	4.66E + 03	2.8995	**NAN**	4.9136
	STD Rank	**5.05E−02**	4.83E−01	0.8744	1.52E + 03	0.4787		0.9791
		1	3	4	6	2		5
F10	AVG	**18.7551**	2.03E + 01	20.4496	20.8793	20.2163	**NAN**	20.2573
	STD Rank	**0.0333**	1.10E−01	0.091	0.2007	0.1735		0.134
		1	4	5	6	2		3
Percentage		**1.6**	**3.4**	**3.3**	**6.3**	**2.8**	**NaN**	**4.1**
Total Rank		**1**	**4**	**3**	**6**	**2**	**–**	**5**

Best results are highlighted in bold

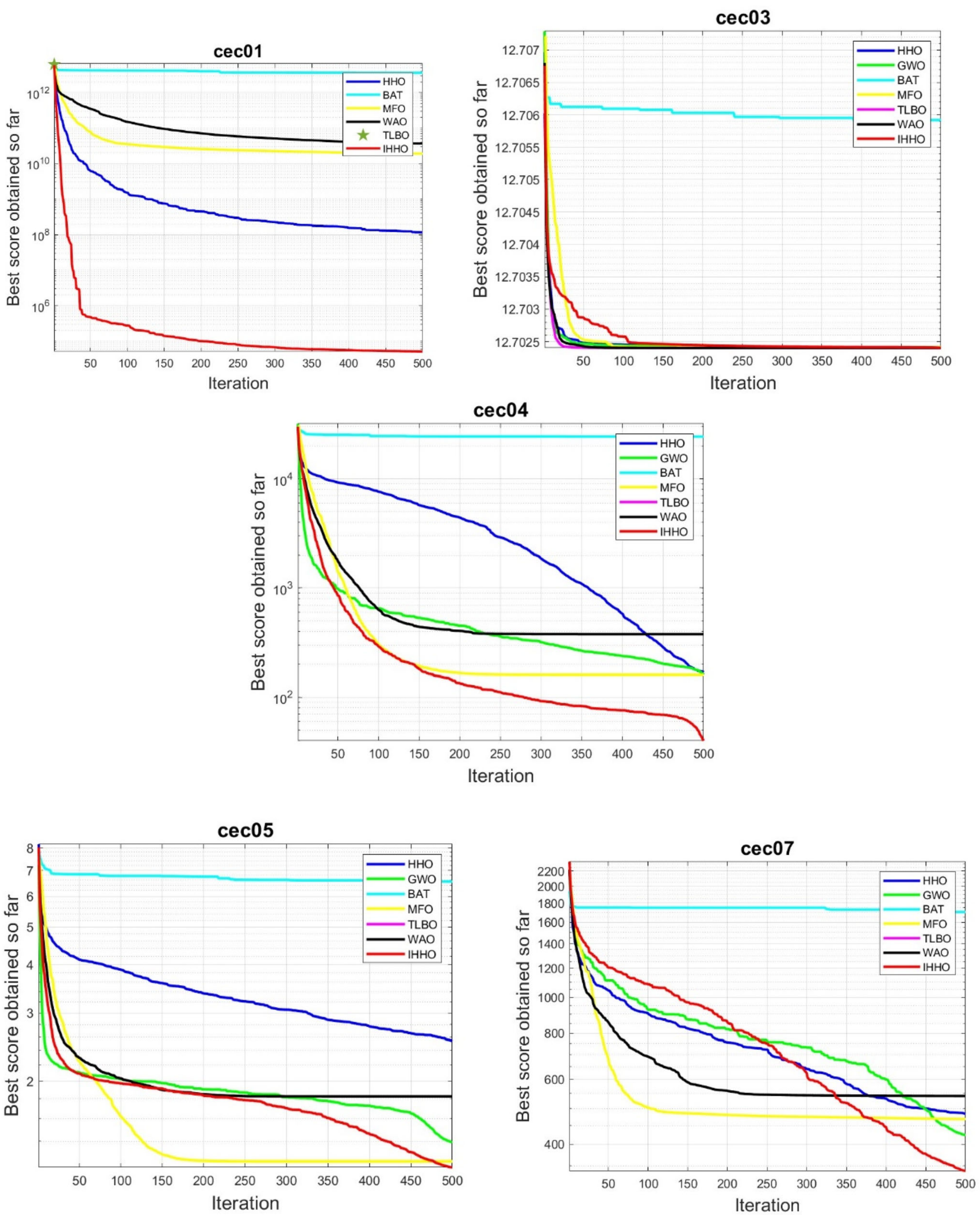
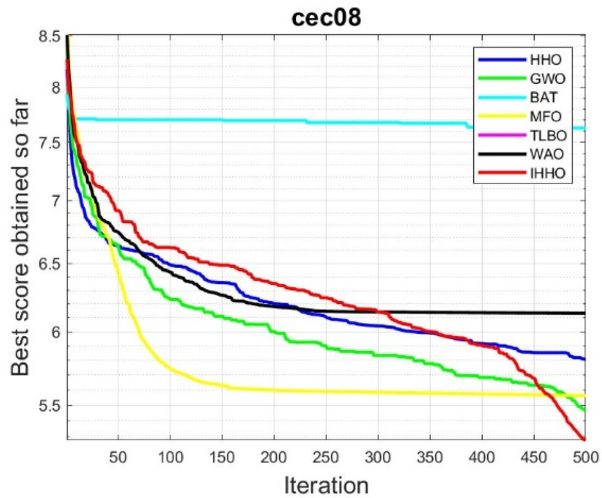
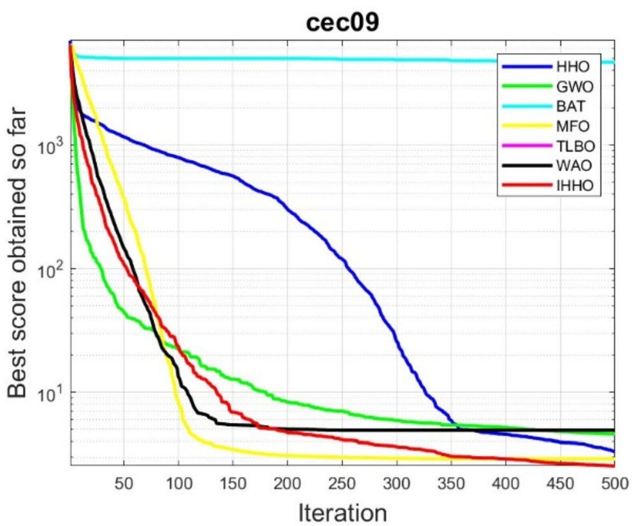


Fig. 18 Convergence curves for IHHO vs other algorithms on CEC2019





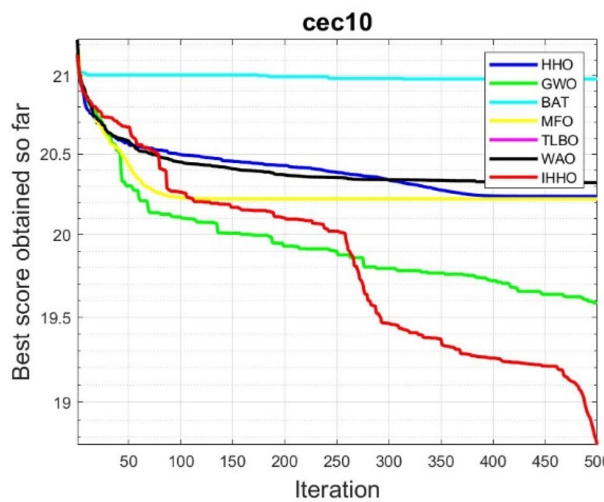


Fig. 18 continued

Table 15 Comparison of results of IHHO vs other modifications of HHO on CEC2019

Benchmark		IHHO	HHO	BHHO[36]	MHHO[37]	LogHHO[38]
F1	AVG	**0.00E + 00**	5.36E + 04	5.34E + 04	4.96E + 04	1.88E + 11
	STD	**0.00E + 00**	5.51E + 03	7.16E + 03	4.13E + 03	1.34E + 11
	Rank	**1**	4	3	2	5
F2	AVG	3.04E + 02	1.74E + 01	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	STD Rank	4.63E + 02	**6.64E−03**	0.0077	0.0053	0.0089
		5	4	**1**	**1**	**1**
F3	AVG	**12.7024**	**12.7024**	**1.27E + 01**	**12.7024**	**1.27E + 01**
	STD	2.60E−06	1.07E−05	**3.83E−09**	4.05E−08	1.57E−05
	Rank	3	3	**1**	3	**1**
F4	AVG	**31.5259**	207.4399	166.6872	159.7486	231.2291
	STD	**9.0423**	78.8245	85.9907	70.2154	85.6502
	Rank	**1**	4	3	2	5
F5	AVG	**1.1784**	2.6186	1.5799	1.6984	2.3964
	STD	**0.11771**	0.56213	0.2194	0.2796	0.6625
	Rank	**1**	5	2	3	4
F6	AVG	8.5407	9.5065	**7.8931**	8.217	9.9884
	STD	1.2269	0.97188	**0.9243**	1.3799	0.9531
	Rank	3	4	**1**	2	5
F7	AVG	**278.1647**	401.4808	436.7906	401.9861	402.6903
	STD	**116.2791**	199.1003	155.4532	275.3689	301.7879
	Rank	**1**	2	5	3	4
F8	AVG	**5.3407**	5.7866	5.5671	5.6283	6.2037
	STD	0.66282	0.55709	**0.4605**	0.7015	0.5425
	Rank	**1**	4	2	3	5
F9	AVG	**2.493**	3.44E + 00	4.60E + 00	3.57E + 00	3.68E + 00
	STD	**5.05E−02**	4.83E−01	0.834	0.6065	0.4296
	Rank	**1**	2	5	3	4
F10	AVG	**18.7551**	2.03E + 01	2.01E + 01	2.01E + 01	2.04E + 01
	STD	**0.0333**	1.10E−01	0.112	1.7089	0.0824
	Rank	**1**	4	2	2	5
Percentage		**1.8**	**3.6**	**2.5**	**2.4**	**3.9**
Total Rank		**1**	**4**	**3**	**2**	**5**

Best results are highlighted in bold

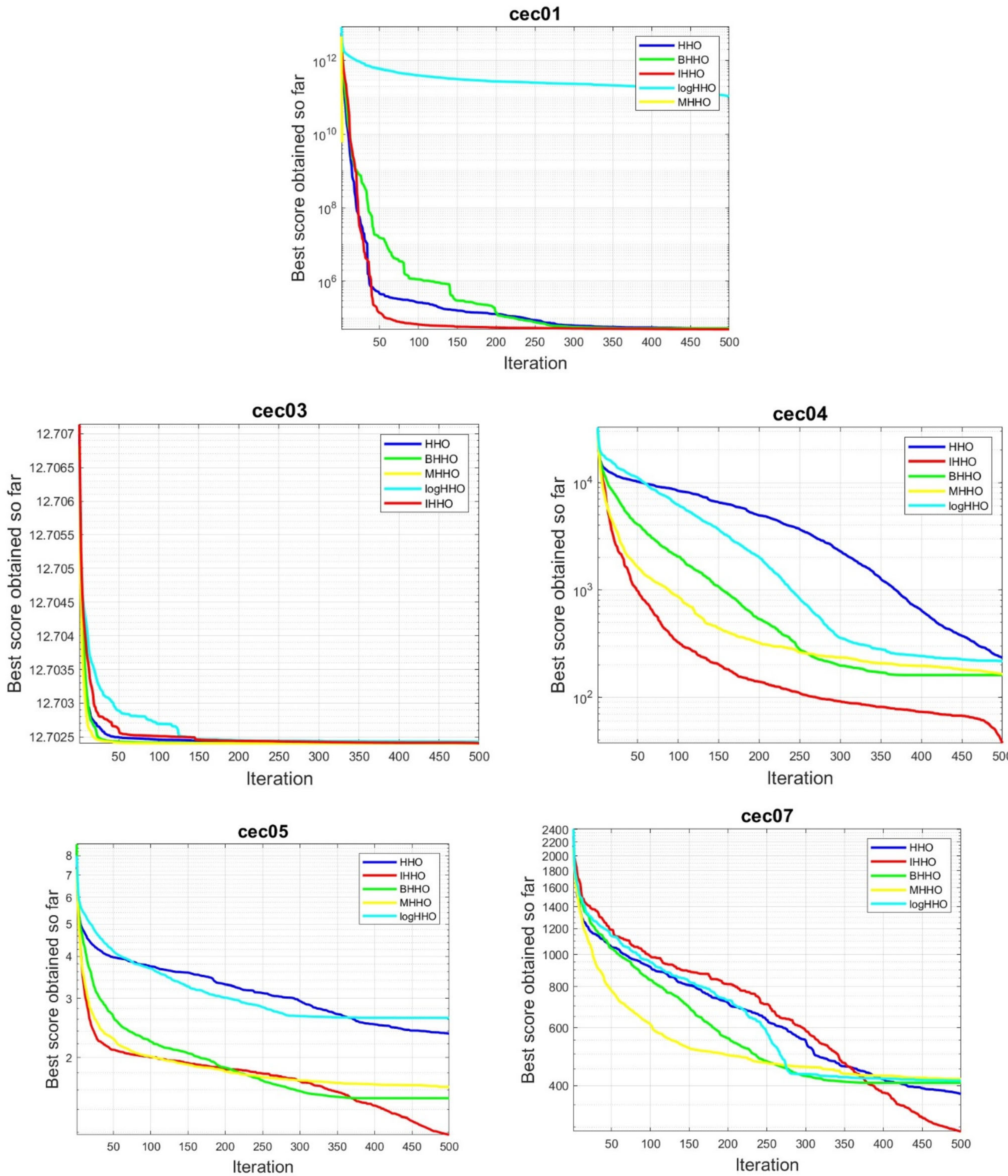


Fig. 19 Convergence curves for IHHO vs HHO modifications on CEC2019

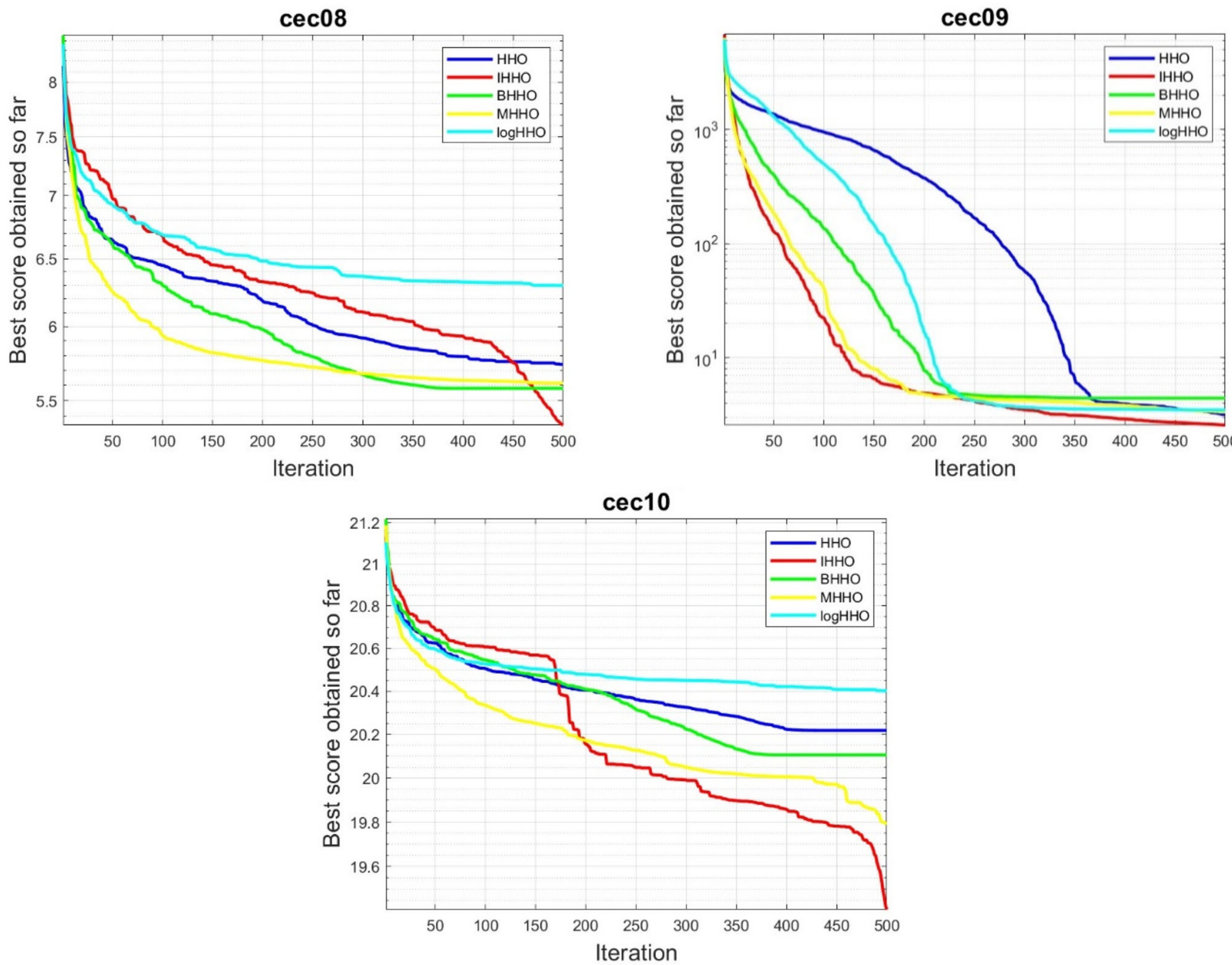


Fig. 19 continued

Table 16 The description of CEC2020

Benchmark	Name	Dim	Range	*F* min
F1	Shifted and Rotated Bent Cigar Function (CEC 2017[86] F1)	10	[−100, 100]	100
F2	Shifted and Rotated Schwefel's Function (CEC 2014[87] F11)	10	[−100, 100]	1100
F3	Shifted and Rotated Lunacek bi-Rastrigin Function (CEC 2017[87]F7)	10	[−100, 100]	700
F4	Expanded Rosenbrock's plus Griewangk's Function (CEC2017[86] f19)	10	[−100, 100]	1900
F5	Hybrid Function 1 (N = 3) (CEC 2014[87] F17)	10	[−100, 100]	1700
F6	Hybrid Function 2 (N = 4) (CEC 2017[86] F16)	10	[−100, 100]	1600
F7	Hybrid Function 3 (N = 5) (CEC 2014[87] F21)	10	[−100, 100]	2100
F8	Composition Function 1 (N = 3) (CEC 2017[86] F22)	10	[−100, 100]	2200
F9	Composition Function 2 (N = 4) (CEC 2017[86] F24)	10	[−100, 100]	2400
F10	Composition Function 3 (N = 5) (CEC 2017[86] F25)	10	[−100, 100]	2500

curves shown in Fig. 18**.** This allows for the comparison to be carried out visually. The figures offer a visual representation of the usual objective values that can be achieved by algorithms after they have been subjected to a certain number of iterations. When looking at the figures, it is obvious that the proposed IHHO converges at a point more quickly than the other used algorithms.

5.3.3 IHHO vs other modifications of HHO

In this paper, the proposed IHHO to other HHO variants, namely BHHO [54], MHHO [37], and LogHHO, was compared[38]. The objective function's mean and standard deviation are shown in Table 15. The results of the various approaches' convergence are also shown in Fig. 19**.**

According to Table 15, the performance of the HHO algorithm is better than that of all other modifications of HHO, with the exception of the CEC02 method. When compared to BHHO, IHHO performs better, with the exception of CEC06. IHHO performs better than MHHO and LogHHO in all functions with the exception of CEC02. IHHO has an overall rank of 1.

A visual comparison of the convergence periods of the IHHO method can be carried out using the convergence curves shown in Fig. 19**.** When looking at the figures, it is obvious that the proposed IHHO approach converges more quickly than the other algorithms that are traditionally used.

5.4 CEC2020 benchmark functions

Algorithms for optimizing a single-objective problem serve as the building blocks from which more complex methods, such as multiobjective, niching, and constrained optimization, are constructed. Therefore, it is crucial to work on improving single-objective optimization methods, as this can have repercussions in other areas [84]. Trials with single-objective benchmark functions are essential to the iterative refinement of these algorithms. New and more difficult algorithms are required as computing power increases. As a result, the CEC'20 Special Session has been designed in real parameter optimization to foster this symbiotic relationship between Methods and issues, which is what drives development [85].

F1–F10 has the same dimensions as a 10-dimensional constrained optimization problem in the [− 100, 100] border range as indicated in Table 16**.**

5.4.1 HHO vs IHHO

In Table 17, after putting IHHO through its paces on the CEC2020 test functions, the results are compared to those obtained using HHO. We did this to establish which option is best. Functions F1–F10 all fall into a 10-dimensional restricted optimization problem with a boundary range of [-100, 100].

As can be seen in Table 17, the proposed algorithm IHHO performed similarly to the original HHO across the board. The results show that IHHO is superior to the classic HHO algorithm across the board. For unimodal functions, the IHHO algorithm excels above other studied modification methods due to its greater capacity to escape the local optimum using the random walk approach.

Figure 20 is a visual comparison of the curves used in the initial HHO method and its subsequent iterations. These examples illustrate the typical objective values that algorithms can achieve after some rounds. Convergence plots typically have the goal function's value on the vertical axis and the number of iterations on the horizontal. Convergence times for the proposed IHHO and the baseline HHO both improve, as shown graphically.

5.4.2 IHHO vs other algorithms

In this section, the proposed IHHO to some established competitors, such as the BAT algorithm [30], MFO [32], TLBO [31], and WOA [33] was compared. Table 18 summarizes the mean and standard deviation of the goal function and contains experimental results that address scalability concerns. Convergence results for several different approaches are also displayed in Fig. 21.

When compared to other modern algorithms, IHHO often yields greater average results. IHHO has the highest average values across the board for all functions, as compared to WOA, MFO, TLBO, and BAT algorithms. On average, IHHO always has the best outcomes. According to Table 18, the proposed IHHO algorithm has superior performance compared to the original HHO method, as well as to all other methods and the rank for all equations of 1.

For unimodal functions, IHHO performs better than other analyzed modification approaches due to its superior ability to break out of the local optimum via a random walk. The IHHO algorithm outperforms its competitors for unimodal functions because the random walk strategy improves the algorithm's ability to escape the local optimum.

Table 17 Comparison of results of IHHO and HHO on CEC2020

Benchmark	Optimizer	Mean	STD	MIN	MAX
F1	IHHO	**2.17E + 04**	**8666.9596**	5703.7119	57,055.175
	HHO	9.91E + 06	6.56E + 06	3.46E + 05	5.15E + 07
F2	IHHO	**1123.0802**	**1.28E + 01**	1.11E + 03	1.31E + 03
	HHO	1198.231	89.4682	1132.6825	1417.3129
F3	IHHO	**740.3267**	**13.5653**	719.1461	785.6197
	HHO	803.3728	15.7372	733.6704	827.7796
F4	IHHO	**7.19E + 03**	**6647.6979**	1971.5397	28,369.1424
	HHO	3.59E + 04	6.38E + 04	4404.9096	57,366.6541
F5	IHHO	**1.76E + 03**	**18.3225**	1728.0332	1810.0079
	HHO	1.80E + 03	7.52E + 01	1738.1283	2038.5601
F6	IHHO	**1.7763e + 03**	**147.1462**	1601.6159	2076.1548
	HHO	1.92E + 03	1.64E + 02	1635.5939	2204.3433
F7	IHHO	**2.28E + 03**	60.2607	2200.0087	2351.5616
	HHO	2352.2351	**30.8292**	2207.2868	2385.9732
F8	IHHO	**2.29E + 03**	**24.7819**	2225.9647	2998.756
	HHO	2376.921	237.0443	2274.969	3132.1365
F9	IHHO	**2719.675**	92.5488	2500.3054	2789.4829
	HHO	2822.6602	**78.5028**	2501.3983	2944.7632
F10	IHHO	**2928.3439**	**23.1189**	2897.752	2970.8463
	HHO	2.93E + 03	6.88E + 01	2898.9264	2954.9231

Best results are highlighted in bold

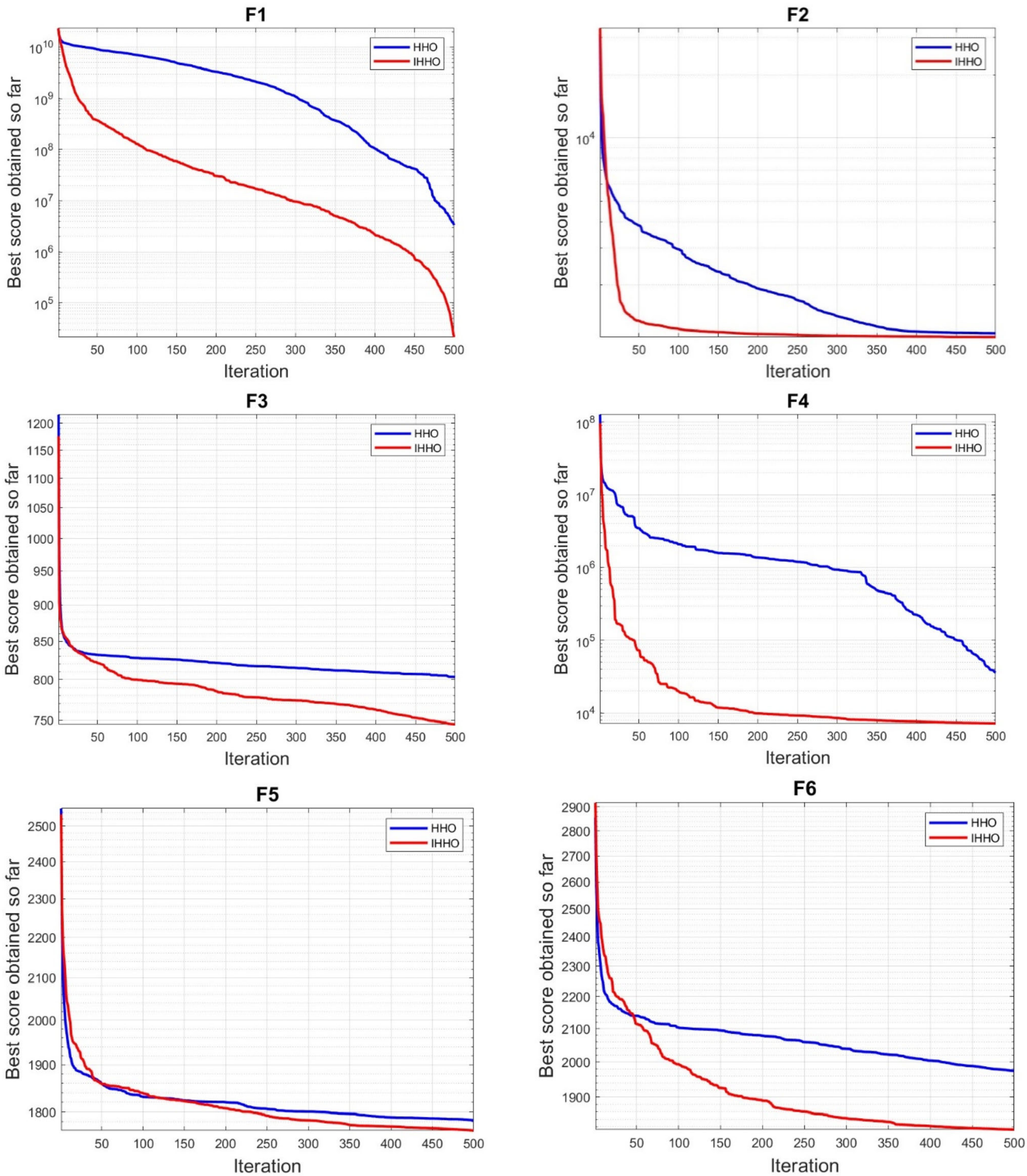


Fig. 20 Convergence curves for IHHO vs HHO on CEC2020

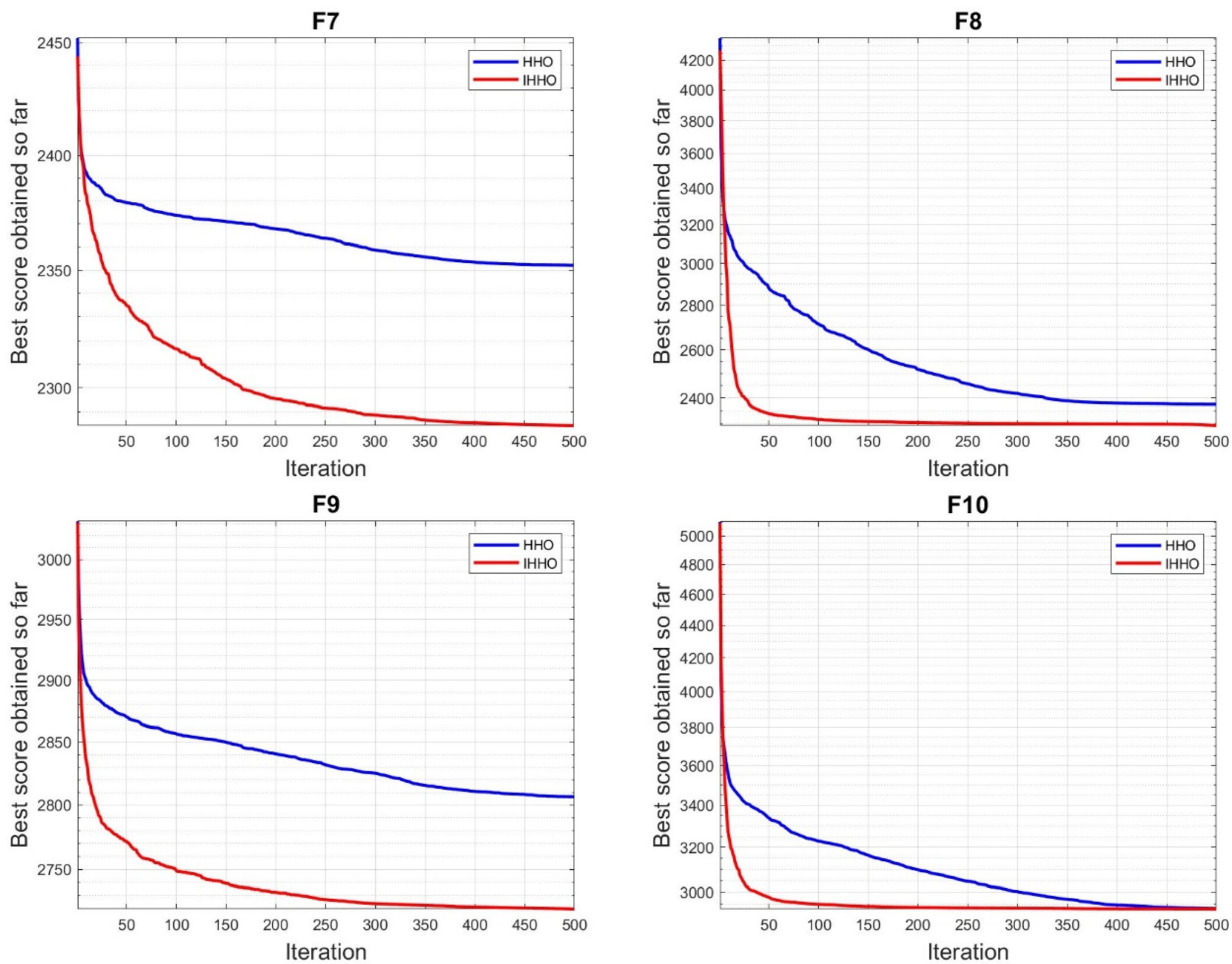


Fig. 20 continued

5.4.3 IHHO vs other modifications of HHO

The proposed IHHO to other HHO variants such as BHHO [40], MHHO[27], and LogHHO [28] were compared. The experimental outcomes that meet scalability concerns in terms of the mean and standard deviation of the objective function are shown in Table 19**.** Similarly, the convergence results for each method are shown in Fig. 22**.**

Using CEC2020 functions, we discovered that the proposed IHHO significantly improved performance over other changes of the original HHO. To get away from the local optimum, IHHO uses a random walk technique that gives it a significant advantage over the other variants tested for the function. According to Table 19, the modified HHO algorithm has superior performance compared to the original HHO method, as well as all other HHO modification algorithms and the rank for all equations of 1.

In contrast to the original HHO approach and its variants, the suggested IHHO is shown to converge faster.

5.5 Standard benchmark functions

An optimization algorithm has to be able to sift through the Search space, identify attractive regions, and then capitalize on those regions to get a global optimum [82]. For an algorithm to converge on the global optimum, it has to strike a balance between exploring new territory and capitalizing on what has already been discovered. In this part, IHHO's efficiency, reliability, and stability were assessed.

The 52 benchmark functions used to assess IHHO's efficacy fall into four categories: (i) 14 unimodal variable-

Table 18 Comparison of results between IHHO and other algorithms on CEC2020

Benchmark		IHHO	HHO	BAT[40]	MFO[42]	TLBO[41]	WOA[43]
F1	AVG	**2.17E + 04**	9.91E + 06	1.97E + 10	2.03E + 08	5.14E + 09	7.45E + 07
	STD	**8666.9596**	6.56E + 06	7.85E + 09	4.69E + 08	3.85E + 09	7.89E + 07
	Rank	**1**	2	6	4	5	3
F2	AVG	**1123.0802**	1198.231	2.09E + 04	1.29E + 03	1.77E + 03	1.22E + 03
	STD	**1.28E + 01**	89.4682	2.46E + 04	340.1241	779.766	61.4991
	Rank	**1**	2	6	4	5	3
F3	AVG	**740.3267**	803.3728	1.06E + 03	743.3572	1.10E + 03	778.9864
	STD	**13.5653**	15.7372	100.0475	14.3218	70.0049	27.5069
	Rank	**1**	4	5	2	6	3
F4	AVG	**7.19E + 03**	3.59E + 04	1.76E + 07	1.91E + 04	1.36E + 04	4.18E + 04
	STD	**6647.6979**	6.38E + 04	7.63E + 07	1.89E + 04	1.67E + 04	6.30E + 04
	Rank	**1**	4	6	3	2	5
F5	AVG	**1.76E + 03**	1.80E + 03	2.24E + 03	1.78E + 03	2.39E + 03	1.84E + 03
	STD	**18.3225**	7.52E + 01	183.3624	57.0341	304.3592	81.4459
	Rank	**1**	3	5	2	6	4
F6	AVG	**1.7763e + 03**	1.92E + 03	2.58E + 03	1.83E + 03	2.97E + 03	1.92E + 03
	STD	147.1462	1.64E + 02	176.5344	**138.7651**	427.2746	136.0335
	Rank	**1**	3	5	2	6	3
F7	AVG	**2.28E + 03**	2352.2351	2.42E + 03	2.31E + 03	2.48E + 03	2.33E + 03
	STD	60.2607	30.8292	50.8977	46.0485	**45.8557**	51.7961
	Rank	**1**	4	5	2	6	3
F8	AVG	**2.29E + 03**	2376.921	3.98E + 03	2.31E + 03	4.58E + 03	2.40E + 03
	STD	24.7819	237.0443	652.9891	**14.2038**	1.85E + 03	304.6141
	Rank	**1**	3	5	2	6	4
F9	AVG	**2719.675**	2822.6602	3.02E + 03	2.76E + 03	3.15E + 03	2.78E + 03
	STD	92.5488	78.5028	112.3605	**50.1124**	94.752	62.1067
	Rank	**1**	4	5	2	6	3
F10	AVG	**2928.3439**	2.93E + 03	4.27E + 03	2.93E + 03	3.14E + 03	2.98E + 03
	STD	23.1189	6.88E + 01	678.1578	**21.6507**	89.0578	39.0756
	Rank	**1**	2	6	2	5	4
Percentage		**1**	3.1	5.4	2.5	5.3	3.5
Total Rank		**1**	3	6	2	5	4

Best results are highlighted in bold

dimension benchmark functions; (ii) 5 unimodal fixed-dimension benchmark functions; (iii) 20 multimodal fixed-dimension benchmark functions; and (iv) 13 multimodal variable-dimension benchmark functions. Tables 20, 21, 22 and 23 list the names of all these methods and their associated attributes.

A table's global minimum is denoted by the column labeled"f_{min}," while the column labeled "Dim" displays the number of variables (design variables) for the functions. Additionally, the variable-dimension unimodal benchmark functions are described in Table 20; the fixed-dimension unimodal benchmark functions are shown in Table 21; the multimodal benchmark functions are described in Table 22; and the variable-dimension multimodal benchmark functions are presented in Table 23.

5.5.1 IHHO vs HHO

In Table 24, a comparison between HHO and IHHO is shown after putting it through its paces on the benchmark functions. This was done so that the best option could be determined. These functions thirty times for a total of five hundred iterations were evaluated.

According to unimodal variable-dimension benchmark functions, the results shown in Table 25 indicate that the performance of HHO is better than that of the original

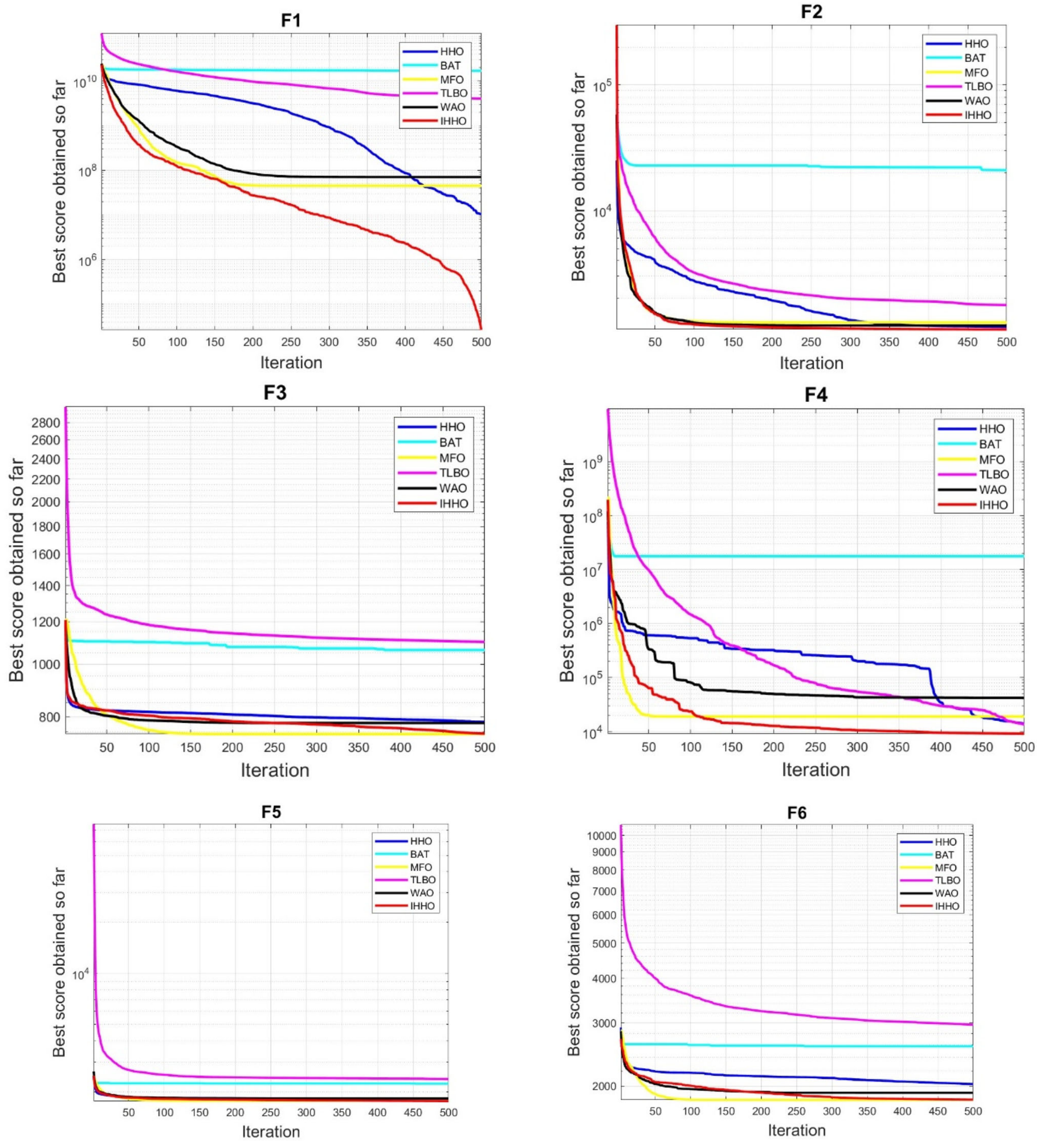


Fig. 21 Convergence curves for IHHO vs other algorithms on CEC2020

HHO, with the exception of F6 HHO and IHHO achieving the fitness function. The fitness function is attained via HHO and IHHO. Since IHHO uses a random walk strategy to break out of the local optimum, it outperforms the other approaches examined for modifying functions. Because of this, IHHO outperforms the alternatives. That is much to credit for IHHO's improved efficiency. Besides F5, F9,

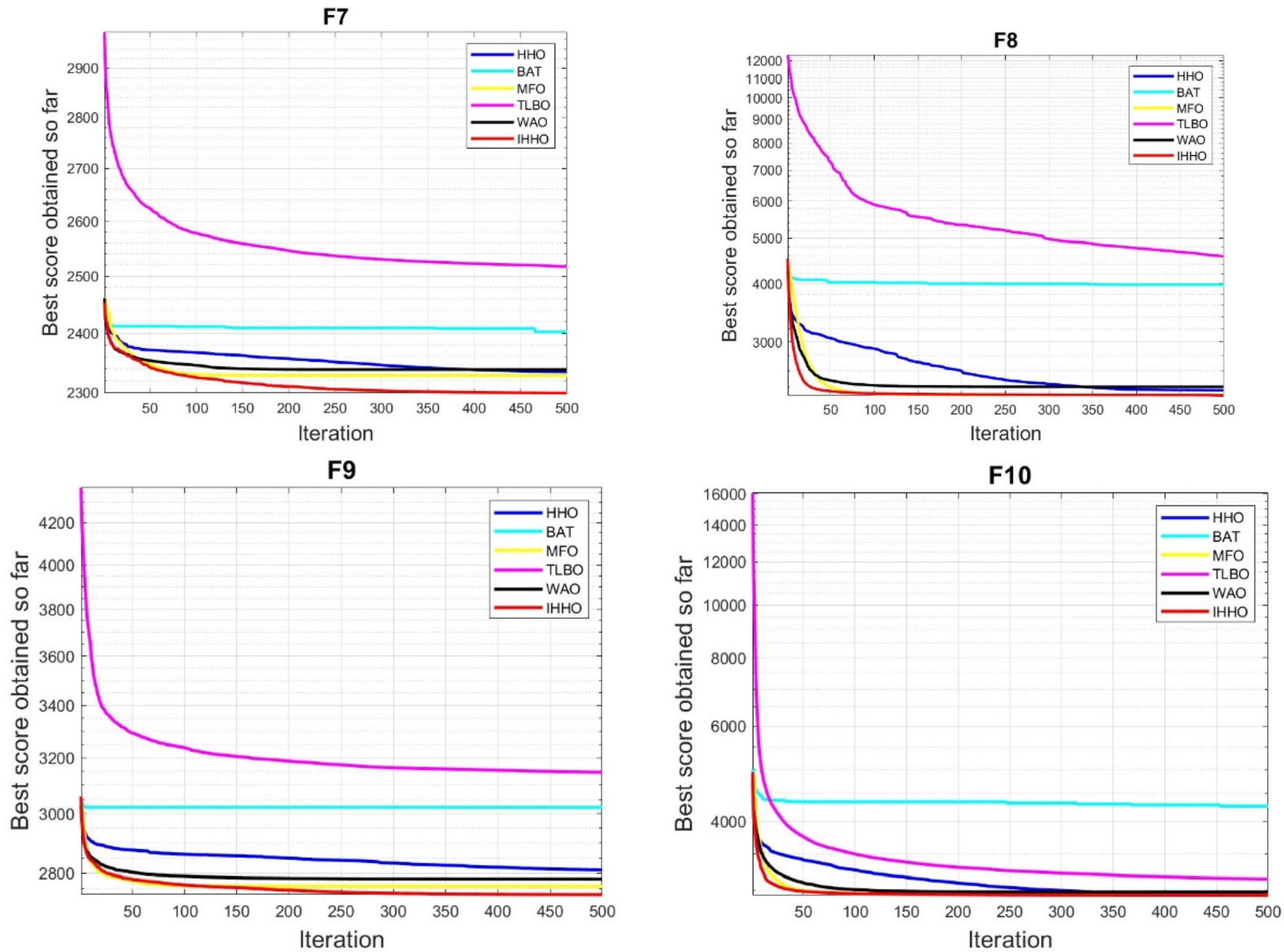


Fig. 21 continued

F11, and F14, IHHO satisfies the fitness function in all other functions.

Looking at the curves in Fig. 23, one can quickly and easily compare the convergence times of the original HHO technique with the original HHO.

Data from Table 24 show that IHHO outperforms the original HHO when compared to unimodal fixed-dimension benchmark functions. IHHO excels at changing functions because it utilizes a random walk method to escape the local optimum. This is why IHHO performs better than its competitors. Having that as a factor has greatly contributed to IHHO's increased productivity.

Convergence periods for the original HHO method and the original HHO may be easily compared from the curves in Fig. 24**.**

Table 24 reveals that when compared to multimodal fixed-dimension benchmark functions, IHHO performs better than the original HHO. Because it employs a random walk strategy to break out of the local optimum, IHHO is particularly effective at reshaping functions. The averages at which the HHO fitness was achieved, IHHO achieved as well, and the commentators that the original HHO could not achieve the required physical fitness, the modification achieved. Because of this, IHHO outperforms its rivals. The presence of such a component is a major reason for IHHO's improved output.

The curves in Fig. 25 allow for a direct comparison of the convergence times of the IHHO technique with the original HHO.

Table 24 reveals that when compared to multimodal fixed-dimension benchmark functions, IHHO performs better than the original HHO. IHHO is so powerful at function transformation because it uses a random walk method to escape the local optimum. Even while some critics claimed that the original HHO couldn't provide the desired level of physical fitness, IHHO did so on average at

Table 19 Comparison of results of IHHO vs other modifications of HHO on CEC2017

Benchmark		IHHO	HHO	BHHO[36]	MHHO[37]	LogHHO[38]
F1	AVG	2.17E + 04	9.91E + 06	3.77E + 06	1.47E + 06	5.52E + 06
	STD	**8666.9596**	6.56E + 06	4.12E + 06	8.42E + 05	7.26E + 06
	Rank	**1**	5	3	2	4
F2	AVG	**1123.0802**	1198.231	1.18E + 03	1.17E + 03	1.21E + 03
	STD	**1.28E + 01**	89.4682	45.4478	43.8321	66.5747
	Rank	**1**	4	3	2	5
F3	AVG	**740.3267**	803.3728	776.8391	784.2181	801.2626
	STD	**13.5653**	15.7372	23.8562	21.698	31.341
	Rank	**1**	5	2	3	4
F4	AVG	**7.19E + 03**	3.59E + 04	1.24E + 04	1.17E + 04	4.63E + 04
	STD	**6647.6979**	6.38E + 04	9.54E + 03	1.50E + 04	1.27E + 05
	Rank	**1**	4	3	2	5
F5	AVG	**1.76E + 03**	1.80E + 03	1.78E + 03	1.80E + 03	1.88E + 03
	STD	**18.3225**	7.52E + 01	25.6485	61.2233	118.0672
	Rank	**1**	3	2	3	5
F6	AVG	**1.7763e + 03**	1.92E + 03	1.88E + 03	1.94E + 03	2.09E + 03
	STD	**147.1462**	1.64E + 02	126.3299	128.9934	205.4413
	Rank	**1**	3	2	4	5
F7	AVG	**2.28E + 03**	2352.2351	2.30E + 03	2.34E + 03	2.35E + 03
	STD	60.2607	30.8292	61.2391	**38.6362**	46.3345
	Rank	**1**	5	2	3	4
F8	AVG	**2.29E + 03**	2376.921	2308.454	2359.7196	2463.2328
	STD	**24.7819**	237.0443	5.0388	11.2163	470.9754
	Rank	**1**	4	2	3	5
F9	AVG	**2719.675**	2822.6602	2765.9147	2798.9617	2867.0919
	STD	92.5488	78.5028	92.1919	74.8073	**58.9812**
	Rank	**1**	4	2	3	5
F10	AVG	**2928.3439**	2.93E + 03	2.94E + 03	2.93E + 03	2.94E + 03
	STD	23.1189	6.88E + 01	25.3672	**16.7946**	32.0362
	Rank	**1**	2	4	2	4
Percentage		**1**	3.9	2.5	2.7	4.6
Total Rank		**1**	4	2	3	5

Best results are highlighted in bold

or above that level. As a result, IHHO is more successful than its competitors. Having this component present is crucial to the increased efficiency of IHHO.

The curves in Fig. 26 allow for a direct comparison of the convergence times of the original HHO technique with the IHHO.

5.5.2 IHHO vs other algorithms

Table 25 contains our comparison of IHHO to various algorithms, namely GWO [39], BAT [40], MFO [42], TLBO [41], and WAO [43] after giving it a thorough workout on the test features to prepare for the engineering problems. This was done so that could make a decision regarding which of the possibilities was preferable. We have performed this evaluation a total of 30 times, making the total number of cycles 500.

According to Table 25, the performance of the IHHO algorithm is better than that of all unimodal variable-dimension benchmark functions, with the exception of the F6: IHHO, HHO, and MFO achieve the required fitness. IHHO, HHO, TLBO, and WAO achieve the requirement fitness for F8. IHHO outperforms BAT by a significant margin when the two are compared to one another.

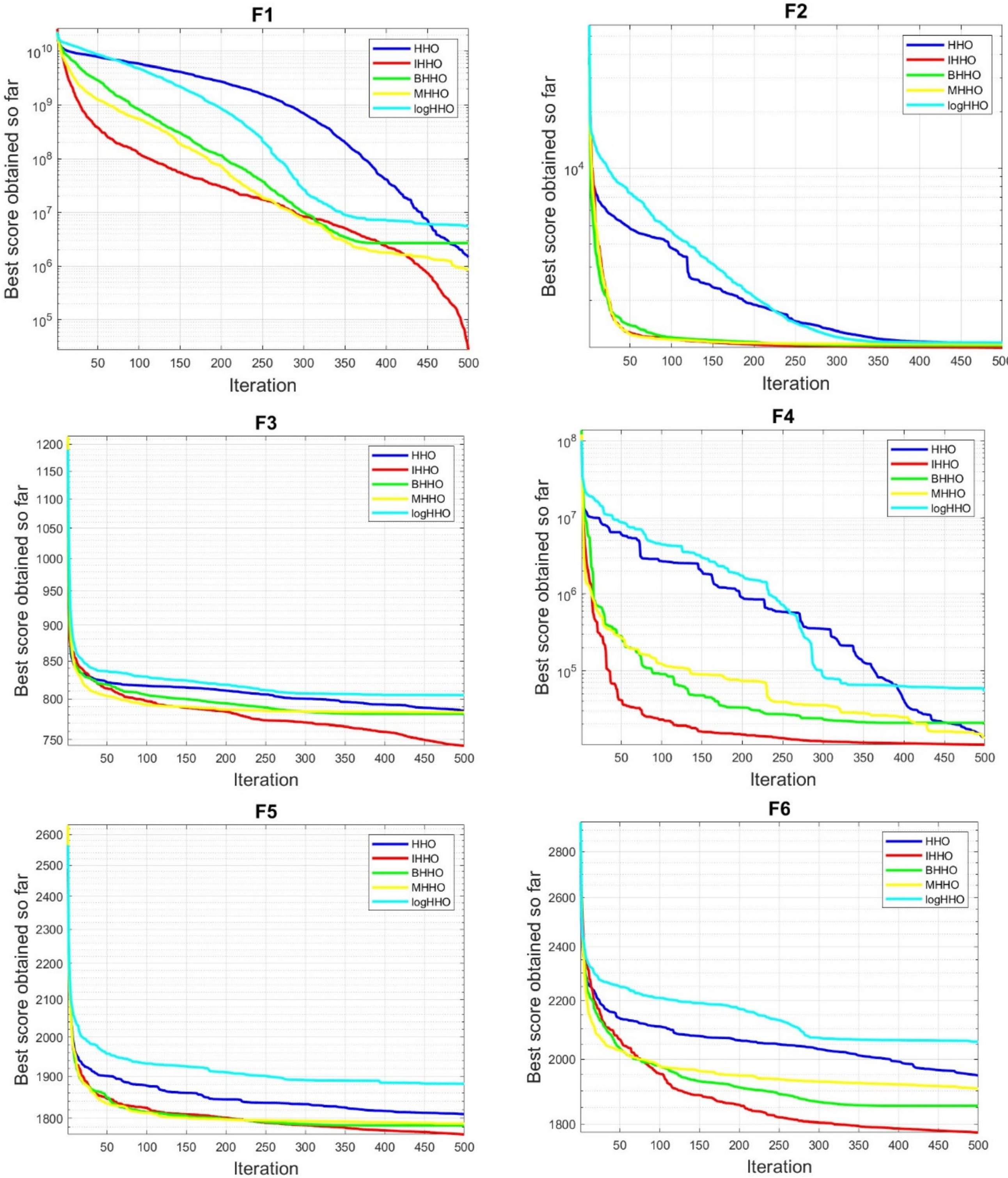


Fig. 22 Convergence curves for IHHO vs HHO modifications on CEC2020

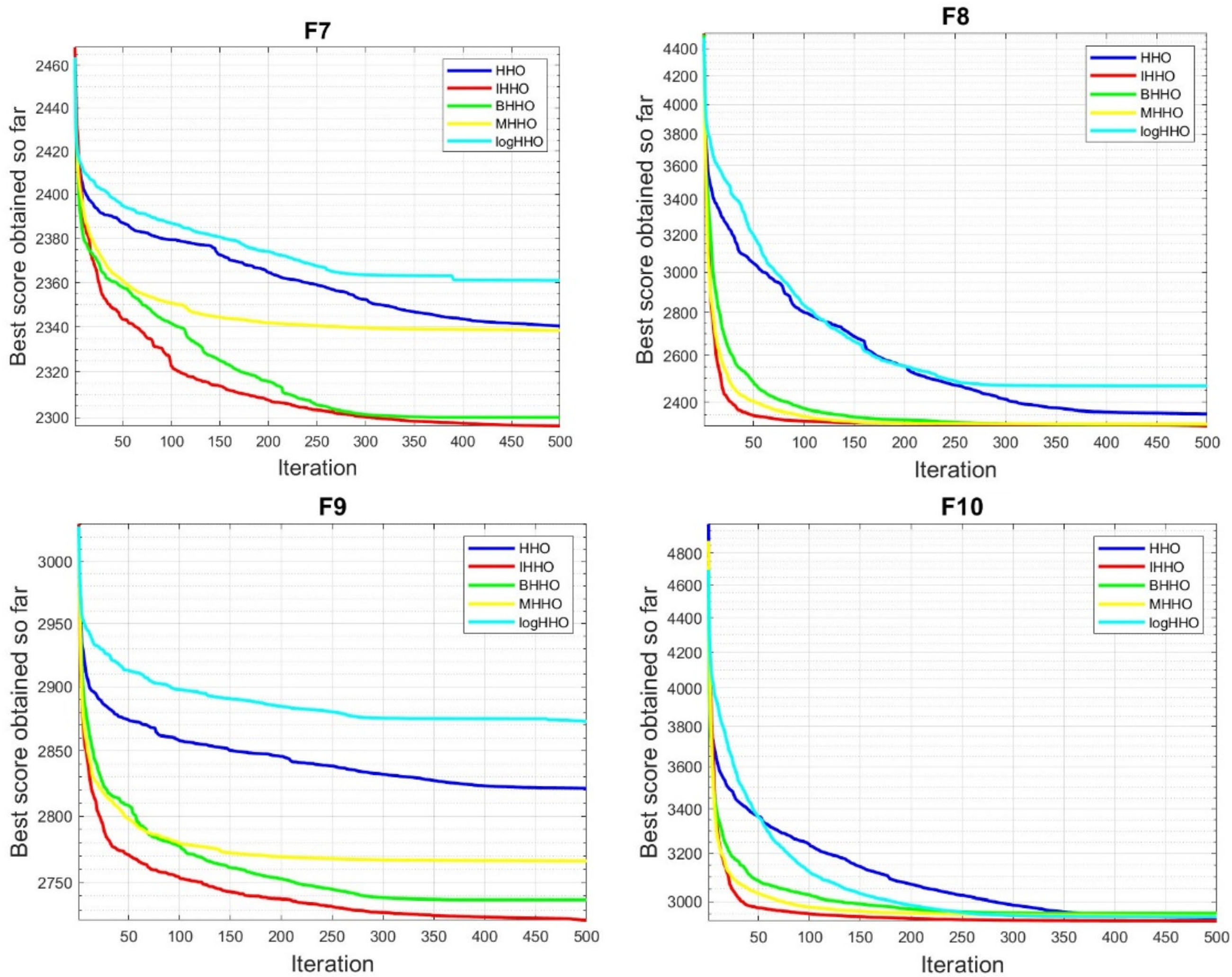


Fig. 22 continued

Because of this, IHHO outperforms its rivals. The presence of such a component is a major reason for IHHO's improved output. IHHO achieved an overall rank of 1.

A visual comparison of the convergence periods of the IHHO method and other algorithms can be carried out using the convergence curves shown in Fig. 27**.**

Table 25 shows that, except for F2, the IHHO method outperforms all of the unimodal fixed-dimension benchmark functions. When comparing IHHO to TLBO, IHHO is the superior option. In particular, the benchmark functions F1, F2, and F4 with a single variable dimension are intractable to TLBO. As a result, IHHO is more successful than its competitors. Having this component present is crucial to the increased efficiency of IHHO. IHHO ranked first among its competitors.

The convergence curves depicted in Fig. 28. can be used to visually examine the differences between the IHHO technique and other algorithms concerning the time required for them to reach convergence.

Table 25 reveals that when compared to multimodal fixed-dimension benchmark functions, IHHO performs better than the original HHO. Because it employs a random walk strategy to break out of the local optimum, IHHO is particularly effective at reshaping functions. The averages at which the HHO fitness was achieved, IHHO achieved as well, and the commentators that the original HHO could not achieve the required physical fitness, the modification achieved. In particular, some benchmark functions with a single variable dimension are intractable to TLBO. IHHO has a rank of 1 for these types of functions.

Table 20 A characterization of benchmark functions with variable dimensions and unimodality

Function no	Function name	Dim	Range	f_{min}
F1	Sphere	30	[−100, 100]	0
F2	Powell Sum	30	[−1, 1]	0
F3	Schwefel's 2.20	30	[−100, 100]	0
F4	Schwefel's 2.21	30	[−100, 100]	0
F5	Step	30	[−100, 100]	0
F6	Stepint	30	[−5.12, 5.12]	−155
F7	Schwefel's 2.22	30	[−100, 100]	0
F8	Schwefel's 2.23	30	[−10, 10]	0
F9	Rosenbrock	30	[−30, 30]	0
F10	Brown	30	[−1, 4]	0
F11	Dixon and Price	30	[−10, 10]	0
F12	Powell Singular	30	[−4, 5]	0
F13	Perm 0, D, Beta	5	[−dim, dim]	0
F14	Sum Squares	30	[−10, 10]	0

Table 21 A characterization of benchmark functions with fixed dimensions and unimodality

Function no	Function name	Dim	Range	f_{min}
F1	Booth	2	[−10, 10]	0
F2	Brent	2	[−10, 10]	0
F3	Matyas	2	[−10, 10]	0
F4	Schaffer N. 4	2	[−100, 100]	0.292579
F5	Wayburn Seader 3	2	[−500, 500]	19.10588

The curves in Fig. 29 allow for a direct comparison of the convergence times of the IHHO technique with other algorithms.

Table 25 shows that IHHO outperforms all of the multimodal variable-dimension benchmark functions except F2 and F12. When comparing IHHO to TLBO, IHHO is the superior option. As a result, IHHO is more successful than its competitors. Having this component present is crucial to the increased efficiency of IHHO.

The convergence curves depicted in Fig. 30 can be used to visually examine the differences between the IHHO technique and other algorithms concerning the time required for them to reach convergence.

5.5.3 IHHO vs other modifications of HHO

As such, the proposed IHHO to existing HHO variations, namely BHHO[26], MHHO [27], and LogHHO [28], was compared. To give all algorithms a fair opportunity, we've decided to set the swarm size at 30, and the stopping condition at 500 iterations. The experimental findings in Table 26 address the issue of scalability. The objective function's mean and standard deviation are shown in the table below. The convergence outcomes of the various methods are shown in Figs. 31, 32, 33, and 34.

According to Table 26, the performance of the IHHO algorithm is better than that of all other modifications of HHO, except the F11 and F13 methods. When compared to MHHO and LogHHO, IHHO performs better for all unimodal variable-dimension benchmark functions. As a result, IHHO is more successful than its competitors. Having this component present is crucial to the increased efficiency of IHHO. A visual comparison of the convergence periods of the IHHO method can be carried out using the Convergence curves shown in Fig. 31**.**

According to Table 26, the performance of the IHHO algorithm is better than or equal to the required fitness of all other modifications of HHO, with the exception of the F1 method. As a result, IHHO is more successful than its competitors. Having this component present is crucial to the increased efficiency of IHHO. Utilizing the convergence curves that are depicted in Fig. 32, one can carry out a visual comparison of the convergence periods of different HHO modifications.

Table 26 reveals that when compared to multimodal fixed-dimension benchmark functions, IHHO performs better than the original HHO and other HHO modifications or achieves the required fitness. Because it employs a random walk strategy to break out of the local optimum, IHHO is particularly effective at reshaping functions.

Table 22 An explanation of benchmark functions for multimodal that are fixed in dimension

Function no	Function name	Dim	Range	f_{min}
F1	Egg Crate	2	[−500,500]	0
F2	Ackley N.3	2	[−32, 32]	−195.629
F3	Adjiaman	2	[−1, 2]	−2.02181
F4	Bird	2	[−2pi, 2pi]	−106.765
F5	Branin RCOS	2	[−5, 5]	0.397887
F6	Cross−in−tray	2	[−10, 10]	−2.06261
F7	Bartels Conn	2	[−500, 500]	1
F8	Bukin 6	2	[(−15, −5), (−5, −3)]	180.3276
F9	Carrom Table	2	[−10, 10]	−24.1568
F10	Chichinadze	2	[−30, 30]	−43.3159
F11	Cross Leg Table	2	[−10, 10]	−1
F12	Easom	2	[−100, 100]	−1
F13	Giunta	2	[−1, 1]	0.060447
F14	Helical Valley	3	[−10, 10]	0
F15	Holder	2	[−10, 10]	−19.2085
F16	Pen Holder	2	[−11, 11]	−0.96354
F17	Test Tube Holder	2	[−10, 10]	−10.8723
F18	Shubert	2	[−10, 10]	−186.731
F19	Shekel	4	[0, 10]	−10.5364
F20	Three-Hump Camel	2	[−5, 5]	0

Table 23 An explanation of benchmark functions for multimodal that are variable in dimension

Function no	Function name	Dim	Range	f_{min}
F1	Schwefer's 2.26	30	[−500,500]	−418.983
F2	Periodic	30	[−10, 10]	0.9
F3	Qing	30	[−500, 500]	0
F4	Alpine N. 1	30	[−10, 10]	0
F5	Xin-She Yang	30	[−5, 5]	0
F6	Trignometric 2	30	[−500, 500]	0
F7	Salomon	30	[−100, 100]	0
F8	Styblinski-Tang	30	[−5, 5]	−1174.98
F9	Xin-She Yang N. 2	30	[−2pi, 2pi]	0
F10	Gen. Penalized	30	[−50, 50]	0
F11	Penalized	30	[−50, 50]	0
F12	Michalewics	30	[0, pi]	−29.6309
F13	Quartic Noise	30	[−1.28, 1.28]	0

Because of this, IHHO outperforms its rivals. The presence of such a component is a major reason for IHHO's improved output. The curves in Fig. 33 allow for a direct comparison of the convergence times of the IHHO technique with different HHO modifications.

According to Table 26, the performance of the HHO algorithm is better than that of all other modifications of HHO, with the exception of the F12 method. When compared to MHHO and LogHHO, IHHO performs better for all unimodal variable-dimension benchmark functions. As

Table 24 Comparison of results of IHHO and HHO on 52 benchmark functions

Benchmark	optimizer	Mean	STD	MIN	MAX
Unimodal variable-dimension					
F1	IHHO	**0**	**0**	0	0
	HHO	1.3961e−91	7.6465e−91	1.2707e−113	4.1882e−90
F2	IHHO	**0**	**0**	0	0
	HHO	2.2961e−124	7.8038e−124	6.7471e−154	3.9174e−123
F3	IHHO	**0**	**0**	0	0
	HHO	1.6936e−49	8.9877e−49	7.6813e−57	4.9275e−48
F4	IHHO	**0**	**0**	0	0
	HHO	2.1191e−50	6.4085e−50	2.3044e−57	3.2259e−49
F5	IHHO	**1.269e−05**	**3.7067e−05**	6.2893e−11	0.00065825
	HHO	0.00020706	0.00034121	1.805e−09	0.00166
F6	IHHO	**−155**	**0**	−155	−155
	HHO	**−155**	**0**	−155	−155
F7	IHHO	**0**	**0**	0	0
	HHO	6.9426e−50	1.8065e−49	1.7956e−59	8.0488e−49
F8	IHHO	**0**	**0**	0	0
	HHO	**0**	**0**	0	0
F9	IHHO	**0.002708**	**0.0049833**	3.6982e−07	0.019291
	HHO	0.013231	0.01834	3.827e−06	0.067915
F10	IHHO	**0**	**0**	0	0
	HHO	1.3167e−98	4.9725e−98	1.2315e−115	2.131e−97
F11	IHHO	0.24973	0.00088237	0.24879	0.25399
	HHO	0.24971	0.00070803	0.24732	0.2515
F12	IHHO	**0**	**0**	0	0
	HHO	7.5986e−97	4.1565e−96	1.935e−114	2.2767e−95
F13	IHHO	**37.259**	**29.048**	0.39056	99.016
	HHO	126.6738	225.3297	0.9636	1075.7969
F14	IHHO	**0**	**0**	0	0
	HHO	4.8407e−96	2.5151e−95	2.6014e−114	1.3782e−94
Unimodal fixed-dimension					
F1	IHHO	**2.0815e−05**	**2.3061e−05**	3.3123e−07	0.00010228
	HHO	5.9535e−05	8.1805e−05	2.1613e−07	0.00033879
F2	IHHO	**1.3839e−87**	**6.8117e−103**	1.3839e−87	1.3839e−87
	HHO	**1.3839e−87**	**6.8117e−103**	1.3839e−87	1.3839e−87
F3	IHHO	**0**	**0**	0	0
	HHO	1.0717e−122	5.8701e−122	1.3295e−162	3.2152e−121
F4	IHHO	**0.29258**	1.2316e−06	0.29258	0.29258
	HHO	**0.29258**	**5.9415e−07**	0.29258	0.29258
F5	IHHO	**19.106**	**8.9581e−05**	19.1059	19.1062
	HHO	19.1062	0.00060229	19.1059	19.1087

Table 24 (continued)

Benchmark	optimizer	Mean	STD	MIN	MAX
Multimodal fixed-dimension					
F1	IHHO	**0**	**0**	0	0
	HHO	1.2975e−109	6.8953e−109	4.7176e−128	3.779e−108
F2	IHHO	**−195.629**	1.5244e−05	−195.629	−195.629
	HHO	**−195.629**	**7.4381e−07**	−195.629	−195.629
F3	IHHO	**−2.0218**	1.9896e−07	−2.0218	−2.0218
	HHO	**−2.0218**	**9.9301e−16**	−2.0218	−2.0218
F4	IHHO	**−106.7634**	0.0013021	−106.7645	−106.7601
	HHO	**−106.7645**	**1.3424e−05**	−106.7645	−106.7645
F5	IHHO	**0.3979**	**1.4258e−05**	0.39789	0.39795
	HHO	0.39789	3.9394e−06	0.39789	0.39791
F6	IHHO	**−2.0626**	2.6641e−07	−2.0626	−2.0626
	HHO	**−2.0626**	2.5808e−08	−2.0626	−2.0626
F7	IHHO	**1**	**0**	1	1
	HHO	**1**	**0**	1	1
F8	IHHO	**180.3276**	**0**	180.3276	180.3276
	HHO	**180.3276**	**0**	180.3276	180.3276
F9	IHHO	**−24.1568**	0.00016531	−24.1568	−24.1562
	HHO	**−24.1568**	**1.4263e−06**	−24.1568	−24.1568
F10	IHHO	**−42.7648**	**0.2223**	−42.9444	−42.4971
	HHO	−42.7355	0.22672	−42.9444	−42.4972
F11	IHHO	**−0.50015**	**0.49812**	−1	−0.0001596
	HHO	−0.036205	0.30403	−1	−0.0021456
F12	IHHO	**−0.99999**	**9.0857e−06**	−1	−0.99996
	HHO	−0.99998	2.1463e−05	−1	−0.99993
F13	IHHO	**0.06447**	**1.8395e−07**	0.06447	0.064471
	HHO	0.064471	3.4941e−07	0.06447	0.064472
F14	IHHO	0.056121	0.079838	3.5001e−05	0.35368
	HHO	**0.02485**	**0.051972**	6.4552e−06	0.24897
F15	IHHO	**−19.2084**	0.00013123	−19.2085	−19.2079
	HHO	**−19.2085**	**3.2574e−12**	−19.2085	−19.2085
F16	IHHO	**−0.96353**	1.6112e−07	−0.96353	−0.96353
	HHO	**−0.96353**	3.1589e−09	−0.96353	−0.96353
F17	IHHO	**−10.8716**	**0.0036257**	−10.8723	−10.8524
	HHO	−10.867	0.008909	−10.8723	−10.8525
F18	IHHO	**−186.7197**	0.014362	−186.7308	−186.6707
	HHO	**−186.7303**	**0.0022392**	−186.7309	−186.7216
F19	IHHO	**−10.529**	**0.012416**	−10.5363	−10.4839
	HHO	−5.1252	1.5814	−5.1285	−5.108
F20	IHHO	**0**	**0**	0	0
	HHO	4.8345e−107	2.6475e−106	1.6585e−131	1.4501e−105

Table 24 (continued)

Benchmark	optimizer	Mean	STD	MIN	MAX
Multimodal variable-dimension					
F1	IHHO	**0.0051495**	**0.0049582**	4.9861e−05	0.015189
	HHO	1.722	9.2336	0.00020745	50.61
F2	IHHO	**0.9**	**4.5168e−16**	0.9	0.9
	HHO	**0.9**	**4.5168e−16**	0.9	0.9
F3	IHHO	**132.2224**	**35.2382**	77.6757	218.557
	HHO	501.6447	113.7039	329.5382	791.1485
F4	IHHO	**0**	**0**	0	0
	HHO	1.0634e−49	5.8038e−49	4.3256e−61	3.1792e−48
F5	IHHO	**1.2369e−284**	**0**	0	3.7108e−283
	HHO	1.2345e−15	6.5949e−15	1.7724e−57	3.6141e−14
F6	IHHO	**1.0003**	**0.00058655**	1	1.0029
	HHO	1.0008	0.0008597	1	1.003
F7	IHHO	**0**	**0**	0	0
	HHO	5.4413e−48	1.8677e−47	3.7222e−58	7.6802e−47
F8	IHHO	**−1174.9747**	0.014211	−1174.985	−1174.9336
	HHO	**−1174.978**	**0.0093998**	−1174.985	−1174.9423
F9	IHHO	**3.5134e−12**	3.5146e−12	3.5124e−12	3.5187e−12
	HHO	**3.5134e−12**	1.7099e−15	3.5124e−12	3.52e−12
F10	IHHO	**9.3135e−06**	**2.3563e−05**	3.9671e−10	0.00010955
	HHO	9.9379e−05	0.00013036	1.6528e−06	0.0005777
F11	IHHO	**1.6483e−06**	**3.3941e−06**	1.4336e−10	1.6366e−05
	HHO	7.4312e−06	1.2137e−05	2.1301e−08	5.8061e−05
F12	IHHO	**−11.5431**	1.4046	−14.3447	−8.7851
	HHO	−11.2914	**1.261**	−14.0072	−9.1013
F13	IHHO	**6.6808e−05**	**6.9955e−05**	1.3118e−06	0.00030964
	HHO	0.00011582	8.191e−05	2.2222e−06	0.00033512

Best results are highlighted in bold

a result, IHHO is more successful than its competitors. Having this component present is crucial to the increased efficiency of IHHO.

A visual comparison of the convergence periods of the HHO method can be carried out using the Convergence curves shown in Fig. 34.

5.6 Engineering problems

5.6.1 Tension/compression spring design

Designing coil springs with the ideal tension and compression is a classic engineering optimization challenge [88]. Figure 35 shows the tension/compression spring design problem (TCSD). Assuming a constant tension/compression load, the goal is to minimize the volume V of the coil spring. There are three potential configurations for this issue:

- the number of usable coils in a spring $P = x_1 \epsilon [2, 15]$
- the size of the winding in centimeters $D = x_2 \epsilon [0.25, 1.3]$
- the size of the wire's diameter $d = x_3 \epsilon [0.005, 2]$

The TCSD problem may be expressed mathematically as follows: [89]

$$\min f(x) = (x_3 + 2) x_2 x_1^2 \tag{25}$$

Subject to:

Table 25 Comparison of results between IHHO and other algorithms on 52 benchmark functions

Benchmark		IHHO	HHO	GWO [39]	BAT [31]	MFO [42]	TLBO [41]	WAO [43]
Unimodal variable-dimension								
F1	AVG	**0**	1.3961e−91	1.8022e−27	37,343.0714	1670.6967	3.2597e−89	4.8564e−74
	STD	**0**	7.6465e−91	3.8421e−27	9257.5225	3789.9264	7.288e−89	1.6938e−73
	Rank	**1**	2	5	7	6	3	4
F2	AVG	**0**	2.2961e−124	1.902e−95	0.37327	6.4086e−10	1.3014e−203	1.6048e−110
	STD	**0**	7.8038e−124	9.4904e−95	0.25834	1.4853e−09	**0**	4.3318e−110
	Rank	**1**	3	5	7	6	2	4
F3	AVG	**0**	1.6936e−49	7.4123e−16	852.4312	98.2681	3.8442e−44	5.6055e−49
	STD	**0**	8.9877e−49	4.4816e−16	133.3365	88.6765	4.9269e−44	2.4127e−48
	Rank	**1**	2	5	7	6	4	3
F4	AVG	**0**	2.1191e−50	7.6834e−07	69.3653	69.8968	1.5e−36	49.5197
	STD	**0**	6.4085e−50	7.6093e−07	7.5904	8.7229	2.3336e−36	33.6296
	Rank	**1**	2	4	6	7	3	5
F5	AVG	**1.269e−05**	0.00020706	0.64605	37,391.8848	2016.7255	4.9657e−05	0.42027
	STD	**3.7067e−05**	0.00034121	0.36754	9192.5316	6691.3898	0.00017474	0.21009
	Rank	**1**	3	5	7	6	2	4
F6	AVG	**−155**	**−155**	−131.1667	−1390.7333	**−155**	−154.1333	**−155**
	STD	**0**	**0**	8.2674	735.764	**0**	0.9732	**0**
	Rank	**1**	**1**	6	7	**1**	5	**1**
F7	AVG	**0**	6.9426e−50	1.8035e−15	2.603928623e + 37	484.9465	9.3136e−44	6.8873e−51
	STD	**0**	1.8065e−49	1.3291e−15	1.349221071e + 38	245.176	8.8213e−44	2.5321e−50
	Rank	**1**	3	5	7	6	4	2
F8	AVG	**0**	**0**	1.223e−91	1,451,309,551.287	5345.6879	**0**	5.1076e−171
	STD	**0**	**0**	3.5588e−91	1,428,255,045.428	29,007.9513	**0**	**0**
	Rank	**1**	**1**	5	7	6	**1**	4
F9	AVG	**0.002708**	0.013231	26.8034	101,550,331.1531	13,129.6221	25.2246	28.0555
	STD	**0.0049833**	0.01834	0.63772	56,718,510.1868	30,865.6438	0.6671	0.56887
	Rank	**1**	2	4	7	6	3	5
F10	AVG	**0**	1.3167e−98	3.173e−30	215.5183	14.4953	3.701e−92	2.7116e−78
	STD	**0**	4.9725e−98	4.2129e−30	310.4292	12.1753	6.91e−92	1.1048e−77
	Rank	**1**	2	5	7	6	3	4
F11	AVG	**0.24973**	**0.24971**	0.66668	694,908.3504	34,894.498	0.66667	0.66696
	STD	0.00088237	0.00070803	2.2816e−05	396,654.3773	114,263.6097	**2.013e−10**	0.00018296
	Rank	2	**1**	4	7	6	3	5

Table 25 (continued)

Benchmark		IHHO	HHO	GWO [39]	BAT [31]	MFO [42]	TLBO [41]	WAO [43]
F12	AVG	**0**	7.5986e−97	1.5647e−05	6550.1938	701.2343	7.8844e−07	1.6308e−06
	STD	**0**	4.1565e−96	1.693e−05	3619.1511	924.3802	1.8043e−06	4.4186e−06
	Rank	**1**	2	5	7	6	3	4
F13	AVG	**37.259**	126.6738	326.2628	88,500.9977	219.1656	1.019420365975e + 89	969.058
	STD	**29.048**	225.3297	1065.9402	246,886.6119	1150.2005	7.188207066634e + 73	1155.8174
	Rank	**1**	2	4	6	3	7	5
F14	AVG	**0**	4.8407e−96	1.2883e−28	5631.4252	512.2511	4.8252e−90	5.444e−74
	STD	**0**	2.5151e−95	1.5634e−28	1868.3183	565.3924	1.4809e−89	2.2568e−73
	Rank	**1**	2	5	7	6	3	4
Percentage		**1.07142857**	**2**	**4.785714286**	**6.857142857**	**5.5**	**3.285714286**	**3.857142857**
Total Rank		**1**	**2**	**5**	**7**	**6**	**3**	**4**
Unimodal fixed-dimension								
F1	AVG	2.0815e−05	5.9535e−05	5.7558e−07	3.2958	**0**	NAN	0.0014649
	STD	2.3061e−05	8.1805e−05	7.2274e−07	5.0174	**0**	NAN	0.0022924
	Rank	3	4	2	6	**1**		5
F2	AVG	**1.3839e−87**	**1.3839e−87**	**1.3839e−87**	3.0956	**1.3839e−87**	NAN	**1.3839e−87**
	STD	**6.8117e−103**	**6.8117e−103**	**6.8117e−103**	5.7217	**6.8117e−103**	NAN	**6.8117e−103**
	Rank	**1**	**1**	**1**	6	**1**		**1**
F3	AVG	**0**	1.0717e−122	1.4753e−103	0.15663	7.0536e−33	2.4907e−179	1.0224e−193
	STD	**0**	5.8701e−122	8.0518e−103	0.2636	3.859e−32	**0**	**0**
	Rank	**1**	4	5	7	6	3	2
F4	AVG	**0.29258**	**0.29258**	**0.29258**	0.37827	**0.29276**	NAN	0.29259
	STD	1.2316e−06	5.9415e−07	**4.4173e−07**	0.054727	**0.00040112**	NAN	2.3168e−05
	Rank	**1**	**1**	**1**	6	5		4
F5	AVG	**19.106**	19.1062	**19.1059**	12,525,493.3395	**19.1059**	146.188	19.1158
	STD	8.9581e−05	0.00060229	1.6316e−05	29,032,950.1484	**1.082e−14**	174.3419	0.0092434
	Rank	3	4	**1**	7	**1**	6	5
Percentage		**1.8**	**2.8**	**2**	**6.4**	**2.8**	**–**	**3.4**
Total Rank		1	**3**	**2**	**6**	**3**	**–**	**5**
Multimodal fixed-dimension								
F1	AVG	**0**	1.2975e−109	2.0635e−212	6.0426	2.7831e−101	NAN	1.6118e−106
	STD	**0**	6.8953e−109	**0**	4.2873	1.5244e−100	NAN	6.4787e−106
	Rank	**1**	3	2	6	5		4
F2	AVG	**−195.629**	**−195.629**	**−195.629**	−185.7	**−195.629**	NAN	**−195.629**
	STD	1.5244e−05	7.4381e−07	1.1993e−07	7.1002	**5.7815e−14**	NAN	1.2607e−07
	Rank	**1**	**1**	**1**	6	**1**		**1**

Table 25 (continued)

Benchmark		IHHO	HHO	GWO [39]	BAT [31]	MFO [42]	TLBO [41]	WAO [43]
F3	AVG	**−2.0218**	**−2.0218**	**−2.0218**	−2.0781	**−2.0218**	NAN	**−2.0218**
	STD	1.9896e−07	9.9301e−16	6.9059e−12	0.19718	1.355e−15	NAN	**8.92e−16**
	Rank	**1**	**1**	**1**	2	**1**		**1**
F4	AVG	**−106.7634**	**−106.7645**	−105.4676	−85.7692	**−106.7645**	NAN	**−106.7645**
	STD	0.0013021	1.3424e−05	4.9355	18.8431	**2.5035e−14**	NAN	5.8234e−05
	Rank	4	**1**	5	6	**1**		**1**
F5	AVG	**0.3979**	**0.39789**	**0.39789**	0.88488	**0.39789**	9.176	**0.39789**
	STD	1.4258e−05	3.9394e−06	9.4347e−07	0.81198	**0**	9.9162	2.3114e−05
	Rank	5	**1**	**1**	6	**1**	7	**1**
F6	AVG	**−2.0626**	**−2.0626**	**−2.0626**	−2.0423	**−2.0626**	NAN	**−2.0626**
	STD	2.6641e−07	2.5808e−08	1.2856e−08	0.038566	**9.0336e−16**	NAN	4.4175e−09
	Rank	**1**	**1**	**1**	6	**1**		1
F7	AVG	**1**	**1**	**1**	2409.2139	**1**	NAN	1
	STD	**0**	**0**	**0**	2608.7417	**0**	NAN	4.1233e−17
	Rank	**1**	**1**	**1**	6	**1**		**1**
F8	AVG	**180.3276**	**1180.3276**	**180.3276**	0.20402	**180.3276**	NAN	**180.3276**
	STD	**0**	**0**	**0**	0.069127	**0**	NAN	**0**
	Rank	**2**	**2**	**2**	1	**2**		**2**
F9	AVG	**−24.1568**	**−24.1568**	−24.1567	−23.1595	**−24.1568**	−11.6341	−24.1565
	STD	0.00016531	1.4263e−06	0.00010797	1.5313	**9.8738e−15**	10.2445	0.00055723
	Rank	**1**	**1**	4	6	**1**	7	5
F10	AVG	−42.7648	−42.7355	−42.884	−36.0901	**−42.9444**	−38.5723	−42.7135
	STD	0.2223	0.22672	0.15433	6.8311	**3.6134e−14**	6.5369	0.2205
	Rank	3	4	2	7	**1**	6	5
F11	AVG	**−0.50015**	−0.036205	−0.03365	−0.00028692	−0.076047	−0.10035	−0.00050469
	STD	0.49812	0.30403	0.18251	**4.1584e−05**	0.024931	0.30501	0.00019414
	Rank	**1**	4	5	7	3	2	6
F12	AVG	−0.99999	−0.99998	**−1**	−0.021348	**−1**	−0.26636	−0.99999
	STD	9.0857e−06	2.1463e−05	5.4995e−07	0.11692	**0**	0.4204	6.1525e−05
	Rank	3	5	**1**	7	**1**	6	3
F13	AVG	**0.06447**	0.064471	**0.06447**	0.082874	**0.06447**	−4.232	**0.06447**
	STD	1.8395e−07	3.4941e−07	1.4756e−08	0.023617	**6.3544e−17**	0.30633	2.8101e−08
	Rank	**1**	6	**1**	7	1	5	**1**
F14	AVG	0.056121	**0.02485**	0.21984	441.1187	0.22735	4.7001	2.088
	STD	0.079838	**0.051972**	1.1385	553.3828	0.44928	6.5339	4.7018
	Rank	2	**1**	3	7	4	6	5

Table 25 (continued)

Benchmark		IHHO	HHO	GWO [39]	BAT [31]	MFO [42]	TLBO [41]	WAO [43]
F15	AVG	**−19.2084**	**−19.2085**	−19.2084	−22.1686	−19.1689	−10.8782	**− 19.2085**
	STD	0.00013123	**3.2574e−12**	5.5976e−05	5.7414	0.21686	6.5776	2.8993e−08
	Rank	**4**	**2**	4	**1**	6	7	**2**
F16	AVG	**−0.96353**	**−0.96353**	**−0.96353**	−0.95371	**−0.96353**	−0.91013	**−0.96353**
	STD	1.6112e−07	3.1589e−09	1.2148e−07	0.01294	**0**	0.07633	2.5537e−07
	Rank	**1**	**1**	**1**	6	**1**	7	**1**
F17	AVG	**−10.8716**	−10.867	−10.8703	−10.3502	−10.8515	−10.7605	−10.8657
	STD	**0.0036257**	0.008909	0.0060327	0.41282	0.032502	0.29379	0.0094869
	Rank	**1**	3	2	7	5	6	4
F18	AVG	**−186.7197**	**−186.7303**	−186.6864	−108.4	−186.7309	−141.8861	−186.7307
	STD	0.014362	0.0022392	0.099328	56.3968	**4.1221e−14**	51.376	0.00047414
	Rank	4	3	5	7	**1**	6	2
F19	AVG	**−10.529**	−5.1252	−9.8135	−1.2053	−7.8423	−4.4846	−7.8384
	STD	**0.012416**	1.5814	2.2381	0.62568	3.6423	2.3076	3.2162
	Rank	**1**	5	2	7	3	6	4
F20	AVG	**0**	4.8345e−107	1.9432e−181	0.25008	3.0353e−105	NAN	1.0124e−76
	STD	**0**	2.6475e−106	**0**	0.23913	1.1525e−104	NAN	5.5274e−76
	Rank	**1**	3	2	6	4		5
Percentage		**1.84891**	**2.34891**	**2.19891**	**5.246095**	**2.09891**	**–**	**2.64891**
Total Rank		**1**	**4**	**3**	**6**	**2**	**–**	**5**
Multimodal fixed-dimension								
F1	AVG	**0.0051495**	1.722	220.0629	210.8508	129.1065	153.0347	111.9213
	STD	**0.0049582**	9.2336	27.3299	151.3665	27.5724	23.4733	48.9784
	Rank	**1**	2	7	6	4	5	3
F2	AVG	**0.9**	**0.9**	1.4297	8.4341	4.4136	3.8335	1.3472
	STD	**4.5168e−16**	**4.5168e−16**	0.33164	1.7521	0.83239	1.8613	0.70971
	Rank	**1**	**1**	4	7	6	5	3
F3	AVG	132.2224	501.6447	1172.6382	80,109,481,734.085	2,083,680,638.5093	**4.1345**	1433.6676
	STD	35.2382	113.7039	593.3588	46,353,618,892.814	11,410,057,520.2504	**15.3963**	543.6294
	Rank	2	3	4	7	6	**1**	5
F4	AVG	**0**	1.0634e−49	0.00047832	50.0535	5.4671	4.6284e−06	0.46375
	STD	**0**	5.8038e−49	0.00054372	8.2233	7.7376	2.5351e−05	2.5401
	Rank	**1**	2	4	7	6	3	5
F5	AVG	**1.2369e−284**	1.2345e−15	6.905e−24	281,824,890,801.55	621,523,678.3356	1.2556e−12	0.00016969
	STD	**0**	6.5949e−15	3.347e−23	689,018,099,127.68	1,666,560,599.8684	5.6459e−12	0.00081592
	Rank	**1**	3	2	7	6	4	5

Table 25 (continued)

Benchmark		IHHO	HHO	GWO [39]	BAT [31]	MFO [42]	TLBO [41]	WAO [43]
F6	AVG	**1.0003**	1.0008	25.6756	1,010,467.5447	33,886.1867	17.3105	75.646
	STD	**0.00058655**	0.0008597	5.6748	299,572.2861	86,324.9242	4.8431	20.4743
	Rank	**1**	2	4	7	6	3	5
F7	AVG	**0**	5.4413e−48	0.18654	19.2454	5.9634	0.10987	0.11656
	STD	**0**	1.8677e−47	0.043417	3.0361	3.8622	0.030513	0.069864
	Rank	**1**	2	5	7	6	3	4
F8	AVG	**−1174.9747**	**−1174.978**	−897.0189	−532.783	−1033.0765	−1022.888	−1103.0024
	STD	0.014211	**0.0093998**	45.8673	66.4812	33.898	32.5882	91.5944
	Rank	2	**1**	6	7	4	5	3
F9	AVG	**3.5134e−12**	3.5146e−12 1.7099e−15	5.587e−08	3.2946e−11	2.4743e−11	3.1006e−11	4.5022e−12
	STD	**1.3473e−15**	2	2.2181e−07	1.9698e−12	3.0665e−12	5.5419e−13	2.1881e−12
	Rank	**1**		7	6	4	5	3
F10	AVG	**9.3135e−06**	9.9379e−05	0.66498	426,117,465.501	670.2591	0.067276	0.54056
	STD	**2.3563e−05**	0.00013036	0.27118	247,458,331.271	3240.8399	0.079853	0.28713
	Rank	**1**	2	5	7	6	3	4
F11	AVG	**1.6483e−06**	7.4312e−06	0.042683	174,524,910.476	1763.0443	0.0034561	0.017917
	STD	**3.3941e−06**	1.2137e−05	0.018153	153,372,104.769	9538.1426	0.018928	0.011744
	Rank	**1**	2	5	7	6	3	4
F12	AVG	−11.5431	−11.2914	−14.4419	−7.2613	−19.8421	**−24.2083**	−11.1289
	STD	1.4046	1.261	3.3939	**0.70315**	1.7589	3.0315	1.2564
	Rank	4	5	3	**7**	2	**1**	6
F13	AVG	**6.6808e−05**	0.00011582	0.0021684	66.5832	5.6651	0.0011005	0.0041038
	STD	**6.9955e−05**	8.191e−05	0.0012567	31.0184	8.0751	0.00037803	0.004235
	Rank	**1**	2	4	7	6	3	5
Percentage		**1.384615385**	**2.230769231**	**4.615384615**	**6.846153846**	**5.230769231**	**3.384615385**	**4.230769231**
Total Rank		**1**	**2**	**5**	**7**	**6**	**3**	**4**

Best results are highlighted in bold

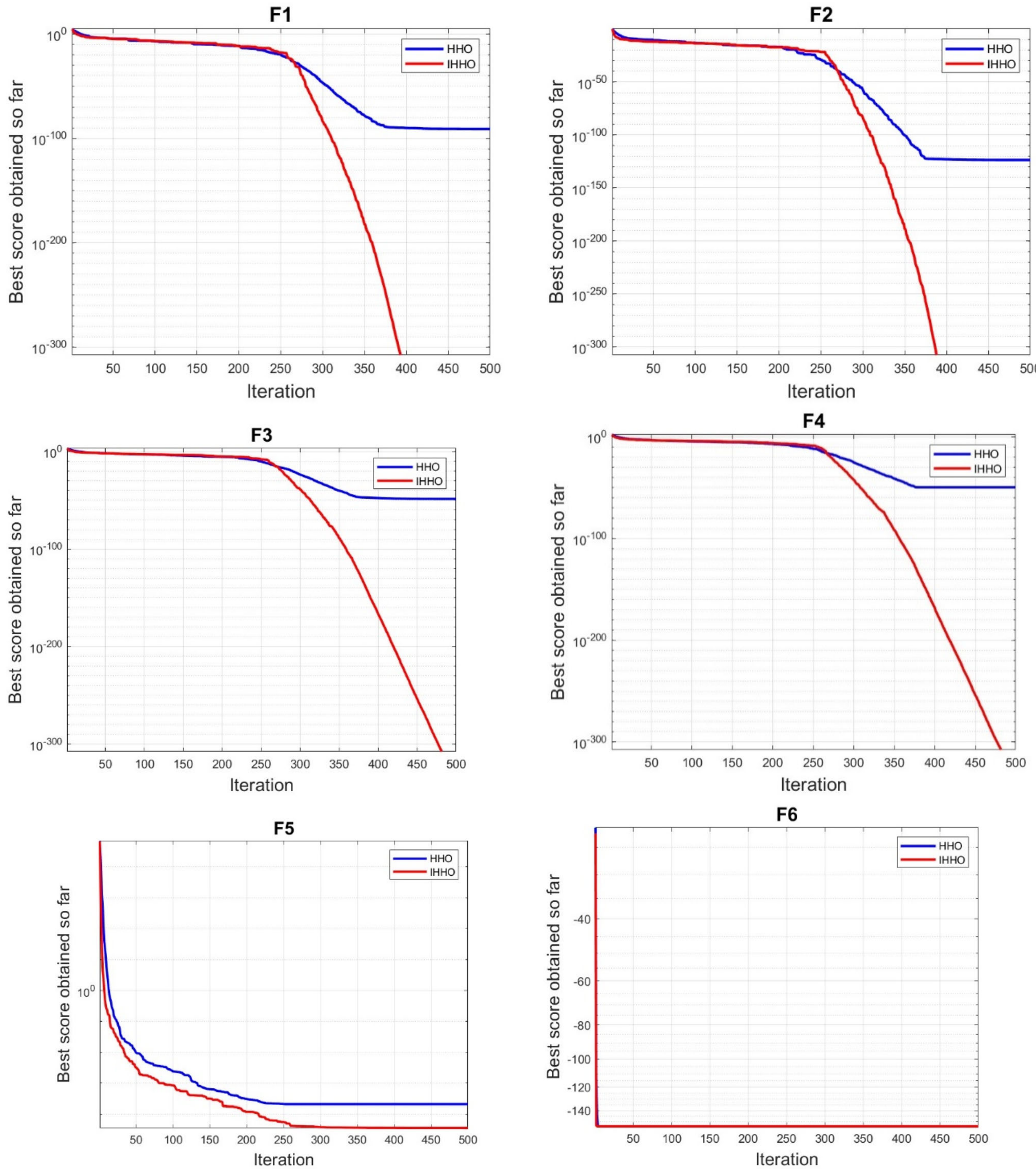


Fig. 23 Unimodal variable-dimension curves

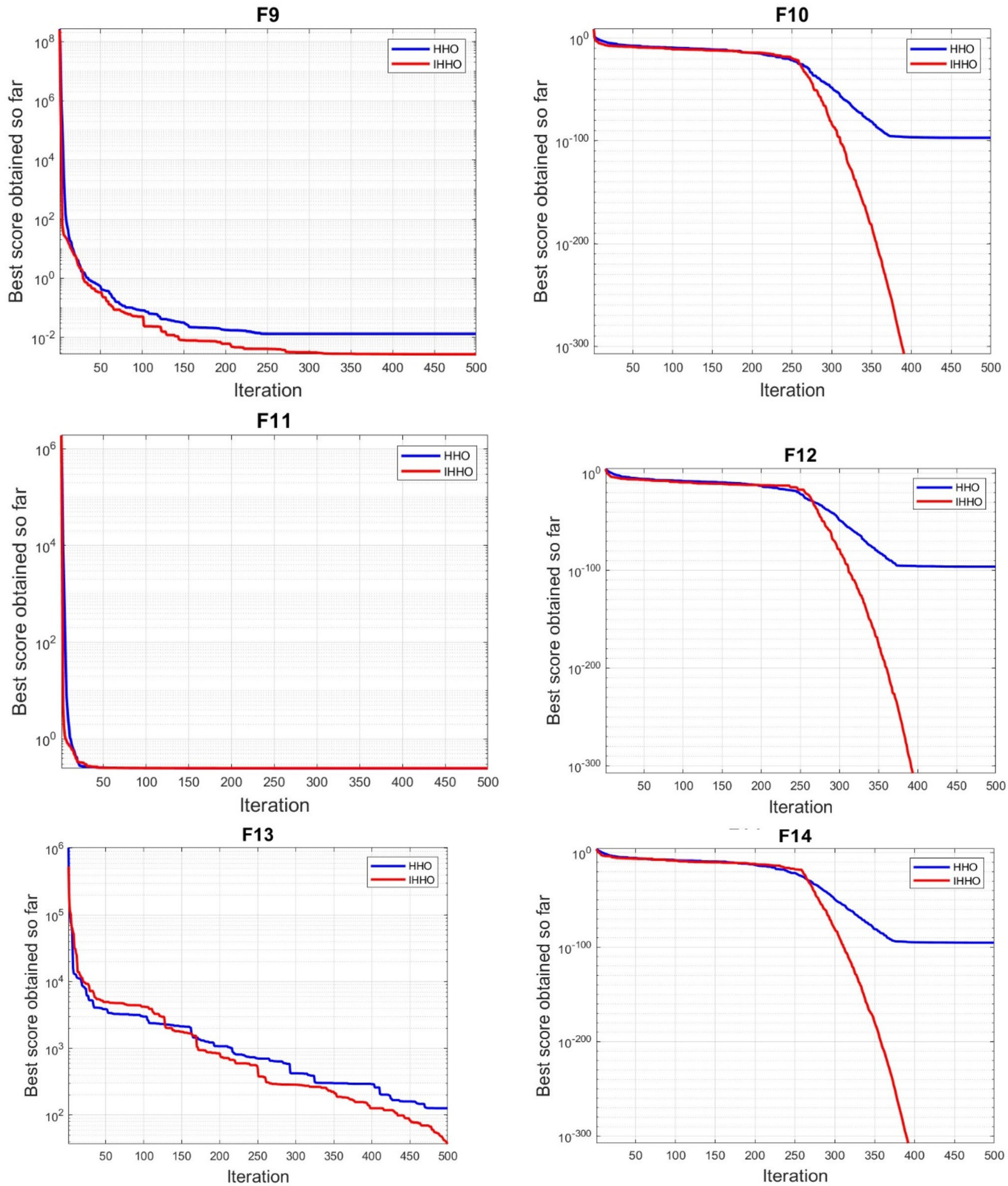


Fig. 23 continued

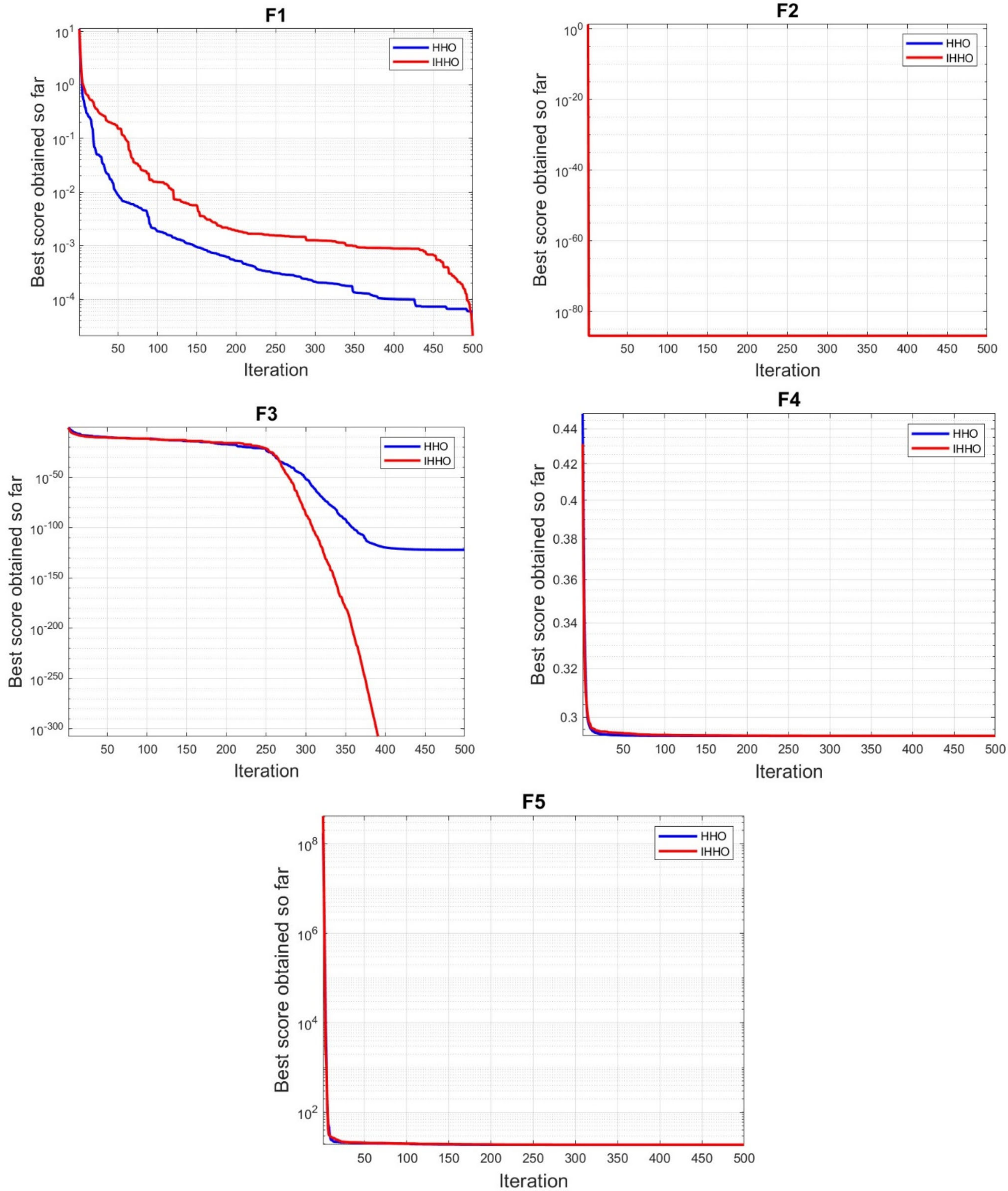


Fig. 24 Convergence curves for IHHO vs HHO on unimodal fixed-dimension curves

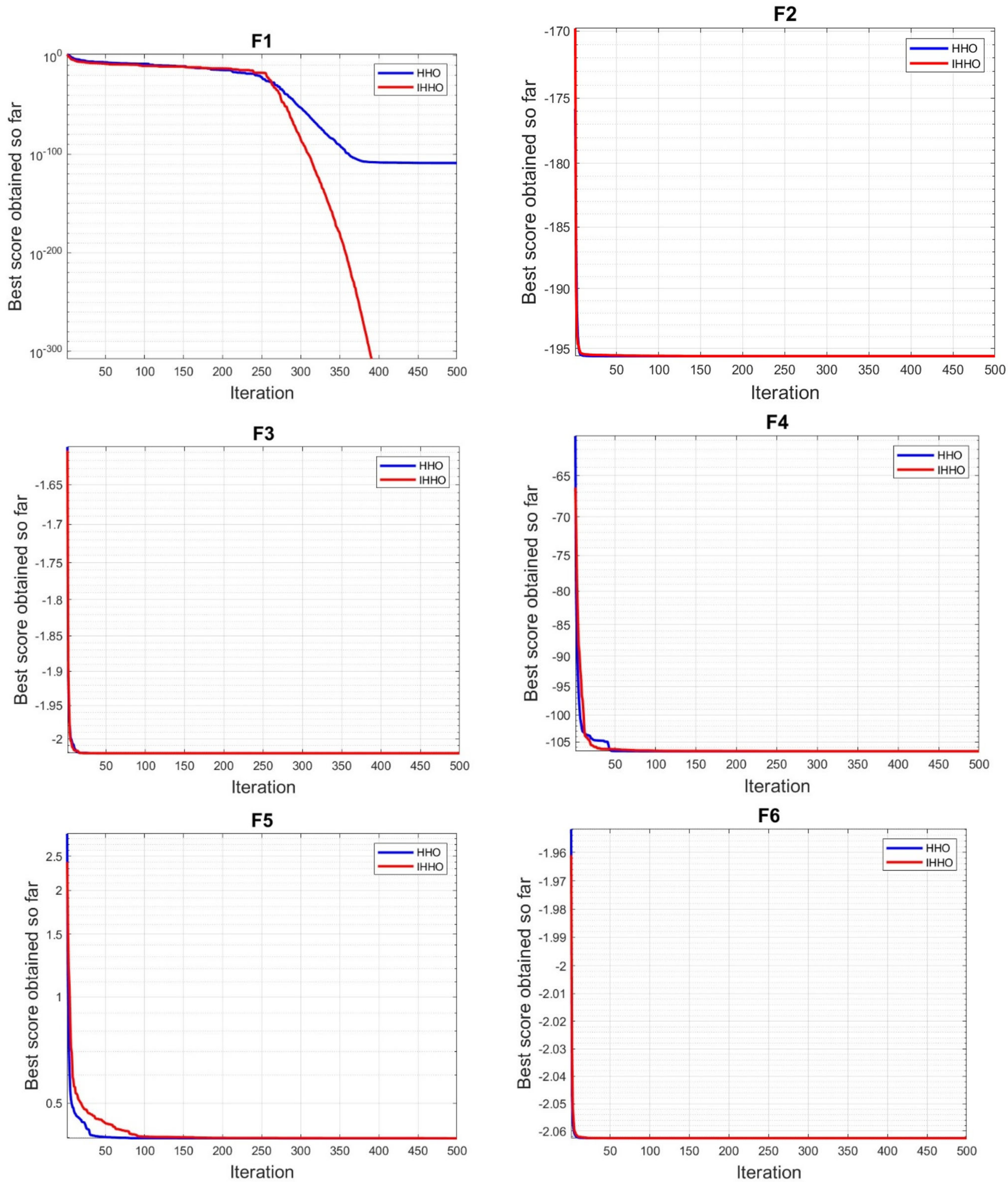


Fig. 25 Convergence curves for IHHO vs HHO on multimodal fixed-dimension curves

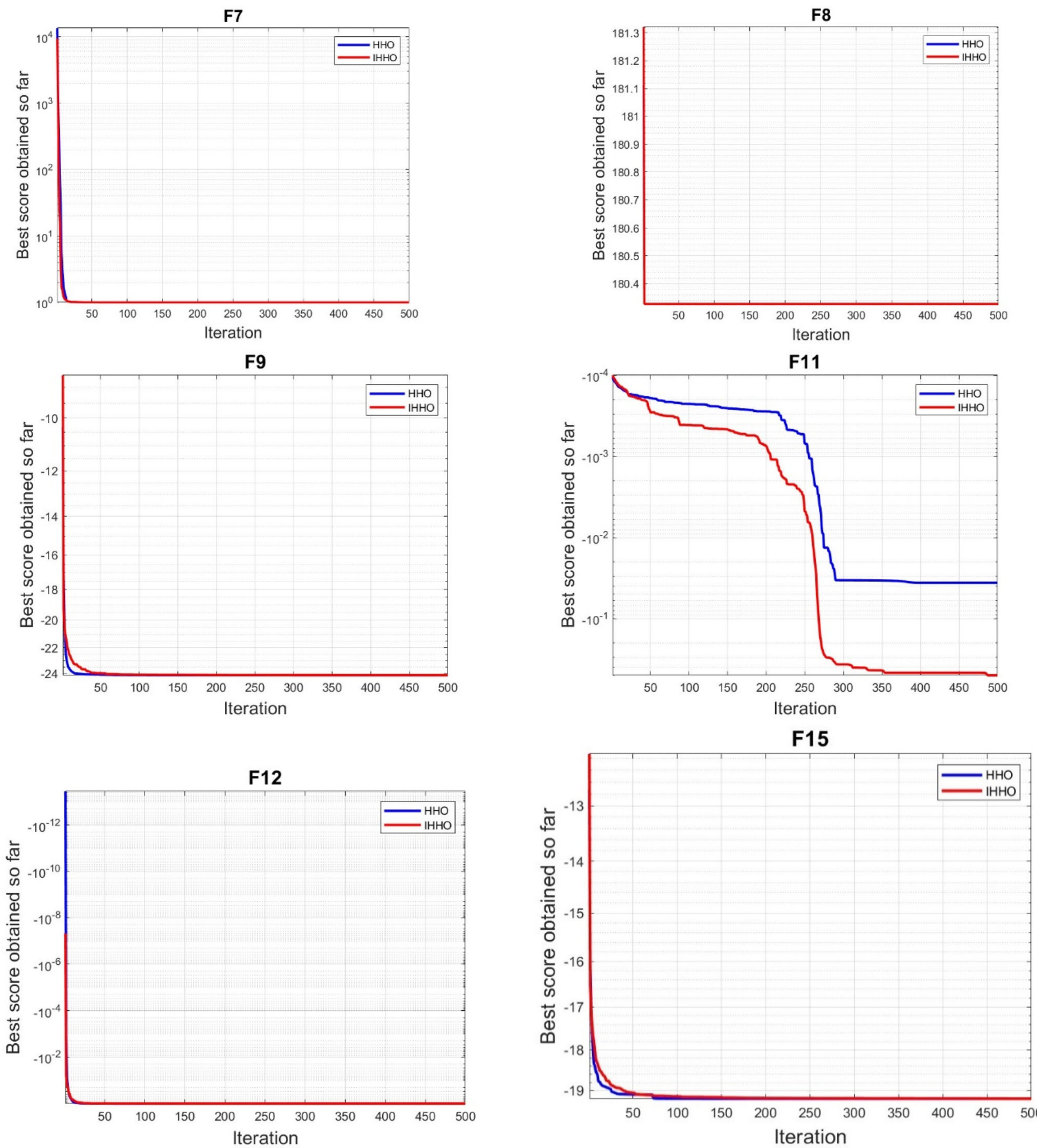


Fig. 25 continued

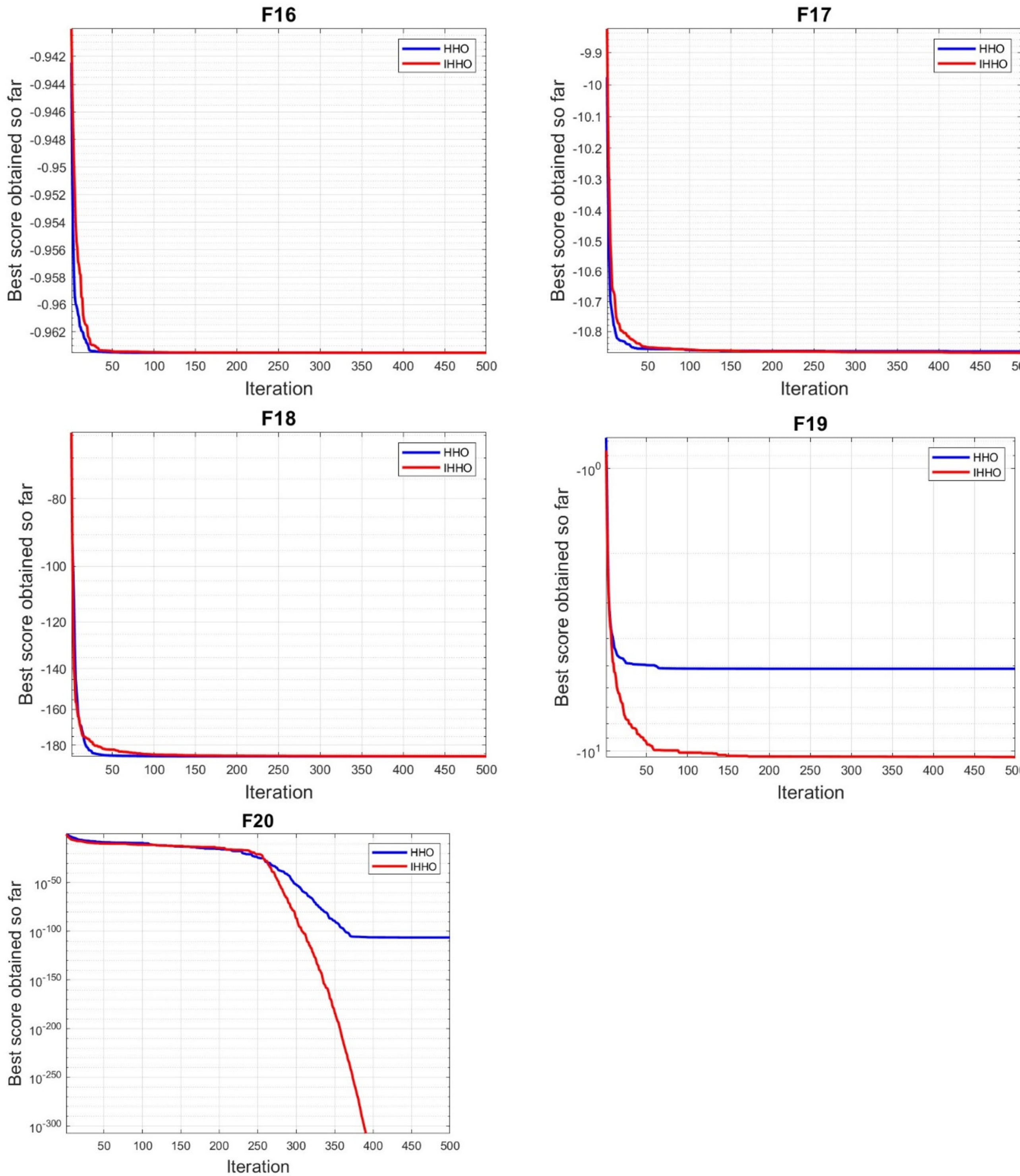


Fig. 25 continued

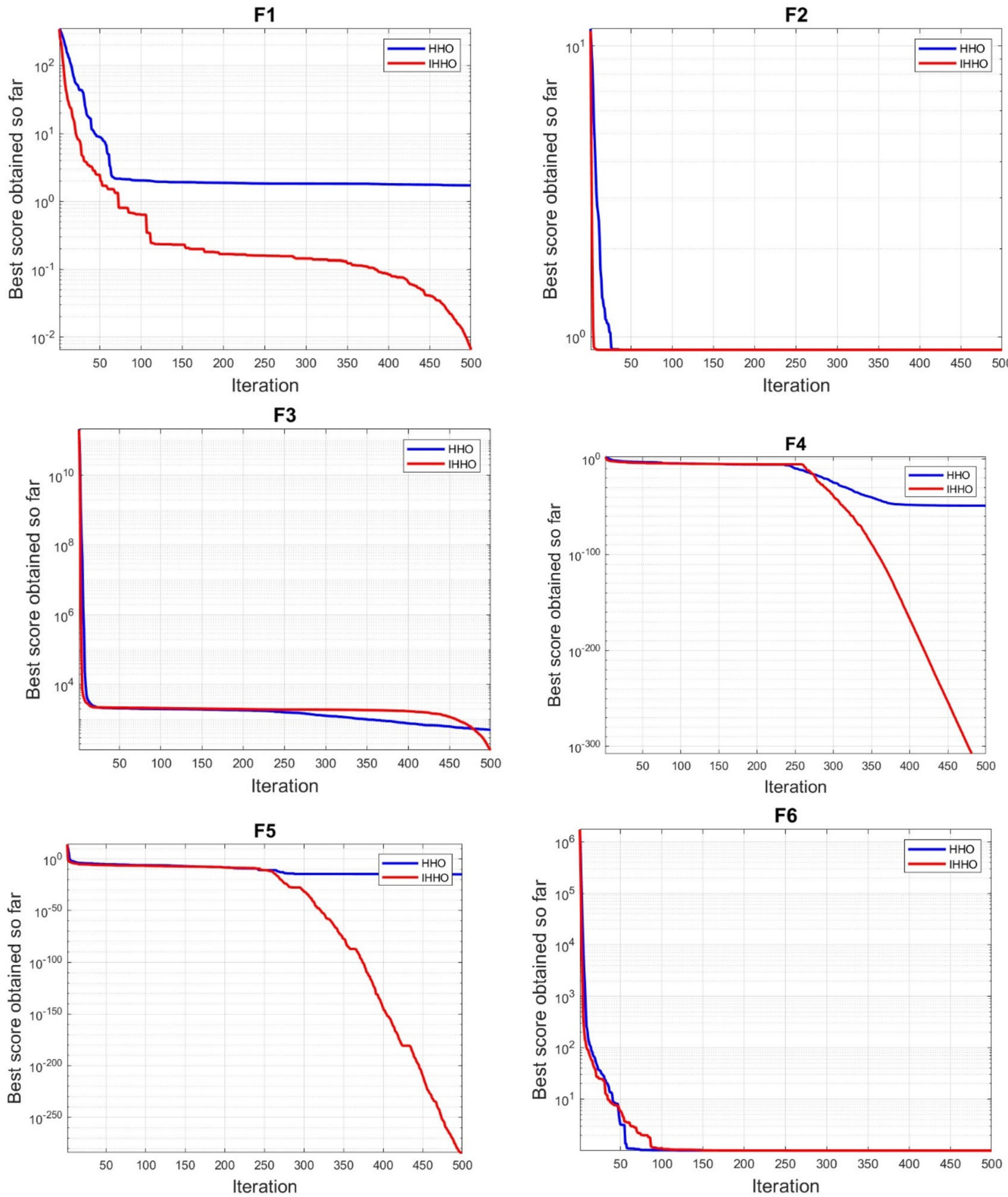


Fig. 26 Convergence curves for IHHO vs HHO on multimodal fixed-dimension curves

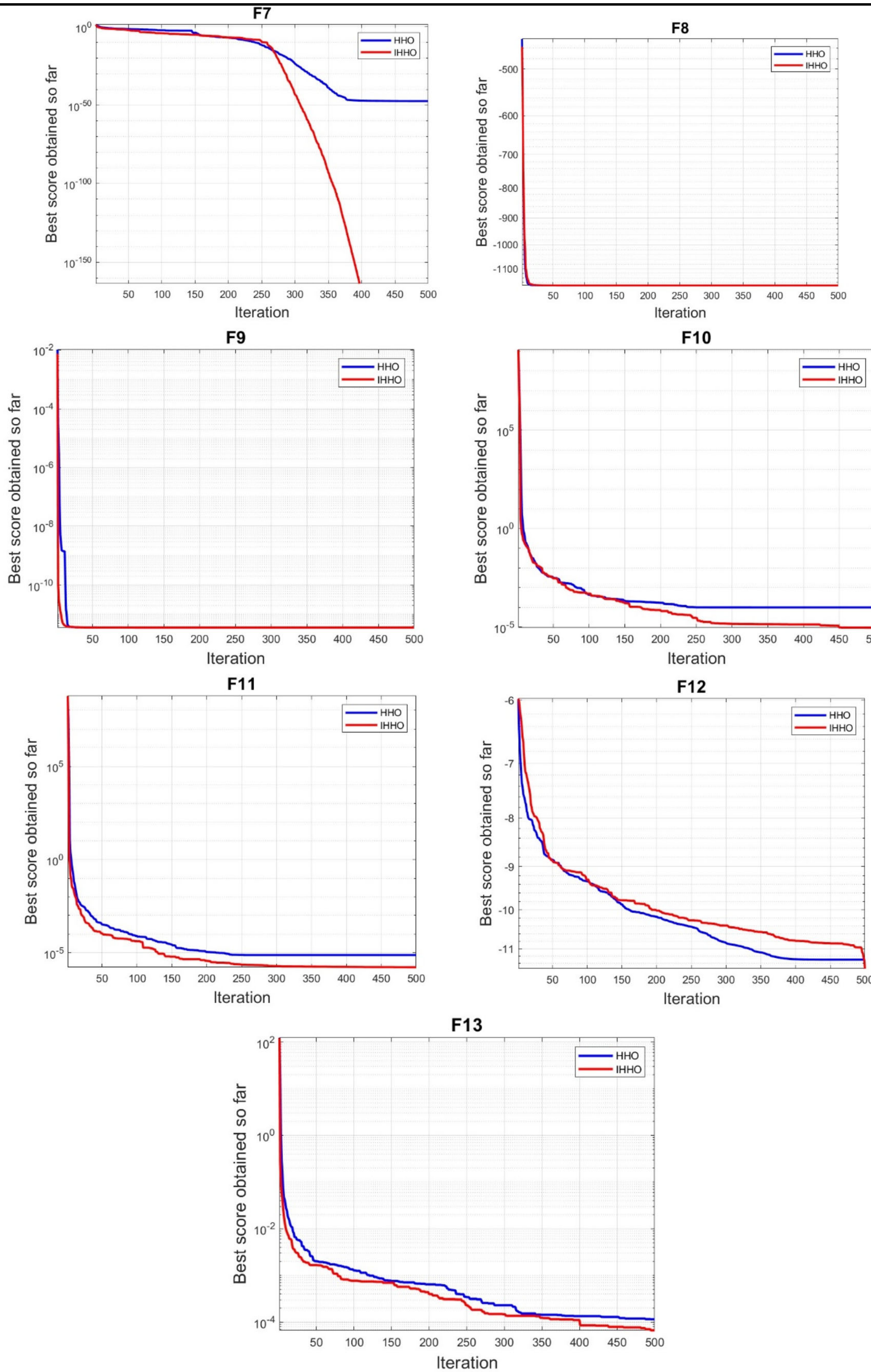


Fig. 26 continued

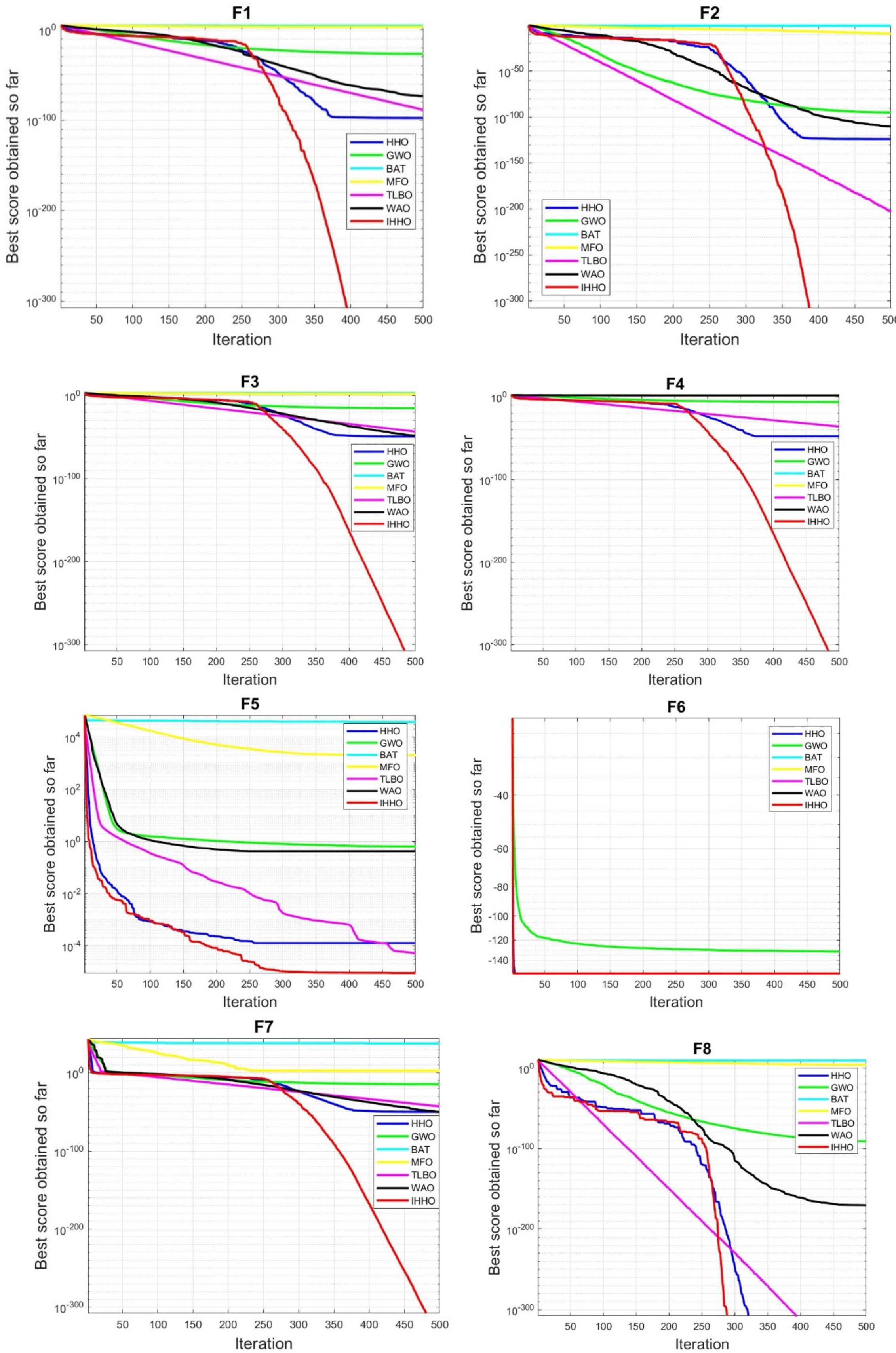


Fig. 27 Convergence curves for IHHO vs other algorithms on unimodal variable-dimension functions

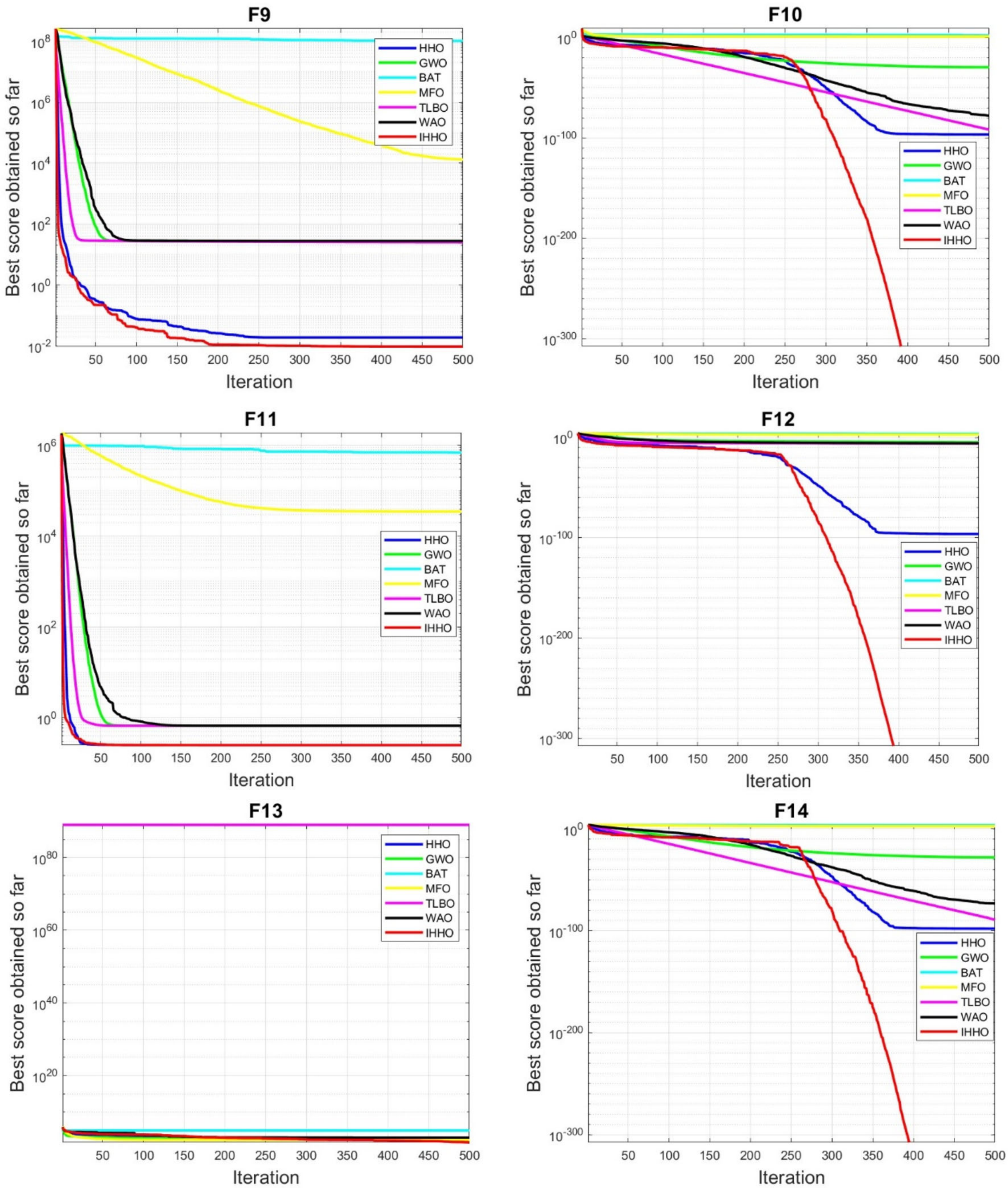


Fig. 27 continued

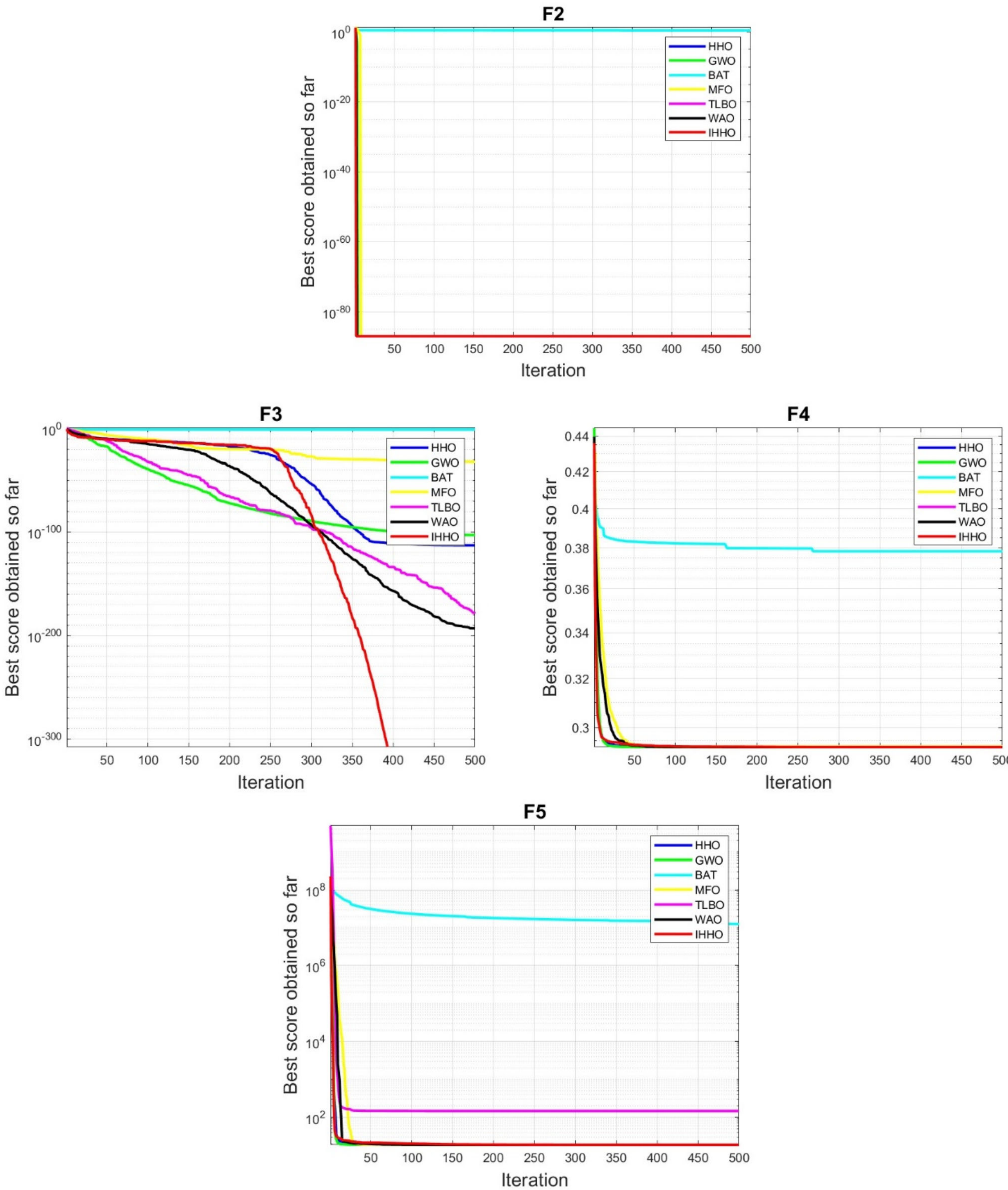


Fig. 28 Convergence curves for IHHO vs other algorithms on Unimodal fixed-dimension functions

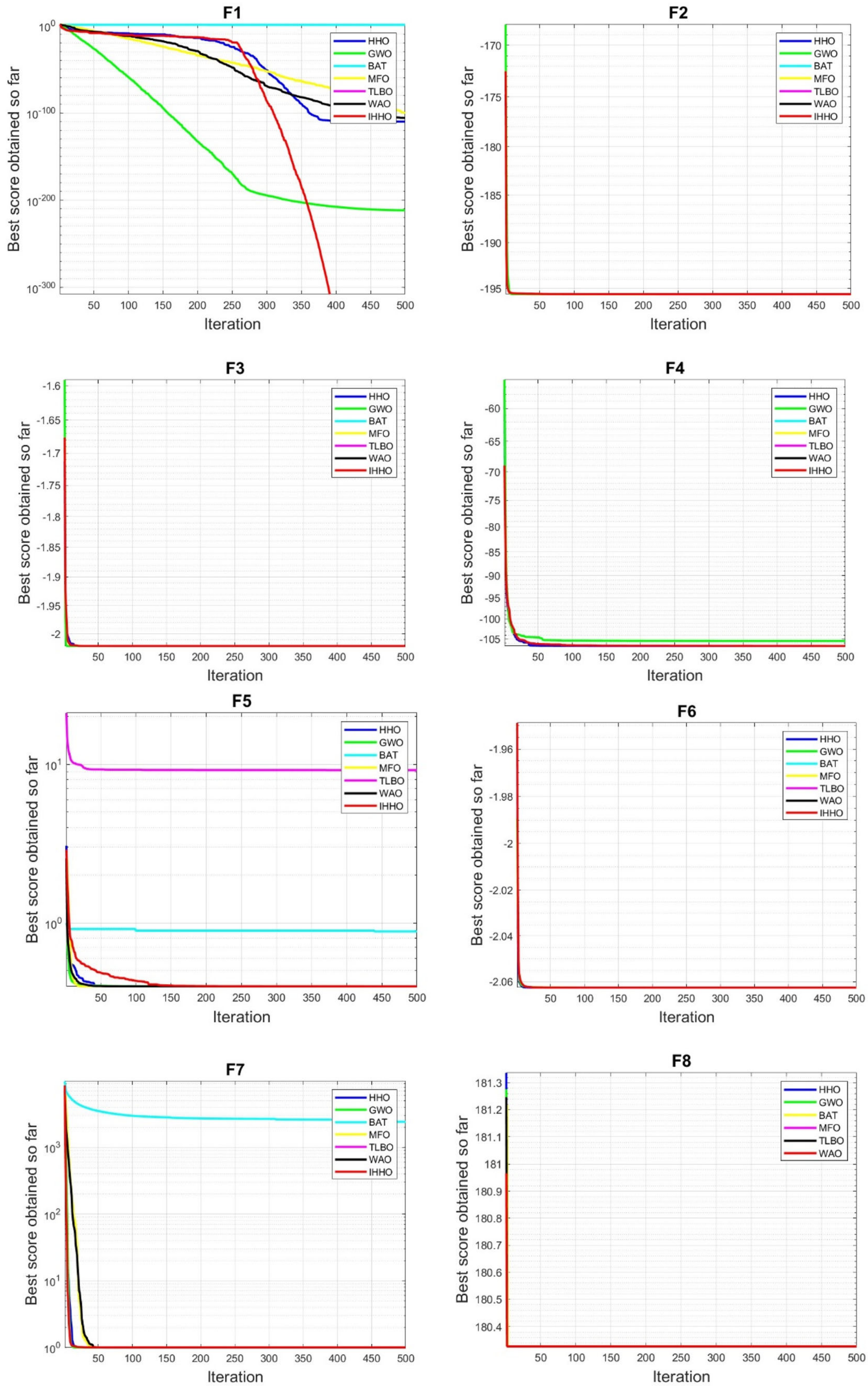


Fig. 29 Convergence curves for IHHO vs other algorithms on multimodal fixed-dimension functions

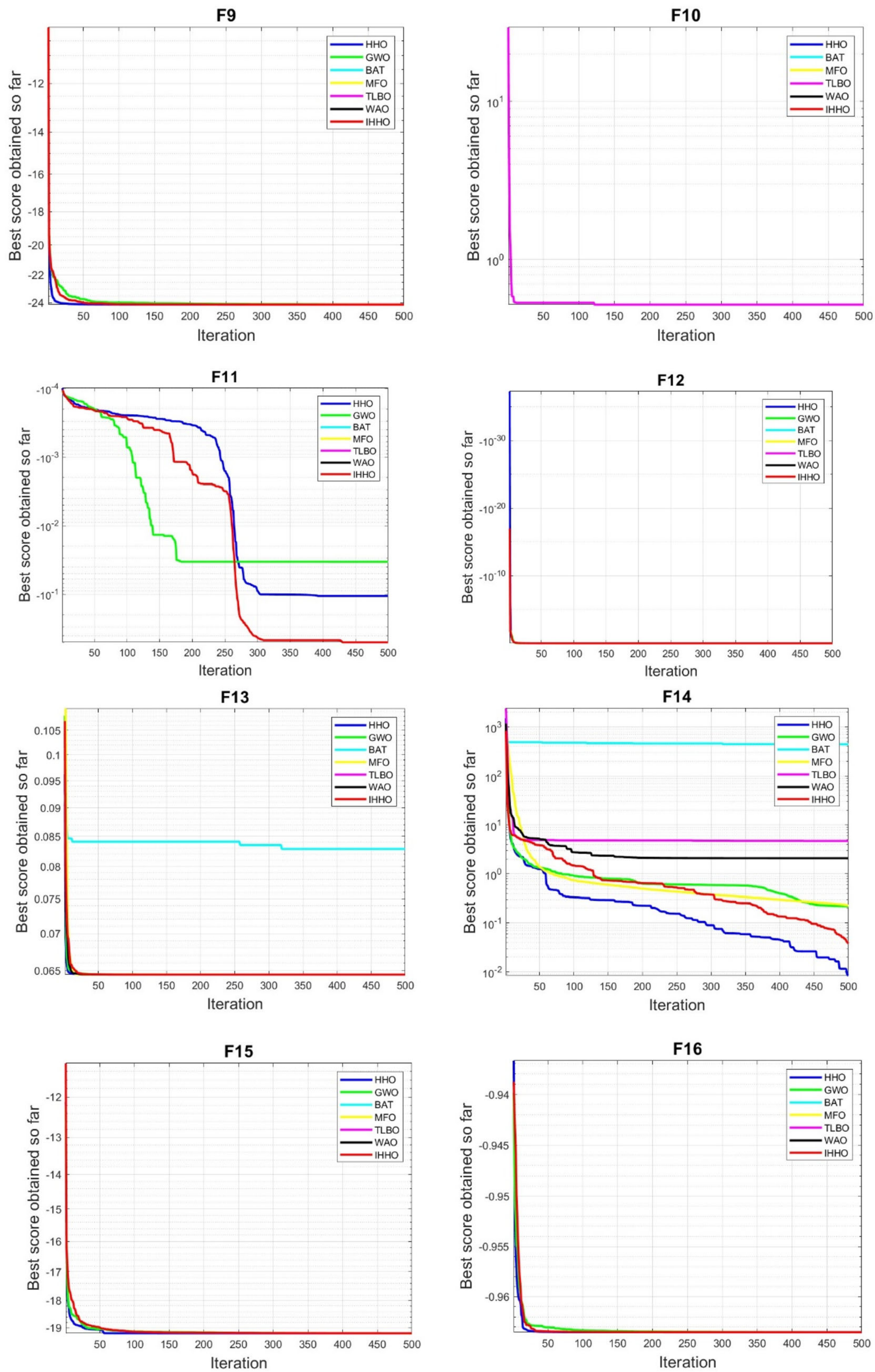


Fig. 29 continued

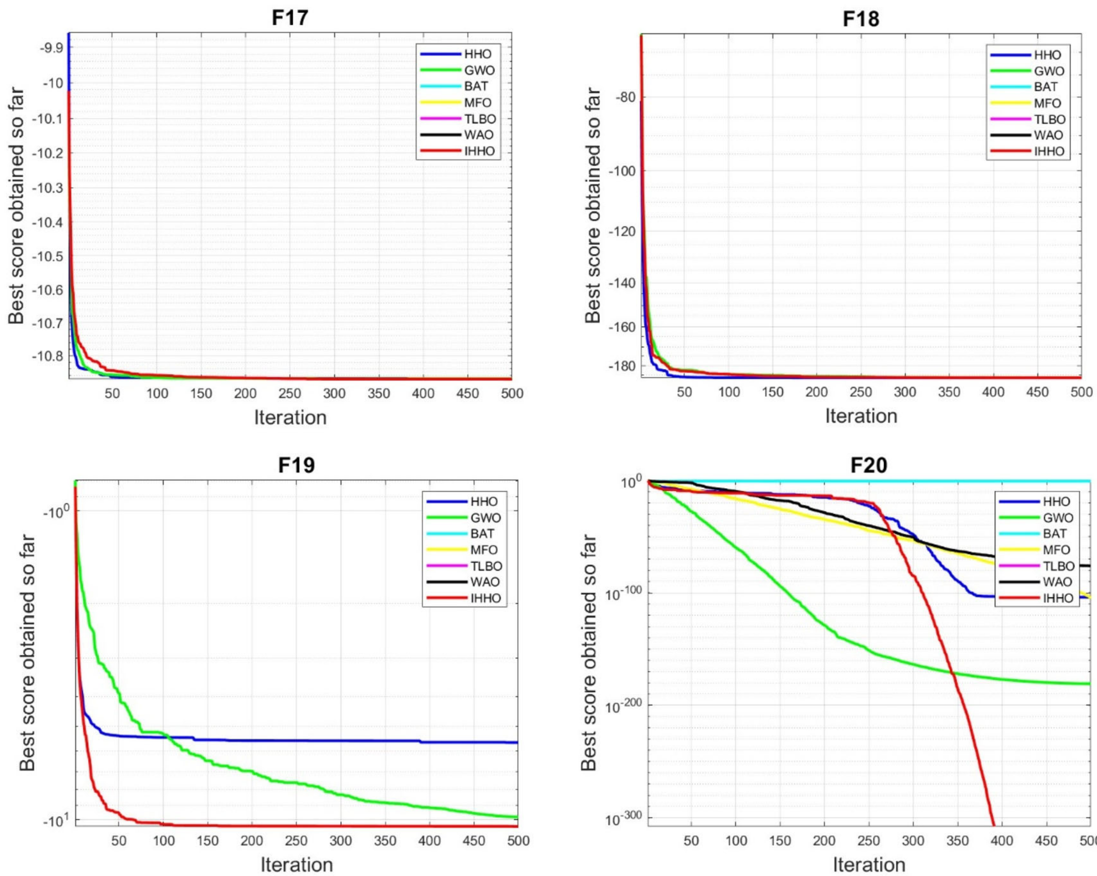


Fig. 29 continued

$$
\begin{aligned}
q_1(x) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \\
q_2(x) &= \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \\
q_3(x) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\
q_4(x) &= \frac{x_2 + x_1}{1.5} - 1 \leq 0
\end{aligned} \tag{26}
$$

5.6.2 Pressure vessel design

Figure 36 shows a pressure vessel created to reduce the entire cost of the pressure vessel's materials, form, and welding [73, 90]. The thickness of the shell ($T_s = x_1$), the depth of the head ($T_h = x_2$), the inner radius ($R = x_3$), and the height of a cylindrical section ($L = x_4$) are the four design parameters. Here is how the pressure vessel problem is formulated:

$$
\begin{aligned}
f_{(x)} = {} & 0.0624 x_1 x_3 x_4 + 1.7781 x_2 x_3^3 + 3.1661 x_1^2 x_4 \\
& + 19.84 x_1^3 x_3
\end{aligned} \tag{27}
$$

Subject to:

$$
\begin{aligned}
g_1(x) &= 0.0193 x_3 - x_1 \leq 0 \\
g_2(x) &= 0.00954 x_3 - x_2 \leq 0 \\
g_3(x) &= 1296000 - \Pi x_3^2 x_4 - \frac{4}{3} \Pi x_3^3 \leq 0 \\
g_4(x) &= x_4 - 240
\end{aligned} \tag{28}
$$

where

$$
\begin{aligned}
& x_1 \epsilon [0.0625, 6.187], x_2 \epsilon [0.0625, 6.187], x_3 \\
& \quad \epsilon [10, 200], x_4 \epsilon [10, 200]
\end{aligned}
$$

5.6.3 Welded beam design

The welded beam design (WBD) issue takes into account several design factors such as design for minimal cost while subjected to shear stress limits (τ), beams' end

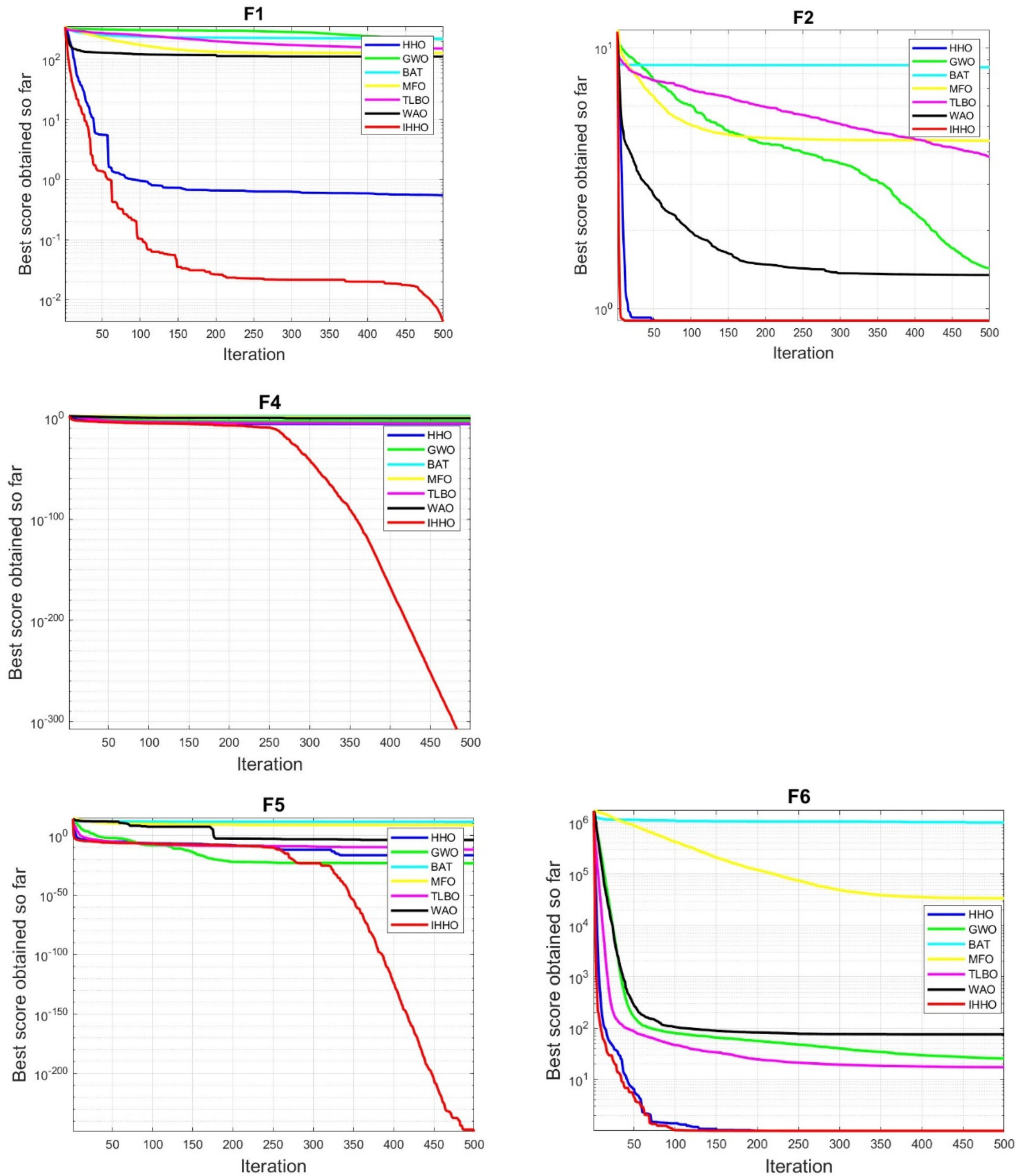


Fig. 30 Convergence curves for IHHO vs other algorithms on multimodal variable-dimension functions

deflection (δ), bending stress in the beam (θ), buckling load on the bar (P_c), and side constraints [91]. According to Fig. 37 WDB took into account four design factors in order to build a welded beam using the fewest possible resources. These parameters are $h(x)$, $i(x_2)$, $t(x_3)$, and $b(x_4)$. This is a mathematical representation of the WDB issue:

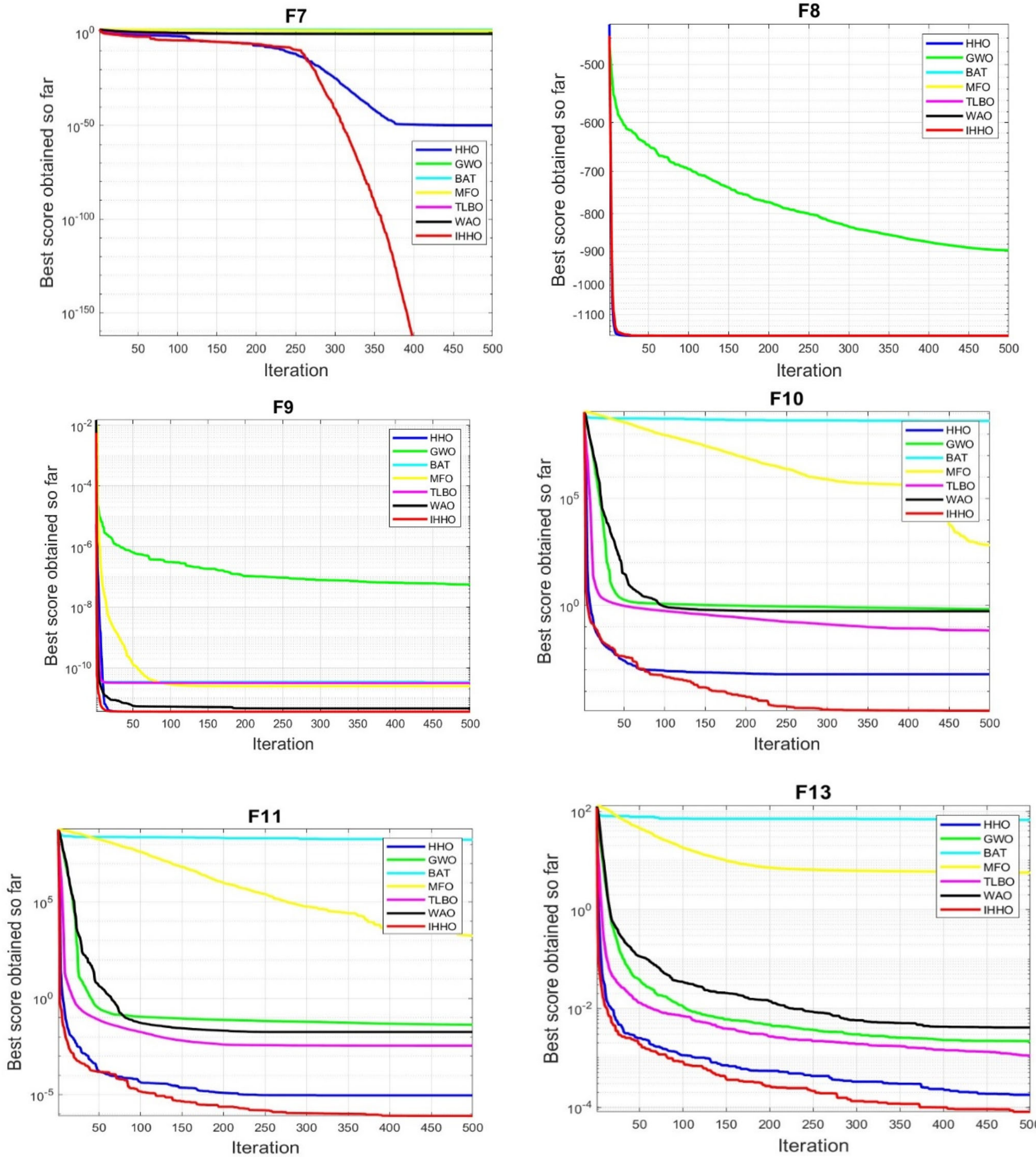


Fig. 30 continued

Table 26 Comparison of results of IHHO vs other modifications of HHO on 52 benchmark functions

Benchmark		IHHO	HHO	BHHO [36]	MHHO [37]	LogHHO [38]
Unimodal variable-dimension						
F1	AVG	**0**	1.3961e−91	6.116e−59	4.5193e−247	7.385e−17
	STD	**0**	7.6465e−91	2.0779e−58	**0**	1.8431e−16
	Rank	**1**	3	4	2	5
F2	AVG	**0**	2.2961e−124	2.3391e−84	**0**	5.7648e−17
	STD	**0**	7.8038e−124	1.0322e−83	**0**	1.9317e−16
	Rank	**1**	3	4	**1**	5
F3	AVG	**0**	1.6936e−49	1.3855e−29	2.7023e−123	4.5853e−25
	STD	**0**	8.9877e−49	6.8401e−29	1.4764e−122	2.4734e−24
	Rank	**1**	3	4	2	5
F4	AVG	**0**	2.1191e−50	2.4949e−31	6.1223e−125	16.2935
	STD	**0**	6.4085e−50	1.1005e−30	3.0248e−124	15.3757
	Rank	**1**	3	4	2	5
F5	AVG	**1.269e−05**	0.00020706	5.01e−05	0.29935	3.9107
	STD	**3.7067e−05**	0.00034121	9.1013e−05	0.28698	1.1734
	Rank	**1**	3	2	4	5
F6	AVG	**−155**	**−155**	**−155**	**−155**	**−155**
	STD	**0**	**0**	**0**	**0**	**0**
	Rank	**1**	**1**	**1**	**1**	**1**
F7	AVG	**0**	6.9426e−50	1.4102e−29	5.8625e−126	1.3336e−25
	STD	**0**	1.8065e−49	5.0969e−29	2.7735e−125	4.2254e−25
	Rank	**1**	3	4	2	5
F8	AVG	**0**	**0**	4.4574e−308	**0**	1.9821e−12
	STD	**0**	**0**	**0**	**0**	8.3108e−12
	Rank	**1**	**1**	**4**	**1**	5
F9	AVG	**0.002708**	0.013231	0.0049096	27.4986	28.8905
	STD	**0.0049833**	0.01834	0.0099083	0.52734	0.040288
	Rank	**1**	3	2	4	5
F10	AVG	**0**	1.3167e−98	7.1392e−61	8.2589e−251	4.4834e−21
	STD	**0**	4.9725e−98	3.9099e−60	**0**	1.0303e−20
	Rank	**1**	3	4	2	5
F11	AVG	0.24973	0.24971	**0.22309**	0.66669	0.99611
	STD	0.00088237	0.00070803	0.031211	**1.8082e−05**	0.006894
	Rank	3	2	1	4	5
F12	AVG	**0**	7.5986e−97	1.0199e−59	5.7654e−247	8.7899e−14
	STD	**0**	4.1565e−96	5.2251e−59	**0**	4.7891e−13
	Rank	**1**	3	4	2	5
F13	AVG	37.259	126.6738	**32.8555**	116.6899	254.5941
	STD	**29.048**	225.3297	57.2093	225.8021	398.7312
	Rank	2	4	**1**	3	5
F14	AVG	**0**	4.8407e−96	4.7614e−60	2.8795e−246	8.1577e−19
	STD	**0**	2.5151e−95	2.593e−59	**0**	2.6108e−18
	Rank	**1**	3	4	2	5

Table 26 (continued)

Benchmark		IHHO	HHO	BHHO [36]	MHHO [37]	LogHHO [38]
Percentage		**1.214285714**	**2.714285714**	**3.071428571**	**2.285714286**	**4.714285714**
Total Rank		**1**	**3**	**4**	**2**	**5**
Unimodal fixed-dimension						
F1	AVG	2.0815e−05	5.9535e−05	**6.5393e−08**	3.7561e−06	8.523e−05
	STD	2.3061e−05	8.1805e−05	**2.1566e−07**	4.7767e−06	0.00011752
	Rank	3	4	**1**	2	5
F2	AVG	**1.3839e−87**	**1.3839e−87**	**1.3839e−87**	**1.3839e−87**	**1.3839e−87**
	STD	**6.8117e−103**	**6.8117e−103**	**6.8117e−103**	**6.8117e−103**	**6.8117e−103**
	Rank	**1**	**1**	**1**	**1**	**1**
F3	AVG	**0**	1.0717e−122	3.3321e−78	1.2509e−314	1.4344e−95
	STD	**0**	5.8701e−122	1.8251e−77	**0**	7.0465e−95
	Rank	**1**	2	4	5	3
F4	AVG	**0.29258**	**0.29258**	**0.29258**	**0.29258**	**0.29258**
	STD	1.2316e−06	5.9415e−07	5.3408e−07	**1.2761e−07**	1.274e−06
	Rank	**1**	**1**	**1**	**1**	**1**
F5	AVG	**19.106**	19.1062	19.1059	19.1059	19.1064
	STD	8.9581e−05	0.00060229	**1.7712e−06**	2.4124e−05	0.00085601
	Rank	**1**	4	**1**	**1**	5
Percentage		**1.4**	2.4	1.6	2	3
Total Rank		**1**	4	2	3	5
Multimodal fixed-dimension						
F1	AVG	**0**	1.2975e−109	2.3129e−73	5.222e−298	2.2797e−20
	STD	**0**	6.8953e−109	1.2668e−72	**0**	8.3495e−20
	Rank	**1**	3	4	2	5
F2	AVG	**−195.629**	**−195.629**	**−195.629**	**−195.629**	**−195.629**
	STD	1.5244e−05	7.4381e−07	**2.6573e−10**	1.011e−09	8.1302e−05
	Rank	**1**	**1**	1	**1**	**1**
F3	AVG	**−2.0218**	**−2.0218**	**−2.0218**	**−2.0218**	**−2.0218**
	STD	1.9896e−07	**9.9301e−16**	1.1662e−15	**9.2199e−16**	6.4964e−08
	Rank	**1**	**1**	**1**	**1**	**1**
F4	AVG	**−106.7634**	**−106.7645**	**−106.7645**	**−106.7645**	−106.7603
	STD	0.0013021	1.3424e−05	**5.4481e−09**	1.2277e−07	0.0057088
	Rank	**1**	**1**	**1**	**1**	5
F5	AVG	**0.3979**	**0.39789**	**0.39789**	**0.39789**	0.39797
	STD	1.4258e−05	3.9394e−06	**1.4848e−09**	8.8691e−09	0.00012145
	Rank	4	**1**	**1**	**1**	5
F6	AVG	**−2.0626**	**−2.0626**	**−2.0626**	**−2.0626**	**−2.0626**
	STD	2.6641e−07	2.5808e−08	**2.3052e−12**	1.0581e−11	2.4307e−06
	Rank	**1**	**1**	**1**	1	**1**
F7	AVG	**1**	**1**	**1**	**1**	**1**
	STD	**0**	**0**	**0**	**0**	1.9097e−15
	Rank	**1**	**1**	**1**	**1**	**1**
F8	AVG	**180.3276**	**180.3276**	**180.3276**	**180.3276**	**180.3276**
	STD	**0**	**0**	**0**	**0**	**0**
	Rank	**1**	**1**	**1**	**1**	**1**
F9	AVG	**−24.1568**	**−24.1568**	**−24.1568**	**−24.1568**	−24.1567
	STD	0.00016531	1.4263e−06	3.4259e−09	2.2034e−07	0.0002296
	Rank	**1**	**1**	**1**	**1**	5

Table 26 (continued)

Benchmark		IHHO	HHO	BHHO [36]	MHHO [37]	LogHHO [38]
F10	AVG	**−42.7648**	−42.7355	−42.8697	−42.7804	−42.6015
	STD	**0.2223**	0.22672	0.16945	0.21916	0.19227
	Rank	**1**	4	3	2	5
F11	AVG	**−0.50015**	−0.036205	−0.070731	−0.006106	−0.00043546
	STD	0.49812	0.30403	0.25261	0.0095452	**0.00043551**
	Rank	**1**	3	2	4	5
F12	AVG	−0.99999	−0.99998	**−1**	**−1**	−0.76653
	STD	9.0857e−06	2.1463e−05	2.4617e−10	**7.2562e−10**	0.43011
	Rank	3	4	**1**	**1**	5
F13	AVG	**0.06447**	0.064471	**0.06447**	**0.06447**	0.064482
	STD	1.8395e−07	3.4941e−07	**3.9337e−11**	**4.4321e−11**	1.7445e−05
	Rank	**1**	4	**1**	**1**	5
F14	AVG	0.056121	0.02485	**0.0064274**	0.01096	0.064131
	STD	0.079838	0.051972	**0.022298**	0.019587	0.10576
	Rank	4	3	**1**	2	5
F15	AVG	**−19.2084**	**−19.2085**	**−19.2085**	**−19.2085**	**−19.2085**
	STD	0.00013123	**3.2574e−12**	1.2288e−10	2.2808e−13	1.7423e−05
	Rank	**1**	**1**	**1**	**1**	**1**
F16	AVG	**−0.96353**	**−0.96353**	**−0.96353**	**−0.96353**	**−0.96353**
	STD	1.6112e−07	3.1589e−09	6.6609e−11	**6.8895e−11**	1.1735e−06
	Rank	**1**	**1**	**1**	**1**	**1**
F17	AVG	**−10.8716**	−10.867	−10.8703	−10.8657	−10.8405
	STD	**0.0036257**	0.008909	0.0060438	0.0094966	0.048889
	Rank	**1**	3	2	4	5
F18	AVG	**−186.7197**	**−186.7303**	**−186.7309**	**−186.7309**	−186.7084
	STD	0.014362	0.0022392	**7.7956e−08**	6.5409e−07	0.056534
	Rank	4	3	**1**	**1**	5
F19	AVG	**−10.529**	−5.1252	−10.5164	−8.3248	−5.1853
	STD	**0.012416**	1.5814	7.3186e−05	2.7536	3.1008
	Rank	**1**	5	2	3	4
F20	AVG	**0**	4.8345e−107	1.1994e−77	5.2277e−283	9.3917e−12
	STD	**0**	2.6475e−106	5.9739e−77	**0**	5.0768e−11
	Rank	**1**	3	4	2	5
Percentage		**1.39891**	**2.09891**	**1.39891**	**1.44891**	**3.39891**
Total Rank		1	4	1	3	5
Multimodal variable-dimension						
F1	AVG	**0.0051495**	1.722	0.7474	29.878	226.5854
	STD	**0.0049582**	9.2336	2.0535	51.2718	28.0164
	Rank	**1**	3	2	4	5
F2	AVG	**0.9**	**0.9**	**0.9**	**0.9**	4.7633
	STD	**4.5168e−16**	**4.5168e−16**	**4.5168e−16**	**4.5168e−16**	0.7702
	Rank	**1**	**1**	**1**	**1**	5
F3	AVG	**132.2224**	501.6447	232.4891	330.992	557.4169
	STD	**35.2382**	113.7039	106.0033	133.9866	142.2815
	Rank	**1**	4	2	3	5
F4	AVG	**0**	1.0634e−49	6.2261e−30	5.282e−131	0.52013
	STD	**0**	5.8038e−49	2.509e−29	2.742e−130	2.8468
	Rank	**1**	3	4	2	5

Table 26 (continued)

Benchmark		IHHO	HHO	BHHO [36]	MHHO [37]	LogHHO [38]
F5	AVG	**1.2369e−284**	1.2345e−15	5.2732e−19	6.177e−36	1837.7999
	STD	**0**	6.5949e−15	2.4521e−18	3.3833e−35	9636.4365
	Rank	**1**	4	3	2	5
F6	AVG	**1.0003**	1.0008	1.0009	37.4598	107.0851
	STD	**0.00058655**	0.0008597	0.0015977	15.2751	28.6798
	Rank	**1**	2	3	4	5
F7	AVG	**0**	5.4413e−48	6.7188e−31	4.2404e−122	0.32691
	STD	**0**	1.8677e−47	2.2332e−30	1.9292e−121	0.14171
	Rank	**1**	3	4	2	5
F8	AVG	**−1174.9747**	**−1174.978**	−1174.9837	−1145.8295	−890.3591
	STD	0.014211	0.0093998	**0.0017942**	31.8021	50.3703
	Rank	3	2	**1**	4	5
F9	AVG	**3.5134e−12**	3.5146e−12 1.7099e−15	3.5381e−12	5.3066e−12	4.7891e−08
	STD	**1.3473e−15**	2	2.0495e−14	2.9595e−12	9.0099e−08
	Rank	**1**		3	4	5
F10	AVG	**9.3135e−06**	9.9379e−05	1.5115e−05	0.24715	1.511
	STD	**2.3563e−05**	0.00013036	2.3959e−05	0.16008	0.60349
	Rank	**1**	3	2	4	5
F11	AVG	**1.6483e−06**	7.4312e−06	3.0467e−06	0.0090666	1.0432
	STD	**3.3941e−06**	1.2137e−05	4.654e−06	0.0054853	2.4001
	Rank	**1**	3	2	4	5
F12	AVG	−11.5431	−11.2914	**−15.3568**	−14.2167	−10.8524
	STD	1.4046	**1.261**	1.8088	1.5774	1.6254
	Rank	3	4	**1**	2	5
F13	AVG	**6.6808e−05**	0.00011582	0.00017865	0.00023069	0.05861
	STD	**6.9955e−05**	8.191e−05	0.00018167	0.00021186	0.070564
	Rank	**1**	2	3	4	5
Percentage		**1.307692308**	**2.769230769**	**2.384615385**	**3.076923077**	**5**
Total Rank		1	3	2	4	5

Best results are highlighted in bold

$$\text{Min } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 - x_2) \quad (29)$$

Subject to:

$$\begin{aligned} g_1(x) &= \tau(x) - 13000 \le 0 \\ g_2(x) &= \delta(x) - 30000 \le 0 \\ g_3(x) &= x_1 - x_4 \le 0 \\ g_4(x) = 0.1047x_1^2 &+ 0.04811x_3x_4(14.0 + x_2) - 5 \le 0 \\ g_5(x) &= 0.125 - x_1 \le 0 \\ x_6(x) &= \delta(x) - 0.25 \le 0 \\ g_7(x) &= 6000 - P_c(x) \le 0 \end{aligned} \quad (30)$$

where

$$\tau(x) = \sqrt{(\tau')_2 + 2\tau'\tau''\frac{x^2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{6000}{\sqrt{2}x_1x_2},$$

$$\tau'' = \frac{MR}{J}$$

$$M = 6000\left(14 + \frac{x_2}{4}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2\right.$$

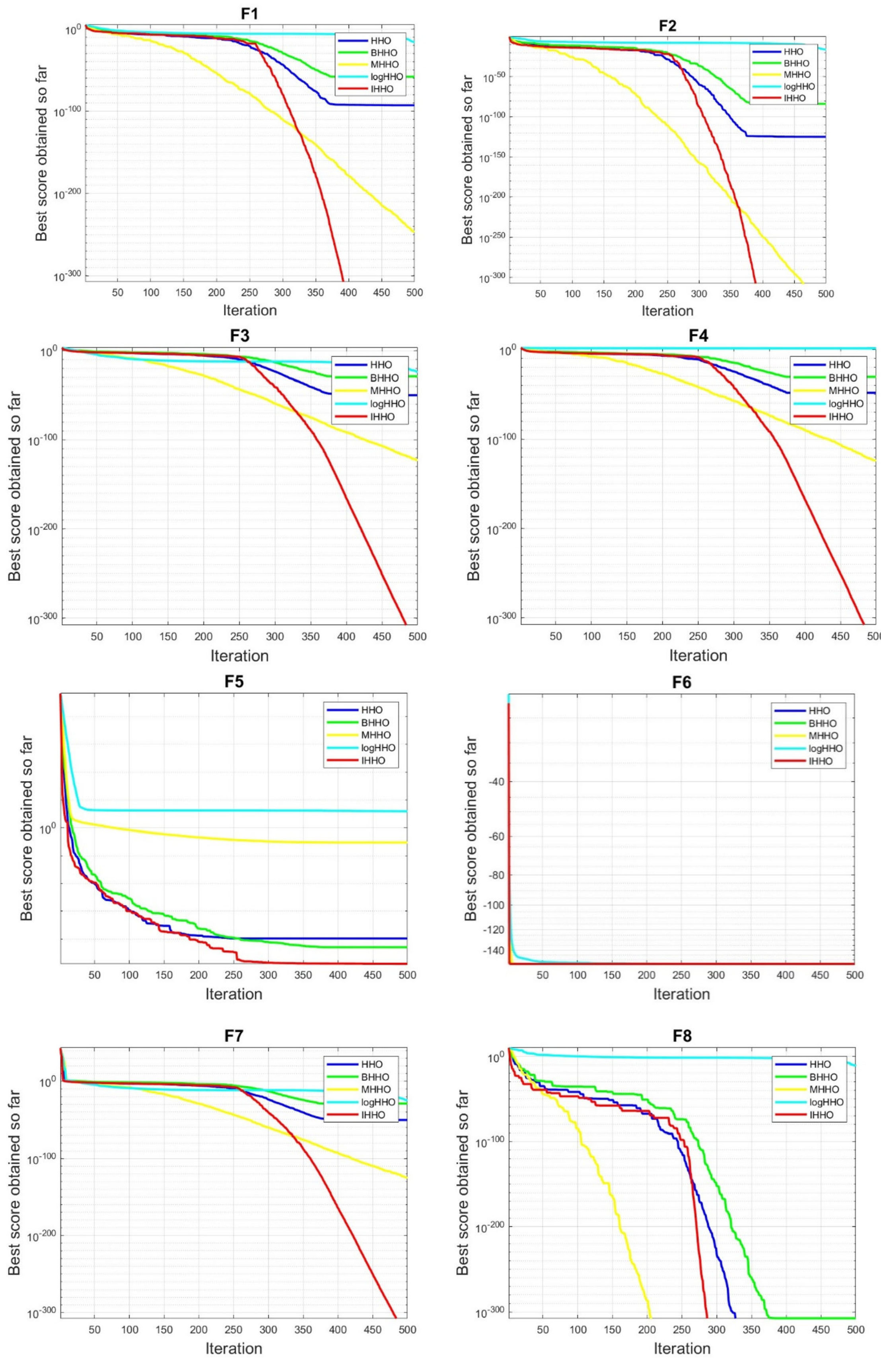


Fig. 31 Convergence curves for IHHO vs HHO modifications on unimodal variable-dimension functions

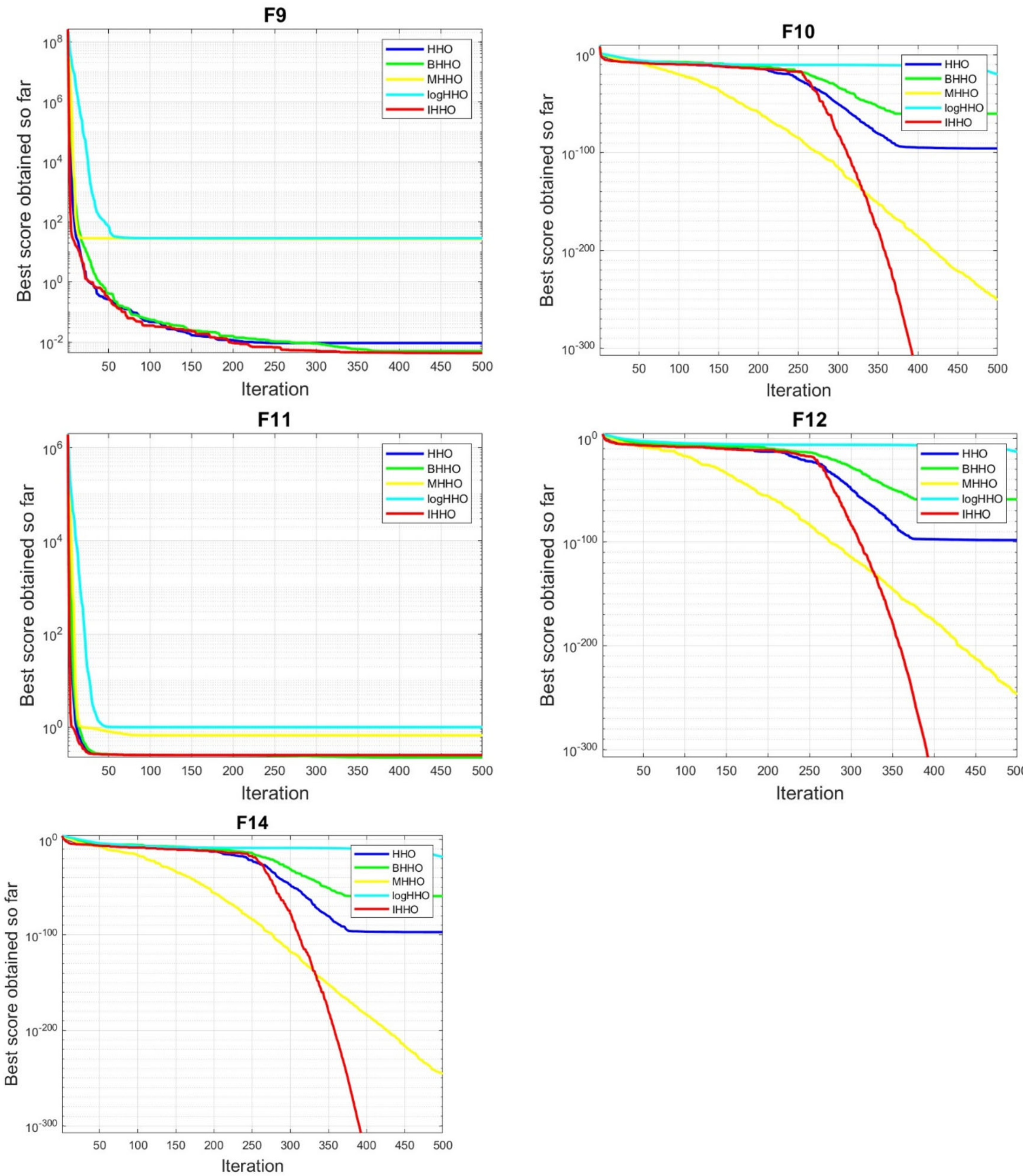


Fig. 30 continued

$$\gamma(\mathrm{x}) = \frac{504000}{x_4 x_3^2}$$

$$\delta(\mathrm{x}) = \frac{2.1952}{x_4 x_3^3}$$

$$Pc(x) = 64746.022(1 - 0.0282346X_3)X_3X_4^3$$

5.6.4 Gear train design problem

The challenge of designing gears with the optimal ratio comprises five different design variables, a nonlinear objective function, and five different nonlinear limitations [92]. The goal of this exercise is to find the gear ratio for the gear train depicted in Fig. 38 with the lowest possible cost [93]. To specify the gear ratio, we say:

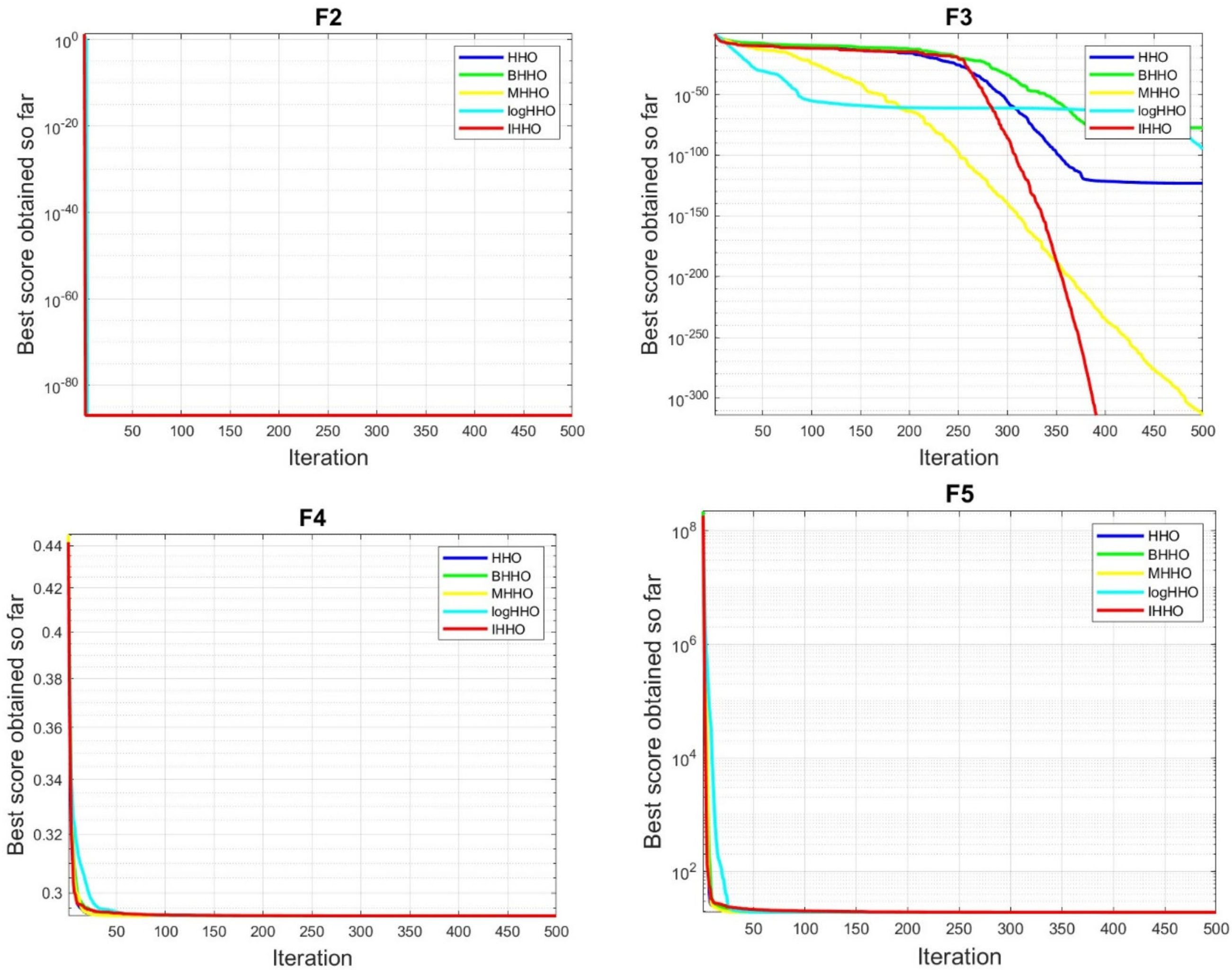


Fig. 32 Convergence curves for IHHO vs HHO modifications on unimodal fixed-dimension functions

$$Gearratio = \frac{\eta_B \eta_D}{\eta_F \eta_A} \tag{31}$$

$$\text{Min}\, f(x) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2 \tag{32}$$

Subject to:

$12 \le x_i \le 60 \quad wherei = 1, \ldots., 4$

5.6.5 Three-bar truss design

Ray and Saini [94] were the first to identify the optimization difficulty inherent in the design of a three-bar truss. As seen in Fig. 39, three bars are preferred in light of this. The goal is to reduce the total bar weight by placing them in this orientation [77]. This issue contains three constrained functions and two design parameters (x_1, x_2). A mathematical formulation of the issue is as follows:

$$\text{Min}\, f(x) = \left(2\sqrt{2}x_1 + x_2\right) xL \tag{33}$$

$$\begin{aligned} g_1 &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0 \\ g_2 &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0 \\ g_3 &= \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \le 0 \end{aligned} \tag{34}$$

where:$x_1 \ge 0, x_2 \le 1$, the constant are $L = 100cm$, $P = 2KN/cm^2$

And $\sigma = 2KN/\text{cm}^2$ [1, 2, 6].

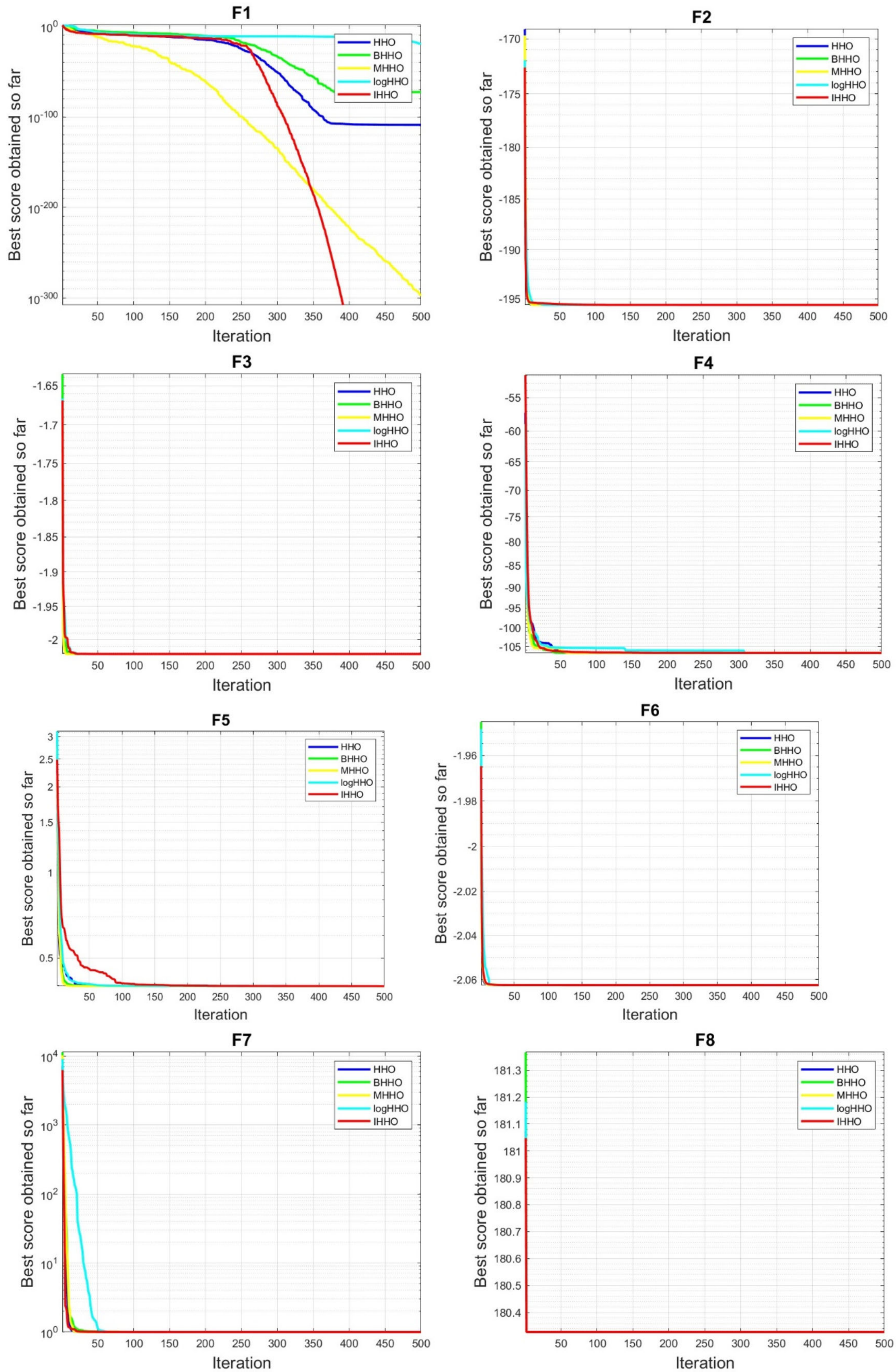


Fig. 33 Convergence curves for IHHO vs HHO modifications on multimodal fixed-dimension functions

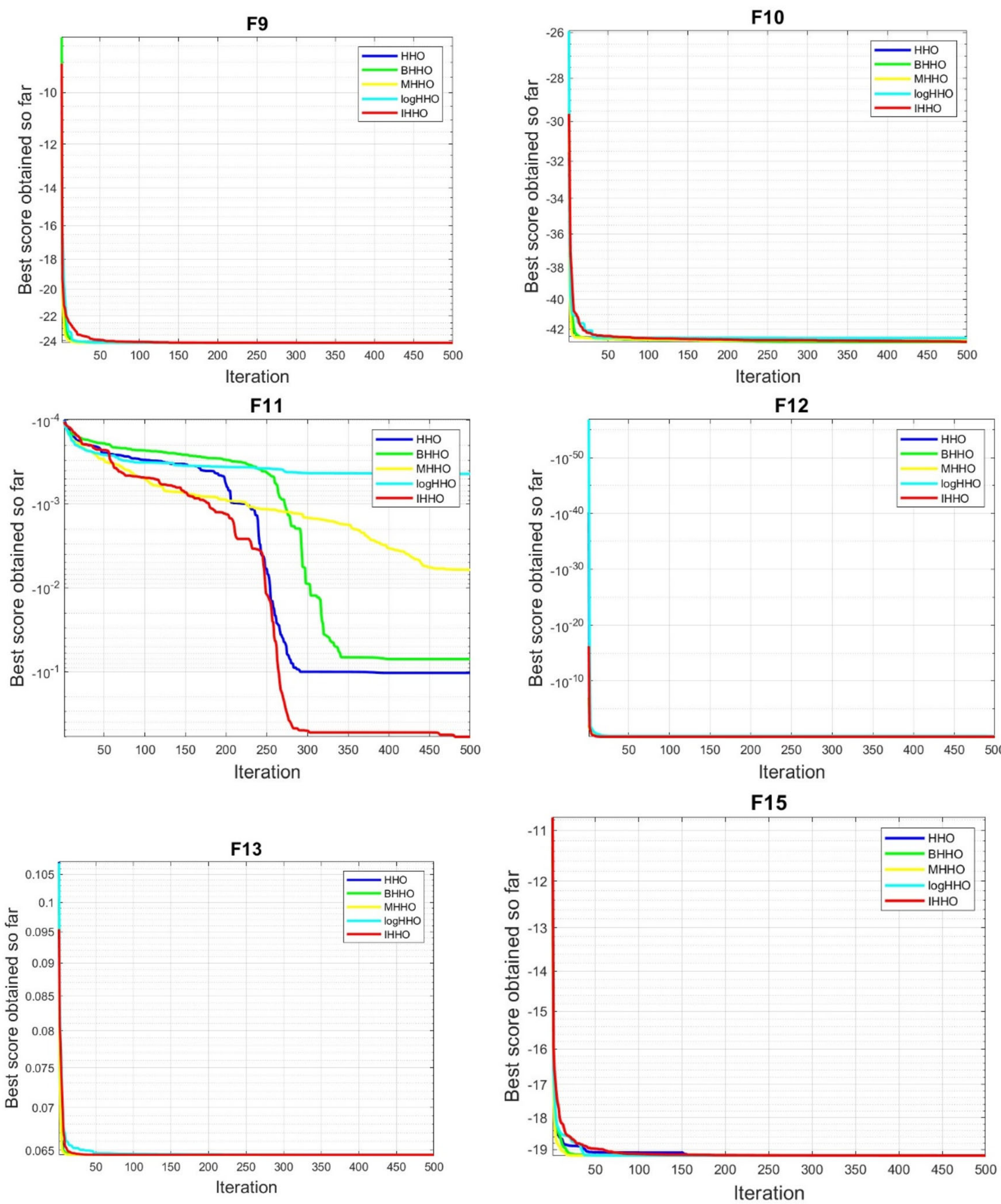


Fig. 33 continued

5.6.6 speed reducer problem

As shown in Fig. 40, there are countless applications for gear reducers because of their versatility and importance in the mechanical transmission of a wide variety of processes [95]. The reducer has several issues that make it less than ideal for use in modern applications, including its heft, high transmission ratio, and low mechanical efficiency [96]. Good power, big transmission ratio, compact size, high mechanical efficiency, and long service life are all requirements for an energy-efficient current reducer [75]. The following constraints are obtained:

$$\text{Min} f(x) = \frac{m_{n1} z_1 (1 + i_1) + m_{n2} z_3 (1 + i_2)}{2cos\beta} \quad (35)$$

Subject to:

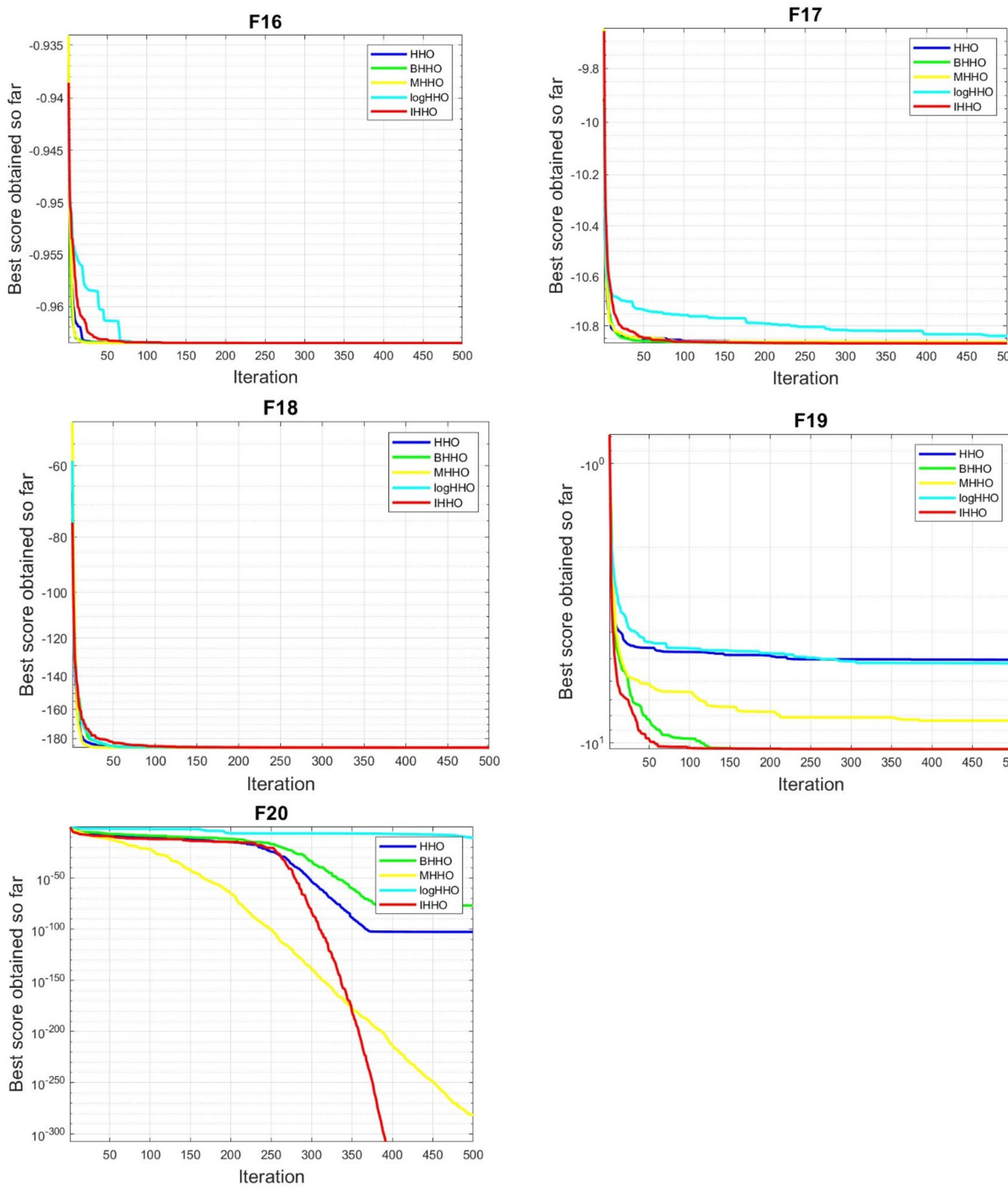


Fig. 33 continued

$$
\begin{aligned}
g_1 &= 3.098\ x\ 10^{-6} x_1^3 x_3^2 x_5 - \cos^3 x_6 \ \geq 0 \\
g_2 &= 1.017\ x\ 10^{-4} x_2^3 x_4^3 - x_5^2 \cos^3 x_6 / x_3^2 \ \geq 0 \\
g_3 &= 9.939\ x\ 10^{-5}.(1 + x_5/x_3) x_1^3 x_3^2 - \cos^3 x_6 \ \geq 0 \\
g_4 &= 1.076\ x\ 10^{-4}.(31.5 + x_5/x_3) x_2^3 x_4^2 - x_5^2 \cos^2 x_6 / x_3^2 \ \geq 0 \\
g_5 &= x_2 x_4 \left(31.5 + \frac{x_5}{x_3}\right) - \frac{x_5}{x_3}.[2(x_1 + 50).\cos x_6 + x_1 x_5] \geq 0
\end{aligned}
\tag{36}
$$

5.6.7 Results

In that part, we will test IHHO with the previous six engineering problems as in Table 27. The problems are tension/compression spring design [98], pressure vessel design [99], welded beam design [100], speed reducer problem [101], gear train design problem [68], and three-bar truss design [103].

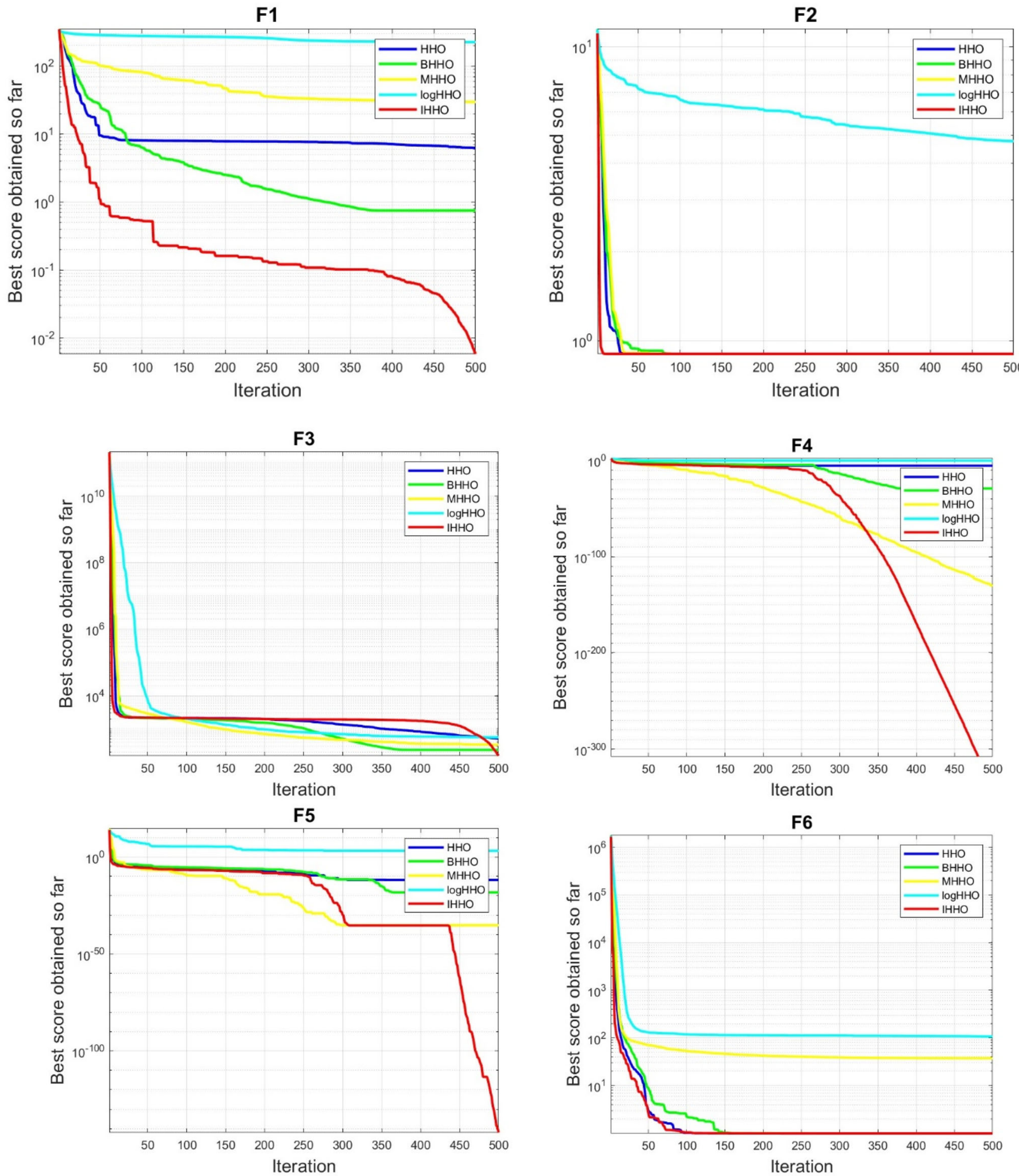


Fig. 34 Convergence curves for IHHO vs HHO modifications on multimodal variable-dimension functions

5.6.7.1 IHHO vs HHO The comparison of IHHO against HHO is presented in Table 28. This was done so that we could decide which of the two options was the better one. However, F2, F3, and F5 have a four-dimensional restricted optimization problem inside the given range for each equation. Because F1, F4, and F6 have different

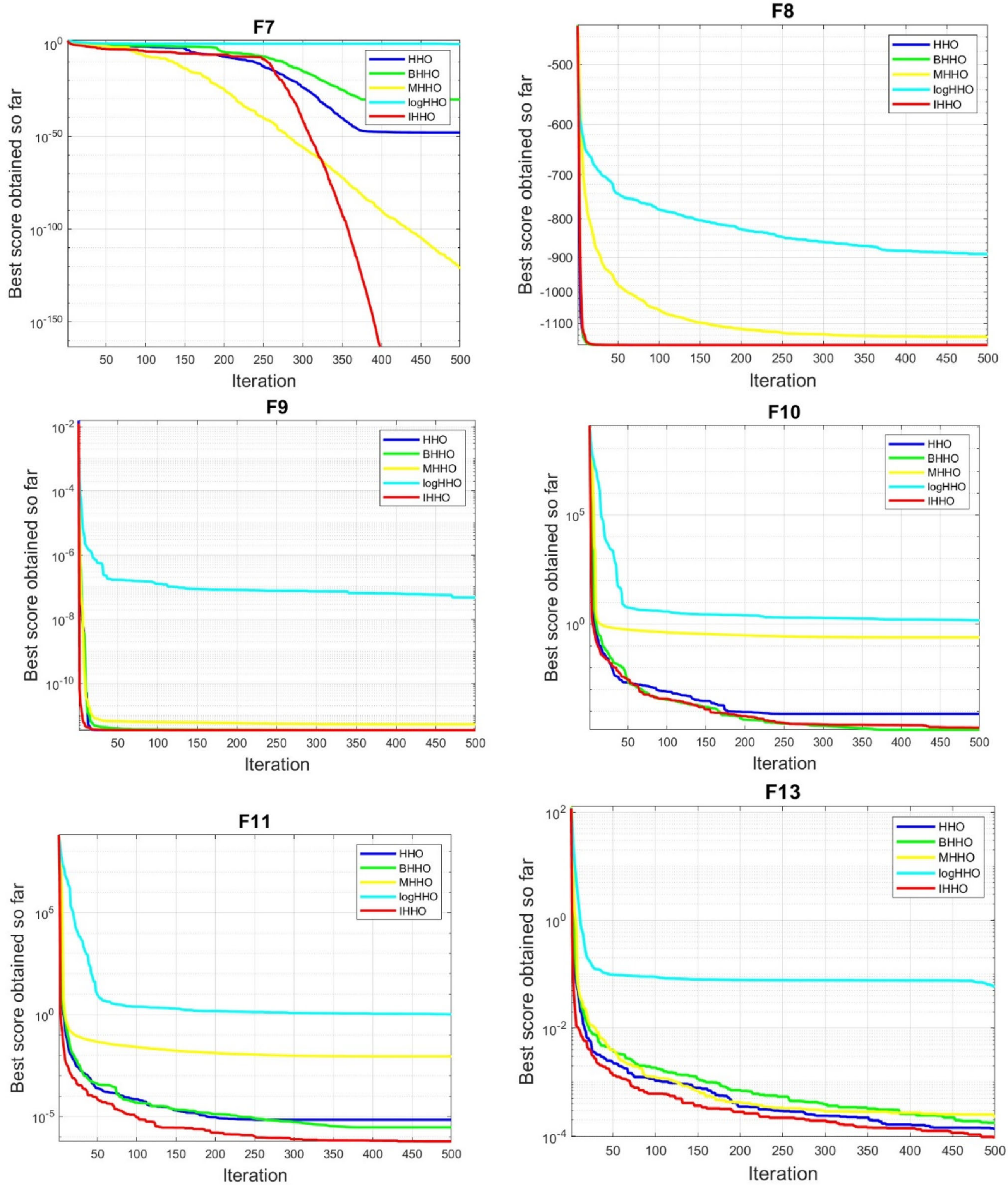


Fig. 34 continued

Fig. 35 Tension/compression spring design problem

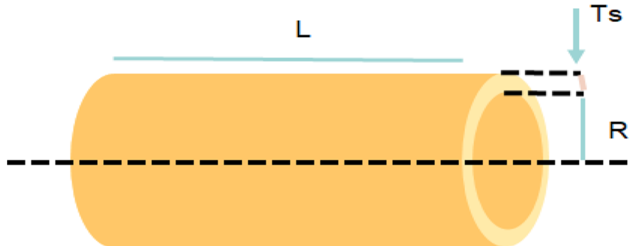


Fig. 36 Pressure vessel design problem

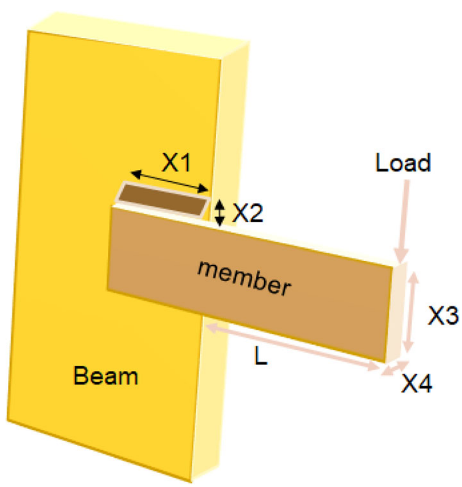


Fig. 37 Welded beam design problem

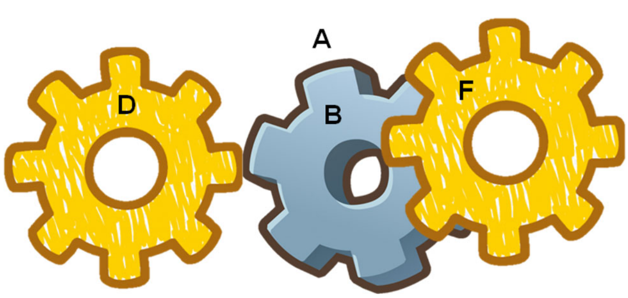


Fig. 38 Gear train design problem

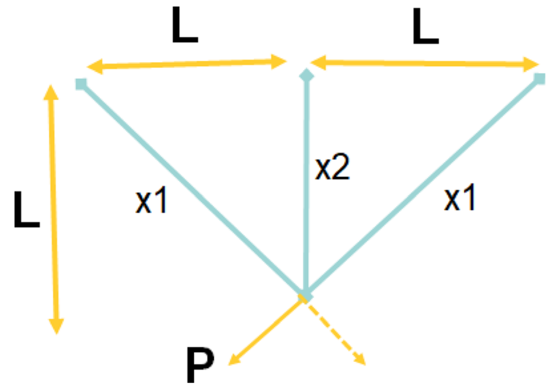


Fig. 39 Three-bar truss design problem

dimensions, this function 30 times for a total of 500 iterations was evaluated.

The results shown in Table 28 indicate that the performance of IHHO is better than that of the original HHO, except for F5 and F6. IHHO outperforms other analyzed methods for changing functions because of its greater capacity to escape the local optimum utilizing the random walk methodology. This gives IHHO a performance advantage over HHO. The higher performance of IHHO can largely be attributed to this factor.

A visual comparison of the convergence times of the IHHO method and the original HHO can be carried out by looking at the curves in Fig. 41, which shows the results of the comparison.

5.6.7.2 IHHO vs other algorithms Table 29 contains our comparison of IHHO to various algorithms, namely BAT [40], MFO [42], TLBO [41], and WOA [43]. This was done so that could make a decision regarding which of the possibilities was preferable. This evaluation a total of 30

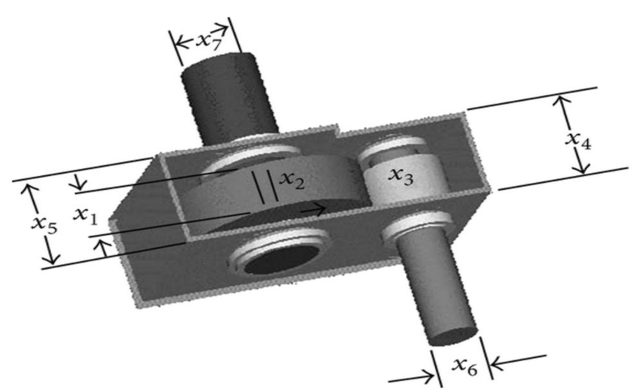


Fig. 40 Speed reducer problem [97]

Table 27 Description of engineering problems

Benchmark	Name	Dim	Range
F1	Tension/compression spring design	3	lb = [0.05, 0.25, 2] ub = [2, 1.3, 15]
F2	Pressure vessel design	4	lb = [0, 0, 10, 10] ub = [99, 99, 200, 200]
F3	Welded beam design	4	lb = [0.1, 0.1, 0.1, 0.1] ub = [2, 2, 10, 10]
F4	speed reducer problem	7	lb = [2.6, 0.7, 17, 7.3, 7.3, 2.9, 5.0] ub = [3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5]
F5	Gear train design problem	4	b = [12, 12, 12, 12] ub = [59, 59, 59, 59]
F6	Three−bar truss design	2	lb = [0,0]
			ub = [1, 1]

Table 28 Comparison of results of IHHO and HHO on six engineering problems

Benchmark	optimizer	Mean	STD	MIN	MAX
F1	IHHO	**0.013667**	**0.00095398**	0.012893	0.018728
	HHO	0.013941	0.00090174	0.012668	0.017774
F2	IHHO	**6512.1456**	**346.3228**	5966.6247	7427.7495
	HHO	1.14E + 04	2.47E + 04	6072.2207	7548.621
F3	IHHO	**2.0472**	**0.2608**	1.7535	2.6917
	HHO	2.1417	0.38059	1.7932	3.2194
F4	IHHO	**3030.5164**	**18.9552**	3013.4418	3140.4497
	HHO	3899.2033	565.4279	3028.2265	4876.9711
F5	IHHO	3.29E−11	7.54E−11	2.3598e−15	7.5594e−10
	HHO	**0**	**0**	0	0
F6	IHHO	265.4281	2.312	263.9116	272.8288
	HHO	**264.0973**	**0.23932**	263.8962	265.3612

Best results are highlighted in bold

times, making the total number of cycles 500 was performed.

According to Table 29, the performance of the HHO algorithm is better than that of other algorithms on all engineering problems, with the exception of the F5 and F6 methods. IHHO outperforms BAT by a significant margin when the two are compared to one another. When compared to MAT, IHHO performs better, with the exception of F6. IHHO outperforms TLBO since the former cannot solve all engineering problems. IHHO performs better than WOA in all functions with the exception of F2.

A visual comparison of the convergence periods of the IHHO method can be carried out using the Convergence curves shown in Fig. 42.

5.6.7.3 IHHO vs other modifications of HHO Here, the proposed IHHO with other HHO variants such as MHHO [37] and LogHHO [38] was evaluated. We have set the swarm size at 30, and the halting criterion at 500 iterations, to ensure that all algorithms get a fair shot. Table 30 displays experimental results regarding the mean and standard deviation of the objective functions. Figure 43 also displays the convergence results for the different approaches.

According to Table 30, the performance of the IHHO algorithm is better than that of all other modifications of HHO, with the exception of the F5, and F6 methods. When compared to MHHO and LogHHO, IHHO performs better, except for the F5, and F6 methods.

A visual comparison of the convergence periods of the IHHO method can be carried out using the convergence curves shown in Fig. 43.

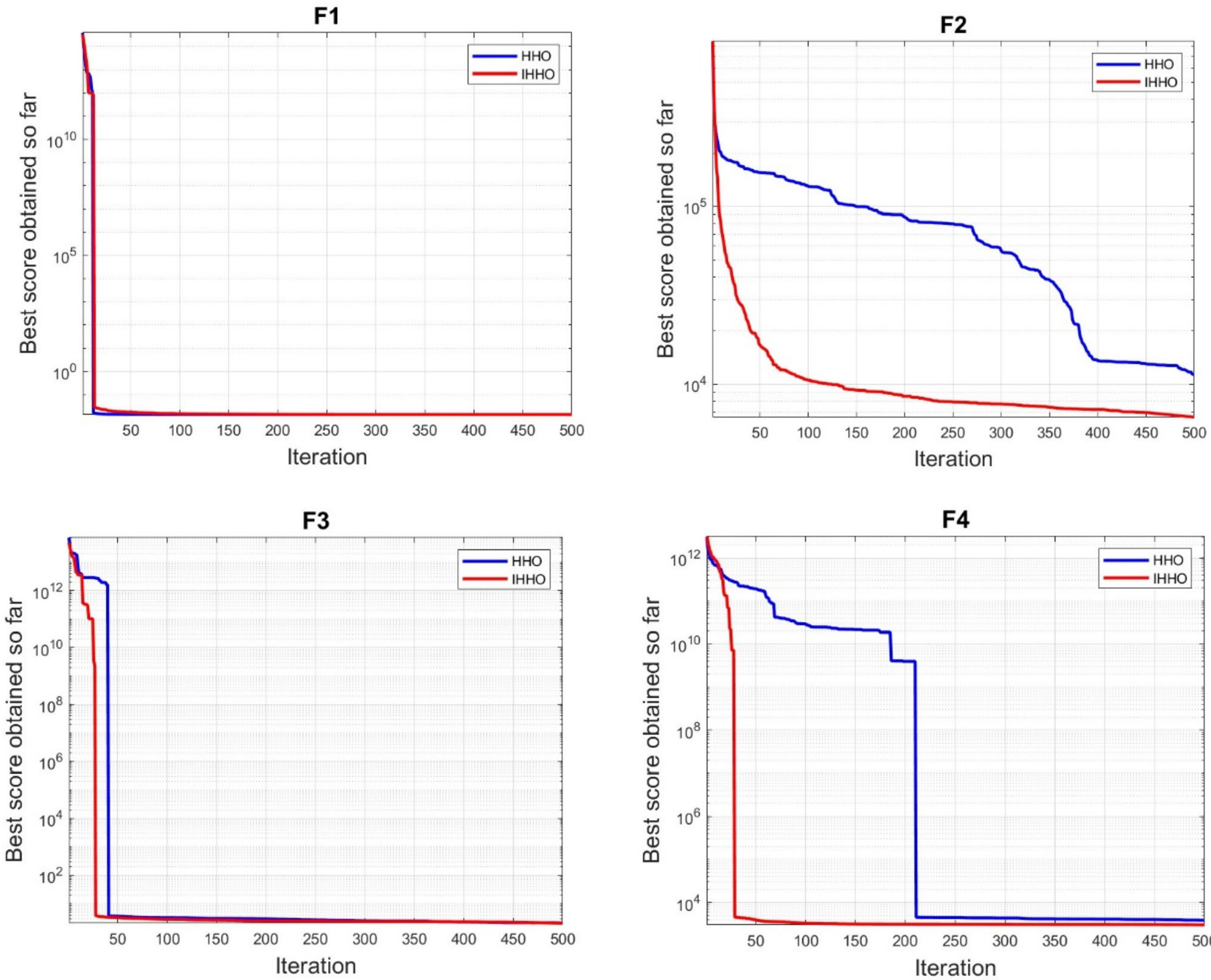


Fig. 41 Convergence curves for IHHO vs HHO on six engineering problems

In CEC 2017, BHHO outperformed other variants of HHO; however, in CEC 2020 and CEC 2019, it was unable to progress and break free from the local solution. While LogHHO performed admirably in CEC 2019, it was unable to break free from the local solution in CEC 2017 and CEC 2020. While MHHO performed well in CEC 2020, it was unsuccessful in CEC 2017 and CEC 2019. But in most benchmark equations (CEC2017, CEC2019, etc.), the improved modification (IHHO) outperformed all other changes that were applied to it.

The suggested approach, IHHO, outperforms other algorithms (GWO, BAT, WOA, TLBO, and MFO), as demonstrated by the numerical results. This is visually demonstrated by various convergence curves, but IHHO was able to overcome these and obtain the global solution in the majority of benchmarks. An analysis of the mean Friedman rank statistical test is used to compare the IHHO rank with other algorithms. The recommended method beats out GWO, BAT, WOA, TLBO, MFO, and three

Table 29 Comparison of results between IHHO and other algorithms on six engineering problems

Benchmark		IHHO	HHO	BAT [81]	MFO [42]	TLBO [41]	WAO [43]
F1	AVG	**0.013667**	0.013941	1.0193e + 13	0.0137	NAN	0.0138
	STD	**0.00095398**	0.00090174	3.9674e + 13	0.0014		0.0014
	Rank	**1**	4	5	2		3
F2	AVG	**6512.1456**	1.14E + 04	4.1358e + 05	6.6533e + 03	NAN	1.2247e + 04
	STD	**346.3228**	2.47E + 04	4.5439e + 05	601.0981		6.5017e + 03
	Rank	**1**	3	5	2		4
F3	AVG	**2.0472**	2.1417	1.8408e + 13	2.8124	NAN	3.1418
	STD	**0.2608**	0.38059	5.8079e + 13	0.1394		0.9008
	Rank	**1**	2	5	3		4
F4	AVG	**3030.5164**	3899.2033	1.4109e + 12	3.0102e + 03	NAN	3.4142e + 03
	STD	**18.9552**	565.4279	1.7415e + 12	7.9191		564.9316
	Rank	2	4	5	1		3
F5	AVG	3.29E−11	**0**	9.7184e−16	0	NAN	2.4713e−23
	STD	7.54E−11	**0**	1.5966e−15	**0**		1.1284e−22
	Rank	5	**1**	4	**1**		3
F6	AVG	265.4281	**264.0973**	268.634	263.9734	**NAN**	265.4553
	STD	2.312	**0.23932**	6.0427	0.1501		1.7689
	Rank	3	**2**	5	1		4
Percentage		2.166666667	2.666666667	4.833333333	1.666666667	–	3.5
Total rank		2	3	5	1	–	4

Best results are highlighted in bold

variants of HHO (BHHO, LogHHO, and MHHO), according to Friedman tests.

6 Conclusion and future work

Metaheuristic optimization algorithms are now commonly used to solve different engineering problems. Despite the available algorithms, new algorithms with extended abilities are continuously proposed to overcome the problems of the existing algorithms. This paper proposes an Improved algorithm based on the well-known Harris Hawks optimization algorithm. The algorithm proposed in this study emphasizes the utilization of random location-based habitats during the exploration phase and the implementation of strategies 1, 3, and 4 during the exploitation phase. The improved algorithm is capable of achieving a good balance between exploration and exploitation due to the modifications of each phase. Our suggested algorithm IHHO is benchmarked against not only the original HHO, but also against state-of-the-art algorithms, namely grey wolf optimization, BAT algorithm, teaching–learning-based optimization, moth-flame optimization, and whale optimization algorithm. IHHO is also compared with other modifications of HHO, namely BHHO, MHHO, and logHHO. Different benchmark functions with varying difficulties CEC2017, CEC2019, CEC2020, and 52 benchmark functions were used. We have also applied these algorithms to six classical real-world engineering problems.

Compared to other variations of HHO, BHHO performed well in CEC 2017 but failed to advance and escape the local solution in CEC 2020 and CEC 2019. LogHHO did well in CEC 2019 but was not able to get ahead and out of the local solution in CEC 2017, and CEC 2020 MHHO did well in CEC 2020, however, it failed in CEC 2017 and CEC 2019. However, the enhanced modification (IHHO) demonstrated exceptional performance in the majority of

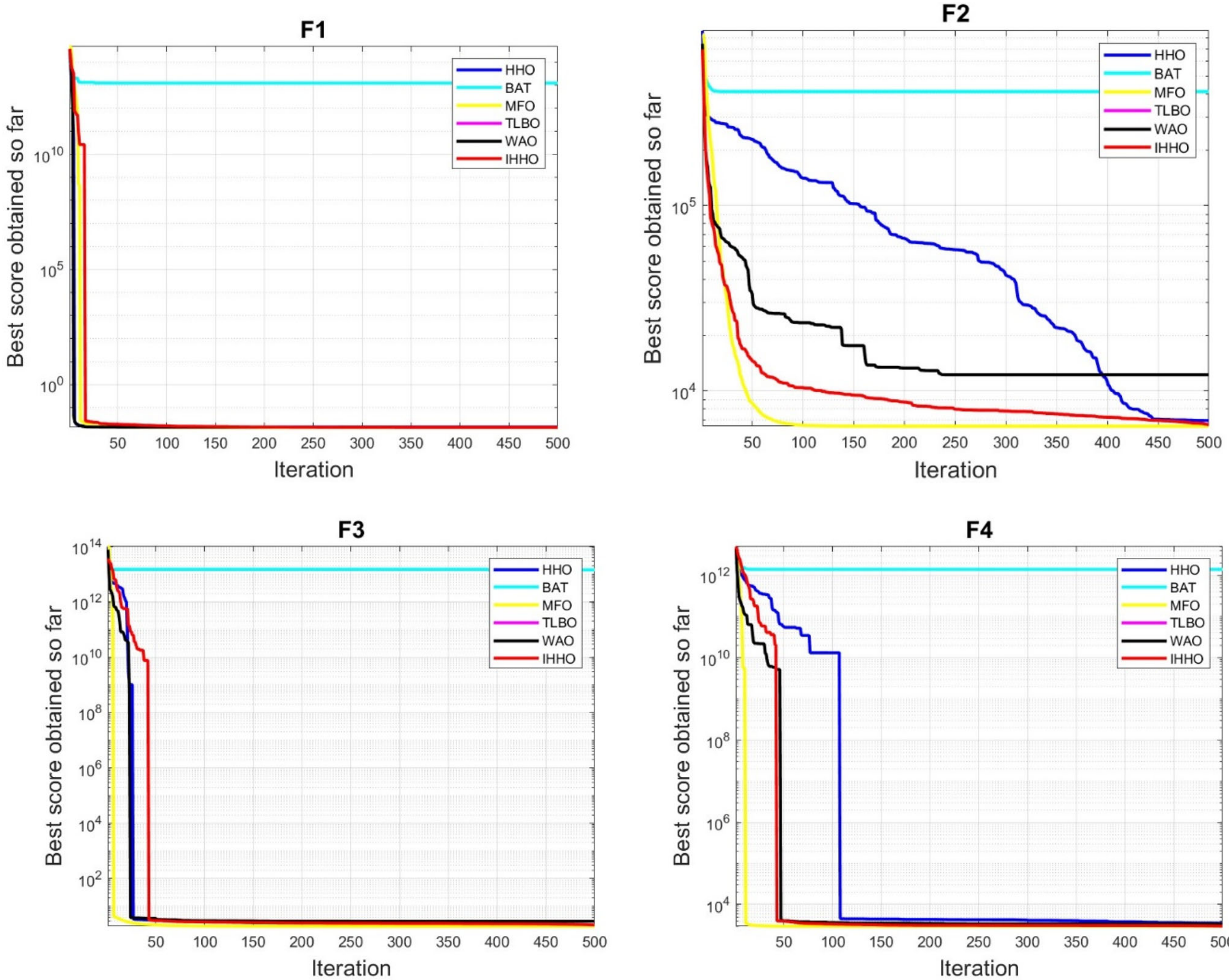


Fig. 42 Convergence curves for IHHO vs other algorithms on six engineering problems

benchmark equations (CEC2017, CEC2019, etc.), surpassing all other modifications implemented on it.

The numerical results show the superiority of the proposed algorithm IHHO over other algorithms (GWO, BAT, WOA, TLBO, and MFO), which is visually proven using different convergence curves but IHHO was able to overcome them and achieve the global solution in most benchmarks. The IHHO rank is compared to other algorithms using a statistical test of the mean Friedman rank. Friedman tests showed that the suggested algorithm outperforms GWO, BAT, WOA, TLBO, MFO, and three HHO variations (BHHO, LogHHO, and MHHO).

As with every proposed algorithm, IHHO has some **limitations**. First, IHHO has solved the problem of sensitivity to parameters in the original HHO. The results of the discussed problems are promising. However, this may be ineffective in other problems. Second, convergence time increases with complexity, which opens a window for

Table 30 Comparison of results of IHHO vs other modifications of HHO on six engineering problems

Benchmark		IHHO	HHO	MHHO [37]	LogHHO [36]
F1	AVG	**0.013667**	0.013941	0.013809	0.013746
	STD	**0.00095398**	0.00090174	0.0011	0.0014
	Rank	**1**	4	3	2
F2	AVG	**6512.1456**	1.14E + 04	6807.9021	10,164.7297
	STD	**346.3228**	2.47E + 04	440.6797	4.7936e + 03
	Rank	**1**	4	2	3
F3	AVG	**2.0472**	2.1417	2.0476	2.1977
	STD	**0.2608**	0.38059	0.2710	0.3709
	Rank	**1**	3	2	4
F4	AVG	**3030.5164**	3899.2033	3086.8416	3841.9831
	STD	**18.9552**	565.4279	6.5047	0.0491
	Rank	**1**	4	2	3
F5	AVG	3.29E−11	**0**	5.23E−32	8.37E−22
	STD	7.54E−11	**0**	8.3580e−29	4.5870e−24
	Rank	4	**1**	2	3
F6	AVG	265.4281	264.0973	**263.9677**	264.092
	STD	2.312	0.23932	**0.1847**	0.2153
	Rank	4	3	**1**	2
Percentage		2	3.166666667	2	2.833333333
Total Rank		1	4	1	3

Best results are highlighted in bold

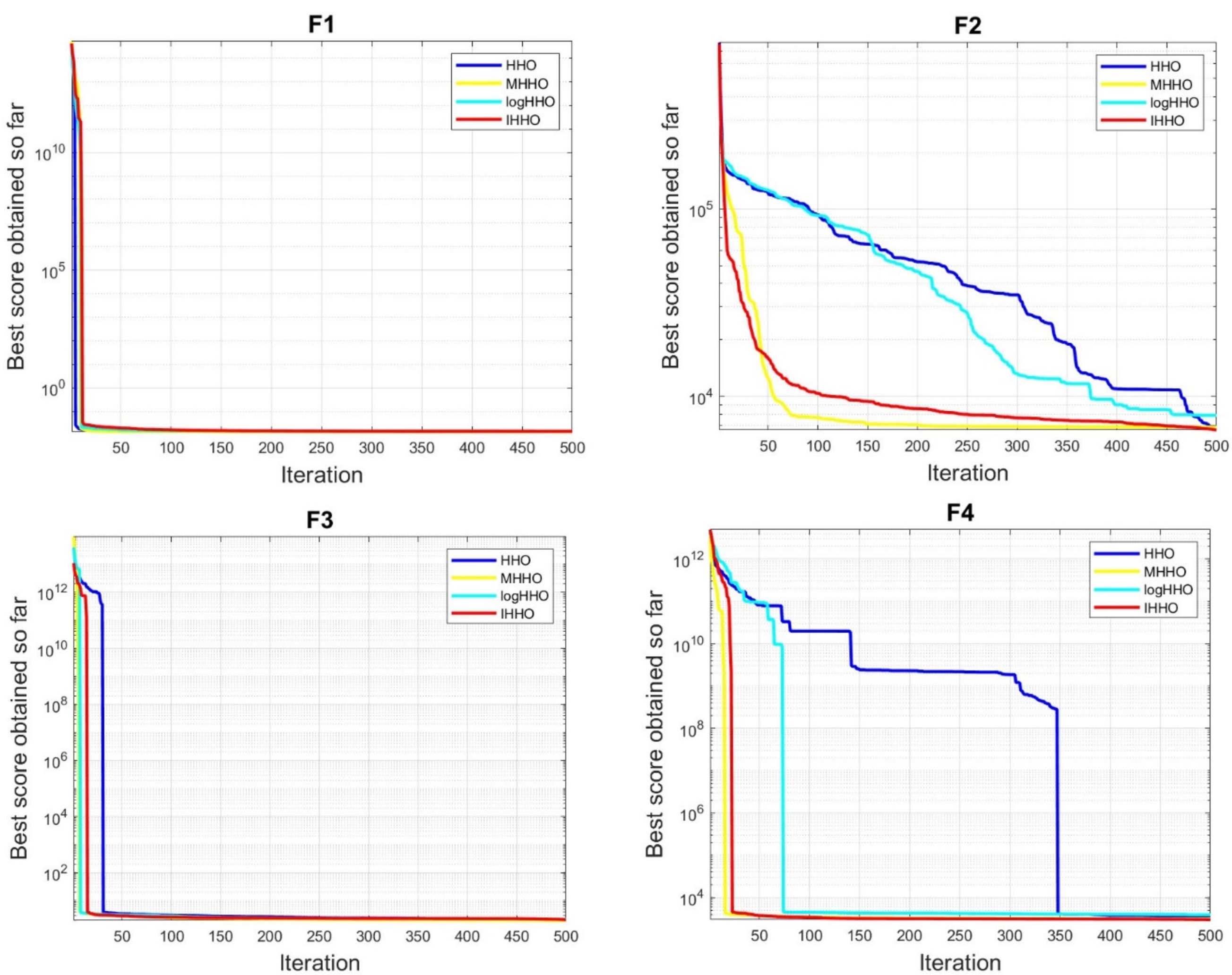


Fig. 43 Convergence curves for IHHO vs HHO modifications on six engineering problems

future research directions to solve this issue. Finally, as this area of research is continuously improving, new algorithms with better performance may be developed.

In future work, we plan to further improve the proposed algorithm. Problem–based adaptive parameters can be used. The binary version of the algorithm can also be developed for classification tasks.

Author's contribution DTA, MMS, AYH, and EMEG contributed to the design and implementation of the research and the analysis of the results. All the authors have participated in writing the manuscript and have revised the final version. All authors read and approved the final manuscript.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and material Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. There are no conflicts of interest. I would like to confirm that there were no known conflicts of interest associated with this publication and that there was no significant financial support for this work that could affect its outcome.

References

1. Khanduja N, Bhushan B (2021) Recent advances and application of metaheuristic algorithms: a survey (2014–2020). In: Malik H, Iqbal A, Joshi P, Agrawal S, Bakhsh FI (eds) Metaheuristic and evolutionary computation: algorithms and applications. In Studies in Computational Intelligence. Springer, Singapore, pp 207–228. https://doi.org/10.1007/978-981-15-7571-6_10
2. Gogna A, Tayal A (2013) Metaheuristics: review and application. J Exp Theor Artif Intell 25(4):503–526. https://doi.org/10.1080/0952813X.2013.782347
3. De León-Aldaco SE, Calleja H, Aguayo Alquicira J (2015) Metaheuristic optimization methods applied to power converters: a review. IEEE Trans Power Electronics 30(12):6791–6803. https://doi.org/10.1109/TPEL.2015.2397311
4. El-Gendy EM, Saafan MM, Elksas MS, Saraya SF, Areed FFG (2020) Applying hybrid genetic–PSO technique for tuning an adaptive PID controller used in a chemical process. Soft Comput 24(5):3455–3474. https://doi.org/10.1007/s00500-019-04106-z
5. Halim AH, Ismail I, Das S (2021) Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. Artif Intell Rev 54(3):2323–2409. https://doi.org/10.1007/s10462-020-09906-6
6. Yang X-S (2010) Nature-inspired metaheuristic algorithms. Luniver Press, Bristol
7. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040. https://doi.org/10.1016/j.cie.2019.106040
8. Ficarella E, Lamberti L, Degertekin SO (2021) Comparison of three novel hybrid metaheuristic algorithms for structural optimization problems. Comput Struct 244:106395. https://doi.org/10.1016/j.compstruc.2020.106395
9. Saafan MM, El-Gendy EM (2021) IWOSSA: an improved whale optimization salp swarm algorithm for solving optimization problems. Expert Syst Appl 176:114901. https://doi.org/10.1016/j.eswa.2021.114901
10. Almufti M (2022) Historical survey on metaheuristics algorithms. Int J Sci World. Accessed: 11, Jul 2022. [Online]. Available: https://www.sciencepubco.com/index.php/IJSW/article/view/29497
11. AlMufti SM (2018) Review on elephant herding optimization algorithm performance in solving optimization problems. Int J Eng Technol. Accessed: 30 Jul 2022. [Online]. Available: https://www.academia.edu/39104104/Review_on_Elephant_Herding_Optimization_Algorithm_Performance_in_Solving_Optimization_Problems
12. Balaha HM, Saafan MM (2021) Automatic exam correction framework (AECF) for the MCQs, essays, and equations matching. IEEE Access 9:32368–32389. https://doi.org/10.1109/ACCESS.2021.3060940
13. Morales-Castañeda B, Zaldívar D, Cuevas E, Fausto F, Rodríguez A (2020) A better balance in metaheuristic algorithms: Does it exist? Swarm Evol Comput 54:100671. https://doi.org/10.1016/j.swevo.2020.100671
14. Alabool HM, Alarabiat D, Abualigah L, Heidari AA (2021) Harris hawks optimization: a comprehensive review of recent variants and applications. Neural Comput Appl 33(15):8939–8980. https://doi.org/10.1007/s00521-021-05720-5
15. Desouky NA, Saafan MM, Mansour MH, Maklad OM (2023) Patient-specific air puff-induced loading using machine learning. Front Bioeng Biotechnol 11:1277970. https://doi.org/10.3389/fbioe.2023.1277970
16. Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117. https://doi.org/10.1016/j.ins.2013.02.041
17. Yousif NR, Balaha HM, Haikal AY, El-Gendy EM (2023) A generic optimization and learning framework for Parkinson disease via speech and handwritten records. J Ambient Intell Human Comput 14(8):10673–10693. https://doi.org/10.1007/s12652-022-04342-6
18. Crespo-Cano R, Cuenca-Asensi S, Fernández E, Martínez-Álvarez A (2019) 'Metaheuristic optimisation algorithms for tuning a bioinspired retinal model. Sensors 19(22):4834. https://doi.org/10.3390/s19224834
19. Adekanmbi O, Green P (2015) Conceptual comparison of population based metaheuristics for engineering problems. Sci World J 2015:e936106. https://doi.org/10.1155/2015/936106

20. Badr AA, Saafan MM, Abdelsalam MM, Haikal AY (2023) Novel variants of grasshopper optimization algorithm to solve numerical problems and demand side management in smart grids. Artif Intell Rev 56(10):10679–10732. https://doi.org/10.1007/s10462-023-10431-5
21. Memari A, Ahmad R, Rahim ARA (2017) Metaheuristic algorithms: guidelines for implementation. JSCDSS 4(6):1–6
22. Balaha HM, Antar ER, Saafan MM, El-Gendy EM (2023) A comprehensive framework towards segmenting and classifying breast cancer patients using deep learning and Aquila optimizer. J Ambient Intell Human Comput 14(6):7897–7917. https://doi.org/10.1007/s12652-023-04600-1
23. Chopard B, Tomassini M (2018) Performance and limitations of metaheuristics. In: Chopard B, Tomassini M (eds) An introduction to metaheuristics for optimization, in Natural Computing Series. Springer International Publishing, Cham, pp 191–203. https://doi.org/10.1007/978-3-319-93073-2_11.
24. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80(5):8091–8126. https://doi.org/10.1007/s11042-020-10139-6
25. Emmerich MTM, Deutz AH (2018) A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat Comput 17(3):585–609. https://doi.org/10.1007/s11047-018-9685-y
26. Reynolds RG, Kinnaird-Heether L (2013) Optimization problem solving with auctions in cultural algorithms. Memetic Comp 5(2):83–94. https://doi.org/10.1007/s12293-013-0112-8
27. Paul AK, Sato T (2017) Localization in wireless sensor networks: a survey on algorithms, measurement techniques, applications and challenges. J Sens Actuat Netw 6(4):24. https://doi.org/10.3390/jsan6040024
28. Nature-inspired metaheuristic algorithms for engineering optimization applications. Accessed: 31 Jul 2022. [Online]. Available: https://link.springer.com/book/https://doi.org/10.1007/978-981-33-6773-9
29. Azad AS, Rahaman MSA, Watada J, Vasant P, Vintaned JAG (2020) Optimization of the hydropower energy generation using Meta-Heuristic approaches: a review. Energy Rep 6:2230–2248. https://doi.org/10.1016/j.egyr.2020.08.009
30. Forestiero A, Mastroianni C, Spezzano G (2008) Reorganization and discovery of grid information with epidemic tuning. Futur Gener Comput Syst 24(8):788–797. https://doi.org/10.1016/j.future.2008.04.001
31. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization. In: Bansal JC, Das KN, Nagar A, Deep K, Ojha AK (eds) Soft computing for problem solving. In Advances in Intelligent Systems and Computing. Springer, Singapore, pp 599–615. https://doi.org/10.1007/978-981-13-1592-3_47.
32. Cicirelli F, Forestiero A, Giordano A, Mastroianni C (2016) Transparent and efficient parallelization of swarm algorithms. ACM Trans Auton Adapt Syst 11(2):14:1–14:26. https://doi.org/10.1145/2897373
33. Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems—ScienceDirect. Accessed: 30 Jul 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0950705118301357
34. Dhiman G, Kaur A (2018) Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. Designs 2:28. https://doi.org/10.3390/designs2030028
35. Forestiero A (2017) Bio-inspired algorithm for outliers detection. Multimed Tools Appl 76(24):25659–25677. https://doi.org/10.1007/s11042-017-4443-1
36. 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. https://doi.org/10.1016/j.enconman.2020.112660
37. Zhang Y, Zhou X, Shih P-C (2020) Modified Harris Hawks optimization algorithm for global optimization problems. Arab J Sci Eng 45(12):10949–10974. https://doi.org/10.1007/s13369-020-04896-7
38. Devarapalli R, Bhattacharyya B (2019) Optimal parameter tuning of power oscillation damper by MHHO algorithm. In: 2019 20th International conference on intelligent system application to power systems (ISAP), pp 1–7. https://doi.org/10.1109/ISAP48318.2019.9065988.
39. 'Grey Wolf Optimization (GWO) (2022)Algorithm | SpringerLink'. Accessed: 22, 2022. [Online]. Available: https://link.springer.com/chapter/https://doi.org/10.1007/978-981-10-5221-7_9
40. 'Bat algorithm: literature review and applications. Int J Bio Inspired Comput. Accessed: 22 Nov 2022. [Online]. Available: https://www.inderscienceonline.com/doi/abs/https://doi.org/10.1504/IJBIC.2013.055093
41. 'Teaching-Learning-Based Optimization Algorithm. SpringerLink'. Accessed: 22 Nov 2022. [Online]. Available: https://link.springer.com/chapter/https://doi.org/10.1007/978-3-319-22732-0_2
42. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
43. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
44. Yu Y, Bian R, Wang T, Wang W, Zhang J (2022) Inverse time overcurrent optimization of distribution network with DG based on improved gray wolf algorithm. Energy Rep 8:237–246. https://doi.org/10.1016/j.egyr.2022.09.095
45. Rodríguez-Esparza E et al (2020) An efficient Harris hawks-inspired image segmentation method. Exp Syst Appl 155:113428. https://doi.org/10.1016/j.eswa.2020.113428
46. Fahmy H, El-Gendy EM, Mohamed MA, Saafan MM (2023) ECH3OA: an enhanced chimp-harris hawks optimization algorithm for copyright protection in color images using watermarking techniques. Knowl-Based Syst 269:110494. https://doi.org/10.1016/j.knosys.2023.110494
47. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: Algorithm and applications. Futur Gener Comput Syst 97:849–872. https://doi.org/10.1016/j.future.2019.02.028
48. Wang S, Jia H, Abualigah L, Liu Q, Zheng R (2021) An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems. Processes 9(9):1551. https://doi.org/10.3390/pr9091551
49. Houssein EH, Neggaz N, Hosney ME, Mohamed WM, Hassaballah M (2021) Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities. Neural Comput Appl 33(20):13601–13618. https://doi.org/10.1007/s00521-021-05991-y
50. Li C, Li J, Chen H, Jin M, Ren H (2021) Enhanced Harris hawks optimization with multi-strategy for global optimization tasks. Exp Syst Appl 185:115499. https://doi.org/10.1016/j.eswa.2021.115499
51. Balaha HM, Hassan AE-S, El-Gendy EM, ZainEldin H, Saafan MM (2023) An aseptic approach towards skin lesion localization and grading using deep learning and harris hawks optimization. Multimed Tools Appl. https://doi.org/10.1007/s11042-023-16201-3
52. Hussain K, Neggaz N, Zhu W, Houssein EH (2021) An efficient hybrid sine-cosine Harris hawks optimization for low and high-

dimensional feature selection. Exp Syst Appl 176:114778. https://doi.org/10.1016/j.eswa.2021.114778
53. Reda M, Onsy A, Elhosseini MA, Haikal AY, Badawy M (2022) A discrete variant of cuckoo search algorithm to solve the Travelling Salesman Problem and path planning for autonomous trolley inside warehouse. Knowl Based Syst 252:109290. https://doi.org/10.1016/j.knosys.2022.109290
54. 'Boosted mutation-based Harris hawks optimizer for parameters identification of single-diode solar cell models - ScienceDirect'. Accessed: 1 Jul, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0196890420301989
55. Saad E, Elhosseini MA, Haikal AY (2019) Culture-based Artificial Bee Colony with heritage mechanism for optimization of Wireless Sensors Network. Appl Soft Comput 79:59–73. https://doi.org/10.1016/j.asoc.2019.03.040
56. Gupta S, Deep K, Heidari AA, Moayedi H, Wang M (2020) Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis. Expert Syst Appl 158:113510. https://doi.org/10.1016/j.eswa.2020.113510
57. 'An Innovative Damped Cuckoo Search Algorithm With a Comparative Study Against Other Adaptive Variants | IEEE Journals & Magazine | IEEE Xplore'. Accessed: 06, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/8805320
58. 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. https://doi.org/10.1016/j.enconman.2020.112470
59. Elhosseini MA, Haikal AY, Badawy M, Khashan N (2019) Biped robot stability based on an A-C parametric Whale Optimization Algorithm. J Comput Sci 31:17–32. https://doi.org/10.1016/j.jocs.2018.12.005
60. Hussain K, Zhu W, Mohd Salleh MN (2019) Long-term memory Harris' Hawk optimization for high dimensional and optimal power flow problems. IEEE Access 7:147596–147616. https://doi.org/10.1109/ACCESS.2019.2946664
61. Fan Q, Chen Z, Xia Z (2020) A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems. Soft Comput 24(19):14825–14843. https://doi.org/10.1007/s00500-020-04834-7
62. 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. https://doi.org/10.1016/j.asoc.2019.106018
63. Du P, Wang J, Hao Y, Niu T, Yang W (2020) A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2.5 and PM10 forecasting. Appl Soft Comput 96:106620. https://doi.org/10.1016/j.asoc.2020.106620
64. 'hHHO-IGWO: A New Hybrid Harris Hawks Optimizer for Solving Global Optimization Problems'. Accessed: 04 Aug 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9051509/
65. Zhong C, Wang M, Dang C, Ke W, Guo S (2020) First-order reliability method based on Harris Hawks Optimization for high-dimensional reliability analysis. Struct Multidisc Optim 62(4):1951–1968. https://doi.org/10.1007/s00158-020-02587-3
66. Shehabeldeen TA, Elaziz MA, Elsheikh AH, Zhou J (2019) Modeling of friction stir welding process using adaptive neuro-fuzzy inference system integrated with harris hawks optimizer. J Market Res 8(6):5882–5892. https://doi.org/10.1016/j.jmrt.2019.09.060
67. Chen H, Heidari AA, Chen H, Wang M, Pan Z, Gandomi AH (2020) Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. Futur Gener Comput Syst 111:175–198. https://doi.org/10.1016/j.future.2020.04.008
68. Qu C, He W, Peng X, Peng X (2020) Harris Hawks optimization with information exchange. Appl Math Model 84:52–75. https://doi.org/10.1016/j.apm.2020.03.024
69. 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. https://doi.org/10.1016/j.engappai.2019.103370
70. Menesy A, Sultan H, Selim A, Ashmawy M, Kamel S (2019) Developing and applying chaotic Harris Hawks optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks. IEEE Access 11:1–1. https://doi.org/10.1109/ACCESS.2019.2961811
71. Thaher T, Heidari AA, Mafarja M, Dong JS, Mirjalili S (2020) Binary Harris Hawks optimizer for high-dimensional, low sample size feature selection. In: Mirjalili S, Faris H, Aljarah I, (eds) Evolutionary machine learning techniques: algorithms and applications, in Algorithms for intelligent systems. Springer, Singapore, pp 251–272. https://doi.org/10.1007/978-981-32-9990-0_12.
72. 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. https://doi.org/10.1016/j.eswa.2021.115805
73. Amiri Golilarz N, Gao H, Demirel H (2019) Satellite Image denoising with Harris Hawks meta heuristic optimization algorithm and improved adaptive generalized gaussian distribution threshold function. IEEE Access 7:57459–57468. https://doi.org/10.1109/ACCESS.2019.2914101
74. Weber K (2002) 'Students' understanding of exponential and logarithmic functions'. In: Second international conference on the teaching of mathematics
75. Aljarah I, Mafarja M, Heidari AA, Faris H, Mirjalili S (2020) Multi-verse optimizer: theory, literature review, and application in data clustering. In: Mirjalili S, Song Dong J, Lewis A (eds), Nature-inspired optimizers: theories, literature reviews and applicationsin studies, in Computational Intelligence. Springer International Publishing, Cham, pp 123–141. https://doi.org/10.1007/978-3-030-12127-3_8.
76. López-Vázquez C, Hochsztain E (2019) Extended and updated tables for the Friedman rank test. Commun Stat Theory Methods 48(2):268–281. https://doi.org/10.1080/03610926.2017.1408829
77. Eisinga R, Heskes T, Pelzer B, Te Grotenhuis M (2017) Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers. BMC Bioinf 18(1):68. https://doi.org/10.1186/s12859-017-1486-2
78. van de Wiel MA (2004) Exact null distributions of quadratic distribution-free statistics for two-way classification. J Stat Plan Inference 120(1):29–40. https://doi.org/10.1016/S0378-3758(02)00498-6
79. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18. https://doi.org/10.1016/j.swevo.2011.02.002
80. Ballester PJ, Stephenson J, Carter JN, Gallagher K (2005) Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. 2005 IEEE Cong Evol Comput 1:498–505. https://doi.org/10.1109/CEC.2005.1554724
81. Yang X-S, He X (2013) Bat algorithm: literature review and applications. Int J Bio Inspired Comput 5(3):141–149. https://doi.org/10.1504/IJBIC.2013.055093
82. Askari Q, Saeed M, Younas I (2020) Heap-based optimizer inspired by corporate rank hierarchy for global optimization.

Expert Syst Appl 161:113702. https://doi.org/10.1016/j.eswa.2020.113702
83. Abdullah JM, Ahmed T (2019) Fitness dependent optimizer: inspired by the bee swarming reproductive process. IEEE Access 7:43473–43486. https://doi.org/10.1109/ACCESS.2019.2907012
84. Evaluating the performance of adaptive gainingsharing knowledge based algorithm on CEC 2020 benchmark problems. In: IEEE Conference Publication | IEEE Xplore'. Accessed: 13 Jan 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9185901
85. [PDF] Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Semantic Scholar. Accessed: 13 Jan 2023. [Online]. Available: https://www.semanticscholar.org/paper/Problem-Definitions-and-Evaluation-Criteria-for-the-Liang-Qu/a70236d1c7df42bf4292cb4762caec732aafd7cd
86. Wu G, Mallipeddi R, Suganthan P (2016) Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization
87. (PDF) Benchmark Functions for the CEC'2013 Special Session and Competition on Large-Scale Global Optimization. Accessed: 13 Jan 2023. [Online]. Available: https://www.researchgate.net/publication/261562928_Benchmark_Functions_for_the_CEC'2013_Special_Session_and_Competition_on_Large-Scale_Global_Optimization
88. Tzanetos A, Blondin M (2023) A qualitative systematic review of metaheuristics applied to tension/compression spring design problem: Current situation, recommendations, and research direction. Eng Appl Artif Intell 118:105521. https://doi.org/10.1016/j.engappai.2022.105521
89. Celik Y, Kutucu H (2018) Solving the tension/compression spring design problem by an improved firefly algorithm. In: Presented at the IDDM. Accessed: 09 Jan 2023. [Online]. Available: https://www.semanticscholar.org/paper/Solving-the-Tension%2FCompression-Spring-Design-by-an-Celik-Kutucu/cb2a064c995dc4f5caa714d82641f3d7e7a3806b
90. Chen Y, Lu C, Fan W, Feng J, Sareh P (2023) Data-driven design and morphological analysis of conical six-fold origami structures. Thin Walled Struct 185:110626. https://doi.org/10.1016/j.tws.2023.110626
91. Kamil AT, Saleh HM, Abd-Alla IH (2021) A multi-swarm structure for particle swarm optimization: solving the welded beam design problem. J Phys Conf Ser 1804(1):012012. https://doi.org/10.1088/1742-6596/1804/1/012012
92. Savsani V, Rao RV, Vakharia DP (2010) Optimal weight design of a gear train using particle swarm optimization and simulated annealing algorithms. Mech Mach Theory 45(3):531–541. https://doi.org/10.1016/j.mechmachtheory.2009.10.010
93. Kaveh A (2013) Optimal analysis of structures by concepts of symmetry and regularity. Springer, Vienna. https://doi.org/10.1007/978-3-7091-1565-7.
94. Ray T, Saini P (2001) Engineering design optimization using a swarm with an Intelligent information sharing among individuals. Eng Optim 33(6):735–748. https://doi.org/10.1080/03052150108940941
95. Chen Y et al (2023) A Unified Inverse Design and Optimization Workflow for the Miura-oRing Metastructure. J Mech Des 145:091704. https://doi.org/10.1115/1.4062667
96. Chen Y, Lu C, Yan J, Feng J, Sareh P (2022) Intelligent computational design of scalene-faceted flat-foldable tessellations. J Comput Des Eng 9(5):1765–1774. https://doi.org/10.1093/jcde/qwac082
97. Design optimization of a speed reducer using deterministic techniques. Accessed: 10 Jan 2023. [Online]. Available: https://www.hindawi.com/journals/mpe/2013/419043/
98. Rao RV, Waghmare GG (2017) A new optimization algorithm for solving complex constrained design optimization problems. Eng Optim 49(1):60–83. https://doi.org/10.1080/0305215X.2016.1164855
99. Shen H, Zhu Y, Niu B, Wu QH (2009) An improved group search optimizer for mechanical design optimization problems. Prog Nat Sci 19(1):91–97. https://doi.org/10.1016/j.pnsc.2008.06.007
100. Comparison of evolutionary-based optimization algorithms for structural design optimization—ScienceDirec. Accessed: 29 Nov 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197612001200?casa_token=Tjor35kZ4LwAAAAA:VTETqG1YUdu2XreFC6J8EeEaCtZIdhYaGXg4UxnWvAKMoL0K0Qb9D2bYNI1RKv8L5sa7dlEBMns
101. Sun Y et al (2020) Gear reducer optimal design based on computer multimedia simulation. J Supercomput 76(6):4132–4148. https://doi.org/10.1007/s11227-018-2255-3
102. Golabi S, Fesharaki JJ, Yazdipoor M (2014) Gear train optimization based on minimum volume/weight design. Mech Mach Theory 73:197–217. https://doi.org/10.1016/j.mechmachtheory.2013.11.002
103. Erdoğan Yildirim A, Karci A (2018) Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm. In: 2018 International conference on artificial intelligence and data processing (IDAP), pp 1–5. https://doi.org/10.1109/IDAP.2018.8620762.