10.1109/ICENCO49778.2020.9357403

Chaotic Harris Hawks Optimization for Unconstrained Function Optimization

Abdelhameed Ibrahim, Hesham Arafat Ali
Computer Engineering and Control Systems Department
Faculty of Engineering
Mansoura University, Mansoura, 35516, Egypt
afai79@mans.edu.eg, h_arafat_ali@mans.edu.eg

Marwa M. Eid, El-Sayed M. El-kenawy
Department of Communications and Electronics
Delta Higher Institute of Engineering and Technology
Mansoura, 35111, Egypt
marwa.3eeed@gmail.com, skenawy@ieee.org

2020 16th International Computer Engineering Conference (ICENCO) | 978-1-7281-6392-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICENCO49778.2020.9357403

***Abstract*—Swarm-based techniques, a form of meta-heuristic techniques, are derived from the swarm system's social conduct in nature. A newly brought optimization algorithm is Harris Hawks Optimization (HHO) that is stimulated through looking conduct of Harris Hawks (agents) of finding food (optimal solution). Balancing between exploitative and exploratory processes of the original HHO algorithm is desirable for achieving better performance to many optimization problems. An algorithm, Chaotic Harris Hawks Optimization (CHHO), is proposed in this work for the unconstrained function optimization. The CHHO algorithm is presented based on adjusting the exploration mechanism of the original HHO algorithm. Ten chaotic maps are used to control this mechanism instead of random adjusting. The unimodal, multimodal, and multimodal based fixed-dimension benchmark functions are used to compare the CHHO algorithm with the HHO algorithm. Visibility of the CHHO algorithm for optimizing the unconstrained benchmark function, with the intelligible effect of the Gauss and Logistic chaotic maps over other maps, is shown in the experiments.**

***Index Terms*—Meta-heuristic, Chaotic maps, Harris hawks optimization, Benchmark functions**

I. Introduction

Real-global optimization issues are complicated with excessive dimensional seek areas, thus, they may be in well-known challenging. Some of the applications of hunting mechanisms of the optimization algorithms can include the following, constrained and unconstrained engineering problems [7], machine learning applications [4], enterprise processes [9], mechanics [6], digital marketing [2], transportation [17], smart decision making [5], image processing applications [12]–[14], [16], and estimation [21]. Optimization refers to locating a suitable most prominent answer for selected trouble amongst many viable ones [20]. An optimization mission is generally converted right into a seek trouble in a multi-dimensional area, which can minimize or maximize an objective function include that assesses the great of a reply nominee, this is generally denoted via way of means of a vector with inside the seek area [8].

Meta-heuristic optimization algorithms are emerged to be famous due to the versatility and simplicity of their simulation process. These optimization algorithms have better performance than the classical and precise optimization techniques, such as Local and Greedy Search, in most of the applications and the investigated problems. The key point behind a set of meta-heuristic rules rely upon its cap potential to make the perfect stability between the exploration of the search space and the exploitation of the optimal solutions it can find [19]. One of the most famous evolutionary set of rules, which is stimulated via way of means of the technique and mechanism of organic evolution, is the Genetic Algorithm (GA) [5]. The GA algorithm is stimulated through the system of herbal selection. To perform the required optimization task in the GA algorithm, a set of genetic operators, including selection, crossover, and mutation, are employed to conform to a preliminary random population.

A form of meta-heuristic algorithms is the swarm-based algorithms that are derived from the swarm system's social conduct in nature. The most applied and famous swarm-primarily based set of rules is the Particle Swarm Optimization (PSO) algorithm [15]. Another critical instance of swarm-primarily based optimizers is the Grey Wolf Optimizer (GWO) which is primarily based on the searching conduct and social hierarchy of gray wolves [3]. The newly brought optimization algorithms are the Salp Swarm Algorithm (SSA) [11], [14], the algorithm of Harris Hawks Optimization (HHO) [10]. The HHO algorithm is stimulated through the looking conduct of Harris Hawks. The exploration segment simulates the manner hawks (search agents) that look for food through perching in random locations and how to be ready to discover prey (best or optimal solution of the investigated problem). The basic algorithm of HHO can achieve the best solution using three main processes of exploration, changing between exploitation and exploration, and exploitation [10], [18].

In this paper, an algorithm, named Chaotic Harris Hawks Optimization (CHHO), is proposed for the unconstrained function optimization. The CHHO algorithm adjusts the behavior of the original HHO algorithm by controlling its exploration mechanism. Ten different chaotic maps are employed for adjusting the HHO exploration mechanism. Tests are planned to compare the proposed CHHO and the first HHO algorithms utilizing twenty-three benchmark functions, named unimodal, multimodal, and multimodal based fixed-dimension. This paper clarifies the initial algorithm of HHO in Section II and the algorithm of CHHO is talked about in Section III. Tests and comparisons have appeared in Section IV, and at last, the conclusion is displayed in Section V.

II. Harris hawks optimization

The exploitative and exploratory process of the algorithm of HHO which is inspired by the Harris hawks attitude in the attacking process is discussed here [10]. The HHO algorithm can be applied to various optimization problems based on the problem of the proper formulation. Figure 1 shows different phases of HHO that will be described in detail in the following.

1) Phase of Exploration: The mechanism of the HHO algorithm exploration process is described in this part. Hawks, in the HHO algorithm, are considered as solutions and the prey is considered as the best solution. The hawks are perch randomly and start to discover the prey (best solution) based on different strategies. Let q parameter is changing randomly within $[0, 1]$ and is updated. Updating the Hawks' positions is simulated for $q \geq 0.5$ or based on the positions of prey for $q < 0.5$ by this equation

$$G(t+1) = \begin{cases} G_{rand}(t) - r_1 \left| G_{rand}(t) - 2r_2 G(t) \right| & q \geq 0.5 \\ \quad (G_{rabbit}(t) - G_m(t)) - & \\ \quad r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{1}$$

where r_1, r_2, r_3, r_4 have random values in [0,1], LB and UB show variables lower and upper bounds, $G(t+1)$ is the updated hawk's position, $G_{rabbit}(t)$ is the prey' position, $G(t)$ is the hawks' position, $G_{rand}(t)$ can be selected randomly, and $G_m(t)$ is the hawks's average position which can be calculated as

$$G_m(t) = \frac{1}{N} \sum_{i=1}^{N} G_i(t) \tag{2}$$

where $G_i(t)$ indicates the hawk's location in iteration t with N hawks.

2) Changing between exploitation and exploration: This process is achieved in the HHO algorithm based on E parameter, which is the prey escaping energy. During the escaping process, E decreases and can be modeled as

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

where T is total iterations, and E_0 is the initial energy state and changes randomly in [-1, 1]. Exploration starts when $|E| \geq 1$, while exploitation starts when $|E| < 1$.

3) Phase of Exploitation: There are four possible strategies in this phase due to the prey escaping strategies and the hawks chasing behaviors. Let r be the chance of prey to successfully get-away (r <0.5) or not be able to get-away (r ≥0.5). To catch the prey, the hawks will carry out a soft or hard besiege which is depending on the prey's energy. The HHO algorithm is able to switch between the soft and hard besiege by initializing the E parameter. Soft besiege can be activated if $|E| \geq 0.5$, and hard besiege if $|E| < 0.5$.

With $|E| \geq 0.5$ and $r \geq 0.5$, the prey still has sufficient energy and is trying to get-away but it cannot. The hawks perform the surprise pounce after softly encircle the prey. This can be expressed by these equations

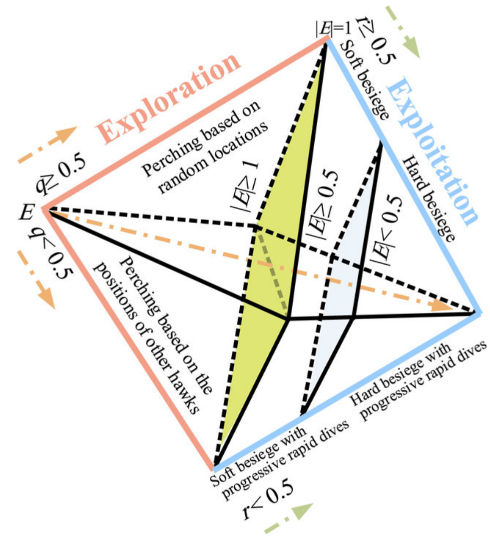

Fig. 1: Algorithm of HHO processes [10].

$$G(t+1) = \Delta G(t) - E \left| JG_{rabbit}(t) - G(t) \right| \tag{4}$$

$$\Delta G(t) = G_{rabbit}(t) - G(t) \tag{5}$$

where $\Delta G(t)$ represents the variation of current location and the prey's position. $J = 2(1 - r_5)$ is the strength of the prey's random jump during get-away process with r_5 is a random number in $[0, 1]$.

With $|E| < 0.5$ and $r \geq 0.5$, the prey's energy is very low to allow it to get-away. To perform surprise pounce, the hawks hardly encircle the prey. In this case, positions are changed as

$$G(t+1) = G_{rabbit}(t) - E \left| \Delta G(t) \right| \tag{6}$$

For $r < 0.5$ and $|E| \geq 0.5$, the prey has more energy that can help it to get-away and before the hawks surprise it, a soft besiege can then be constructed. The strategy for changing the hawks' positions in this process is expressed as

$$G(t+1) = \begin{cases} X & if F(X) < F(G(t)) \\ Y & if F(Y) < F(G(t)) \end{cases} \tag{7}$$

where X and Y can be calculated as in the following equations

$$X = G_{rabbit}(t) - E \left| JG_{rabbit}(t) - G(t) \right| \tag{8}$$

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

where Y parameter will be calculated using the LF (levy flight)-based patterns. Vector S is random with $1 \times D$ size for a dimension D. LF function can be expressed as

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

where $\beta = 1.5$ as a fixed value, and u, v are changed randomly in $[0, 1]$.

For $r < 0.5$ and $|E| < 0.5$, the prey's energy is not enough to allow it to get-away and before the hawks surprise pounce, a

Algorithm 1 : Basic HHO algorithm [10]

Initialization agents $G_i(i = 1, 2, 3 \ldots, N)$, size N, total iterations t_m, fitness function F, and $t = 1$
while $t \leq t_m$ **do**
  Calculating Hawks' fitness values F_N
  Setting G_{rabbit} as optimal solution
  for (each agent (G_i)) **do**
    Updating J and E_0 as $J = 2(1 - rand())$ and $E_0 = 2rand() - 1$
    Updating E by Eq. (3)
    if $(|E| \geq 1)$ **then**
      Updating the new positions by Eq. (1)
    end if
    if $(|E| < 1)$ **then**
      if $(|E| \geq 0.5\ r \geq 0.5)$ **then**
        Updating the new positions by Eq. (4)
      else if $(|E| < 0.5$ and $r \geq 0.5)$ **then**
        Updating the new positions by Eq. (6)
      else if $(|E| \geq 0.5$ and $r < 0.5)$ **then**
        Updating the new positions by Eq. (7)
      else if $(|E| < 0.5$ and $r < 0.5)$ **then**
        Updating the new positions by Eq. (11)
      end if
    end if
  end for
  Set t = t + 1
end while
Return G_{rabbit}

hard besiege can be structured in this case. The hawks reduce the location distance of reaching the best solution based on this equation

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

where X and Y can be calculated as follows

$$X = G_{rabbit}(t) - E\,|JG_{rabbit}(t) - G_m(t)| \tag{12}$$

$$Y = X + S \times LF(D) \tag{13}$$

where $G_m(t)$ is calculated by Eq. (2). Based on the LF-based patterns location footprints, X or Y will be assigned as the next position. The HHO algorithm pseudo-code is demonstrated in Algorithm (1).

III. Proposed Chaotic Harris Hawks Optimization

The balancing between the exploitative and exploratory phases of the original HHO algorithm is desirable for achieving better performance to many optimization problems. The HHO algorithm achieves the best solution using three processes of exploration, changing between exploration and exploitation, and exploitation. The proposed CHHO algorithm, discussed in this section, is designed based on adjusting the exploration mechanism of the original HHO algorithm using

TABLE I: Mathematical Forms of Chaotic Maps

No.	Map	Mathematical Form	Range
CHHO1	Chebyshev	$o_{k+1} = cos(scos^{-1}(o_k))$	[-1,1]
CHHO2	Circle	$o_{k+1} = mod(o_k + d - (\frac{c}{2\pi})sin(2\pi o_k), 1), c = 0.5, d = 0.2$	[0,1]
CHHO3	Gauss	$o_{k+1} = \begin{cases} 1, & o_k = 0, \\ \frac{1}{mod(o_k,1)}, & otherwise. \end{cases}$	[0,1]
CHHO4	Iterative	$o_{k+1} = sin(\frac{c\pi}{o_k}), c = 0.7$	[-1,1]
CHHO5	Logistic	$o_{k+1} = co_k(1 - o_k), c = 4$	[0,1]
CHHO6	Piecewise	$o_{k+1} = \begin{cases} \frac{o_k}{l}, & 0 \leq o_k < l, \\ \frac{o_k - l}{0.5 - l}, & l \leq o_k < 0.5, \\ \frac{1 - l - o_k}{0.5 - l}, & 0.5 \leq o_k < 1 - l, \\ \frac{1 - o_k}{l}, & 1 - l \leq o_k < 1, \end{cases} , l = 0.4$	[0,1]
CHHO7	Sine	$o_{k+1} = \frac{c}{\pi} sin(\pi o_k), c = 4$	[0,1]
CHHO8	Singer	$o_{k+1} = c(7.86o_k - 23.3o_k^2 + 28.75o_k^3 - 13.302875o_k^4), c = 1.07$	[0,1]
CHHO9	Sinusoidal	$o_{k+1} = co_k^2 sin(\pi o_k), c = 2.3$	[0,1]
CHHO10	Tent	$o_{k+1} = \begin{cases} \frac{o_k}{0.7}, & o_k < 0.7, \\ \frac{10}{3}(1 - o_k), & otherwise. \end{cases}$	[0,1]

different chaotic maps. Ten chaotic maps, named Chebyshev (CHHO1), Circle (CHHO2), Gauss (CHHO3), Iterative (CHHO4), Logistic (CHHO5), Piecewise (CHHO6), Sine (CHHO7), Singer (CHHO8), Sinusoidal (CHHO9), and Tent (CHHO10), are employed in CHHO to control this exploration mechanism instead of using random adjusting in HHO. Forms of the chaotic maps are shown mathematically in Table (I).

Mathematically, the hawks within the CHHO algorithm are considered the solutions, and the leading solution is considered as the prey. The hawks are perch to detect the best solution. Since the q parameter has an equal chance for these strategies, it can be adjusted by the chaotic maps as follows

$$q = o_j, j = 1, 2, \ldots, 10 \tag{14}$$

where o_j is one of chaotic maps from the ten maps in Table (I). Then, the hawks updating positions are modeled as in Eq. (15) for based on the q parameter as follows

$$G(t+1) = \begin{cases} G_{rand}(t) - r_1\,|G_{rand}(t) - 2r_2G(t)| & q \geq 0.5 \\ (G_{rabbit}(t) - G_m(t)) - \\ r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{15}$$

where r_1, r_2, r_3, r_4 are changed randomly in $[0, 1]$, q is adjusted by Eq. (14) for different chaotic maps, $G(t+1)$ is the new hawk's position, $G_{rabbit}(t)$ is the prey's position, $G(t)$ is the hawks' position, $G_{rand}(t)$ can be selected randomly, and G_m is the hawks's average position using Eq. (2):

The changing process between exploitation and exploration, in addition to, the exploitation process are applied in the CHHO algorithm in the same way as in the original HHO algorithm. The proposed CHHO algorithm pseudo-code is demonstrated in Algorithm (2). In the proposed CHHO algorithm, the chaotic steps (from step 9 to step 11) are inserted to adjust the exploration mechanism of the original HHO algorithm by assigning q to a chaotic map as shown in Eq. (14) and then updating the hawks' positions, based on the selected chaotic map, as in Eq. (15). By analyzing the CHHO algorithm, computational complexity can be expressed as O $(t_m \times N)$ and can be O $(t_m \times N \times d)$ for d dimension problems.

IV. Experimental Results

The proposed CHHO algorithm performance is evaluated in this section. The suggested chaotic algorithm (CHHO) is

Algorithm 2 : Proposed CHHO algorithm

Initialization agents $G_i (i = 1, 2, 3 \ldots, N)$, size N, total iterations t_m, fitness function F, and $t = 1$
while $t \leq t_m$ **do**
  Calculating Hawks' fitness values F_N
  Setting G_{rabbit} as optimal solution
  for (each agent (G_i)) **do**
    Updating J and E_0 as
    $J = 2(1 - rand())$ and $E_0 = 2rand() - 1$
    Updating E by Eq. (3)
    if $(|E| \geq 1)$ **then**
      Getting value of chaotic map o_j
      Updating q using chaotic map o_j using Eq. (14)
      Updating the new positions by Eq. (15)
    end if
    if $(|E| < 1)$ **then**
      if $(|E| \geq 0.5$ $r \geq 0.5)$ **then**
        Updating the new positions by Eq. (4)
      else if $(|E| < 0.5$ and $r \geq 0.5)$ **then**
        Updating the new positions by Eq. (6)
      else if $(|E| \geq 0.5$ and $r < 0.5)$ **then**
        Updating the new positions by Eq. (7)
      else if $(|E| < 0.5$ and $r < 0.5)$ **then**
        Updating the new positions by Eq. (11)
      end if
    end if
  end for
  Set t = t + 1
end while
Return G_{rabbit}

compared to the basic HHO algorithm [10] for optimizing unconstrained functions. Table II shows the configuration of both algorithms for a fair comparison. This table presents the values for different parameters of the optimization algorithms and shows that the q parameter is assigned differently in the compared algorithms. CHHO adjusts the q parameter using different chaotic maps, while HHO sets it randomly in $[0, 1]$. Other parameter settings are set to be the same. Commonly used benchmark functions [1] are tested in the experiment. Table III describes the seven unimodal functions. The description of six multimodal functions is shown in Table III, while Table V describe ten multimodal based fixed-dimension functions.

The convergence curves of the proposed CHHO algorithm, based on ten Chaotic Maps, and the original HHO algorithm using the benchmark functions of unimodal (f_1-f_7), multimodal (f_8-f_{13}), and multimodal(f_{14}-f_{23}) with fixed-dimension are shown in Figure 2. For the unimodal functions, the convergence curves of f_1, f_2, f_3, and f_4 have fast convergence which improves the algorithm behavior by the **CHHO3** (Gauss map). The convergence curves of functions f_5, f_6, and f_7 have better performance based on the **CHHO5** (Logistic map). For the multimodal functions (f_8-f_{23}), different chaotic maps show better convergence than the original algorithm for

TABLE II: Configuration of CHHO and HHO algorithms

Parameter	Value
No. agents	10
No. iterations	100
Test functions	F1 - F23
Dimension	According to the function
No. repetitions of runs	10
q parameter	in HHO, Random [0,1] in CHHO from Eq. 14
E parameter	from Eq. 3
E_0 parameter	Random [-1,1]
r parameter	Random [0,1]
r_1, r_2, r_3, r_4, and r_5 parameters	Random [0,1]
Variables lower bound	According to the function
Variables upper bound	According to the function

TABLE III: Description of unimodal benchmark functions

Function	D	Range
$f_1(w) = \sum_{i=1}^{n} w^2$	30	[-100, 100]
$f_2(w) = \sum_{i=1}^{n} \lvert w_i \rvert + \prod_{i=1}^{n} \lvert w_i \rvert$	30	[-10, 10]
$f_3(w) = \sum_{i=1}^{n} (\sum_{j=1}^{i} w_i)^2$	30	[-100, 100]
$f_4(w) = max_i\{\lvert w_i \rvert, 1 \leq i \leq D\}$	30	[-100, 100]
$f_5(w) = \sum_{i=1}^{D-1} [100(w_{i+1} - w_i^2)^2 - (w_i - 1)^2]$	30	[-30, 30]
$f_6(w) = \sum_{i=1}^{D} (w_i + 0.5)^2$	30	[-100, 100]
$f_7(w) = \sum_{i=1}^{D} i w_i^4 + rand[0, 1]$	30	[-1.28, 1.28]

different functions. In most functions, **CHHO3** and **CHHO5** maps achieve fast convergence than other chaotic maps. In summary, the results of the comparison based on the twenty-three unimodal and multimodal reference functions show the visibility of the proposed CHHO algorithm for the unconstrained function optimization with an obvious effect of Gauss and Logistic chaotic maps.

V. Conclusion

A Chaotic Harris Hawks Optimization (CHHO) algorithm for the unconstrained function optimization is proposed in this paper. The CHHO algorithm is developed based on adjusting the exploration mechanism of the original HHO algorithm. Ten chaotic maps are used to adjust this mechanism instead of random initializing. The proposed CHHO algorithm efficiency is tested, in the experiments, and compared with the basic HHO algorithm. The comparison is based on twenty-three unimodal and multimodal benchmark functions. The experiments confirm the quality of the proposed CHHO algorithm for the unconstrained benchmark functions. One of the main future directions of this work is to apply the CHHO algorithm for the segmentation and classification problems with different types of images, including breast cancer, X-ray chest images, and CT scans. Another future direction is to test a binary CHHO conversion algorithm for the problem of feature selection with a variety of datasets.

References

[1] Sankalap Arora, Harpreet Singh, Manik Sharma, Sanjeev Sharma, and Priyanka Anand. A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access*, 7:26343–26361, 2019.

[2] Yongyang Cheng, Shuai Zhao, Bo Cheng, Shoulu Hou, Yulong Shi, and Junliang Chen. Modeling and optimization for collaborative business process towards IoT applications. *Mobile Information Systems*, 2018:1–17, September 2018.

TABLE IV: Description of multimodal benchmark functions

Function	D	Range	f_{min}
$f_{08}(w) = \sum_{i=1}^{D} -w_i \sin\ (\sqrt{\lvert w_i \rvert})$	30	[-500, 500]	-12569.487
$f_{09}(w) = \sum_{i=1}^{D} [w_i^2 - 10\cos(2\pi w_i) + 10]$	30	[-5.12, 5.12]	0
$f_{10}(w) = -20\exp(-0.2\sqrt{\sum_{i=1}^{D} w_i^2}) - \exp(\frac{1}{d}\sum_{i=1}^{D} \cos(2\pi w_i)) + 20 + \eta$	30	[-32, 32]	0
$f_{11}(w) = \frac{1}{4000}\sum_{i=1}^{D} w_i^2 - \prod_{i=1}^{D} \cos(\frac{w_i}{\sqrt{i}}) + 1$	30	[-600, 600]	0
$f_{12}(w) = \frac{\pi}{D}\{10\sin^2(\pi y_i) + \sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1) + (yD-1)^2$ $+ \sum_{i=1}^{D} u(w_i, 10, 100, 4)]\}$ $y_i = 1 + \frac{w_i+1}{4}, \quad u(w_i,h,k,m) = \begin{cases} k(w_i-h)^m & w_i > h \\ 0 & -h < w_i < h \\ k(-w_i-h)^m & w_i < -h \end{cases}$	30	[-50, 50]	0
$f_{13}(w) = 0.1\{10\sin^2(3\pi y_i) + \sum_{i=1}^{D-1}(w_i-1)^2[1+10\sin^2(3\pi y_i+1)]$ $+ (w_n-1)^2[1+\sin^2(2\pi w_n)]\} + \sum_{i=1}^{n} u(w_i, 5, 100, 4)$	30	[-50, 50]	0

TABLE V: Description of multimodal based fixed-dimension benchmark functions

Function	D	Range	f_{min}
$f_{14}(w) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(w_i-h_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
$f_{15}(w) = \sum_{i=1}^{11}\left[h_i - \frac{w_1(b_i^2+b_i w_2)}{b_i^2+b_i w_3+w_4}\right]^2$	4	[-5, 5]	0.00030
$f_{16}(w) = 4w_1^2 - 2.1w_1^4 + \frac{1}{3}w_1^6 + w_1 w_2 - 4w_2^2 + 4w_2^4$	2	[-5, 5]	-1.0316
$f_{17}(w) = \left(w_2 - \frac{5.1}{4\pi^2}w_1^2 + \frac{5}{\pi}w_1 + -6\right)^2 + 10\left(1-\frac{1}{8\pi}\right)\cos w_1 + 10$	2	[-5, 5]	0.398
$f_{18}(w) = [1 + (w_1+w_2+1)^2(19 - 14w_1 + 3w_1^2 - 14w_2 + 6w_1w_2 + 3w_2^2)]$ $\times [30 + (2w_1 - 3w_2)^2 w(18 - 32w_1 + 12w_1^2 + 48w_2 - 36w_1w_2 + 27w_2^2)]$	2	[-2, 2]	3
$f_{19}(w) = -\sum_{i=1}^{4} b_i \exp(-\sum_{j=1}^{3} h_{ij}(w_j - p_{ij})^2)$	3	[1, 3]	-3.86
$f_{20}(w) = -\sum_{i=1}^{4} b_i \exp(-\sum_{j=1}^{6} h_{ij}(w_j - p_{ij})^2)$	6	[0, 1]	-3.32
$f_{21}(w) = -\sum_{i=1}^{5}\left[(w-h_i)(w-h_i)^T + b_i\right]^{-1}$	4	[0, 10]	-10.1532
$f_{22}(w) = -\sum_{i=1}^{7}\left[(w-h_i)(w-h_i)^T + b_i\right]^{-1}$	4	[0, 10]	-10.4028
$f_{23}(w) = -\sum_{i=1}^{10}\left[(w-h_i)(w-h_i)^T + b_i\right]^{-1}$	4	[0, 10]	-10.5363

[3] El-Sayed El-Kenawy and Marwa Eid. Hybrid gray wolf and particle swarm optimization for feature selection. *International Journal of Innovative Computing, Information and Control*, 16(3):831–844, 2020.

[4] El-Sayed M. El-Kenawy, Marwa Metwally Eid, Mohamed Saber, and Abdelhameed Ibrahim. MbGWO-SFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection. *IEEE Access*, 8(1):107635–107649, 2020.

[5] El-Sayed M. El-kenawy, Abdelhameed Ibrahim, Seyedali Mirjalili, Marwa M. Eid, and Sherif E. Hussein. Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images. *IEEE Access*, 8:179317 – 179335, 2020.

[6] E. S. M. T. El-Knawy and A. I. El-Desoky. Trust measurement for online auctions: proposal of new model. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 12(2):385–394, 2016.

[7] Mohamad M. Fouad, Ali Ibrahim El-Desouky, Rami Al-Hajj, and El-Sayed M. El-Kenawy. Dynamic group-based cooperative optimization algorithm. *IEEE Access*, 8:148378–148403, 2020.

[8] E. M. Hassib, A. I. El-Desouky, E. M. El-Kenawy, and S. M. El-Ghamrawy. An imbalanced big data mining framework for improving optimization algorithms performance. *IEEE Access*, 7:170774–170795, 2019.

[9] Eslam. M. Hassib, Ali. I. El-Desouky, Labib. M. Labib, and El-Sayed M. El-kenawy. Woa+brnn: An imbalanced big data classification framework using whale optimization and deep neural network. *Soft Computing*, 24(8):5573–5592, March 2019.

[10] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849–872, August 2019.

[11] Abdelhameed Ibrahim, Ali Ahmed, Sherif Hussein, and Aboul Ella Hassanien. Fish image segmentation using salp swarm algorithm. In Aboul Ella Hassanien, Mohamed F. Tolba, Mohamed Elhoseny, and Mohamed Mostafa, editors, *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 42–51, Cham, 2018. Springer International Publishing.

[12] Abdelhameed Ibrahim and El-Sayed M. El kenawy (2020). Applications and datasets for superpixel techniques: A survey. *Journal of Computer Science and Information Systems*, 15 (6 October 2020).

[13] Abdelhameed Ibrahim and El-Sayed M. El kenawy (2020). Image segmentation methods based on superpixel techniques: A survey. *Journal of Computer Science and Information Systems*, 15 (6 October 2020).

[14] Abdelhameed Ibrahim, Shaimaa Mohammed, Hesham Arafat Ali, and Sherif E. Hussein. Breast cancer segmentation from thermal images based on chaotic salp swarm algorithm. *IEEE Access*, 8(1):122121–122134, 2020.

[15] Abdelhameed Ibrahim, Mostafa Noshy, Hesham Arafat Ali, and Mahmoud Badawy. PAPSO: A power-aware VM placement technique based on particle swarm optimization. *IEEE Access*, 8(1):81747–81764, 2020.

[16] Abdelhameed Ibrahim, Alaa Tharwat, Tarek Gaber, and Aboul Ella Hassanien. Optimized superpixel and AdaBoost classifier for human thermal face recognition. *Signal, Image and Video Processing*, 12(4):711–719, November 2017.

[17] Yang Li, Jinlong Wang, Dongbo Zhao, Guoqing Li, and Chen Chen. A two-stage approach for combined heat and power economic emission dispatch: Combining multi-objective optimization with integrated decision making. *Energy*, 162:237–254, November 2018.

[18] Ahmed S. Menesy, Hamdy M. Sultan, Ali Selim, Mohamed G. Ashmawy, and Salah Kamel. Developing and applying chaotic harris hawks optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks. *IEEE Access*, 8:1146–1159, 2020.

[19] Seyedali Mirjalili, Jin Song Dong, Andrew Lewis, and Ali Safa Sadiq. Particle swarm optimization: Theory, literature review, and application in airfoil design. In *Nature-Inspired Optimizers*, pages 167–184. Springer International Publishing, February 2019.

[20] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51 – 67, 2016.

[21] El-Sayed M.T.El-kenawy. Solar radiation machine learning production depend on training neural networks with ant colony optimization algorithms. *International Journal of Advanced Research in Computer and Communication Engineering(IJARCCE)*, 7(5):1–4, 2018.

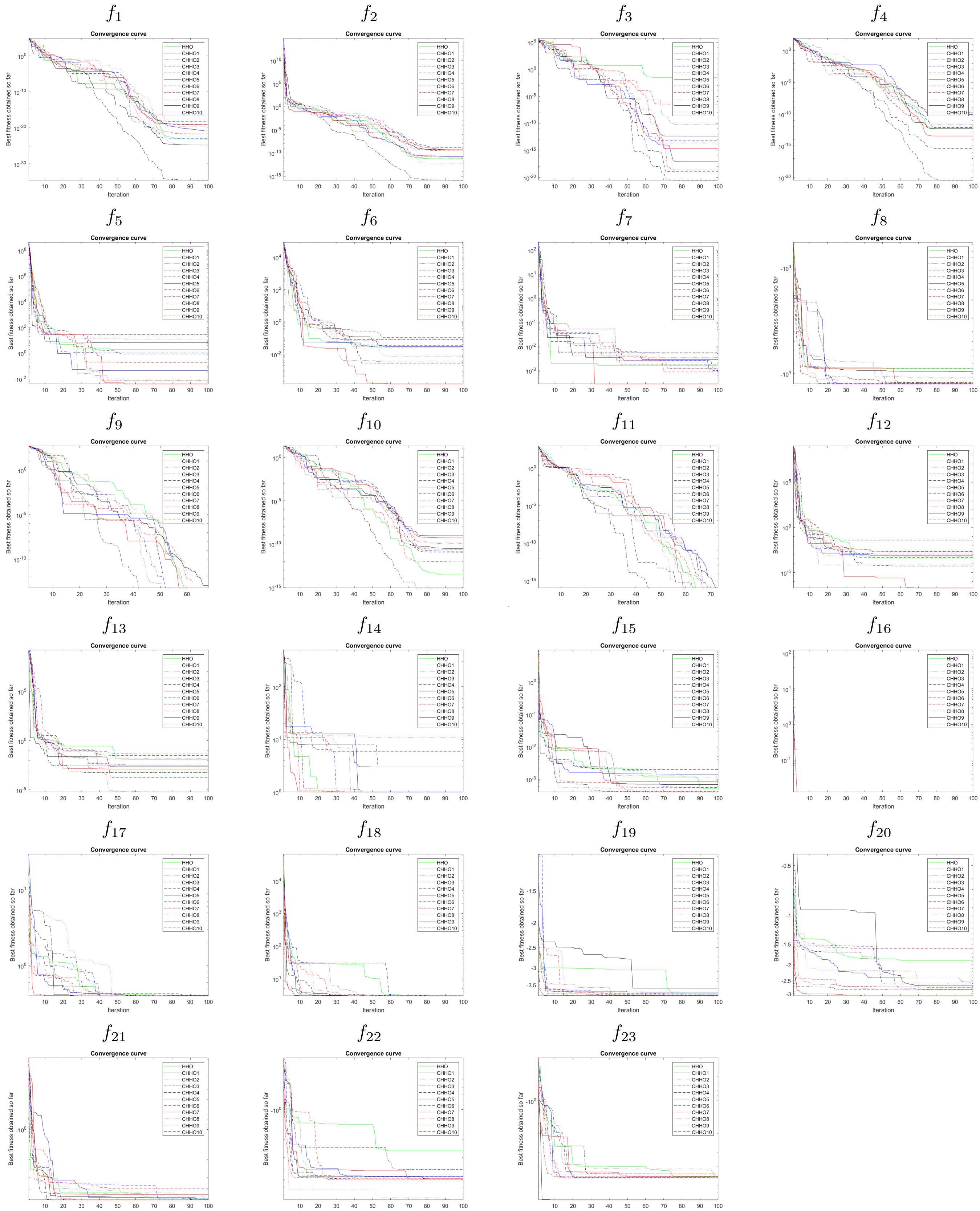

Fig. 2: Convergence curves of the proposed CHHO, based on chaotic maps, and the original HHO algorithms based on the benchmark functions of unimodal (f_1-f_7), multimodal (f_8-f_{13}), and multimodal based fixed-dimension (f_{14}-f_{23}) .