

Article

# Harris Hawks Optimization with Multi-Strategy Search and Application

Shangbin Jiao <sup>1,2</sup>, Chen Wang <sup>1,\*</sup>, Rui Gao <sup>1,2,3</sup>, Yuxing Li <sup>1</sup>  and Qing Zhang <sup>1</sup>

<sup>1</sup> School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China; jiaoshangbin@xaut.edu.cn (S.J.); gaorui@bjw1xy.edu.cn (R.G.); liyuxing@xaut.edu.cn (Y.L.); zhangqing@xaut.edu.cn (Q.Z.)

<sup>2</sup> Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an 710048, China

<sup>3</sup> School of Electronic & Electrical Engineering, Baoji University of Arts and Sciences, Baoji 721016, China

\* Correspondence: 1180311029@stu.xaut.edu.cn

**Abstract:** The probability of the basic HHO algorithm in choosing different search methods is symmetric: about 0.5 in the interval from 0 to 1. The optimal solution from the previous iteration of the algorithm affects the current solution, the search for prey in a linear way led to a single search result, and the overall number of updates of the optimal position was low. These factors limit Harris Hawks optimization algorithm. For example, an ease of falling into a local optimum and the efficiency of convergence is low. Inspired by the prey hunting behavior of Harris's hawk, a multi-strategy search Harris Hawks optimization algorithm is proposed, and the least squares support vector machine (LSSVM) optimized by the proposed algorithm was used to model the reactive power output of the synchronous condenser. Firstly, we select the best Gauss chaotic mapping method from seven commonly used chaotic mapping population initialization methods to improve the accuracy. Secondly, the optimal neighborhood perturbation mechanism is introduced to avoid premature maturity of the algorithm. Simultaneously, the adaptive weight and variable spiral search strategy are designed to simulate the prey hunting behavior of Harris hawk to improve the convergence speed of the improved algorithm and enhance the global search ability of the improved algorithm. A numerical experiment is tested with the classical 23 test functions and the CEC2017 test function set. The results show that the proposed algorithm outperforms the Harris Hawks optimization algorithm and other intelligent optimization algorithms in terms of convergence speed, solution accuracy and robustness, and the model of synchronous condenser reactive power output established by the improved algorithm optimized LSSVM has good accuracy and generalization ability.

**Keywords:** Harris Hawks optimization algorithm; chaotic mapping; multi-strategy strategy; least squares support vector machine; synchronous condenser



**Citation:** Jiao, S.; Wang, C.; Gao, R.; Li, Y.; Zhang, Q. Harris Hawks Optimization with Multi-Strategy Search and Application. *Symmetry* **2021**, *13*, 2364. <https://doi.org/10.3390/sym13122364>

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 5 November 2021

Accepted: 30 November 2021

Published: 8 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Along with the significant increase in the processing power of computer hardware and software, a large number of excellent meta-heuristics were created in the intelligent computing field [1–5]. Meta-heuristics are a large class of algorithms developed in contrast to optimization and heuristics. Optimization algorithms are dedicated to finding the optimal solution to a problem, but they are often difficult to implement due to the unresolvability of the problem [6,7]. Heuristic algorithms are dedicated to customizing algorithms through intuitive experience and problem information, but are often difficult to generalize due to their specialized nature. Compared to these two algorithms, meta-heuristic algorithms are more general and do not require deep adaptation to the problem, and although they do not guarantee optimal solutions, they can generally obtain optimal solutions under acceptable spatial and temporal conditions, although the degree of deviation from the optimal solution is difficult to estimate [8–12].

The main optimization strategies of meta-heuristic algorithms are summarized as follows: (1) diversification of exploration in a wide range to ensure the global optimal solution; and (2) intensification and exploitation in a local range to obtain an optimal solution as close to the optimal solution as possible [13]. The main difference between various meta-heuristic algorithms is how to strike a balance between the two. Almost all meta-heuristic algorithms have the following characteristics: (1) they are inspired by some phenomena in nature, such as simulated physics, biology, and biological behavior; (2) they use stochastic strategies; (3) they do not use the gradient resolution information of the objective function; and (4) they have several parameters that need to be adapted to the problem, and (5) they have good parallel and autonomous exploration. Meta-heuristic algorithms have been widely used in all aspects of social production and life. Many related research papers are published every year in the fields of production scheduling [14,15], engineering computing [16,17], management decision-making [18,19], machine learning (ML) [20,21], system control [22], and many other disciplines.

P-meta-heuristics are categorized into four main groups [23,24]: (1) simulated physical process algorithm, (2) evolutionary algorithm, (3) simulated swarm intelligence algorithm, and (4) human behavior [25–27]. Algorithms for simulating physical processes include simulated annealing (SA) [28], gravitational search algorithm [29] which simulates earth gravity, artificial chemical reaction optimization algorithm [30], heat transfer search, which simulates the heat transfer search process in thermodynamics [31], Gases Brownian motion optimization, which simulates the phenomenon of Brownian motion in physics [32], Henry gas solubility optimization, which simulates Henry gas solubility process [33]; and evolutionary algorithm. In 1975, American professor Holland proposed the genetic algorithm (GA) based on the Darwinian evolutionary theory and the mechanism of superiority and inferiority in nature. GA [34], evolution strategies [35] and differential evolution [36], genetic programming [37], and Biogeography-Based Optimizer [38]; simulated population intelligence algorithms: the Artificial Bee Colony (ABC) algorithm [39] based on the honey bee harvesting mechanisms, Firefly Algorithm based on the flickering behavior of fireflies [40], Beetle Antennae Search algorithm based on the foraging principle of aspen bark beetles [41], Grey Wolf Optimization (GWO) algorithm inspired by the hierarchy and predatory behavior of gray wolf packs [42], and Virus Colony Search algorithm [43], which is based on the proliferation and infection strategies of viruses to survive and reproduce in the cellular environment through host cells. Simulation of human behavior: Tabu Search [44], Socio Evolution and Learning Optimization [45], Teaching Learning Based Optimization [46], and Imperialist Competitive Algorithm [47].

Harris Hawks Optimization (HHO) [24] is a swarm intelligence optimization algorithm proposed by Heidari et al. in 2019 to simulate the prey hunting process of Harris's hawks in nature. The algorithm was inspired by the three phases of Harris's hawks' predatory behavior: search, search-exploitation conversion, and exploitation. The algorithm has a simple principle, fewer parameters, and better global search capability. Therefore, HHO has been applied in image segmentation [48], neural network training [49], motor control [50] and other fields. However, similar to other swarm intelligence optimization algorithms, HHO has the disadvantages of slow convergence speed, low optimization accuracy, and easily falls into local optimum when solving complex optimization problems. For example, the literature [51] used the information exchange mechanism to enhance the population diversity, thus improving the convergence speed of the HHO algorithm with information exchange (IEHHO) algorithm. The limitation of the IEHHO algorithm is how to set the parameters of the proposed algorithm. Zhang et al. [52] introduced an exponentially decreasing strategy to update the energy factor to increase the exploration and exploitation capability obtained by the relatively higher values of escaping energy; Elgamel et al. [53] made two improvements: (1) they applied chaotic mapping in the initialization phase of HHO; and (2) they used the SA algorithm as the current best solution to improve HHO exploitation; Shiming Song et al. [54] applied Gaussian mutation and a dimension decision strategy of the cuckoo search method into this algorithm to increase

the HHO's performance. The mechanism of cuckoo search was useful in improving the convergence speed of the search agents as well as sufficient excavation of the solutions in the search area, while the Gaussian mutation strategy performed well in increasing the accuracy and jumping of the local optimum.

However, according to the no free lunch theory [55], one meta-heuristics algorithm cannot always perform as the best on all operations. The original HHO method could not fully balance the exploration and exploitation phases, which resulted in insufficient global search capability and slow convergence of the HHO method. To alleviate these adverse effects, we propose an improved algorithm model called chaotic multi-strategy search HHO (CSHHO), which introduces chaotic mapping and global search strategy, to solve single-objective optimization problems efficiently.

Here, the initialization phase of HHO is replaced by chaotic mapping, which allows the population initialization phase to be evenly distributed in the upper and lower bounds to enhance the population diversity, simultaneously enabling the population to approach the prey location faster, which accelerates the convergence speed of the algorithm. The adaptive weights are added to the position update formula in the exploration phase of HHO to dynamically adjust the influence of the global optimal solution. In the update phase of the HHO, the optimal neighborhood perturbation strategy is introduced to prevent the algorithm from falling into the local optimal solution and to solve the premature aging phenomenon.

To verify the superior performance of the CSHHO algorithm, this experiment first tests the effect of common chaotic mappings of the HHO algorithm's performance. The selected chaotic mappings are Sinusoidal, Tent, Kent, Cubic, Logistic, Gauss, and Circle, and the experimental results will show that Gauss chaotic mapping improves the accuracy of the HHO algorithm to the greatest extent. Second, the HHO algorithm based on Gauss chaotic mapping with multi-strategy search is tested. Then, it is compared with other classic and state-of-the-art algorithms on 23 classic test functions and 30 IEEE CEC2017 competition functions to verify the significant superiority of the proposed paradigm over other algorithms by Friedman test and Bonferroni–Holm corrected Wilcoxon signed-rank test. Finally, CSHHO is applied to model the reactive power output problem of a synchronous condenser based on LSSVM. The complete results will show that the effectiveness of the proposed optimizer is better than other models in the experiment.

The remainder of this paper is organized as follows: Section 2 introduces the basic theory and structure of the original HHO algorithm. Section 3 introduces the chaotic operator and Global search strategy to integrate it into the original optimizer. Section 4 conducts a full range of experiments on the proposed method and demonstrates the experimental results. It further discusses the proposed method based on experimental results. Section 5 applies the proposed method to the LSSVM-based synchronous condenser reactive power output problem. Finally, Section 6 summarizes the study and proposes research ideas for the future.

## 2. Harris Hawks Optimization Algorithm

The HHO algorithm is a swarm intelligence optimization algorithm that is widely used in solving optimization problems. The main idea of the algorithm is derived from the cooperative behavior and chasing strategy of Harris's hawk when catching prey in nature [24]. In the process of prey capture, the HHO algorithm is divided into two segments according to the physical energy  $E$  of the prey at the time of escape: exploration and exploitation phases, as shown in Figure 1. During the exploration phase, Harris's hawks randomly select a perching location to observe and monitor their prey.

$$X'_i(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X_i(t)|, q \geq 0.5 \\ X_{rabbit}(t) - X_m(t) - r_3 [\mathbf{lb} + r_4(\mathbf{ub} - \mathbf{lb})], q < 0.5 \end{cases} \quad (1)$$

where  $X_{rabbit}(t)$  and  $X_{rand}(t)$  denote the position of the prey and the individual position at time  $t$ , respectively, and  $q$  is a random number between  $(0, 1)$ , the average individual position:

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

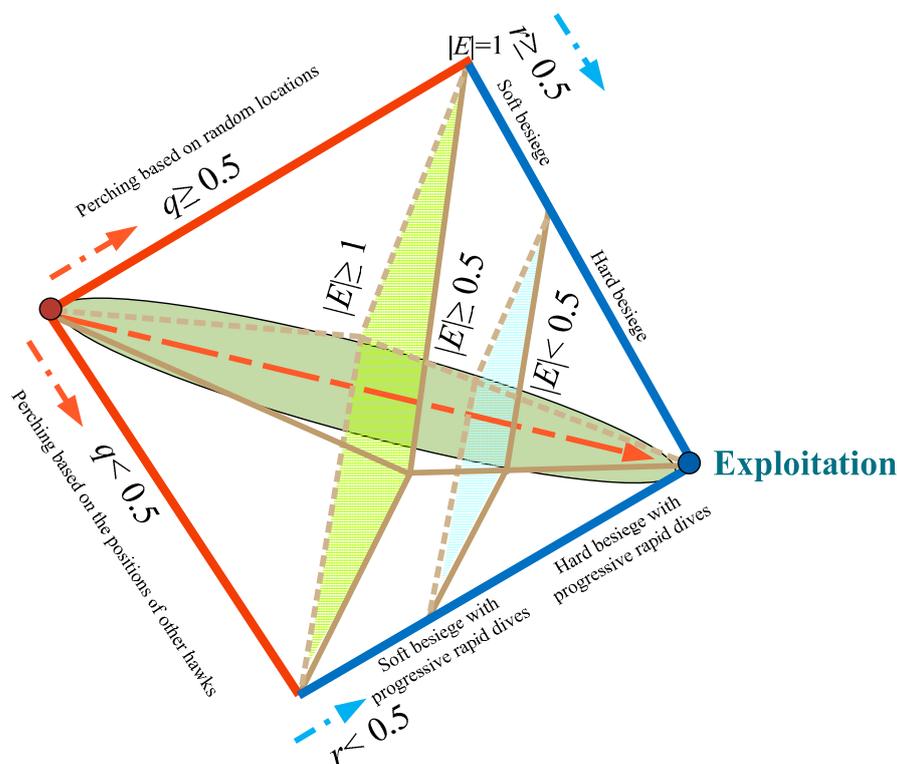


Figure 1. Description of each stage of HHO.

As the physical capacity of the prey decreases, the exploration phase changes to the exploitation phase, the prey’s physical energy factor  $E$  is as follows:

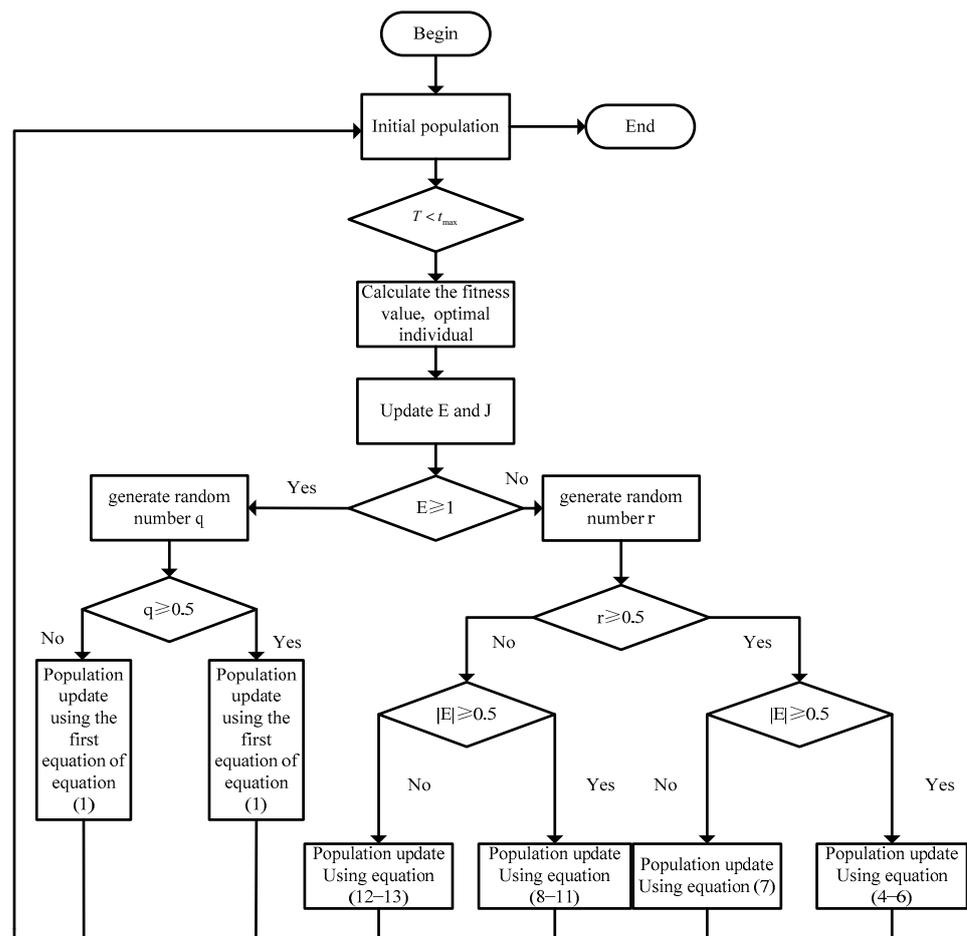
$$E = 2E_0 \left( 1 - \frac{t}{T} \right) \tag{3}$$

In the exploitation phase, the Harris’s hawk launches a surprise attack on the target prey found in the exploration phase, and the prey tries to escape when it encounters danger. Let the randomly generated prey escape probability be  $r$ , when  $r < 0.5$  the prey successfully escapes; when  $r > 0.5$  the prey does not successfully escape. According to the magnitude of  $r$  and  $|E|$ , four different location update strategies were proposed in the exploitation phase (see Table 1).

According to the position update condition in the HHO algorithm, the position of the Harris hawk is updated continuously, the fitness value was calculated according to the position of the Harris hawk, and if the fitness threshold was reached, the algorithm was finished. Otherwise, the algorithm continued to execute, and if the maximum number of iterations was reached, the algorithm was finished, and the optimal solution was obtained (See Figure 2).

**Table 1.** Exploitation phase of HHO algorithm.

Strategy	The value of $r$ and $ E $ $r \geq 0.5$ and $ E  \geq 0.5$	
Soft besiege	$X'_i(t+1) = \Delta X_i(t) - E(t)  J X_{rabbit}(t) - X_i(t) $	(4)
	$\Delta X_i(t) = X_{rabbit}(t) - X_i(t)$	(5)
	$J = 2(1 - r_5)$	(6)
Hard besiege	$r \geq 0.5$ and $ E  < 0.5$	
	$X'_i(t+1) = X_{rabbit}(t) - E(t)  \Delta X_i(t) $	(7)
Soft besiege with progressive rapid dives	$r < 0.5$ and $ E  \geq 0.5$	
	$X'_i(t+1) = \begin{cases} Y & f(Y) < f(X_i(t)) \\ Z & f(Z) < f(X_i(t)) \end{cases}$	(8)
	$Y = X_{rabbit}(t) - E(t)  J X_{rabbit}(t) - X_i(t) $	(9)
	$Z = Y + S \times LF(D)$	(10)
	$LF(\cdot) = 0.01 \times \frac{u}{ v ^{\frac{1}{\beta}}} \times \left( \frac{\Gamma(1+\beta) \times \sin(0.5\beta\pi)}{\Gamma(0.5(1+\beta)) \times \beta \times 2^{(\beta-1)/2}} \right)^{\frac{1}{\beta}}$	(11)
Hard besiege with progressive rapid dives	$r < 0.5$ and $ E  < 0.5$	
	$X'_i(t+1) = \begin{cases} Y & f(Y) < f(X_i(t)) \\ Z & f(Z) < f(X_i(t)) \end{cases}$	(12)
	$Y = X_{rabbit}(t) - E(t)  J X_{rabbit}(t) - X_m(t) $	(13)



**Figure 2.** Flow chart of HHO algorithm.

### 3. HHO Algorithm Based on Multi-Search Strategy

#### 3.1. Reasons for Improving the Basic HHO Algorithm

Harris’s hawks generally gather high in trees to hunt for prey. In the process of hunting for prey, they often hover in a spiral to capture prey; when approaching prey,

they rush towards their prey at a faster speed until the distance from the prey is small. They slow down and adjust their body posture to increase the probability of their catching prey [56–58]. This mechanism is important in the HHO algorithm. The exploration phase of the basic HHO algorithm uses Equations (1)–(3), where the optimal solution from the previous iteration of the algorithm affected the current solution and caused the algorithm to fall into a local optimum. The search for prey in a linear way led to a single search result. From all iterations of the algorithm, the optimal position of the current algorithm was only updated when the algorithm searched for a better solution than the current one, and the overall number of updates of the optimal position was low, which led to a decrease in the efficiency of the algorithm's search. In reality, when a Harris's hawk chases its prey, it hovers and descends in a spiral manner to catch its prey adaptively, showing better agility when hunting.

Here, the optimal neighborhood disturbance strategy was introduced to enhance the convergence speed of the algorithm and avoid premature maturity of the algorithm. The adaptive weighting and variable spiral position update strategies were introduced to enhance the global search capability of the algorithm by simulating the predation process of Harris's hawks in nature. To make the initial solution generated in the population initialization phase of the HHO algorithm cover the solution space as much as possible, we selected the best chaotic mapping method for HHO among seven commonly used chaotic mapping population initialization methods. It was used as the population initialization method to improve the algorithm. Hence, the above four methods are used to improve the global search capability of the HHO algorithm and to increase the speed of Harris Hawk's search for the optimal solution.

### 3.2. Chaotic Mapping

Chaotic is a deterministic stochastic method found in non-periodic, non-convergence and bounded nonlinear dynamic systems. Mathematically, chaotic is the randomness of a simple deterministic dynamic system, and a chaotic system is considered as the source of randomness. The essence of chaotic is obviously random and unpredictable, and it also has regularity [59].

As an important part of the population initialization algorithm, its result directly affects the convergence speed and quality of the algorithm [60,61]. For example, uniform distribution has more complete coverage of solution space than random distribution, and it is easier to obtain good initial solutions. A classical HHO algorithm uses random population initialization operation, which cannot cover the whole solution space. A chaotic sequence has ergodicity, randomness, and regularity in a certain range. Compared with random search, chaotic sequence searches the search space thoroughly with higher probability, which enables the algorithm to go beyond the local optimum and maintain the diversity of the population. Based on the above analysis, to obtain a good initial solution position and speed up the convergence of the population, seven common chaotic mappings Sinusoidal, Tent, Kent, Cubic, Logistic, Gauss, and Circle were selected [62–69] and used to initialize the population of HHO algorithm. The results were analyzed and the optimal one for the HHO algorithm selected as the population initialization method for the improved algorithm. The following were the mathematical formulas of the 10 chaotic mappings:

(1) Sinusoidal chaotic mapping:

$$x_{k+1} = P \cdot \sin(2\pi x_k) \quad (14)$$

where  $P$  was the control parameter, here  $P = 2.3$ ,  $x_0 = 0.7$ , Equation (14) was written as

$$x_{k+1} = \sin(\pi x_k) \quad (15)$$

(2) Tent chaotic mapping

$$x_{k+1} = \begin{cases} 2x_k & x_k < 0.5 \\ 2(1 - x_k) & x_k \geq 0.5 \end{cases} \tag{16}$$

(3) Kent chaotic mapping

$$\begin{cases} x_{k+1} = x_k/\mu & 0 < x_k < \mu \\ x_{k+1} = (1 - x_k)/(1 - \mu) & \mu \leq x_k < 1 \end{cases} \tag{17}$$

The control parameter  $\mu \in (0, 1)$ , when  $\mu = 0.5$ , the system was Short Period State,  $\mu = 0.5$  was not taken here. When using the chaotic mapping, the initial value  $x_0$  had to not be the same as the system parameters  $\mu$ , otherwise the system evolved into a periodic system. Here, we took  $\mu = 0.4$ .

(4) Cubic chaotic mapping

The standard Cubic chaotic mapping function was expressed as

$$x_{k+1} = bx_k^3 - cx_k \tag{18}$$

where  $b$  and  $c$  were the influence factors of chaotic mapping. The range of Cubic chaotic mapping was different for different values to  $b$  and  $c$ . When  $c = 3$ , the sequence generated by Cubic mapping was chaotic. Also when  $b = 1$ ,  $x_n \in (-2, 2)$ ; when  $b = 4$ ,  $x_n \in (-1, 1)$ . Here, we took  $b = 4$  and  $c = 3$ .

(5) Logistic chaotic mapping

$$x_{k+1} = Px_k(1 - x_k) \tag{19}$$

when  $P = 4$ , the generation number of Logistic chaotic mapping was between  $(0, 1)$ .

(6) Gauss chaotic mapping

$$x_{k+1} = \begin{cases} 0 & x_k = 0 \\ \frac{1}{x_k \bmod (1)} & \text{otherwise} \end{cases} \tag{20}$$

$$\frac{1}{x_k \bmod (1)} = \frac{1}{x_k} - \left\lfloor \frac{1}{x_k} \right\rfloor$$

(7) Circle chaotic mapping

$$x_{k+1} = \text{mod} \left( x_k + 0.2 - \left( \frac{0.5}{2\pi} \right) \sin(2\pi x_k), 1 \right) \tag{21}$$

The three steps to initialize the population of the HHO using the seven chaotic mappings were:

**Step 1:** Randomly generate  $M$  Harris hawks in  $D$ -dimensional space, i.e.,  $Y = (y_1, y_2, y_3, \dots, y_n)$   $y_i \in (-1, 1) i = 1, 2, \dots, n$ .

**Step 2:** Iterate each dimension of each Harris hawk  $M$  times, resulting in  $M$  Harris hawks.

**Step 3:** After all Harris hawk iterations were completed, chaotic mapping (21) was applied to the solution space.

$$x_{id} = lb_d + (1 + y_{id}) \times \frac{ub_d - lb_d}{2} \tag{22}$$

where  $ub$  was the upper bound of the exploration space,  $lb$  the lower bound of the exploration space; the  $d$ -dimensional coordinates of the  $i$ -th Harris hawk were represented by  $y_{id}$ , which was generated using Equations (14)–(21); the coordinates of the  $i$ -th Harris hawk in the  $d$ -dimensional of the exploration space were  $x_{id}$ , which was generated using Equation (22).

Here, we first proposed the HHO algorithm based on seven different chaotic initialization strategies, respectively, chaotic initialization Harris hawks optimization (CIHHO) algorithm. Obviously, the implementation of CIHHO is basically the same as that of HHO, except that the initialization in Step 2 generates  $m$  individual Harris hawks using Equations (14)–(21), and then maps the positions of these  $m$  Harris hawks to the search space of the population using Equation (22).

### 3.3. Adaptive Weight

Inspired by the predation process of the Harris's hawk hunting strategy, we added an adaptive weight to the position update of Harris's hawk that changed with the number of iterations. In the early stage of the exploration phase of HHO, the influence of the optimal Harris's hawk position on the current individual position adjustment was weakened to improve the global search ability of the algorithm in the early stage. As the number of iterations increased, the influence of the optimal Harris's hawk position gradually increased, so that other Harris hawks could quickly converge to the optimal Harris hawk position and improve the convergence speed of the whole algorithm. According to the variation of the number of updates in the HHO algorithm, the adaptive weight composed of the number of iterations  $t$  were chosen as follows:

$$w(t) = 0.2 \cos\left(\frac{\pi}{2} \cdot \left(1 - \frac{t}{t_{\max}}\right)\right) \quad (23)$$

Such adaptive weight  $w(t)$  had a property of nonlinear variation between  $[0, 1]$ , due to the variation property of the  $\cos$  function between  $[0, \frac{\pi}{2}]$ , so that the weights were small at the beginning stage of the exploration phase, but changed slightly faster; at the end of the exploration phase their values were larger, but the speed of change would slow down, so that the convergence of the algorithm was fully guaranteed. The improved HHO algorithm position update formula is:

$$X(t+1) = \begin{cases} w(t)X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)| & q \geq 0.5 \\ w(t)(X_{rabbit}(t) - X_m(t)) - r_3(lb + r_4(ub - lb)) & q < 0.5 \end{cases} \quad (24)$$

The position update after the introduction of adaptive weights dynamically adjusted the weight size according to the increase of the number of iterations, so that the randomly selected Harris's hawk position  $X_{rand}(t)$  and the optimal average Harris's hawk position  $X_{rabbit}(t) - X_m(t)$  in the population guide the individual Harris's hawks differently at different times. As the number of iterations increased, the Harris's hawk population would move closer to the optimal position, and the larger weights would speed up the movement of Harris's hawk positions, which accelerated the convergence of the algorithm.

### 3.4. Variable Spiral Position Update

In the search phase of the HHO algorithm, Harris's hawk randomly searched for prey in two equal-opportunity strategies based on target location and its own location. However, in nature, Harris's hawks generally hover in a spiral shape to search for prey. To simulate the real process of prey search in nature, we introduced a variable spiral position update strategy in the search phase of the HHO algorithm, so that the Harris's hawk would adjust the distance of each position update according to the spiral shape between the target position and its own position (see Figure 3).

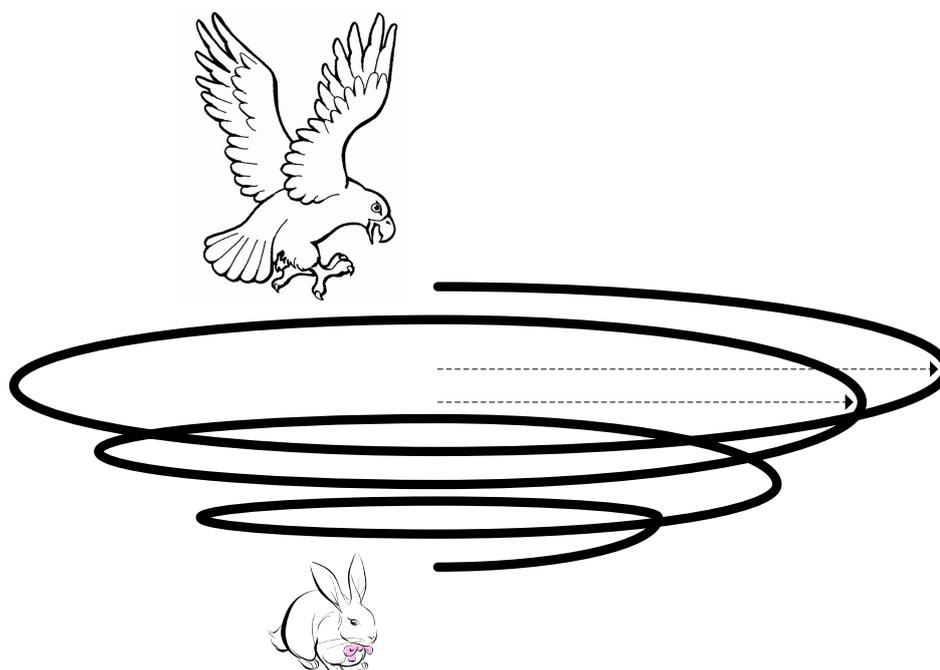


Figure 3. Spiral position update.

In the exploration phase of the HHO algorithm, Equation (1), a constant  $b$  was introduced to control the shape of the spiral; if this parameter was set to a constant, each time the Harris’s hawk position updated a different spiral arc for speed adjustment would follow. However, if  $b$  was set to a constant value, the spiral movement of the Harris’s hawk would be too singular when searching for prey, and it would follow a fixed spiral line to approach the target every time, which would easily fall into the misconception of local optimal solution and weaken the global exploration ability of the algorithm. To address this, we introduced the idea of variable spiral search to enable the Harris’s hawk to develop more diverse search path strategies for location update and design the parameter  $b$  as a variable that changes with the number of iterations to dynamically adjust the shape of the spiral when the Harris’s hawk explores, to increase the ability of the Harris’s hawk to explore unknown areas; thus, improving the global search capability of the algorithm. After combining the adaptive weights, the new spiral position update was created (see Equation (25)).

The  $b$  parameter was designed based on the mathematical model of the spiral, and the spiral shape was dynamically adjusted by introducing the number of iterations on the basis of the original spiral model. The  $b$  parameter was designed in such a way that the spiral shape changed from large to small as the number of iterations increased. Early in the exploration phase of the HHO algorithm, the Harris’s hawk searches the target with a larger spiral shape, the Harris hawk explores the global optimal solution as much as possible to improve the global optimal search capability of the algorithm; later in the exploration phase of the HHO algorithm, the Harris’s hawk searched the target with a small spiral shape to improve the algorithm’s search accuracy.

$$X(t + 1) = \begin{cases} w(t)X_{rand}(t) - b|X_{rand}(t) - 2r_2X(t)| & q \geq 0.5 \\ w(t)(X_{rabbit}(t) - X_m(t)) - b(lb + r_4(ub - lb)) & q < 0.5 \\ b = e^{5(\pi(1 - \frac{t}{t_{max}}))} \end{cases} \quad (25)$$

### 3.5. Optimal Neighborhood Disturbance

When updating the position, the Harris's hawk generally takes the current optimal position as the target of this iteration. In the whole iteration, the optimal position is updated only when there is a better position; thus, the total number of updates was not many, which led to the inefficiency of the algorithm search. In this regard, the optimal neighborhood disturbance strategy was introduced to search the neighborhood of the optimal position randomly to find a better global value, which could not only improve the convergence speed of the algorithm, but also avoided premature maturity of the algorithm. The optimal position generated a random disturbance to increase its search of the nearby space, and the neighborhood disturbance formula was:

$$\tilde{X}(t) = \begin{cases} X^*(t) + 0.5 \cdot h \cdot X^*(t), & g < 0.5 \\ X^*(t), & g \geq 0.5 \end{cases} \quad (26)$$

where  $h$  and  $g$  were random numbers uniformly generated between  $[0, 1]$ ;  $X(t)$  was the new position generated. For the generated neighborhood positions, a greedy strategy was used to determine whether to keep them, and the formula was:

$$X^*(t) = \begin{cases} \tilde{X}(t), & f(\tilde{X}(t)) < f(X^*(t)) \\ X^*(t), & f(X^*(t)) \leq f(\tilde{X}(t)) \end{cases} \quad (27)$$

where  $f(x)$  was the position adaptation value of  $x$ . If the generated position was better than the original position, it would be replaced with the original position to make it the global optimum. Otherwise, the optimal position remained unchanged.

### 3.6. Computational Complexity

The computational complexity of the population initialization process of the classical HHO algorithm is  $O(N)$ , and the computational complexity of the updated mechanism was  $O(T \times N) + O(T \times N \times D)$ , so the computational complexity of the classical HHO algorithm was  $O(N \times (T + TD + 1))$ , where  $T$  was the maximum number of iterations and  $D$  the dimension of the specific problem. The computational complexity of the population initialization process of the CSHHO algorithm was  $O(ND)$ , and the computational complexity of the update mechanism was the same as that of the classical HHO algorithm, so the computational complexity of the CSHHO algorithm was  $O(N \times (T + TD + D))$ , where  $T$  was the maximum number of iterations, and  $D$  the dimensionality of the specific problem.

### 3.7. Algorithm Procedure

Algorithm 1 shows the procedure of the CSHHO optimization algorithm: See Algorithm 1.

**Algorithm 1:** CSHHO algorithm

**Input:** The population size  $N$ , maximum number of iterations  $T$ .

**Output:** The location of rabbit and its fitness value.

Using Seven chaotic maps to initialize the population:

Through chaotic variables  $y_i^k \in [0, 1], k = 1, 2, \dots, M$ .  $M$  indicates the initial population dimension and Equations (14)–(21) generate initial chaotic vector;

Inverse mapping to get the initial population of the corresponding solution space through Equation (22);

**While** stopping condition is not meet **do**

Calculate the fitness values of hawks;

Set  $X_{rabbit}$  as best location of rabbit;

**For each**  $X_i$  **do**

Update the  $E$  using Equation (3);

**if**  $|E| \gg 1$  **then**

Update the vector  $X_i$  using Equations (25) and (2); Random generate parameters:  $r_1, r_2$ ;

$b = e^{5 \times (\pi(1 - \frac{t}{t_{max}}))}$ ;

**If**  $q \geq 0.5$  **then**

$X_i(t+1) = \omega(t) \times X_{rand}(t) - b \times |X_{rand}(t) - 2 \times r_2 \times X_i(t)|$ ;

**end**

**if**  $q < 0.5$  **then**

$X_i(t+1) = \omega(t) \times (X_{rabbit}(t) - X_m(t)) - b \times |lb + r_4 \times (ub - lb)|$ ;

**end**

**end**

**if**  $|E| < 1$  **then**

**if**  $r \geq 0.5$  and  $|E| \geq 0.5$  **then**

Update the vector  $X_i$  using Equations (4)–(6);

**end**

**else if**  $r \geq 0.5$  and  $|E| < 0.5$  **then**

Update the vector  $X_i$  using Equation (7);

**end**

**else if**  $r < 0.5$  and  $|E| \geq 0.5$

Update the vector  $X_i$  using Equations (8)–(11);

**end**

**else if**  $r < 0.5$  and  $|E| < 0.5$

Update the vector  $X_i$  using Equations (12) and (13);

**end**

**end**

**end**

Optimal neighborhood disturbance using Equations (27) and (28);

$$\tilde{X}(t) = \begin{cases} X^*(t) + 0.5 \times h \times X^*(t), & g < 0.5 \\ X^*(t), & g > 0.5 \end{cases};$$

$$X^*(t) = \begin{cases} \tilde{X}(t), & f(\tilde{X}(t)) < f(X^*(t)) \\ X^*(t), & f(X^*(t)) > f(\tilde{X}(t)) \end{cases};$$

**end**

**Return**  $X_{rabbit}$ ;

#### 4. Experiments and Discussion

In this section, to test and verify the performance of our optimizer proposed, namely CSHHO, a different category of experiments were designed. According to the randomness of the HHO algorithm, this section used a necessary and acceptable set of test functions to ensure that the superior results of the CSHHO algorithm did not happen by accident. Therefore, this section used two different benchmark test suites: classical 23 well-known benchmark functions [70,71] and standard IEEE CEC 2017 [72]. All experiments were as follows:

Experiment 1: First, seven chaotic mappings were used as the initialization method of HHO population and tested separately. Second, the seven data sets are analyzed, and

the optimal chaotic mapping was selected as the population initialization method of the improved algorithm.

Experiment 2: First, on the basis of Experiment 1, a combination test of adaptive weighting mechanism, variable spiral position update and optimal neighborhood disturbance mechanism was executed. Second, we analyzed and compared the CSHHO algorithm with other recently proposed meta-heuristic algorithms such as HHO [24], WOA [38], SCA [73], and Chicken Swarm Optimization (CSO) [74]. Third, we analyzed and compared the CSHHO algorithm with developed advanced variants such as HHO with dimension decision logic and Gaussian mutation (GCHHO) [54] and Hybrid PSO Algorithm with Adaptive Step Search (DEPSOASS) [75] and Gravitational search algorithm with linearly decreasing gravitational constant (Improved GSA) [76] and Dynamic Generalized Opposition-based Learning Fruit Fly Algorithm (DGOBLFOA) [77]. Fourth, based on the third step, the IEEE CEC 2017 was used to perform an algorithm-based accuracy scalability test with test dimensions  $D = 50$ ,  $D = 100$ .

To ensure the fairness of the experiments, the experiments were evaluated using the same parameters, and all population sizes  $N$  were set to 30, and dimension  $D$  was set to 30; each algorithm on each test instance was performed over 50 independent runs. In each run, the function error value  $\log(F(x) - F(x^*))$ , where  $F(x)$  was mean value found at all of the run, and  $F(x^*)$  was the optimal value recorded in 23 benchmark functions. The average error (Mean) and standard deviation (Std) of the function error values were considered as two performance metrics for evaluating the performance of the algorithm in all runs. The experimental environment: CPU Intel(R) Xeon(R) CPU E5-2680 v3 (2.50 GHz), RAM 16.00 GB, MATLAB R2019b.

#### 4.1. Benchmark Functions Verification

All experiments were performed using the classical 23 test functions [70,71] to test the performance of each algorithm in terms of convergence speed and search accuracy. These Benchmark functions were divided into three categories, including unimodal (UM) and multi-modal (MM). F1–F7 were the UM functions, which had unique global optimality and were used to test the exploitation performance of optimization algorithms. F8–F23 were the MM functions, which were used to test the exploration performance of the optimization algorithm and LO avoidance potentials. As the complexity of the test functions increased, the tested algorithms were more likely to fall into local optima, and all the test functions were used to evaluate the performance of the tested algorithms in various aspects. The convergence curves and test values of the corresponding test functions are given. Appendix A shows the classical 23 test functions.

IEEE CEC 2017 functions were also used in Experiment 2 to evaluate the scalability of CSHHO, other meta-heuristic algorithms, and developed HHO advanced variants. IEEE CEC 2017 Benchmark functions were classified into four categories, consisting of three UM functions (F1–F3), seven MM functions (F4–F10), 10 hybrid functions (F11–F20), and 10 composite functions (F21–F30). To evaluate the scalability of each algorithm more comprehensively, the dimensions of Benchmark functions were set to  $D = 50$ ,  $D = 100$ , and Table 2 records the corresponding accuracy values. It also shows the function formulas for IEEE CEC 2017.

In addition, to compare the performance of various algorithms, the rank was used to rank the mean values of all the algorithms in the simulation experiment in the order of lowest to highest. The lower the rank, the better the algorithm was compared to other algorithms; conversely, the higher the rank, the worse the algorithm was compared to other algorithms. Wilcoxon signed-rank test [78] was used to detect whether there was a significant performance difference among all algorithms, the  $p$ -value was corrected by Bonferroni–Holm correction [79]; moreover, the Friedman test [80] was used to rank the superiority of all the algorithms.

**Table 2.** The information of IEEE CEC2017.

ID	Description	Type	Dimension	Range	Optimum
F1	Shifted and Rotated Bent Cigar Function	Unimodal	30, 50, 100	[−100, 100]	100
F2	Shifted and Rotated Zakharov function	Unimodal	30, 50, 100	[−100, 100]	300
F3	Shifted and Rotated Rosenbrock's function	Multimodal	30, 50, 100	[−100, 100]	400
F4	Shifted and Rotated Rastrigin's function	Multimodal	30, 50, 100	[−100, 100]	500
F5	Shifted and Rotated Expanded Scaffer's F6 function	Multimodal	30, 50, 100	[−100, 100]	600
F6	Shifted and Rotated Lunacek Bi-Rastrigin function	Multimodal	30, 50, 100	[−100, 100]	700
F7	Shifted and Rotated Non-Continuous Rastrigin's function	Multimodal	30, 50, 100	[−100, 100]	800
F8	Shifted and Rotated Lévy function	Multimodal	30, 50, 100	[−100, 100]	900
F9	Shifted and Rotated Schwefel's function	Multimodal	30, 50, 100	[−100, 100]	1000
F10	Hybrid Function 1 (N = 3)	Hybrid	30, 50, 100	[−100, 100]	1100
F11	Hybrid Function 2 (N = 3)	Hybrid	30, 50, 100	[−100, 100]	1200
F12	Hybrid Function 3 (N = 3)	Hybrid	30, 50, 100	[−100, 100]	1300
F13	Hybrid Function 4 (N = 4)	Hybrid	30, 50, 100	[−100, 100]	1400
F14	Hybrid Function 5 (N = 4)	Hybrid	30, 50, 100	[−100, 100]	1500
F15	Hybrid Function 6 (N = 4)	Hybrid	30, 50, 100	[−100, 100]	1600
F16	Hybrid Function 6 (N = 5)	Hybrid	30, 50, 100	[−100, 100]	1700
F17	Hybrid Function 6 (N = 5)	Hybrid	30, 50, 100	[−100, 100]	1800
F18	Hybrid Function 6 (N = 5)	Hybrid	30, 50, 100	[−100, 100]	1900
F19	Hybrid Function 6 (N = 6)	Hybrid	30, 50, 100	[−100, 100]	2000
F20	Composition Function 1 (N = 3)	Composition	30, 50, 100	[−100, 100]	2100
F21	Composition Function 2 (N = 3)	Composition	30, 50, 100	[−100, 100]	2200
F22	Composition Function 3 (N = 4)	Composition	30, 50, 100	[−100, 100]	2300
F23	Composition Function 4 (N = 4)	Composition	30, 50, 100	[−100, 100]	2400
F24	Composition Function 5 (N = 5)	Composition	30, 50, 100	[−100, 100]	2500
F25	Composition Function 6 (N = 5)	Composition	30, 50, 100	[−100, 100]	2600
F26	Composition Function 7 (N = 6)	Composition	30, 50, 100	[−100, 100]	2700
F27	Composition Function 7 (N = 6)	Composition	30, 50, 100	[−100, 100]	2800
F28	Composition Function 9 (N = 3)	Composition	30, 50, 100	[−100, 100]	2900
F29	Composition Function 10 (N = 3)	Composition	30, 50, 100	[−100, 100]	3000

We used the values of the Friedman test to rank all the algorithms involved in the comparison, if the values of the Friedman test were the same, then the rankings were averaged. Here, the Friedman test was performed on the classical 23 test functions, and the test values were recorded in the average ranking values (ARV) column.

#### 4.2. Efficiency Analysis of the Improvement Strategy

First, in the population initialization phase, we selected the Gauss mapping, which had the highest impact on the accuracy of the HHO algorithm, as the population initialization method of CSHHO from seven commonly used chaotic mappings. Second, a global optimization strategy was used to optimize the HHO algorithm, which consisted of three components, including adaptive weight strategy, variable spiral update strategy and optimal neighborhood disturbance strategy. To verify the performance improvement of the HHO algorithm by the two improvements, six algorithms were used for comparison:

1. HHO without any modification, i.e., basic HHO;
2. WOA without any modification, i.e., basic WOA;
3. SCA without any modification, i.e., basic SCA;
4. CSO without any modification, i.e., basic CSO;
5. HHO with dimension decision logic and Gaussian mutation (GCHHO);
6. Hybrid PSO Algorithm with Adaptive Step Search (DEPSOASS);
7. Gravitational search algorithm with linearly decreasing gravitational constant (Improved GSA);
8. Dynamic Generalized Opposition-based Learning Fruit Fly Algorithm (DGOBLFOA).

#### 4.2.1. Influence of Seven Common Chaotic Mappings on HHO Algorithm

In order to select the best effective chaotic mapping method among seven well-known chaotic mapping methods, which enables us to obtain the best initial solution position and speed up the convergence of the Harris Hawk algorithm population, sinusoidal chaotic mapping, Tent chaotic mapping, Kent chaotic mapping, Cubic chaotic mapping, Logistic chaotic mapping, Gauss chaotic mapping, Circle chaotic mapping were initialized to the population of the HHO algorithm, respectively, forming Sinusoidal-HHO, Tent-HHO, Kent-HHO, Cubic-HHO, Logistic-HHO, Gauss-HHO, Circle-HHO, and we compared the accuracy of these seven algorithms.

Table 3 presents the results of the seven algorithms for 23 classical test functions. The results included Best, Worst, Mean, Rank, and Std for each algorithm run 50 times independently.

Table 4 shows the Bonferroni–Holm corrected probability values  $p$  obtained from the Wilcoxon signed-rank test for the seven chaotic mapping HHO algorithms. Symbols “+|=−” represent the number of algorithms that were better, similar, or worse than Gauss-HHO. The ARV at Table 5 is the value of the Friedman test for the seven chaotic mapping HHO algorithms.

Table 3 shows the data with better experimental results in bold. By analyzing the experimental results, we concluded that under the UM functions (F1–F7), Sinusoidal chaotic mapping achieved optimal results in F3, F5, F7 test functions, Circle chaotic mapping achieved optimal results in F1, F4 test functions, and Sinusoidal chaotic mapping had the most influence on the HHO algorithm, followed by Gauss chaotic mapping and Circle chaotic mapping. Under the MM functions (F8–F23), Gauss chaotic mapping had the most influence on the HHO algorithm. Circle chaotic mapping, Sinusoidal chaotic mapping, Tent chaotic mapping, and Kent chaotic mapping obtained the best results in F21, F15, F20, F13, and F23 test functions, respectively; the results of the seven chaotic mappings were compared in 23 test functions, The Gauss chaotic mapping obtained the most optimal solutions.

Table 4 shows the Bonferroni–Holm correction  $p$ -values of Wilcoxon signed rank test with 5% confidence level, “+|=−” indicates whether Gauss-HHO was worse consistent or better with Circle-HHO, Sinusoidal-HHO, Tent-HHO, Kent-HHO, Cubic-HHO and Logistic-HHO. Analyzing the Bonferroni–Holm corrected  $p$ -value of Wilcoxon signed rank test and the value of “+|=−” in each row of the table, better results were obtained using Gauss-HHO based among the 23 tested functions; the experimental results of HHO algorithms based on seven chaotic mappings, respectively, were evaluated comprehensively using Friedman’s test at Table 5, compared with the other six chaotic mappings population initialization methods. Gauss-HHO obtained the best results in terms of average ranking, indicating that for the HHO optimization algorithm, Gauss chaotic mapping not only had the randomness, ergodicity and initial value sensitivity of the chaotic mapping itself, but also the population initialization of the HHO optimization algorithm using Gauss chaotic mapping. The Gauss chaos map was used to initialize the population of the HHO optimization algorithm, and to obtain a better optimization accuracy.

Table 3. Results of a comparison with seven chaotic mappings on HHO.

Benchmark		Circle	Sinusoidal	Tent	Kent	Cubic	Logistic	Gauss
F1	Mean	$4.37 \times 10^{-110}$	$2.73 \times 10^{-96}$	$7.91 \times 10^{-101}$	$3.68 \times 10^{-99}$	$8.75 \times 10^{-100}$	$7.28 \times 10^{-99}$	$1.4 \times 10^{-109}$
	Std	$3.05 \times 10^{-109}$	$1.09 \times 10^{-95}$	$3.91 \times 10^{-100}$	$1.82 \times 10^{-98}$	$4.06 \times 10^{-99}$	$5.13 \times 10^{-98}$	$9.69 \times 10^{-109}$
	Rank	1	7	3	5	4	6	2
Best\Worst		$2.15 \times 10^{-108} \setminus$ $6.44 \times 10^{-135}$	$4.36 \times 10^{-95} \setminus$ $3.01 \times 10^{-115}$	$2.21 \times 10^{-99} \setminus$ $1.1 \times 10^{-119}$	$9.39 \times 10^{-98} \setminus$ $3.69 \times 10^{-116}$	$2.11 \times 10^{-98} \setminus$ $1.4 \times 10^{-116}$	$3.63 \times 10^{-97} \setminus$ $1.96 \times 10^{-116}$	$6.85 \times 10^{-108} \setminus$ $1.18 \times 10^{-137}$
	Mean	$6.4 \times 10^{-59}$	$3.88 \times 10^{-53}$	$3.45 \times 10^{-53}$	$7.39 \times 10^{-53}$	$8.12 \times 10^{-53}$	$8.59 \times 10^{-53}$	<b><math>8.05 \times 10^{-61}</math></b>
F2	Std	$3.12 \times 10^{-58}$	$1.74 \times 10^{-52}$	$1.5 \times 10^{-52}$	$4.9 \times 10^{-52}$	$5.03 \times 10^{-52}$	$4.9 \times 10^{-52}$	<b><math>2.37 \times 10^{-60}</math></b>
	Rank	2	4	3	5	6	7	1
	Best\Worst	$1.97 \times 10^{-57} \setminus$ $5.03 \times 10^{-77}$	$1.20 \times 10^{-51} \setminus$ $6.33 \times 10^{-60}$	$8.88 \times 10^{-52} \setminus$ $1.72 \times 10^{-61}$	$3.47 \times 10^{-51} \setminus$ $1.86 \times 10^{-64}$	$3.56 \times 10^{-51} \setminus$ $2.76 \times 10^{-68}$	$3.4391 \times 10^{-51} \setminus$ $3.91 \times 10^{-62}$	<b><math>1.0834 \times 10^{-59} \setminus</math></b> <b><math>3.80 \times 10^{-68}</math></b>
F3	Mean	$2.90 \times 10^{-86}$	$2.51 \times 10^{-86}$	$5.22 \times 10^{-82}$	$5.28 \times 10^{-78}$	$3.08 \times 10^{-83}$	$9.34 \times 10^{-86}$	$7.29 \times 10^{-78}$
	Std	$2.01 \times 10^{-85}$	$9.38 \times 10^{-86}$	$3.13 \times 10^{-81}$	$3.73 \times 10^{-77}$	$2.15 \times 10^{-82}$	$3.43 \times 10^{-85}$	$5.15 \times 10^{-77}$
	Rank	2	1	5	6	4	3	7
Best\Worst		$1.42 \times 10^{-84} \setminus$ $8.08 \times 10^{-104}$	$3.51 \times 10^{-85} \setminus$ $9.55 \times 10^{-104}$	$2.18 \times 10^{-80} \setminus$ $1.01 \times 10^{-108}$	$2.63 \times 10^{-76} \setminus$ $5.02 \times 10^{-101}$	$1.52 \times 10^{-81} \setminus$ $2.2 \times 10^{-107}$	$1.90 \times 10^{-84} \setminus$ $5.74 \times 10^{-105}$	$3.64 \times 10^{-76} \setminus$ $1.58 \times 10^{-107}$
	Mean	$8.48 \times 10^{-56}$	$1.23 \times 10^{-50}$	$7.91 \times 10^{-51}$	$8.72 \times 10^{-51}$	$3.65 \times 10^{-50}$	$9.39 \times 10^{-50}$	$9.3 \times 10^{-53}$
F4	Std	$4.24 \times 10^{-55}$	$4.28 \times 10^{-50}$	$5.22 \times 10^{-50}$	$3.6 \times 10^{-50}$	$2.4 \times 10^{-49}$	$6.51 \times 10^{-49}$	$6.49 \times 10^{-52}$
	Rank	1	5	3	4	6	7	2
	Best\Worst	$2.64 \times 10^{-54} \setminus$ $9.93 \times 10^{-69}$	$1.91 \times 10^{-49} \setminus$ $1.3 \times 10^{-55}$	$3.69 \times 10^{-49} \setminus$ $2.77 \times 10^{-59}$	$1.97 \times 10^{-49} \setminus$ $4.06 \times 10^{-58}$	$1.70 \times 10^{-48} \setminus$ $2.16 \times 10^{-60}$	$4.60 \times 10^{-48} \setminus$ $4.39 \times 10^{-58}$	$4.59 \times 10^{-51} \setminus$ $1.13 \times 10^{-65}$
F5	Mean	$8.55 \times 10^{-2}$	$4.17 \times 10^{-2}$	$5.42 \times 10^{-2}$	$4.65 \times 10^{-2}$	$7.29 \times 10^{-2}$	$4.85 \times 10^{-2}$	$8.33 \times 10^{-2}$
	Std	$5.63 \times 10^{-1}$	$4.90 \times 10^{-3}$	$9.52 \times 10^{-3}$	$7.56 \times 10^{-3}$	$9.62 \times 10^{-3}$	$8.98 \times 10^{-3}$	$5.64 \times 10^{-1}$
	Rank	7	1	4	2	5	3	6
Best\Worst		$3.9896 \setminus$ $4.3262 \times 10^{-5}$	$0.021867 \setminus$ $3.3302 \times 10^{-6}$	$0.048302 \setminus$ $8.4348 \times 10^{-7}$	$0.0425 \setminus$ $8.3416 \times 10^{-7}$	$0.043385 \setminus$ $3.4475 \times 10^{-5}$	$0.051141 \setminus$ $6.6967 \times 10^{-6}$	$3.9896 \setminus$ $6.6344 \times 10^{-6}$
	Mean	$3.34 \times 10^{-5}$	$7.87 \times 10^{-5}$	$5.47 \times 10^{-5}$	$4.72 \times 10^{-5}$	$6.85 \times 10^{-5}$	$4.45 \times 10^{-5}$	<b><math>2.48 \times 10^{-5}</math></b>
F6	Std	$6.12 \times 10^{-5}$	$1.10 \times 10^{-4}$	$9.45 \times 10^{-5}$	$5.54 \times 10^{-5}$	$8.89 \times 10^{-56}$	$6.00 \times 10^{-5}$	<b><math>3.13 \times 10^{-5}</math></b>
	Rank	2	7	5	4	6	3	1
	Best\Worst	$3.42 \times 10^{-4} \setminus$ $1.6931 \times 10^{-8}$	$4.27 \times 10^{-4} \setminus$ $7.8651 \times 10^{-8}$	$3.71 \times 10^{-4} \setminus$ $2.4303 \times 10^{-9}$	$3.16 \times 10^{-4} \setminus$ $3.9886 \times 10^{-8}$	$4.33 \times 10^{-4} \setminus$ $1.2627 \times 10^{-8}$	$2.77 \times 10^{-4} \setminus$ $2.6747 \times 10^{-8}$	<b><math>1.48 \times 10^{-4} \setminus</math></b> <b><math>1.1473 \times 10^{-10}</math></b>
F7	Mean	$1.02 \times 10^{-4}$	$7.60 \times 10^{-5}$	$8.41 \times 10^{-5}$	$9.33 \times 10^{-5}$	$7.99 \times 10^{-5}$	$7.62 \times 10^{-5}$	$9.17 \times 10^{-5}$
	Std	$9.77 \times 10^{-5}$	$4.66 \times 10^{-5}$	$1.02 \times 10^{-4}$	$9.06 \times 10^{-5}$	$8.17 \times 10^{-5}$	$7.35 \times 10^{-5}$	$8.83 \times 10^{-5}$
	Rank	7	1	4	6	3	2	5
Best\Worst		$4.09 \times 10^{-4} \setminus$ $1.85 \times 10^{-6}$	$1.92 \times 10^{-4} \setminus$ $1.62 \times 10^{-6}$	$5.44 \times 10^{-4} \setminus$ $1.26 \times 10^{-6}$	$3.88 \times 10^{-4} \setminus$ $4.19 \times 10^{-6}$	$5.07 \times 10^{-4} \setminus$ $1.77 \times 10^{-6}$	$4.39 \times 10^{-4} \setminus$ $8.40 \times 10^{-7}$	$4.36 \times 10^{-4} \setminus$ $1.72 \times 10^{-6}$
	Mean	$-1.26 \times 10^4$	$-1.26 \times 10^4$	$-1.26 \times 10^4$	$-1.26 \times 10^4$	$-1.25 \times 10^4$	$-1.26 \times 10^4$	<b><math>-1.26 \times 10^4</math></b>
F8	Std	52.1	0.232	0.453	0.508	0.378	0.226	<b>30.7</b>
	Rank	1	1	1	1	7	1	1
	Best\Worst	$-1.23 \times 10^4 \setminus$ $-1.25 \times 10^4$	$-1.25 \times 10^4 \setminus$ $-1.25 \times 10^4$	$-1.25 \times 10^4 \setminus$ $-1.25 \times 10^4$	$-1.25 \times 10^4 \setminus$ $-1.25 \times 10^4$	$-9.90 \times 10^3 \setminus$ $-1.25 \times 10^4$	$-1.25 \times 10^4 \setminus$ $-1.25 \times 10^4$	<b><math>-1.23 \times 10^4 \setminus</math></b> <b><math>-1.25 \times 10^4</math></b>

Table 3. Cont.

Benchmark		Circle	Sinusoidal	Tent	Kent	Cubic	Logistic	Gauss
F9	Mean	0	0	0	0	0\0	0	0
	Std	0	0	0	0	0	0	0
	Rank	1	1	1	1	1	1	1
	Best\Worst	0\0	0\0	0\0	0\0	0\0	0\0	0\0
F10	Mean	$8.88 \times 10^{-16}$						
	Std	$1.99 \times 10^{-31}$						
	Rank	1	1	1	1	1	1	1
	Best\Worst	$8.88 \times 10^{-16}$ \ $8.88 \times 10^{-16}$						
F11	Mean	0	0	0	0	0	0	0
	Std	0	0	0	0	0	0	0
	Rank	1	1	1	1	1	1	1
	Best\Worst	0\0	0\0	0\0	0\0	0\0	0\0	0\0
F12	Mean	$1.31 \times 10^{-6}$	$2.79 \times 10^{-6}$	$4.21 \times 10^{-6}$	$2.75 \times 10^{-6}$	$4.02 \times 10^{-6}$	$4.18 \times 10^{-6}$	$1.18 \times 10^{-6}$
	Std	$1.49 \times 10^{-6}$	$3.23 \times 10^{-6}$	$8.73 \times 10^{-6}$	$3.36 \times 10^{-6}$	$5.88 \times 10^{-6}$	$5.65 \times 10^{-6}$	$1.52 \times 10^{-6}$
	Rank	2	4	7	3	5	6	1
	Best\Worst	$7.06 \times 10^{-6}$ \ $5.43 \times 10^{-9}$	$1.46 \times 10^{-5}$ \ $2.19 \times 10^{-8}$	$5.17 \times 10^{-5}$ \ $1.09 \times 10^{-8}$	$1.75 \times 10^{-5}$ \ $2.96 \times 10^{-9}$	$3.25 \times 10^{-5}$ \ $2.30 \times 10^{-9}$	$2.80 \times 10^{-5}$ \ $5.59 \times 10^{-9}$	$6.68 \times 10^{-6}$ \ $5.16 \times 10^{-10}$
F13	Mean	$2.66 \times 10^{-4}$	$3.31 \times 10^{-4}$	$3.91 \times 10^{-5}$	$2.56 \times 10^{-5}$	$4.15 \times 10^{-5}$	$2.68 \times 10^{-5}$	$4.62 \times 10^{-4}$
	Std	$1.55 \times 10^{-3}$	$2.07 \times 10^{-3}$	$5.02 \times 10^{-5}$	$4.26 \times 10^{-5}$	$5.82 \times 10^{-54}$	$3.68 \times 10^{-5}$	$2.17 \times 10^{-3}$
	Rank	5	6	3	1	4	2	7
	Best\Worst	$0.010987$ \ $1.4769 \times 10^{-7}$	$0.014637$ \ $1.0906 \times 10^{-7}$	$0.00023743$ \ $4.4639 \times 10^{-8}$	$0.00026746$ \ $5.4412 \times 10^{-7}$	$0.00024897$ \ $2.9199 \times 10^{-11}$	$0.00016986$ \ $8.1121 \times 10^{-8}$	$0.010987$ \ $5.0129 \times 10^{-8}$
F14	Mean	1.20	$9.97 \times 10^{-1}$	$9.98 \times 10^{-1}$	1.22	1.02	1.12	1.57
	Std	$9.76 \times 10^{-1}$	$5.61 \times 10^{-16}$	$5.61 \times 10^{-16}$	$9.82 \times 10^{-1}$	$1.41 \times 10^{-1}$	$7.10 \times 10^{-1}$	1.49
	Rank	5	1	2	6	3	4	7
	Best\Worst	$5.93$ \ $9.98 \times 10^{-1}$	$9.98 \times 10^{-1}$ \ $9.98 \times 10^{-1}$	$9.98 \times 10^{-1}$ \ $9.98 \times 10^{-1}$	$5.93$ \ $9.98 \times 10^{-1}$	$1.99$ \ $9.98 \times 10^{-1}$	$5.93$ \ $9.98 \times 10^{-1}$	$5.93$ \ $9.98 \times 10^{-1}$
F15	Mean	$3.24 \times 10^{-4}$	$3.19 \times 10^{-4}$	$3.22 \times 10^{-4}$	$3.27 \times 10^{-4}$	$3.72 \times 10^{-4}$	$3.28 \times 10^{-4}$	$3.25 \times 10^{-4}$
	Std	$1.45 \times 10^{-5}$	$1.03 \times 10^{-5}$	$1.70 \times 10^{-5}$	$1.93 \times 10^{-5}$	$2.24 \times 10^{-4}$	$1.79 \times 10^{-5}$	$1.23 \times 10^{-5}$
	Rank	3	1	2	5	7	6	4
	Best\Worst	$3.82 \times 10^{-4}$ \ $3.08 \times 10^{-4}$	$3.48 \times 10^{-4}$ \ $3.08 \times 10^{-4}$	$3.72 \times 10^{-4}$ \ $3.08 \times 10^{-4}$	$3.92 \times 10^{-4}$ \ $3.08 \times 10^{-4}$	$1.63 \times 10^{-3}$ \ $3.08 \times 10^{-4}$	$3.86 \times 10^{-4}$ \ $3.08 \times 10^{-4}$	$3.95 \times 10^{-4}$ \ $3.08 \times 10^{-4}$
F16	Mean	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03
	Std	$1.35 \times 10^{-15}$						
	Rank	1	1	1	1	1	1	1
	Best\Worst	$-1.03$ \ $-1.03$	$-1.03$ \ $-1.031$	$-1.03$ \ $-1.03$	$-1.03$ \ $-1.03$	$-1.03$ \ $-1.03$	$-1.03$ \ $-1.03$	$-1.03$ \ $-1.03$

Table 3. Cont.

Benchmark		Circle	Sinusoidal	Tent	Kent	Cubic	Logistic	Gauss
F17	Mean	$3.98 \times 10^{-1}$	$3.9 \times 10^{-9}$					
	Std	$7.97 \times 10^{-8}$	$4.58 \times 10^{-8}$	$2.57 \times 10^{-7}$	$1.65 \times 10^{-7}$	$3.38 \times 10^{-7}$	$7.58 \times 10^{-8}$	$4.17 \times 10^{-8}$
	Rank	2	2	2	2	2	2	1
	Best\Worst	$3.97 \times 10^{-1} \setminus$ $3.97 \times 10^{-1}$	$3.97 \times 10^{-1} \setminus$ $0.397 \times 10^{-1}$	$3.97 \times 10^{-1} \setminus$ $0.397 \times 10^{-1}$				
F18	Mean	3	3	3	3	3	3	3
	Std	$1.41 \times 10^{-8}$	$1.31 \times 10^{-7}$	$1.98 \times 10^{-8}$	$5.48 \times 10^{-8}$	$1.41 \times 10^{-8}$	$7.00 \times 10^{-8}$	$4.63 \times 10^{-8}$
	Rank	1	1	1	1	1	1	1
	Best\Worst	3\3	3\3	3\3	3\3	3\3	3\3	3\3
F19	Mean	-3.86	-3.86	-3.86	-3.86	-3.86	-3.86	-3.86
	Std	$2.09 \times 10^{-3}$	$2.52 \times 10^{-3}$	$2.35 \times 10^{-3}$	$2.43 \times 10^{-3}$	$1.61 \times 10^{-3}$	$3.02 \times 10^{-3}$	$1.32 \times 10^{-3}$
	Rank	1	1	1	1	1	1	1
	Best\Worst	-3.85\ -3.86	-3.85\ -3.86	-3.85\ -3.86	-3.85\ -3.86	-3.86\ -3.86	-3.85\ -3.86	-3.85\ -3.86
F20	Mean	-3.12	-3.12	-3.14	-3.12	-3.13	-3.11	-3.12
	Std	$1.10 \times 10^{-1}$	$1.12 \times 10^{-1}$	$1.10 \times 10^{-1}$	$1.10 \times 10^{-1}$	$1.10 \times 10^{-1}$	$1.12 \times 10^{-1}$	$9.95 \times 10^{-2}$
	Rank	3	3	1	3	2	7	3
	Best\Worst	-2.88\ -3.31	-2.80\ -3.30	-2.89\ -3.30	-2.86\ -3.30	-2.90\ -3.30	-2.82\ -3.30	-2.81\ -3.27
F21	Mean	-8.77	-7.05	-7.04	-8.75	-5.36	-7.03	-8.45
	Std	2.23	2.47	2.46	2.21	1.21	2.45	2.36
	Rank	1	4	5	2	7	6	3
	Best\Worst	-5.05\ -1.01 $\times 10^1$	-5.04\ -1.01 $\times 10^1$	-5.05\ -1.01 $\times 10^1$	-5.05\ -1.01 $\times 10^1$			
F22	Mean	-8.44	-8.62	-7.06	-8.95	-5.49	-7.04	-9.06
	Std	2.54	2.45	2.55	2.32	1.38	2.53	1.26
	Rank	4	3	5	2	7	6	1
	Best\Worst	-5.08\ -1.04 $\times 10^1$	-5.07\ -1.04 $\times 10^1$	-5.08\ -1.04 $\times 10^1$	-5.09\ -1.04 $\times 10^1$	-5.08\ -1.04 $\times 10^1$	-5.08\ -1.04 $\times 10^1$	-5.09\ -1.04 $\times 10^1$
F23	Mean	-9.27	-8.61	-6.84	-9.54	-5.38	-7.34	-8.92
	Std	2.23	2.53	2.52	1.95	1.33	2.63	2.4
	Rank	2	4	6	1	7	5	3
	Best\Worst	-5.12\ -1.05 $\times 10^1$	-5.123\ -1.05 $\times 10^1$	-5.12\ -1.05 $\times 10^1$	-5.12\ -1.05 $\times 10^1$	-2.41\ -1.05 $\times 10^1$	-5.13\ -1.05 $\times 10^1$	-5.12\ -1.05 $\times 10^1$

Table 4. The Bonferroni–Holm corrected  $P$ -values of Wilcoxon’s signed-rank test.

Benchmark	Gauss	Circle		Sinusoidal		Tent		Kent		Cubic		Logistic	
		Corrected $p$ -Value	+ \= \-	Corrected $p$ -Value	+ \= \-	Corrected $p$ -Value	+ \= \-	Corrected $p$ -Value	+ \= \-	Corrected $p$ -Value	+ \= \-	Corrected $p$ -Value	+ \= \-
F1	N/A	$2.12 \times 10^1$	28\0\22	$1.043 \times 10^{-7}$	50\0\0	$5.40 \times 10^{-7}$	48\0\2	$1.04 \times 10^{-7}$	50\0\0	$1.08 \times 10^{-7}$	49\0\1	$6.77 \times 10^{-7}$	48\0\2
F2	N/A	$3.05 \times 10^1$	26\0\24	$1.043 \times 10^{-7}$	50\0\0	$1.20 \times 10^{-7}$	49\0\1	$1.08 \times 10^{-7}$	49\0\1	$1.20 \times 10^{-7}$	48\0\2	$1.03 \times 10^{-7}$	50\0\0
F3	N/A	$3.23 \times 10^1$	25\0\25	$3.20 \times 10^1$	24\0\26	$2.27 \times 10^1$	29\0\21	$2.01 \times 10^0$	32\0\18	$6.39 \times 10^0$	31\0\19	$8.66 \times 10^0$	28\0\22
F4	N/A	$1.64 \times 10^1$	23\0\27	$1.02 \times 10^{-5}$	47\0\3	$2.74 \times 10^{-5}$	46\0\4	$3.24 \times 10^{-7}$	48\0\2	$3.71 \times 10^{-5}$	46\0\4	$3.61 \times 10^{-7}$	48\0\2
F5	N/A	$2.02 \times 10^1$	30\0\20	$2.79 \times 10^1$	27\0\23	$2.79 \times 10^1$	27\0\23	$2.85 \times 10^1$	26\0\24	$6.12 \times 10^0$	31\0\19	$2.59 \times 10^1$	28\0\22
F6	N/A	$2.66 \times 10^1$	23\0\27	$4.67 \times 10^{-2}$	36\0\14	$8.66 \times 10^0$	33\0\17	$1.74 \times 10^0$	32\0\18	$2.25 \times 10^{-2}$	37\0\13	$6.18 \times 10^0$	32\0\18
F7	N/A	$3.34 \times 10^1$	26\0\24	$3.13 \times 10^1$	28\0\22	$3.22 \times 10^1$	23\0\27	$2.72 \times 10^1$	23\0\27	$3.26 \times 10^1$	23\0\27	$3.12 \times 10^1$	24\0\26
F8	N/A	$1.64 \times 10^1$	20\0\30	$3.26 \times 10^1$	24\1\25	$2.03 \times 10^1$	21\0\29	$3.27 \times 10^1$	24\0\26	$3.25 \times 10^1$	28\1\21	$2.71 \times 10^1$	20\1\29
F9	N/A	$2.49 \times 10^1$	0\50\0	$2.10 \times 10^1$	0\50\0	$1.70 \times 10^1$	0\50\0	$1.30 \times 10^1$	0\50\0	$9.00 \times 10^0$	0\50\0	$5.00 \times 10^0$	0\50\0
F10	N/A	$2.40 \times 10^1$	0\50\0	$2.00 \times 10^1$	0\50\0	$1.60 \times 10^1$	0\50\0	$1.20 \times 10^1$	0\50\0	$8.00 \times 10^0$	0\50\0	$4.00 \times 10^0$	0\50\0
F11	N/A	$2.30 \times 10^1$	0\50\0	$1.90 \times 10^1$	0\50\0	$1.50 \times 10^1$	0\50\0	$1.10 \times 10^1$	0\50\0	$7.00 \times 10^0$	0\50\0	$3.00 \times 10^0$	0\50\0
F12	N/A	$3.02 \times 10^1$	31\0\19	$3.32 \times 10^{-1}$	33\0\17	$5.13 \times 10^{-1}$	36\0\14	$9.84 \times 10^{-1}$	31\0\19	$1.26 \times 10^{-2}$	37\0\13	$6.20 \times 10^{-2}$	36\0\14
F13	N/A	$2.84 \times 10^1$	22\0\28	$2.91 \times 10^1$	26\0\24	$8.84 \times 10^0$	29\0\21	$3.15 \times 10^1$	23\0\27	$2.71 \times 10^1$	27\0\23	$3.29 \times 10^1$	22\0\28
F14	N/A	$1.26 \times 10^1$	2\39\9	$4.30 \times 10^{-1}$	0\41\9	$4.30 \times 10^{-1}$	0\41\9	$1.32 \times 10^1$	3\39\8	$1.23 \times 10^0$	1\40\9	$4.88 \times 10^0$	1\40\9
F15	N/A	$2.49 \times 10^1$	20\0\30	$5.13 \times 10^{-1}$	14\0\36	$1.31 \times 10^1$	18\0\32	$2.79 \times 10^1$	21\0\29	$1.99 \times 10^1$	29\0\21	$2.27 \times 10^1$	27\0\23
F16	N/A	$2.20 \times 10^1$	0\50\0	$1.80 \times 10^1$	0\50\0	$1.40 \times 10^1$	0\50\0	$1.00 \times 10^1$	0\50\0	$6.00 \times 10^0$	0\50\0	$2.00 \times 10^0$	0\50\0
F17	N/A	$3.23 \times 10^1$	10\28\12	$2.68 \times 10^1$	11\28\11	$2.20 \times 10^1$	12\27\11	$9.10 \times 10^0$	20\21\9	$9.18 \times 10^0$	22\17\11	$3.25 \times 10^1$	8\31\11
F18	N/A	$3.22 \times 10^1$	0\48\2	$3.28 \times 10^1$	5\42\3	$3.06 \times 10^1$	2\45\3	$3.32 \times 10^1$	5\42\3	$3.20 \times 10^1$	0\48\2	$2.76 \times 10^1$	2\45\3
F19	N/A	$3.28 \times 10^1$	25\0\25	$1.32 \times 10^1$	30\0\20	$3.23 \times 10^1$	27\0\23	$3.26 \times 10^1$	25\0\25	$2.92 \times 10^1$	25\0\25	$3.24 \times 10^1$	22\0\28
F20	N/A	$3.34 \times 10^1$	25\0\25	$3.00 \times 10^1$	23\0\27	$2.12 \times 10^1$	20\0\30	$3.00 \times 10^1$	23\0\27	$3.24 \times 10^1$	24\0\26	$3.27 \times 10^1$	24\0\26
F21	N/A	$2.03 \times 10^1$	21\0\29	$2.15 \times 10^0$	32\0\18	$2.87 \times 10^{-1}$	37\0\13	$2.85 \times 10^1$	28\0\22	$6.425 \times 10^{-5}$	40\0\10	$8.45 \times 10^{-2}$	38\0\12
F22	N/A	$2.20 \times 10^1$	29\0\21	$1.89 \times 10^1$	30\0\20	$4.39 \times 10^{-2}$	37\0\13	$2.79 \times 10^1$	28\0\22	$3.068 \times 10^{-7}$	47\0\3	$1.22 \times 10^{-2}$	39\0\11
F23	N/A	$2.75 \times 10^1$	23\0\27	$3.05 \times 10^1$	28\0\22	$2.31 \times 10^{-1}$	31\0\19	$3.26 \times 10^1$	24\0\26	$9.225 \times 10^{-6}$	40\0\10	$1.24 \times 10^0$	30\0\20

**Table 5.** Average ranks obtained by each method in the Friedman test.

Chaotic Mapping	Average Rankings
<b>Gauss</b>	<b>3.50</b>
Circle	3.74
Sinusoidal	3.89
Tent	3.85
Kent	4.85
Cubic	4.57
Logistic	3.61

#### 4.2.2. Comparison with Conventional Techniques

The Gauss mapping was used for population initialization. Then the adaptive weight mechanism, variable spiral position update mechanism, and adaptive neighborhood disturbance mechanism were introduced to form the CSHHO algorithm. In order to verify the effectiveness of the CSHHO algorithm against the emerging swarm intelligence optimization algorithms in recent years. In this subsection, the CSHHO algorithm is compared with recently published meta-heuristics, including HHO [24], WOA [24], SCA [65] and CSO [66] to calculate the average precision mean and stability Std of each algorithm. The performance of CSHHO was tested against other optimization algorithms using a nonparametric test: the Bonferroni–Holm corrected Wilcoxon signed rank test. Finally, the non-parametric test method (i.e., the Friedman test) was used to calculate the ARV values of all the participating algorithms and rank them together. As in Experiment 1, this experiment was also based on the test set of 23 classical test functions (see Table 1). The details of the experiment were consistent with the description at the beginning of this section, and Table 6 shows the detailed experimental results. Additionally, Table 7 gives the corrected Wilcoxon signed-rank test based on the 5% confidence level, the “+|=−” value: the number of CSHHO results that were worse, similar, better or than the comparison algorithm for each test function run 50 times, the result based on the Friedman test was at table.

The optimal results of the tested algorithms under the current Benchmark function are marked in bold. Analyzing the data in Table 6, CSHHO had a strong optimization capability compared to the traditional optimization algorithms. Under the MM functions (F8–F23), CSHHO obtained good results with the best optimization results under the Benchmarks of F9–F13, F15–F19, F21–F23, and CSHHO explored the most optimal region of the above Benchmark and outperformed the other compared optimization algorithms in terms of search performance. The CSHHO explored the above Benchmark optimal regions and outperformed the other participating optimization algorithms in terms of exploration performance. It tied for first place in Benchmark F9, F11, and F11. This showed that CSHHO had strong exploration ability and LO avoidance potentials.

Table 7 was analyzed to determine if there was a significant difference between the other algorithms and CSHHO. The “+|=−” column indicates the number of results that are less than, similar to, or greater than CSHHO for each of the HHO, WOA, SCA, and CSO algorithms run 50 times in each test function. CSHHO has 21 test functions with better results than HHO, CSHHO has 22 test functions with better results than WOA, CSHHO has 23 test functions with better results than SCA, CSHHO has 22 test functions with better results than CSO; the results of the corrected 5% confidence level Wilcoxon signed rank test were analyzed. If the  $p$ -value was greater than 0.05, the algorithm was considered to be the same as CSHHO; otherwise, it was considered to be significantly different. In most cases the  $p$ -values of Wilcoxon signed-rank test  $< 0.5$ , indicating that the CSHHO algorithm was significantly different from the other compared algorithms, all results were corrected by Bonferroni–Holm correction; At Table 8, analysis of the Friedman test value showed that the value of CSHHO was 2.57 lower than the traditional optimization algorithm. The CSHHO algorithm’s performance was better than other meta-heuristic algorithms.

Table 6. Results of a comparison with classic meta-heuristic algorithms.

Benchmark		CSHHO	HHO	WOA	SCA	CSO
F1	Mean	$1.96 \times 10^{-113}$	$1.24 \times 10^{-88}$	$6.15 \times 10^{-74}$	$2.92 \times 10^{-13}$	$2.82 \times 10^{-19}$
	Std	$1.38 \times 10^{-112}$	$8.75 \times 10^{-88}$	$2.79 \times 10^{-73}$	$1.33 \times 10^{-12}$	$7.98 \times 10^{-19}$
	Rank	1	2	3	5	4
F2	Best\Worst	$1.44 \times 10^{-138} \setminus 9.78 \times 10^{-112}$	$5.93 \times 10^{-111} \setminus 6.19 \times 10^{-87}$	$1.16 \times 10^{-87} \setminus 1.85 \times 10^{-72}$	$2.12 \times 10^{-20} \setminus 8.05 \times 10^{-12}$	$4.81 \times 10^{-26} \setminus 4.63 \times 10^{-18}$
	Mean	$1.66 \times 10^{-59}$	$1 \times 10^{-49}$	$8.97 \times 10^{-51}$	$1.04 \times 10^{-10}$	$1.05 \times 10^{-18}$
	Std	$1.06 \times 10^{-58}$	$4.8 \times 10^{-49}$	$4.65 \times 10^{-50}$	$2.43 \times 10^{-10}$	$2.76 \times 10^{-18}$
F3	Rank	1	3	2	5	4
	Best\Worst	$7.47 \times 10^{-58} \setminus 8.54 \times 10^{-70}$	$1.86 \times 10^{-74} \setminus 1.28 \times 10^{-73}$	$4.54 \times 10^4 \setminus 1.25 \times 10^4$	$2.19 \times 10^{-4} \setminus 1.31 \times 10^{-3}$	$6.13 \times 10^3 \setminus 3.20 \times 10^3$
	Mean	$4.96 \times 10^{-88}$	$1.86 \times 10^{-74}$	$4.54 \times 10^4$	0.000219	$6.13 \times 10^3$
F4	Std	$2.81 \times 10^{-87}$	$1.28 \times 10^{-73}$	$1.25 \times 10^4$	$1.31 \times 10^{-3}$	$3.2 \times 10^3$
	Rank	1	2	5	3	4
	Best\Worst	$5.85 \times 10^{-108} \setminus 1.96 \times 10^{-86}$	$4.26 \times 10^{-99} \setminus 9.05 \times 10^{-73}$	$1.81 \times 10^4 \setminus 7.29 \times 10^4$	$1.75 \times 10^{-12} \setminus 9.26 \times 10^{-3}$	$3.04 \times 10^2 \setminus 1.65 \times 10^4$
F5	Mean	$1.05 \times 10^{-55}$	$7.88 \times 10^{-49}$	53.4	$7.27 \times 10^{-5}$	21.6
	Std	$6.13 \times 10^{-55}$	$4.62 \times 10^{-48}$	$2.82 \times 10^1$	$1.38 \times 10^{-4}$	$1.28 \times 10^1$
	Rank	1	2	5	3	4
F6	Best\Worst	$1.83 \times 10^{-68} \setminus 4.25 \times 10^{-54}$	$2.8 \times 10^{-58} \setminus 3.25 \times 10^{-47}$	$9.79 \times 10^{-1} \setminus 8.84 \times 10^1$	$2.56 \times 10^{-7} \setminus 6.9 \times 10^{-4}$	$0.0748 \setminus 4.11 \times 10^1$
	Mean	$8.57 \times 10^{-3}$	$1.02 \times 10^{-2}$	$2.79 \times 10^1$	$1.11 \times 10^1$	$4.77 \times 10^1$
	Std	$2.23 \times 10^{-2}$	$1.43 \times 10^{-2}$	$4.98 \times 10^{-1}$	$2.78 \times 10^1$	$9.63 \times 10^1$
F7	Rank	1	2	4	3	5
	Best\Worst	$1.29 \times 10^{-8} \setminus 0.132$	$1.25 \times 10^{-5} \setminus 0.0723$	26.9\28.8	6.48\204	27.1\583
	Mean	$6.73 \times 10^{-7}$	$1.59 \times 10^{-4}$	0.374	0.377	3.73
F8	Std	$1.85 \times 10^{-6}$	$2.76 \times 10^{-4}$	$2.44 \times 10^{-1}$	$1.42 \times 10^{-1}$	$4.16 \times 10^{-1}$
	Rank	1	2	3	4	5
	Best\Worst	$4.40 \times 10^{-14} \setminus 8.8 \times 10^{-6}$	$4.85 \times 10^{-9} \setminus 0.00181$	0.0422\1.25	0.117\0.627	3.05\4.99
F9	Mean	$5.98 \times 10^{-5}$	$2.02 \times 10^{-4}$	$3.67 \times 10^{-3}$	$1.84 \times 10^{-3}$	$9.11 \times 10^{-2}$
	Std	$7.15 \times 10^{-5}$	$3.26 \times 10^{-4}$	$3.79 \times 10^{-3}$	$2.02 \times 10^{-3}$	$1.82 \times 10^{-1}$
	Rank	1	2	4	3	5
F10	Best\Worst	$2.03 \times 10^{-7} \setminus 4.65 \times 10^{-4}$	$3.13 \times 10^{-7} \setminus 2.10 \times 10^{-3}$	$9.46 \times 10^{-6} \setminus 1.75 \times 10^{-2}$	$6.53 \times 10^{-5} \setminus 1.02 \times 10^{-2}$	$2.66 \times 10^{-3} \setminus 1.07$
	Mean	$-1.24 \times 10^4$	$-1.26 \times 10^4$	$-1.04 \times 10^4$	$-2.24 \times 10^3$	$-7.12 \times 10^3$
	Std	$2.71 \times 10^2$	$9.50 \times 10^1$	$1.67 \times 10^3$	$1.52 \times 10^2$	$6.62 \times 10^2$
F11	Rank	2	1	3	5	4
	Best\Worst	$-1.26 \times 10^4 \setminus -1.15 \times 10^4$	$-1.26 \times 10^4 \setminus -1.19 \times 10^4$	$-1.26 \times 10^4 \setminus -7.76 \times 10^3$	$-2.61 \times 10^3 \setminus -1.81 \times 10^3$	$-8.89 \times 10^3 \setminus -5.91 \times 10^3$
	Mean	0	0	$3.41 \times 10^{-15}$	0.572	0.725
F12	Std	0	0	$1.78 \times 10^{-14}$	2.03	3.34
	Rank	1	1	3	4	5
	Best\Worst	0\0	0\0	$0 \setminus 1.14 \times 10^{-13}$	$0 \setminus 1.12 \times 10^1$	$0 \setminus 2.11 \times 10^1$
F13	Mean	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$4.3 \times 10^{-15}$	$5.46 \times 10^{-8}$	$9.39 \times 10^{-11}$
	Std	$1.99 \times 10^{-31}$	$1.99 \times 10^{-31}$	$2.68 \times 10^{-15}$	$1.41 \times 10^{-7}$	$2.5 \times 10^{-10}$
	Rank	1	1	3	5	4
F14	Best\Worst	$8.88 \times 10^{-16} \setminus 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \setminus 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \setminus 7.99 \times 10^{-15}$	$6.45 \times 10^{-11} \setminus 7.1 \times 10^{-7}$	$2.82 \times 10^{-13} \setminus 1.66 \times 10^{-9}$
	Mean	0	0	$1.97 \times 10^{-2}$	$9.43 \times 10^{-2}$	$1.22 \times 10^{-2}$

Table 6. Cont.

Benchmark		CSHHO	HHO	WOA	SCA	CSO
F12	Std	0	0	$7.55 \times 10^{-2}$	$1.56 \times 10^{-1}$	$4.49 \times 10^{-2}$
	Rank	1	1	4	5	3
	Best\Worst	0\0	0\0	$0\4.34 \times 10^{-1}$	$0\7.65 \times 10^{-1}$	$0\2.47 \times 10^{-1}$
	Mean	$5.26 \times 10^{-7}$	$9.97 \times 10^{-6}$	$2.69 \times 10^{-2}$	$7.80 \times 10^{-2}$	229
	Std	$1.2 \times 10^{-6}$	$1.36 \times 10^{-5}$	0.04	$2.80 \times 10^{-2}$	$1.1 \times 10^3$
F13	Rank	1	2	3	4	5
	Best\Worst	$1.68 \times 10^{-11}\5 \times 10^{-6}$	$3.07 \times 10^{-8}\7.06 \times 10^{-5}$	$4.58 \times 10^{-3}\2.64 \times 10^{-1}$	$3.41 \times 10^{-2}\1.63 \times 10^{-1}$	$1.57 \times 10^{-1}\7.34 \times 10^3$
	Mean	$1.64 \times 10^{-5}$	$1.26 \times 10^{-4}$	$5.05 \times 10^{-1}$	$2.64 \times 10^{-1}$	5.48
	Std	$2.86 \times 10^{-5}$	$2.49 \times 10^{-4}$	$2.57 \times 10^{-1}$	$8.66 \times 10^{-2}$	$2.07 \times 10^1$
	Rank	1	2	4	3	5
F14	Best\Worst	$6.14 \times 10^{-10}\1.23 \times 10^{-4}$	$3.21 \times 10^{-7}\1.31 \times 10^{-3}$	$9.14 \times 10^{-2}\1.19$	$8.80 \times 10^{-2}\5.17 \times 10^{-1}$	$1.38\1.47 \times 10^2$
	Mean	1.39	1.37	3.06	1.63	1.34
	Std	1.23	0.796	3.15	0.935	1.45
	Rank	3	2	5	4	1
	Best\Worst	$9.98 \times 10^{-1}\5.93$	$9.98 \times 10^{-1}\5.93$	$0.9.98 \times 10^{-1}\10.8$	$0.9.98 \times 10^{-1}\2.98$	$0.9.98 \times 10^{-1}\10.8$
F15	Mean	$3.40 \times 10^{-4}$	$3.60 \times 10^{-4}$	$7.04 \times 10^{-4}$	$9.44 \times 10^{-4}$	$7.90 \times 10^{-4}$
	Std	$3.39 \times 10^{-5}$	$1.45 \times 10^{-4}$	$4.57 \times 10^{-4}$	$3.72 \times 10^{-4}$	$2.78 \times 10^{-4}$
	Rank	1	2	3	5	4
	Best\Worst	$3.14 \times 10^{-4}\5.55 \times 10^{-4}$	$3.08 \times 10^{-4}\1.34 \times 10^{-3}$	$3.10 \times 10^{-4}\2.25 \times 10^{-3}$	$3.63 \times 10^{-4}\1.59 \times 10^{-3}$	$3.19 \times 10^{-4}\1.62 \times 10^{-3}$
	Mean	-1.03	-1.03	-1.03	-1.03	-1.03
F16	Std	$1.35 \times 10^{-15}$	$1.98 \times 10^{-8}$	$2.4 \times 10^{-8}$	$2.16 \times 10^{-5}$	$2.6 \times 10^{-6}$
	Rank	1	2	3	5	4
	Best\Worst	-1.03\ -1.03	-1.03\ -1.03	-1.03\ -1.03	-1.03\ -1.03	-1.03\ -1.03
	Mean	$3.98 \times 10^{-1}$	$0.398 \times 10^{-1}$	$0.398 \times 10^{-1}$	$0.399 \times 10^{-1}$	$0.398 \times 10^{-1}$
	Std	$1.88 \times 10^{-8}$	$2.62 \times 10^{-5}$	$2.15 \times 10^{-5}$	$7.21 \times 10^{-4}$	$1.86 \times 10^{-5}$
F17	Rank	1	1	1	1	1
	Best\Worst	$3.98 \times 10^{-1}\3.98 \times 10^{-1}$	$3.98 \times 10^{-1}\3.98 \times 10^{-1}$	$3.98 \times 10^{-1}\3.98 \times 10^{-1}$	$3.98 \times 10^{-1}\0.401$	$3.98 \times 10^{-1}\3.98 \times 10^{-1}$
	Mean	3.00	3.00	3.00	3.00	3.00
	Std	$1.15 \times 10^{-7}$	$1.01 \times 10^{-6}$	$1.62 \times 10^{-4}$	$5.88 \times 10^{-5}$	$2.09 \times 10^{-4}$
	Rank	1	1	1	1	1
F18	Best\Worst	3.00\3.00	3.00\3.00	3.00\3.00	3.00\3.00	3.00\3.00
	Mean	-3.86	-3.86	-3.85	-3.85	-3.86
	Std	$1.83 \times 10^{-3}$	$3.29 \times 10^{-3}$	$1.71 \times 10^{-2}$	$2.18 \times 10^{-3}$	$1.01 \times 10^{-2}$
	Rank	1	1	1	1	1
	Best\Worst	-3.86\ -3.86	-3.86\ -3.85	-3.86\ -3.75	-3.86\ -3.85	-3.86\ -3.8
F19	Mean	-3.13	-3.07	-3.26	-3.01	-3.24
	Std	$1.05 \times 10^{-1}$	$1.41 \times 10^{-1}$	$8.66 \times 10^{-2}$	$1.27 \times 10^{-1}$	$7.31 \times 10^{-2}$
	Rank	3	4	1	5	2
	Best\Worst	-3.31\ -2.74	-3.30\ -2.73	-3.32\ -3.04	-3.19\ -2.59	-3.32\ -3.02
	Mean	-9.65	-5.35	-8.24	-3.02	-7.64
F20	Std	1.38	1.18	2.54	1.83	3.02
F21						

Table 6. Cont.

Benchmark		CSHHO	HHO	WOA	SCA	CSO
F22	Rank	1	4	2	5	3
	Best\Worst	-10.2\ -5.02	-10.1\ -5.04	-10.2\ -2.63	-5.82\ -0.497	-10.2\ -2.55
	Mean	-9.45	-5.46	-7.83	-3.59	-8.19
	Std	1.94	1.29	.97	1.79	2.98
F23	Rank	1	4	3	5	2
	Best\Worst	-10.4\ -5.03	-10.1\ -5.03	-10.4\ -2.76	-8.23\ -0.906	-10.4\ -2.74
	Mean	-9.97	-5.04	-7.59	-4.1	-8.05
	Std	1.46	1.02	3.29	1.59	3.28
	Rank	1	4	3	5	2
	Best\Worst	-10.5\ -5.09	-9.96\ -1.65	-10.5\ -1.67	-9.42\ -0.945	-10.5\ -2.37

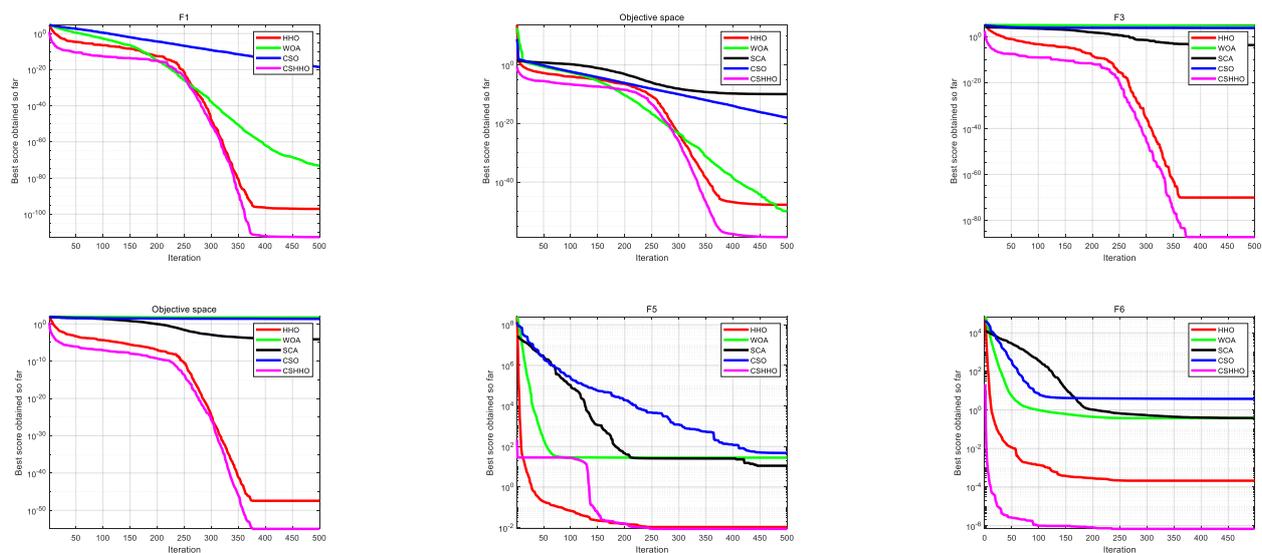
Table 7. The Bonferroni-Holm corrected p-values of Wilcoxon’s signed-rank test with classic meta-heuristic algorithms.

Benchmark	CSHHO	HHO		WOA		SCA		CSO	
		Corrected p-Value	+ \-						
F1	N/A	$6.80 \times 10^{-8}$	0\0\50	$6.50 \times 10^{-8}$	0\0\50	$5.89 \times 10^{-8}$	0\0\50	$4.84 \times 10^{-8}$	0\0\50
F2	N/A	$6.73 \times 10^{-8}$	0\0\50	$6.42 \times 10^{-8}$	0\0\50	$5.82 \times 10^{-8}$	0\0\50	$4.76 \times 10^{-8}$	0\0\50
F3	N/A	$1.38 \times 10^{-6}$	5\0\45	$6.35 \times 10^{-8}$	0\0\50	$5.74 \times 10^{-8}$	0\0\50	$4.69 \times 10^{-8}$	0\0\50
F4	N/A	$6.65 \times 10^{-8}$	0\0\50	$6.27 \times 10^{-8}$	0\0\50	$5.67 \times 10^{-8}$	0\0\50	$4.61 \times 10^{-8}$	0\0\50
F5	N/A	$5.24 \times 10^{-1}$	18\0\32	$6.20 \times 10^{-8}$	0\0\50	$5.59 \times 10^{-8}$	0\0\50	$4.53 \times 10^{-8}$	0\0\50
F6	N/A	$6.57 \times 10^{-8}$	0\0\50	$6.12 \times 10^{-8}$	0\0\50	$5.52 \times 10^{-8}$	0\0\50	$4.46 \times 10^{-8}$	0\0\50
F7	N/A	$4.11 \times 10^{-4}$	11\0\39	$4.27 \times 10^{-8}$	1\0\49	$4.18 \times 10^{-8}$	1\0\49	$4.38 \times 10^{-8}$	0\0\50
F8	N/A	$1.22 \times 10^{-3}$	35\0\15	$1.77 \times 10^{-6}$	8\0\42	$5.44 \times 10^{-8}$	0\0\50	$4.31 \times 10^{-8}$	0\0\50
F9	N/A	$3.00 \times 10^0$	0\50\0	$4.00 \times 10^0$	0\48\2	$6.92 \times 10^{-7}$	0\8\42	$8.13 \times 10^{-1}$	0\45\5
F10	N/A	$3.00 \times 10^0$	0\50\0	$3.17 \times 10^{-6}$	0\15\35	$5.37 \times 10^{-8}$	0\0\50	$4.23 \times 10^{-8}$	0\0\50
F11	N/A	$2.00 \times 10^0$	0\50\0	$8.13 \times 10^{-1}$	0\45\5	$1.10 \times 10^{-7}$	0\3\47	$3.17 \times 10^{-3}$	0\36\14
F12	N/A	$6.92 \times 10^{-7}$	5\0\45	$6.05 \times 10^{-8}$	0\0\50	$5.29 \times 10^{-8}$	0\0\50	$4.16 \times 10^{-8}$	0\0\50
F13	N/A	$1.44 \times 10^{-3}$	11\0\39	$5.97 \times 10^{-8}$	0\0\50	$5.21 \times 10^{-8}$	0\0\50	$4.08 \times 10^{-8}$	0\0\50
F14	N/A	$3.93 \times 10^0$	6\30\14	$1.15 \times 10^{-2}$	6\16\28	$4.60 \times 10^{-3}$	6\2\42	$3.93 \times 10^0$	5\38\7
F15	N/A	$3.57 \times 10^0$	26\0\24	$1.81 \times 10^{-7}$	5\0\45	$5.14 \times 10^{-8}$	0\0\50	$4.18 \times 10^{-8}$	1\0\49
F16	N/A	$4.00 \times 10^0$	0\48\2	$2.25 \times 10^0$	0\47\3	$6.94 \times 10^{-8}$	0\0\50	$1.15 \times 10^{-2}$	0\38\12
F17	N/A	$3.58 \times 10^{-5}$	1\17\32	$2.72 \times 10^{-6}$	1\11\38	$5.06 \times 10^{-8}$	0\0\50	$3.09 \times 10^{-2}$	3\32\15
F18	N/A	$2.96 \times 10^{-1}$	2\36\12	$6.87 \times 10^{-8}$	0\0\50	$6.94 \times 10^{-8}$	0\0\50	$5.00 \times 10^{-1}$	2\41\7
F19	N/A	$3.02 \times 10^{-2}$	13\0\37	$1.16 \times 10^{-4}$	10\0\40	$5.33 \times 10^{-8}$	1\0\49	$6.77 \times 10^{-1}$	21\0\29
F20	N/A	$2.96 \times 10^{-1}$	19\0\31	$1.04 \times 10^{-5}$	41\0\9	$2.34 \times 10^{-4}$	13\0\37	$5.59 \times 10^{-5}$	38\0\12
F21	N/A	$2.23 \times 10^{-7}$	3\0\47	$1.10 \times 10^0$	22\0\28	$4.99 \times 10^{-8}$	0\0\50	$2.49 \times 10^{-4}$	10\0\40
F22	N/A	$2.59 \times 10^{-7}$	7\0\43	$1.54 \times 10^0$	25\0\25	$5.33 \times 10^{-8}$	1\0\49	$9.79 \times 10^{-2}$	16\0\34
F23	N/A	$5.43 \times 10^{-8}$	2\0\48	$1.02 \times 10^{-1}$	18\0\32	$4.91 \times 10^{-8}$	0\0\50	$4.73 \times 10^{-4}$	14\0\36

**Table 8.** Average rankings obtained by classic meta-heuristic in the Friedman test, and the best result is shown in boldface.

Chaotic Mapping	Average Rankings
<b>CSHHO</b>	<b>2.57</b>
HHO	3.67
WOA	4.74
SCA	5.96
CSO	5.00

Figure 4 shows the convergence curves of CSHHO and the traditional optimization algorithms HHO, WOA, SCA and CSO under 23 Benchmark functions, including the performance under UM functions (F1–F7), MM functions (F8–F23), MM functions (F8–F23). The function error value was defined as, where  $F(x)$  was the mean value found at all of iterations, and  $F(x^*)$  was the optimal value recorded in 23 benchmark functions. Among them, under UM functions (F1–F7), the CSHHO algorithm converged with higher accuracy and converged faster than other algorithms, indicating that the development performance of CSHHO algorithm was improved compared with other algorithms; under MM functions (F8–F23), the CSHHO algorithm converged with higher accuracy and converged faster than other algorithms, indicating that the development performance of CSHHO algorithm was improved compared with other algorithms. From F8–F23 CSHHO algorithm did not fall into the local optimum region and could not escape; in F9, F11, F16 CSHHO converged faster in F12, F13, F17, F19, F21–F23. Although the convergence curve of the CSHHO algorithm was smoother and converged slower in the initial iterations as the algorithm searched further. In F10, F15 the CSHHO algorithm not only explores the dominant region with faster convergence speed, but also leads the rest of the algorithms in terms of search accuracy. Therefore, CSHHO algorithm benefits from Gauss chaotic mapping that enhances the population initialization of the algorithm, as well as adaptive weighting mechanism, variable spiral position updating mechanism and adaptive neighborhood disturbance mechanism that enhance the exploration and exploitation ability of the algorithm, CSHHO is less likely to fall into the current search region and increase the ability to jump out of the local optimal region.

**Figure 4.** Cont.

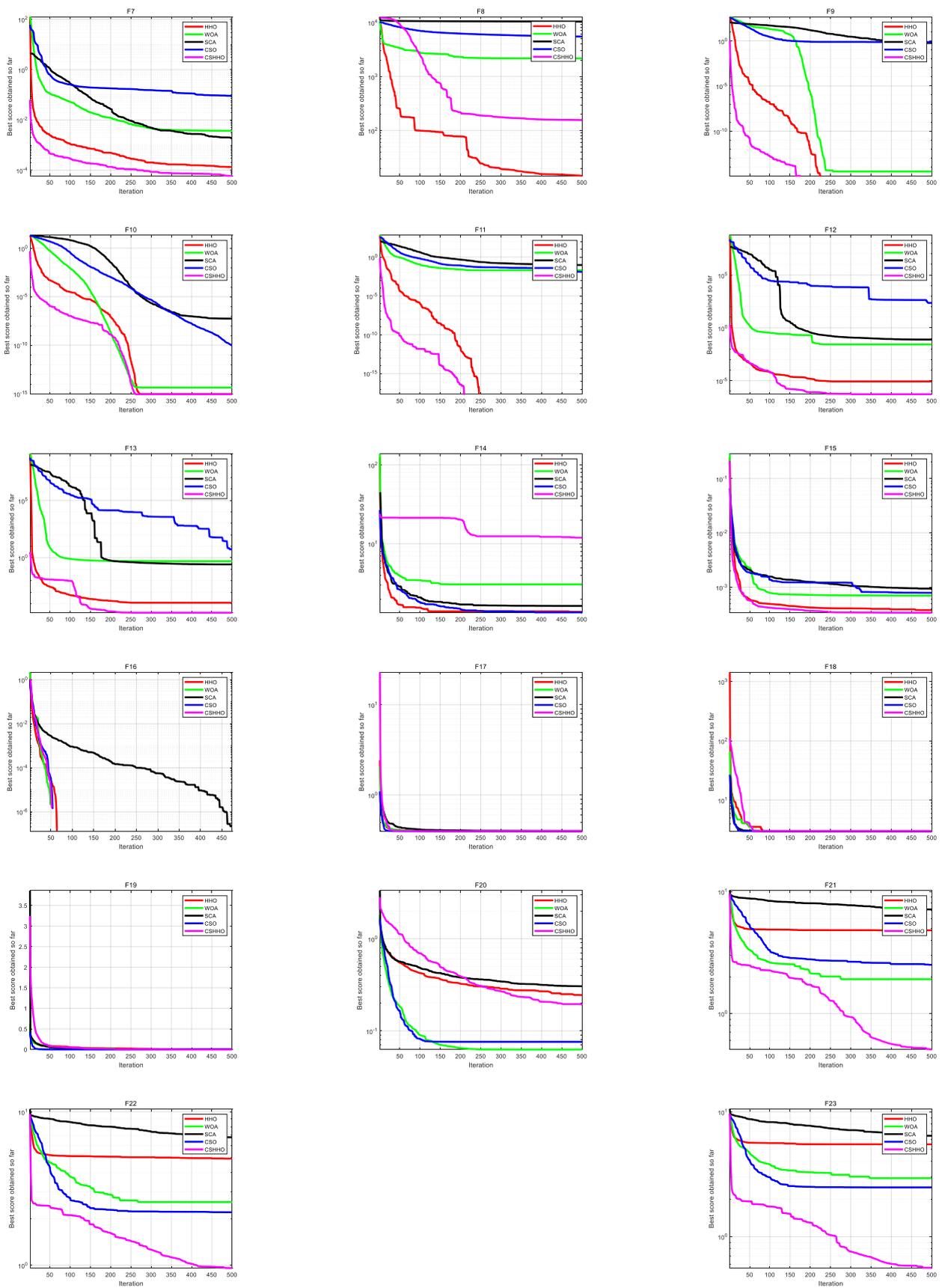


Figure 4. Convergence curves of CSHHO and the classic meta-heuristic algorithms.

#### 4.2.3. Comparison with HHO Variants

In order to verify the effectiveness of the CSHHO algorithm against HHO variants in recent years, this subsection compares the CSHHO algorithm with the recently published advanced HHO variants: GCHHO and DEPSOASS and Improved GSA and DGOBLFOA, observing the mean accuracy mean and stability Std of each algorithm. Table 9 presents the results of the experiments. Next, using nonparametric tests: the Wilcoxon signed-rank test and the Friedman test were used to synthetically assess the performance differences between CSHHO and GCHHO and DEPSOASS and Improved GSA and DGOBLFOA. The configuration of the experiments is the same as in Section 4.2.2, “+ \= \-”: the number of CSHHO results obtained from 50 runs in each test function that are worse, similar, or better than the comparison algorithm. Table 10 presents the results of the Bonferroni–Holm correction of Wilcoxon signed-rank test experiments, Table 11 presents the results of the Friedman test experiments.

Analyzing the data in Table 9, CSHHO has some advantages over the advanced HHO variants: under the UM functions (F1–F7), CSHHO outperforms the other algorithms in F1–F4, F6, F7, which indicates that CSHHO has further enhanced global best-finding capability compared to the advanced HHO variants. Under the MM functions (F8–F23) the CSHHO’s performance outperforms the remaining algorithms in F9, F10, F11, F16, F17, F18, F19, F21, which indicates that CSHHO can explore the peak-to-peak information deeply and effectively to avoid the algorithm from entering the local optimum. In summary, CSHHO has a good ability to develop and explore and avoid local optima.

Table 10 shows the results of the corrected Wilcoxon signed rank test at 5% confidence level and Friedman’s test for CSHHO and GCHHO and DEPSOASS and Improved GSA and DGOBLFOA. The p-values indicate whether the numerical results of the algorithms involved in the comparison are significant compared to CSHHO; if the p-value is greater than 0.05, the numerical results of the algorithm are considered the same as CSHHO; otherwise, it is considered to be significantly different. Analyzing the Bonferroni–Holm corrected p-value column values in Table 10, under the 23 classical test functions, only GCHHO is less different from CSHHO in the F9–F11 and F16–18 test functions, and in the rest of the test functions. Moreover, at Table 11 upon analyzing the results of the Friedman test, the value of CSHHO is 1.70, which is lower than others. The result indicates that CSHHO has an advantage over the above algorithms in optimization.

Figure 5 shows the convergence curves of CSHHO and the variants of the optimization algorithm GCHHO and DEPSOASS and Improved GSA and DGOBLFOA under 23 Benchmark Functions, including the performance under UM functions (F1–F7), MM functions (F8–F23). The function error value is defined as, where  $F(x)$  is the mean value found at all of iterations, and  $F(x^*)$  is the optimal value is recorded in 23. The CSHHO algorithm under UM functions (F1–F7) has improved convergence accuracy in F1–F4, F6, F7 compared to other algorithms, and convergence speed is better than other algorithms in F1, F2, F3, F4, F5, F6, F7; under MM functions (F8–F23), CSHHO algorithm does not fall into the local optimal region and cannot escape, in F9, F10, F11, F16, F17, F18, F19, F21. CSHHO can explore the dominant region well and is ahead of other algorithms in terms of search accuracy. In F9, F10, F11, F12, F13, F21, F22, F23, CSHHO has smooth convergence curves and faster convergence speed. In F16–F20, although the convergence speed of the CSHHO algorithm is slow, the convergence curve does not produce large fluctuations, which indicates that the CSHHO algorithm has good search ability, and it does not fall into local optimum and cannot jump out. Functions. In summary, the CSHHO algorithm’s performance is further improved compared to GCHHO algorithm and other advanced algorithms. Compared to these advanced variants, CSHHO algorithm are effective.

Table 9. Results of a comparison with the advanced meta-heuristic algorithms.

Benchmark		CSHHO	GCHHO	DEPSOASS	Improved GSA	DGOBLFOA
F1	Mean	$1.96 \times 10^{-113}$	$3.91 \times 10^{-97}$	$1.45 \times 10^{-3}$	$2.84 \times 10^1$	$2.88 \times 10^{-8}$
	Std	$1.38 \times 10^{-112}$	$2.55 \times 10^{-96}$	$1.45 \times 10^{-3}$	$2.84 \times 10^1$	$2.88 \times 10^{-8}$
	Rank	1	2	4	5	3
	Best\Worst	$9.78 \times 10^{-112} \setminus 1.44 \times 10^{-138}$	$1.80 \times 10^{-95} \setminus 1.25 \times 10^{-119}$	$3.22 \times 10^{-3} \setminus 3.86 \times 10^{-4}$	$4.68 \times 10^1 \setminus 1.51 \times 10^1$	$1.42 \times 10^{-6} \setminus 0.00 \times 10^0$
F2	Mean	$1.66 \times 10^{-59}$	$1.62 \times 10^{-50}$	$1.88 \times 10^{-1}$	$2.81 \times 10^1$	$2.54 \times 10^{-1}$
	Std	$1.06 \times 10^{-58}$	$9.45 \times 10^{-50}$	$1.88 \times 10^{-1}$	$2.81 \times 10^1$	$2.54 \times 10^{-1}$
	Rank	1	2	3	5	4
	Best\Worst	$7.47 \times 10^{-58} \setminus 8.54 \times 10^{-70}$	$6.63 \times 10^{-49} \setminus 3.57 \times 10^{-60}$	$2.84 \times 10^{-1} \setminus 1.19 \times 10^{-1}$	$3.46 \times 10^1 \setminus 1.44 \times 10^1$	$1.03 \times 10^0 \setminus 8.72 \times 10^{-4}$
F3	Mean	$4.96 \times 10^{-88}$	$1.46 \times 10^{-65}$	$5.37 \times 10^{-1}$	$3.32 \times 10^2$	$1.18 \times 10^{-2}$
	Std	$2.81 \times 10^{-87}$	$1.01 \times 10^{-64}$	$5.37 \times 10^{-1}$	$3.32 \times 10^2$	$1.18 \times 10^{-2}$
	Rank	1	2	4	5	3
	Best\Worst	$1.96 \times 10^{-86} \setminus 5.85 \times 10^{-108}$	$7.16 \times 10^{-64} \setminus 1.16 \times 10^{-91}$	$1.44 \times 10^0 \setminus 9.11 \times 10^{-2}$	$5.38 \times 10^2 \setminus 1.39 \times 10^2$	$4.66 \times 10^{-1} \setminus 1.45 \times 10^{-24}$
F4	Mean	$1.05 \times 10^{-55}$	$1.82 \times 10^{-48}$	$3.49 \times 10^{-2}$	$2.31 \times 10^0$	$1.74 \times 10^{-4}$
	Std	$6.13 \times 10^{-55}$	$9.06 \times 10^{-48}$	$3.49 \times 10^{-2}$	$2.31 \times 10^0$	$1.74 \times 10^{-4}$
	Rank	1	2	4	5	3
	Best\Worst	$4.25 \times 10^{-54} \setminus 1.83 \times 10^{-68}$	$5.94 \times 10^{-47} \setminus 2.66 \times 10^{-58}$	$5.40 \times 10^{-2} \setminus 1.75 \times 10^{-2}$	$2.83 \times 10^0 \setminus 1.44 \times 10^0$	$8.57 \times 10^{-3} \setminus 1.29 \times 10^{-20}$
F5	Mean	$8.57 \times 10^{-3}$	$7.14 \times 10^{-4}$	$2.76 \times 10^1$	$1.19 \times 10^4$	$2.87 \times 10^1$
	Std	$2.23 \times 10^{-2}$	$7.02 \times 10^{-4}$	$2.76 \times 10^1$	$1.19 \times 10^4$	$2.87 \times 10^1$
	Rank	2	1	3	5	4
	Best\Worst	$1.32 \times 10^{-1} \setminus 1.29 \times 10^{-8}$	$3.28 \times 10^{-3} \setminus 2.96 \times 10^{-5}$	$2.95 \times 10^1 \setminus 2.59 \times 10^1$	$2.60 \times 10^4 \setminus 1.29 \times 10^3$	$3.20 \times 10^1 \setminus 2.79 \times 10^1$
F6	Mean	$6.73 \times 10^{-7}$	$1.20 \times 10^{-6}$	$1.30 \times 10^{-3}$	$2.83 \times 10^1$	$5.35 \times 10^0$
	Std	$1.85 \times 10^{-6}$	$1.61 \times 10^{-6}$	$1.30 \times 10^{-3}$	$2.83 \times 10^1$	$5.35 \times 10^0$
	Rank	1	2	3	5	4
	Best\Worst	$8.80 \times 10^{-6} \setminus 4.40 \times 10^{-14}$	$1.02 \times 10^{-5} \setminus 1.47 \times 10^{-7}$	$3.27 \times 10^{-3} \setminus 3.91 \times 10^{-4}$	$4.15 \times 10^1 \setminus 1.43 \times 10^1$	$6.03 \times 10^0 \setminus 4.32 \times 10^0$
F7	Mean	$5.98 \times 10^{-5}$	$1.94 \times 10^{-4}$	$1.26 \times 10^{-1}$	$1.04 \times 10^2$	$6.10 \times 10^{-1}$
	Std	$7.15 \times 10^{-5}$	$1.92 \times 10^{-4}$	$1.26 \times 10^{-1}$	$1.04 \times 10^2$	$6.10 \times 10^{-1}$
	Rank	1	2	3	5	4
	Best\Worst	$4.65 \times 10^{-4} \setminus 2.03 \times 10^{-7}$	$7.84 \times 10^{-4} \setminus 5.01 \times 10^{-6}$	$2.21 \times 10^{-1} \setminus 7.33 \times 10^{-2}$	$1.52 \times 10^2 \setminus 3.44 \times 10^0$	$1.44 \times 10^0 \setminus 1.01 \times 10^{-1}$
F8	Mean	$-1.24 \times 10^4$	$-1.25 \times 10^4$	$-2.78 \times 10^3$	$-2.70 \times 10^3$	$-3.70 \times 10^2$
	Std	$2.71 \times 10^2$	$1.67 \times 10^2$	$-2.78 \times 10^3$	$-2.70 \times 10^3$	$1.69 \times 10^2$
	Rank	2	1	3	4	5
	Best\Worst	$-1.15 \times 10^4 \setminus -1.26 \times 10^4$	$-1.18 \times 10^4 \setminus -1.26 \times 10^4$	$-1.79 \times 10^3 \setminus -3.79 \times 10^3$	$-1.74 \times 10^3 \setminus -3.42 \times 10^3$	$-1.79 \times 10^3 \setminus -3.46 \times 10^3$
F9	Mean	0.00	0.00	$3.72 \times 10^1$	$2.54 \times 10^2$	$8.95 \times 10^0$
	Std	0.00	0.00	$3.72 \times 10^1$	$2.54 \times 10^2$	$8.95 \times 10^0$
	Rank	1	1	4	5	3
	Best\Worst	0.00\0.00	0.00\0.00	$6.02 \times 10^1 \setminus 2.00 \times 10^1$	$2.91 \times 10^2 \setminus 1.96 \times 10^2$	$2.81 \times 10^1 \setminus 2.18 \times 10^{-3}$
F10	Mean	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$3.06 \times 10^{-2}$	$5.11 \times 10^0$	$4.23 \times 10^{-2}$
	Std	$1.99 \times 10^{-31}$	$1.99 \times 10^{-31}$	$3.06 \times 10^{-2}$	$5.11 \times 10^0$	$4.23 \times 10^{-2}$
	Rank	1	1	3	5	4
	Best\Worst	$8.88 \times 10^{-16} \setminus 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \setminus 8.88 \times 10^{-16}$	$4.31 \times 10^{-2} \setminus 1.67 \times 10^{-2}$	$6.09 \times 10^0 \setminus 3.64 \times 10^0$	$6.08 \times 10^{-1} \setminus 8.88 \times 10^{-16}$
F11	Mean	$0.00 \times 10^0$	$0.00 \times 10^0$	$5.90 \times 10^{-4}$	$7.85 \times 10^{-1}$	$1.09 \times 10^{-9}$

Table 9. Cont.

Benchmark		CSHHO	GCHHO	DEPSOASS	Improved GSA	DGOBLFOA
F12	Std	$0.00 \times 10^0$	$0.00 \times 10^0$	$5.90 \times 10^{-4}$	$7.85 \times 10^{-1}$	$1.09 \times 10^{-9}$
	Rank	1	1	4	5	3
	Best\Worst	<b>0.00\0.00</b>	0.00\0.00	$1.97 \times 10^{-2}$ \( $2.27 \times 10^{-5}$	$9.59 \times 10^{-1}$ \( $4.58 \times 10^{-1}$	$5.46 \times 10^{-8}$ \( $0.00 \times 10^0$
	Mean	$5.26 \times 10^{-7}$	$8.05 \times 10^{-8}$	$9.89 \times 10^{-6}$	$1.29 \times 10^0$	$6.98 \times 10^{-1}$
	Std	$1.20 \times 10^{-6}$	$9.54 \times 10^{-8}$	$9.89 \times 10^{-6}$	$1.29 \times 10^0$	$6.98 \times 10^{-1}$
F13	Rank	2	1	3	5	4
	Best\Worst	$5.00 \times 10^{-6}$ \( $1.68 \times 10^{-11}$	$5.02 \times 10^{-7}$ \( $5.77 \times 10^{-9}$	$2.58 \times 10^{-5}$ \( $4.76 \times 10^{-6}$	$2.32 \times 10^0$ \( $6.10 \times 10^{-1}$	$1.01 \times 10^0$ \( $3.67 \times 10^{-1}$
	Mean	$1.64 \times 10^{-5}$	$1.26 \times 10^{-6}$	$1.69 \times 10^{-3}$	$6.34 \times 10^0$	$2.44 \times 10^0$
	Std	$2.86 \times 10^{-5}$	$1.48 \times 10^{-6}$	$1.69 \times 10^{-3}$	$6.34 \times 10^0$	$2.44 \times 10^0$
	Rank	2	1	3	5	4
F14	Best\Worst	$1.23 \times 10^{-4}$ \( $6.14 \times 10^{-10}$	$6.48 \times 10^{-6}$ \( $4.00 \times 10^{-8}$	$1.12 \times 10^{-2}$ \( $6.80 \times 10^{-5}$	$9.67 \times 10^0$ \( $2.90 \times 10^0$	$2.88 \times 10^0$ \( $1.60 \times 10^0$
	Mean	$1.39 \times 10^0$	$9.98 \times 10^{-1}$	$1.52 \times 10^0$	$7.07 \times 10^0$	$4.34 \times 10^0$
	Std	$1.23 \times 10^0$	$5.61 \times 10^{-16}$	$1.52 \times 10^0$	$7.07 \times 10^0$	$4.34 \times 10^0$
	Rank	2	1	3	5	4
	Best\Worst	$5.93 \times 10^0$ \( $9.98 \times 10^{-1}$	$9.98 \times 10^{-1}$ \( $9.98 \times 10^{-1}$	$3.02 \times 10^0$ \( $9.98 \times 10^{-1}$	$1.55 \times 10^1$ \( $1.01 \times 10^0$	$1.18 \times 10^1$ \( $9.98 \times 10^{-1}$
F15	Mean	$3.40 \times 10^{-4}$	$3.17 \times 10^{-4}$	$9.62 \times 10^{-4}$	$1.39 \times 10^{-3}$	$2.61 \times 10^{-2}$
	Std	$3.39 \times 10^{-5}$	$4.50 \times 10^{-5}$	$9.62 \times 10^{-4}$	$1.39 \times 10^{-3}$	$2.61 \times 10^{-2}$
	Rank	2	1	3	4	5
	Best\Worst	<b><math>5.55 \times 10^{-4}</math>\(<math>3.14 \times 10^{-4}</math></b>	$5.41 \times 10^{-4}$ \( $3.07 \times 10^{-4}$	$4.55 \times 10^{-3}$ \( $5.50 \times 10^{-4}$	$3.18 \times 10^{-3}$ \( $1.03 \times 10^{-3}$	$6.88 \times 10^{-2}$ \( $1.81 \times 10^{-3}$
	Mean	<b><math>-1.03 \times 10^0</math></b>	$-1.03 \times 10^0$	$-1.03 \times 10^0$	$-1.02 \times 10^0$	$-1.03 \times 10^0$
F16	Std	<b><math>1.35 \times 10^{-15}</math></b>	$1.35 \times 10^{-15}$	$-1.03 \times 10^0$	$-1.02 \times 10^0$	$-1.03 \times 10^0$
	Rank	1	1	1	5	1
	Best\Worst	<b><math>-1.03 \times 10^0</math>\(<math>-1.03 \times 10^0</math></b>	$-1.03 \times 10^0$ \( $-1.03 \times 10^0$	$-1.03 \times 10^0$ \( $-1.03 \times 10^0$	$-9.22 \times 10^{-1}$ \( $-1.03 \times 10^0$	$-1.03 \times 10^0$ \( $-1.03 \times 10^0$
	Mean	<b><math>3.98 \times 10^{-1}</math></b>	$3.98 \times 10^{-1}$	$3.98 \times 10^{-1}$	$4.08 \times 10^{-1}$	$5.30 \times 10^{-1}$
	Std	<b><math>1.88 \times 10^{-8}</math></b>	$1.12 \times 10^{-16}$	$3.98 \times 10^{-1}$	$4.08 \times 10^{-1}$	$5.30 \times 10^{-1}$
F17	Rank	1	1	1	4	5
	Best\Worst	<b><math>3.98 \times 10^{-13}</math>\(<math>98 \times 10^{-1}</math></b>	$3.98 \times 10^{-1}$ \( $3.98 \times 10^{-1}$	$3.98 \times 10^{-1}$ \( $3.98 \times 10^{-1}$	$4.35 \times 10^{-1}$ \( $3.98 \times 10^{-1}$	$1.70 \times 10^0$ \( $4.00 \times 10^{-1}$
	Mean	<b><math>3.00 \times 10^0</math></b>	$3.00 \times 10^0$	$3.00 \times 10^0$	$4.91 \times 10^0$	$3.50 \times 10^0$
	Std	<b><math>1.15 \times 10^{-7}</math></b>	$0.00 \times 10^0$	$3.00 \times 10^0$	$4.91 \times 10^0$	$3.50 \times 10^0$
	Rank	1	1	1	5	4
F18	Best\Worst	<b><math>3.00 \times 10^0</math>\(<math>3.00 \times 10^0</math></b>	$3.00 \times 10^0$ \( $3.00 \times 10^0$	$3.00 \times 10^0$ \( $3.00 \times 10^0$	$1.25 \times 10^1$ \( $3.01 \times 10^0$	$6.51 \times 10^0$ \( $3.01 \times 10^0$
	Mean	<b><math>-3.86 \times 10^0</math></b>	$-3.86 \times 10^0$	$-3.86 \times 10^0$	$-3.45 \times 10^0$	$-3.50 \times 10^0$
	Std	<b><math>1.83 \times 10^{-3}</math></b>	$2.24 \times 10^{-15}$	$-3.86 \times 10^0$	$-3.45 \times 10^0$	$-3.50 \times 10^0$
	Rank	1	1	1	5	4
	Best\Worst	<b><math>-3.86 \times 10^0</math>\(<math>-3.86 \times 10^0</math></b>	$-3.86 \times 10^0$ \( $-3.86 \times 10^0$	$-3.86 \times 10^0$ \( $-3.86 \times 10^0$	$-2.90 \times 10^0$ \( $-3.86 \times 10^0$	$-2.69 \times 10^0$ \( $-3.85 \times 10^0$
F19	Mean	$-3.13 \times 10^0$	$-3.25 \times 10^0$	$-3.31 \times 10^0$	$-1.69 \times 10^0$	$-2.09 \times 10^0$
	Std	$1.05 \times 10^{-1}$	$5.88 \times 10^{-2}$	$-3.31 \times 10^0$	$-1.69 \times 10^0$	$-2.09 \times 10^0$
	Rank	3	2	1	5	4
	Best\Worst	$-2.74 \times 10^0$ \( $-3.31 \times 10^0$	$-3.20 \times 10^0$ \( $-3.32 \times 10^0$	$-3.18 \times 10^0$ \( $-3.32 \times 10^0$	$-6.72 \times 10^{-1}$ \( $-2.55 \times 10^0$	$-1.11 \times 10^0$ \( $-3.10 \times 10^0$
	Mean	<b><math>-9.65 \times 10^0</math></b>	$-6.07 \times 10^0$	$-6.85 \times 10^0$	$-4.05 \times 10^0$	$-4.49 \times 10^0$
F20	Std	<b><math>1.38 \times 10^0</math></b>	$2.06 \times 10^0$	$-6.85 \times 10^0$	$-4.05 \times 10^0$	$-4.49 \times 10^0$

Table 9. Cont.

Benchmark		CSHHO	GCHHO	DEPSOASS	Improved GSA	DGOBLFOA
F22	Rank	1	3	2	5	4
	Best\Worst	$-5.02 \times 10^0 \setminus -1.02 \times 10^1$	$-5.06 \times 10^0 \setminus -1.02 \times 10^1$	$-2.68 \times 10^0 \setminus -1.02 \times 10^1$	$-1.99 \times 10^0 \setminus -7.93 \times 10^0$	$-2.67 \times 10^0 \setminus -7.66 \times 10^0$
	Mean	$-9.45 \times 10^0$	$-5.94 \times 10^0$	$-1.01 \times 10^1$	$-3.55 \times 10^0$	$-4.08 \times 10^0$
	Std	$1.94 \times 10^0$	$1.97 \times 10^0$	$-1.01 \times 10^1$	$-3.55 \times 10^0$	$-4.08 \times 10^0$
F23	Rank	2	3	1	5	4
	Best\Worst	$-5.03 \times 10^0 \setminus -1.04 \times 10^1$	$-5.09 \times 10^0 \setminus -1.04 \times 10^1$	$-6.48 \times 10^0 \setminus -1.04 \times 10^1$	$-1.93 \times 10^0 \setminus -7.27 \times 10^0$	$-1.67 \times 10^0 \setminus -8.98 \times 10^0$
	Mean	$-9.97 \times 10^0$	$-5.67 \times 10^0$	$-1.03 \times 10^1$	$-4.34 \times 10^0$	$-4.48 \times 10^0$
	Std	$1.46 \times 10^0$	$1.64 \times 10^0$	$-1.03 \times 10^1$	$-4.34 \times 10^0$	$-4.48 \times 10^0$
	Rank	2	3	1	5	4
	Best\Worst	$-5.09 \times 10^0 \setminus -1.05 \times 10^1$	$-5.13 \times 10^0 \setminus -1.05 \times 10^1$	$-5.42 \times 10^0 \setminus -1.05 \times 10^1$	$-2.61 \times 10^0 \setminus -7.03 \times 10^0$	$-2.43 \times 10^0 \setminus -9.12 \times 10^0$

Table 10. The Bonferroni-Holm corrected *p*-values of Wilcoxon’s signed-rank test with the advanced meta-heuristic algorithms.

Benchmark		GCHHO		DEPSOASS		GSA		FOA	
		Corrected <i>p</i> -Value	+ \= \-						
F1	N/A	$3.07 \times 10^{-8}$	1\0\49	$6.57 \times 10^{-8}$	0\0\50	$5.06 \times 10^{-8}$	0\0\50	$2.99 \times 10^{-8}$	2\0\48
F2	N/A	$6.80 \times 10^{-8}$	0\0\50	$6.50 \times 10^{-8}$	0\0\50	$4.99 \times 10^{-8}$	0\0\50	$4.01 \times 10^{-8}$	0\0\50
F3	N/A	$6.73 \times 10^{-8}$	0\0\50	$6.42 \times 10^{-8}$	0\0\50	$4.91 \times 10^{-8}$	0\0\50	$3.93 \times 10^{-8}$	0\0\50
F4	N/A	$6.65 \times 10^{-8}$	0\0\50	$6.35 \times 10^{-8}$	0\0\50	$4.84 \times 10^{-8}$	0\0\50	$3.85 \times 10^{-8}$	0\0\50
F5	N/A	$3.46 \times 10^{-1}$	25\0\25	$6.27 \times 10^{-8}$	0\0\50	$4.76 \times 10^{-8}$	0\0\50	$3.78 \times 10^{-8}$	0\0\50
F6	N/A	$4.84 \times 10^{-3}$	9\0\41	$6.20 \times 10^{-8}$	0\0\50	$4.69 \times 10^{-8}$	0\0\50	$3.70 \times 10^{-8}$	0\0\50
F7	N/A	$4.72 \times 10^{-4}$	12\0\38	$6.12 \times 10^{-8}$	0\0\50	$4.61 \times 10^{-8}$	0\0\50	$3.63 \times 10^{-8}$	0\0\50
F8	N/A	$3.70 \times 10^{-1}$	35\0\15	$6.05 \times 10^{-8}$	0\0\50	$4.53 \times 10^{-8}$	0\0\50	$3.55 \times 10^{-8}$	50\0\0
F9	N/A	$4.00 \times 10^0$	0\50\0	$5.97 \times 10^{-8}$	0\0\50	$4.46 \times 10^{-8}$	0\0\50	$3.48 \times 10^{-8}$	0\0\50
F10	N/A	$4.00 \times 10^0$	0\50\0	$5.89 \times 10^{-8}$	0\0\50	$4.38 \times 10^{-8}$	0\0\50	$4.71 \times 10^{-8}$	0\2\48
F11	N/A	$3.00 \times 10^0$	0\50\0	$5.82 \times 10^{-8}$	0\0\50	$4.31 \times 10^{-8}$	0\0\50	$8.59 \times 10^{-2}$	0\42\8
F12	N/A	$8.60 \times 10^{-1}$	26\0\24	$5.74 \times 10^{-8}$	0\0\50	$4.23 \times 10^{-8}$	0\0\50	$3.40 \times 10^{-8}$	0\0\50
F13	N/A	$7.25 \times 10^{-4}$	36\0\14	$5.67 \times 10^{-8}$	0\0\50	$4.16 \times 10^{-8}$	0\0\50	$3.33 \times 10^{-8}$	0\0\50
F14	N/A	$3.13 \times 10^{-1}$	6\44\0	$1.02 \times 10^{-7}$	2\0\48	$1.20 \times 10^{-3}$	5\1\44	$1.39 \times 10^{-7}$	1\0\49
F15	N/A	$3.99 \times 10^{-6}$	48\0\2	$5.59 \times 10^{-8}$	0\0\50	$4.08 \times 10^{-8}$	0\0\50	$3.25 \times 10^{-8}$	0\0\50
F16	N/A	$2.00 \times 10^0$	0\50\0	$5.52 \times 10^{-8}$	0\0\50	$6.75 \times 10^{-8}$	0\0\50	$3.17 \times 10^{-4}$	0\26\24
F17	N/A	$8.75 \times 10^{-1}$	4\46\0	$5.44 \times 10^{-8}$	0\0\50	$4.71 \times 10^{-8}$	0\2\48	$3.17 \times 10^{-8}$	0\0\50
F18	N/A	$8.75 \times 10^{-1}$	4\46\0	$5.37 \times 10^{-8}$	0\0\50	$6.88 \times 10^{-8}$	0\0\50	$3.10 \times 10^{-8}$	0\0\50
F19	N/A	$6.88 \times 10^{-8}$	50\0\0	$5.29 \times 10^{-8}$	0\0\50	$3.04 \times 10^{-7}$	43\0\7	$3.02 \times 10^{-8}$	0\0\50
F20	N/A	$1.10 \times 10^{-6}$	45\0\5	$5.21 \times 10^{-8}$	0\0\50	$2.99 \times 10^{-8}$	49\0\1	$3.08 \times 10^{-8}$	1\0\49
F21	N/A	$1.83 \times 10^{-5}$	13\0\37	$3.08 \times 10^{-8}$	2\0\48	$1.89 \times 10^{-3}$	19\0\31	$3.26 \times 10^{-8}$	2\0\48
F22	N/A	$4.72 \times 10^{-4}$	13\0\37	$5.14 \times 10^{-8}$	0\0\50	$1.55 \times 10^{-3}$	44\0\6	$2.95 \times 10^{-8}$	0\0\50
F23	N/A	$2.67 \times 10^{-7}$	9\0\41	$3.07 \times 10^{-8}$	1\0\49	$2.46 \times 10^{-5}$	47\0\3	$2.97 \times 10^{-8}$	1\0\49

**Table 11.** Average rankings obtained by advanced meta-heuristic in the Friedman test, and the best result is shown in boldface.

Chaotic Mapping	Average Rankings
<b>CSHHO</b>	<b>1.70</b>
GCHHO	1.83
DEPSOASS	2.76
GSA	4.87
FOA	3.85

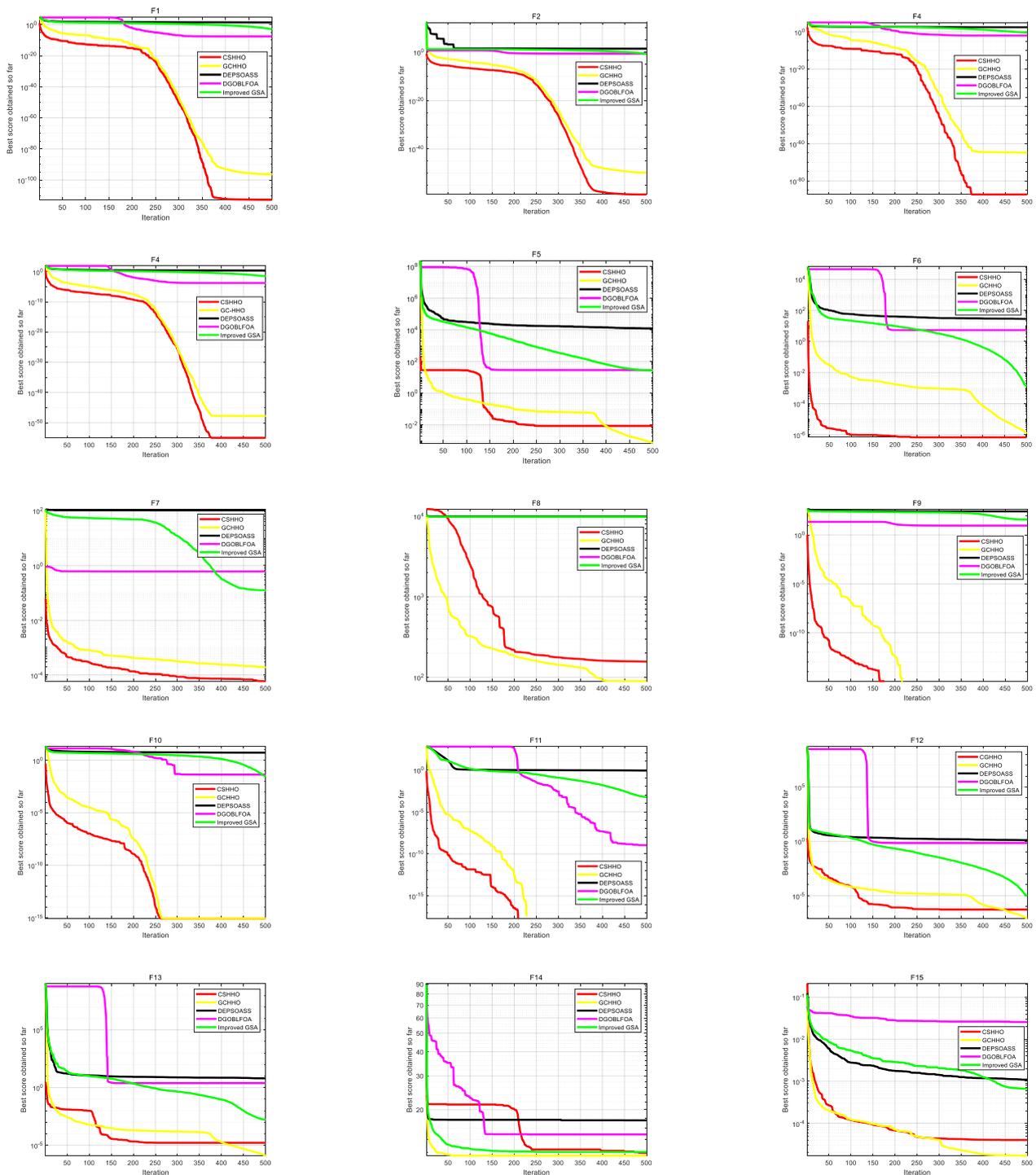
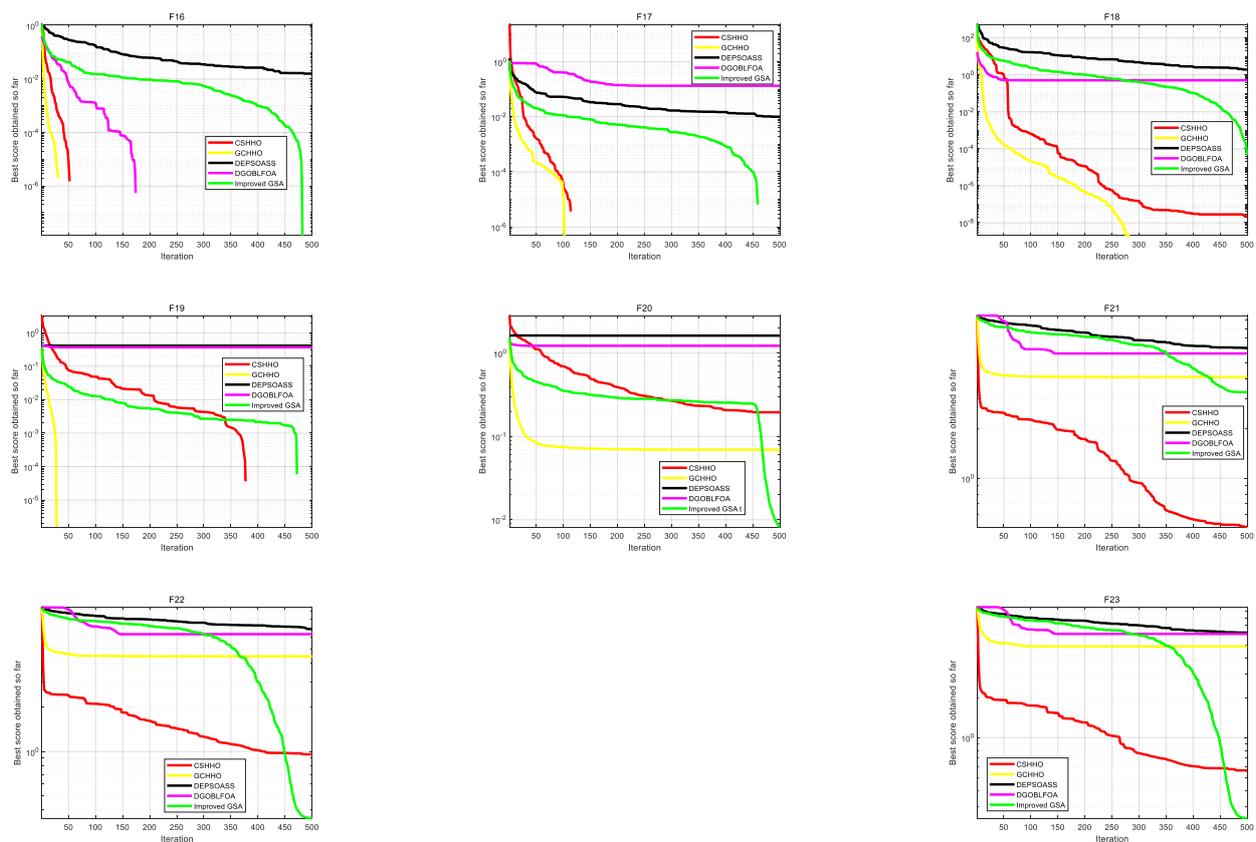


Figure 5. Cont.



**Figure 5.** Convergence curves of CSHHO and the advanced meta-heuristic algorithms.

#### 4.2.4. Scalability Test on CSHHO

Dimensional data are an important basis for analyzing the influence of the number of factors to be optimized on the algorithm, and the purpose of the scalability test is to further verify the overall performance and stability of the optimization model. The experimental subjects in this section are CSHHO, HHO. 29 CEC2017 functions [72] based on 50 and 100 dimensions, respectively, are used for scalability experiments. In this experiment, the experimental parameters and the experimental environment are consistent with the previous experiments except that the dimensionality settings are different from the previous experiments, and Table 12 shows the experimental results using Mean and Std.

The best numerical results in CSHHO and HHO are set in bold, and the numerical results of both equivalents are in bold. Under the UM function (F1–F2), the numerical results of CSHHO are overall better than those of HHO, and CSHHO continues to maintain some advantage as the number of dimensions increases; under the MM function test (F3–F9), CSHHO performs better than HHO in 50 and 100 dimensions. The CSHHO's performance in the 50 and 100 dimensions was generally better than that of HHO in the 50 dimensions in F6 and F7 and in the 100 dimensions in F4. In the hybrid function (F10–F19), CSHHO performs better in the rest of the test sets, except for F11 and F12, where it performs lower than HHO. In the composition function (F20–F29), it still has good ability, and the accuracy in functions F21, F22 and F26 is higher than HHO, and the effect in functions F22, F23, F24, F26 is the same as HHO, and the effect in functions F23 (50 dimensions), F26 (100 dimensions) is lower than HHO. In general, compared with HHO, CSHHO can better balance the exploration and exploitation process as the number of dimensions increases.

Table 12. Experimental results of scalability tests in different dimensions.

Benchmark	Metric	50		100	
		CSHHO	HHO	CSHHO	HHO
F1	Mean\Std	$3.49 \times 10^7 \setminus 1.19 \times 10^8$	$3.30 \times 10^8 \setminus 9.91 \times 10^7$	$1.08 \times 10^{10} \setminus 2.06 \times 10^9$	$1.10 \times 10^{10} \setminus 2.09 \times 10^9$
F2	Mean\Std	$9.88 \times 10^4 \setminus 1.66 \times 10^4$	$1.03 \times 10^5 \setminus 1.38 \times 10^4$	$2.78 \times 10^5 \setminus 1.73 \times 10^4$	$2.90 \times 10^5 \setminus 1.42 \times 10^4$
F3	Mean\Std	$8.65 \times 10^2 \setminus 1.16 \times 10^2$	$9.00 \times 10^2 \setminus 1.52 \times 10^2$	$2.89 \times 10^2 \setminus 4.96 \times 10^2$	$3.01 \times 10^3 \setminus 5.84 \times 10^2$
F4	Mean\Std	$8.95 \times 10^2 \setminus 2.82 \times 10^1$	$8.96 \times 10^2 \setminus 3.47 \times 10^1$	$1.56 \times 10^3 \setminus 4.55 \times 10^1$	$1.56 \times 10^3 \setminus 5.35 \times 10^1$
F5	Mean\Std	$6.71 \times 10^2 \setminus 4.36 \times 10^0$	$6.74 \times 10^2 \setminus 4.16 \times 10^0$	$6.85 \times 10^2 \setminus 2.82 \times 10^0$	$6.85 \times 10^2 \setminus 3.25 \times 10^0$
F6	Mean\Std	$1.84 \times 10^3 \setminus 9.07 \times 10^1$	$1.84 \times 10^3 \setminus 7.88 \times 10^1$	$3.71 \times 10^3 \setminus 1.41 \times 10^2$	$3.69 \times 10^3 \setminus 1.46 \times 10^2$
F7	Mean\Std	$1.20 \times 10^3 \setminus 3.34 \times 10^1$	$1.20 \times 10^3 \setminus 3.11 \times 10^1$	$2.01 \times 10^3 \setminus 7.14 \times 10^1$	$2.01 \times 10^3 \setminus 6.02 \times 10^1$
F8	Mean\Std	$2.52 \times 10^4 \setminus 2.86 \times 10^3$	$2.54 \times 10^4 \setminus 3.63 \times 10^3$	$5.24 \times 10^4 \setminus 5.53 \times 10^3$	$5.72 \times 10^4 \setminus 5.52 \times 10^3$
F9	Mean\Std	$9.41 \times 10^3 \setminus 9.83 \times 10^2$	$9.49 \times 10^3 \setminus 8.76 \times 10^2$	$2.21 \times 10^4 \setminus 1.46 \times 10^3$	$2.28 \times 10^4 \setminus 2.12 \times 10^3$
F10	Mean\Std	$1.65 \times 10^3 \setminus 1.47 \times 10^2$	$1.66 \times 10^3 \setminus 1.21 \times 10^2$	$5.03 \times 10^4 \setminus 1.59 \times 10^4$	$5.06 \times 10^4 \setminus 1.06 \times 10^4$
F11	Mean\Std	$2.17 \times 10^8 \setminus 1.28 \times 10^8$	$1.88 \times 10^8 \setminus 1.05 \times 10^8$	$1.58 \times 10^9 \setminus 5.29 \times 10^8$	$1.68 \times 10^9 \setminus 5.46 \times 10^8$
F12	Mean\Std	$3.82 \times 10^6 \setminus 1.44 \times 10^6$	$5.20 \times 10^6 \setminus 4.28 \times 10^6$	$1.75 \times 10^7 \setminus 7.23 \times 10^6$	$1.62 \times 10^7 \setminus 6.97 \times 10^6$
F13	Mean\Std	$1.54 \times 10^6 \setminus 1.01 \times 10^6$	$2.07 \times 10^6 \setminus 2.53 \times 10^6$	$4.81 \times 10^6 \setminus 1.82 \times 10^6$	$5.41 \times 10^6 \setminus 1.64 \times 10^6$
F14	Mean\Std	$6.00 \times 10^5 \setminus 2.65 \times 10^5$	$6.35 \times 10^5 \setminus 2.81 \times 10^5$	$3.36 \times 10^6 \setminus 9.38 \times 10^5$	$4.35 \times 10^6 \setminus 4.49 \times 10^6$
F15	Mean\Std	$4.38 \times 10^3 \setminus 6.23 \times 10^2$	$4.52 \times 10^3 \setminus 6.15 \times 10^2$	$8.55 \times 10^3 \setminus 1.00 \times 10^3$	$8.56 \times 10^3 \setminus 8.52 \times 10^2$
F16	Mean\Std	$3.78 \times 10^3 \setminus 3.76 \times 10^2$	$3.90 \times 10^3 \setminus 4.41 \times 10^2$	$6.73 \times 10^3 \setminus 7.03 \times 10^2$	$6.84 \times 10^3 \setminus 7.62 \times 10^2$
F17	Mean\Std	$5.39 \times 10^5 \setminus 4.79 \times 10^6$	$4.63 \times 10^6 \setminus 4.40 \times 10^6$	$5.75 \times 10^6 \setminus 2.56 \times 10^6$	$6.62 \times 10^6 \setminus 3.52 \times 10^6$
F18	Mean\Std	$5.82 \times 10^5 \setminus 6.11 \times 10^5$	$1.12 \times 10^6 \setminus 7.19 \times 10^5$	$1.20 \times 10^7 \setminus 4.90 \times 10^6$	$1.55 \times 10^7 \setminus 6.97 \times 10^6$
F19	Mean\Std	$3.63 \times 10^2 \setminus 3.07 \times 10^2$	$3.39 \times 10^3 \setminus 3.20 \times 10^2$	$6.05 \times 10^3 \setminus 5.46 \times 10^2$	$6.06 \times 10^3 \setminus 4.89 \times 10^2$
F20	Mean\Std	$2.46 \times 10^3 \setminus 8.32 \times 10^1$	$2.85 \times 10^3 \setminus 6.82 \times 10^1$	$4.17 \times 10^3 \setminus 1.82 \times 10^2$	$4.13 \times 10^3 \setminus 1.86 \times 10^2$
F21	Mean\Std	$1.03 \times 10^4 \setminus 1.09 \times 10^3$	$1.14 \times 10^4 \setminus 1.04 \times 10^3$	$2.56 \times 10^4 \setminus 1.24 \times 10^3$	$2.56 \times 10^4 \setminus 1.48 \times 10^3$
F22	Mean\Std	$3.74 \times 10^3 \setminus 2.31 \times 10^2$	$3.74 \times 10^3 \setminus 1.77 \times 10^2$	$5.35 \times 10^3 \setminus 3.78 \times 10^2$	$5.35 \times 10^3 \setminus 2.91 \times 10^2$
F23	Mean\Std	$4.20 \times 10^3 \setminus 2.15 \times 10^2$	$4.18 \times 10^3 \setminus 1.76 \times 10^2$	$7.25 \times 10^3 \setminus 5.84 \times 10^2$	$7.25 \times 10^3 \setminus 4.62 \times 10^2$
F24	Mean\Std	$3.28 \times 10^3 \setminus 6.13 \times 10^1$	$3.30 \times 10^3 \setminus 7.30 \times 10^1$	$4.73 \times 10^3 \setminus 2.76 \times 10^2$	$4.73 \times 10^3 \setminus 2.60 \times 10^2$
F25	Mean\Std	$1.08 \times 10^4 \setminus 1.79 \times 10^3$	$1.10 \times 10^4 \setminus 1.63 \times 10^3$	$2.85 \times 10^4 \setminus 2.46 \times 10^3$	$2.86 \times 10^4 \setminus 4.26 \times 10^3$
F26	Mean\Std	$4.44 \times 10^3 \setminus 5.43 \times 10^2$	$4.44 \times 10^3 \setminus 3.71 \times 10^2$	$5.70 \times 10^3 \setminus 1.25 \times 10^3$	$5.43 \times 10^3 \setminus 6.12 \times 10^2$
F27	Mean\Std	$3.82 \times 10^3 \setminus 1.43 \times 10^2$	$3.89 \times 10^3 \setminus 1.77 \times 10^2$	$5.79 \times 10^3 \setminus 4.35 \times 10^2$	$5.86 \times 10^3 \setminus 4.88 \times 10^2$
F28	Mean\Std	$6.09 \times 10^3 \setminus 7.55 \times 10^2$	$6.38 \times 10^3 \setminus 7.33 \times 10^2$	$1.10 \times 10^4 \setminus 1.01 \times 10^3$	$1.11 \times 10^4 \setminus 9.60 \times 10^2$
F29	Mean\Std	$5.23 \times 10^7 \setminus 1.82 \times 10^7$	$5.25 \times 10^7 \setminus 1.55 \times 10^7$	$1.52 \times 10^8 \setminus 6.75 \times 10^7$	$1.54 \times 10^8 \setminus 7.05 \times 10^7$

## 5. Engineering Application

In this chapter, the proposed CSHHO is applied to model the reactive power output of a synchronous condenser. Due to the large number of UHV DC transmission projects in the power system, to ensure DC to DC power consumption and peaking demand, the scale of conventional units on the receiving AC grid is reduced, the dynamic reactive power support capability of the system is weakened, and the voltage stability margin is reduced [81]. This requires dynamic reactive power compensation devices to have instantaneous reactive power support characteristics in case of system failure, and the synchronous condenser to have reactive power output characteristics to meet the dynamic reactive power compensation requirements of the grid [82]. Modeling the reactive power support capability of a synchronous condenser is of great theoretical significance and practical value for the reactive power control of converter stations in high-voltage DC transmission systems with synchronous condenser.

The existing research methods for modeling the reactive power output of synchronous condenser are the mathematical analytical model calculation method and the experimental result fitting method [83–86] and both require large computational effort and have low accuracy, but few papers have studied the application of LSSVM in modeling the reactive power output of synchronous condenser. The advantages of the least squares support vector machine (LSSVM) are that it is less likely to fall into local minima and has high generalization ability [87]. Researchers have used various intelligent optimization algorithms to find the optimal results of kernel function parameters and regularization parameters, including the GA [88], Particle Swarm Optimization Algorithm (PSO) [89], Free Search Algorithm (FS) [90], Ant Colony Optimization Algorithm (ACO) [91], ABC Algorithm [92], GWO algorithm [93] and Backtracking Search Optimization Algorithm (BSA) [94], etc. However, the traditional swarm optimization algorithm is prone to the defects of falling into local optimum and low convergence accuracy in the search process. According to the results above, the CSHHO not only reduces the probability of the algorithm falling into local optimum and improves the convergence accuracy of the algorithm, but also has the advantages of the basic Harris Hawk optimization algorithm: 1. the steadiness of the searching cores; 2. the fruitfulness in the initial iterations; 3. the progressive selection scheme [5].

This paper proposes a CSHHO-LSSVM-based reactive power modeling method based on the numerical characteristics and global search capability of CSHHO. The optimal values of the penalty parameters, kernel function parameters, and loss function parameters of the LSSVM are found by using CSHHO to build the CSHHO-LSSVM model for the reactive power output of the synchronous condenser.

### 5.1. Principle of LSSVM

Support Vector Machine (SVM) is an ML method based on statistical learning theory, with kernel function as the core, which implicitly maps the data in the original space to the high-dimensional feature space, then finds the linear relationship in the feature space [87].

LSSVM is a regression algorithm that extends the basic SVM. Compared with the SVM algorithm, LSSVM requires fewer parameters and is more stable. LSSVM simplifies the complex constraints, which makes the improved SVM more capable of handling data. Moreover, by setting the error sum of squares as the loss function of the algorithm, LSSVM enhances the performance of regression prediction and improves the prediction accuracy. Simultaneously, the complexity of the algorithm is reduced, which reduces the processing time of the algorithm and provides more flexibility. LSSVM uses a nonlinear model on basis of SVM:

$$f(x) = (\omega, \phi(x)) + b \quad (28)$$

The input data were  $(x_i, y_i) i = 1, \dots, l$  where  $x_i \in R^d$  denoted the different elements,  $d$  denoted the dimension,  $y_i \in R$  the expected value of the output, and  $l$  the total number

of inputs.  $\phi(x)$  denoted the mapping function. In summary, the LSSVM optimization objective was:

$$\min \frac{1}{2} \|\omega\|^2 + \frac{1}{2} \gamma \sum_{i=1}^l e_i^2 \quad (29)$$

$$s.t. \omega^T \phi(x_i) + b + e_i = y_i (i = 1, \dots, l).$$

where  $e_i$  denoted the error, the magnitude of which determined the prediction accuracy;  $e \in R^{l \times 1}$  denoted the error vector,  $\gamma$  denoted the regularization parameter  $r$ , which determined the magnitude of the error. Adding a Lagrangian multiplier to Equation (29),  $\lambda \in R^{l \times 1}$ , Equation (30) was expressed as:

$$\min J = \frac{1}{2} \|\omega\|^2 + \frac{1}{2} \gamma \sum_{i=1}^l e_i^2 - \sum_{i=1}^l \lambda_i (\omega^T \phi(x_i) + b + e_i - y_i) \quad (30)$$

From the KKT condition, we obtained:

$$\begin{cases} \frac{\partial J}{\partial \omega} = 0 \rightarrow \sum_{i=1}^l \lambda_i \phi(x_i) \\ \frac{\partial J}{\partial b} = 0 \rightarrow \sum_{i=1}^l \lambda_i = 0 \\ \frac{\partial J}{\partial e_i} = 0 \rightarrow \lambda_i = \gamma e_i, i = 1, 2, \dots, l \\ \frac{\partial J}{\partial \lambda_i} = 0 \rightarrow \omega^T \phi(x_i) + b + e_i - y_i = 0, i = 1, 2, \dots, l \end{cases} \quad (31)$$

By eliminating the slack variables  $e_i$  and weight vectors  $\omega$ , the optimization problem was linearized:

$$\begin{bmatrix} 0 & Q^T \\ Q & K + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} b \\ A \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \quad (32)$$

where  $A = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ ,  $Q = [1, 1, \dots, 1]^T$ , was an  $l \times 1$  dimensional column vector,  $Y = [y_1, y_2, \dots, y_N]^T$ . According to the Mercer condition,  $K$  denoted a kernel function:  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$   $i, j = 1, 2, \dots, N$ . The Radial Basis Function kernel function was chosen for the model:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \sigma > 0 \quad (33)$$

Therefore, the nonlinear prediction model was expressed by Equation (34):

$$y = \sum_{i=1}^l \lambda_i K(x_i, x) + b \quad (34)$$

When predicting with least squares support vector regression models, the penalty factor and radial basis kernel function parameters were the two parameters to be solved.

## 5.2. Simulation and Verification

The reactive power regulation results of a synchronous condenser based on PSCAD/EMTDC simulation software were used as training samples and test samples. Table 11 shows the main parameters. The data with serial numbers 9, 14, 26 and 35 in Table 11 were taken as the test samples, and the rest were the training samples.

First, the data were preprocessed, and the LSSVM was trained by using CSHHO to find the penalty parameters, kernel function parameters and optimal parameters of the loss function  $(\gamma, \sigma, S)$ , and the LSSVM was predicted by applying the test sample to the LSSVM, and the regression fitted prediction model was output. The algorithm flow is shown in Figure 6. Figure 7 compares the output results of the LSSVM model of the test sample and the CSHHO-LSSVM model were compared with those of the test sample and

the errors of the LSSVM model of the test sample and the CSHHO-LSSVM model. The output regression of CSHHO-LSSVM model was better fitted and had higher accuracy and radial basis kernel function parameters.

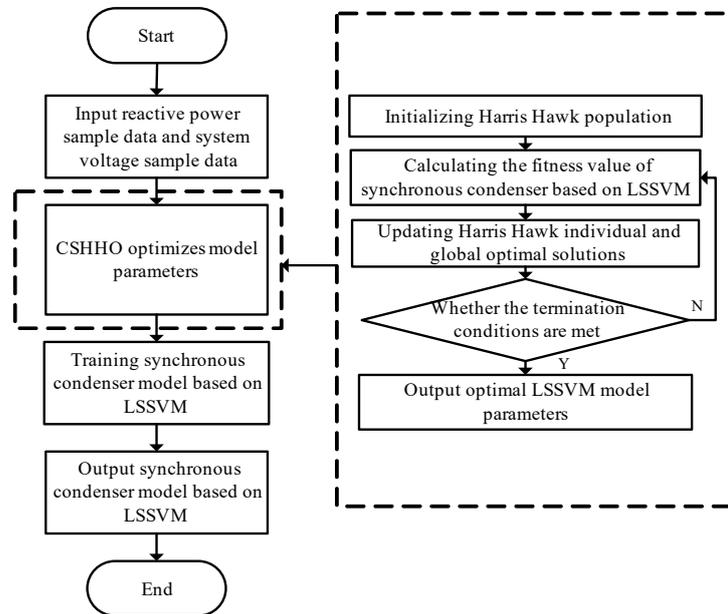


Figure 6. CSHHO Optimizes Synchronous condenser Reactive Power Output based LSSVM.

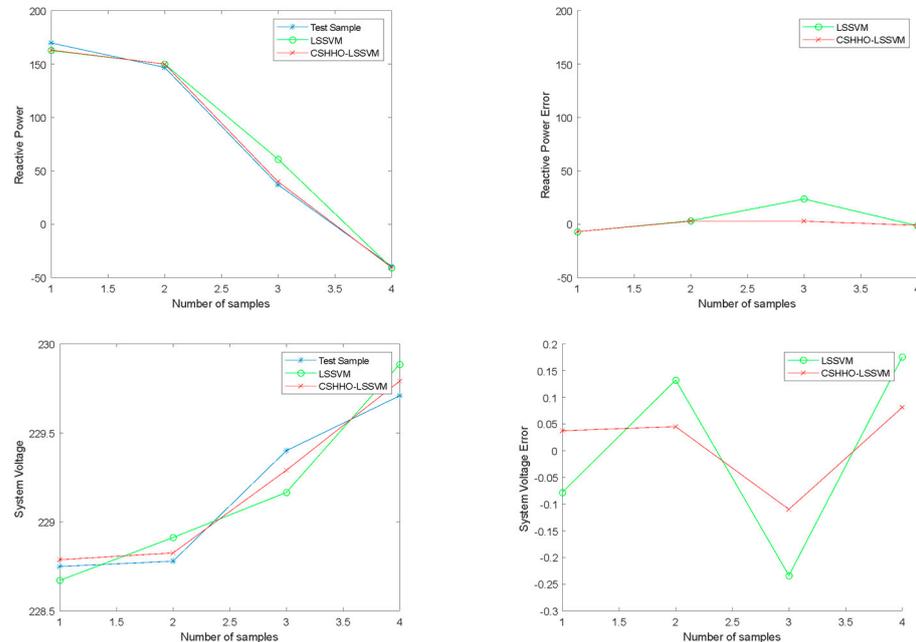


Figure 7. Output and error comparison diagram of LSSVM model and CSHHO-LSSVM model.

To verify the generalization ability of the CSHHO-LSSVM model, it was evaluated by absolute deviation, as shown in Table 13. The table shows the absolute deviation range of CSHHO-LSSVM model from 0.0123 to 0.989, indicating that the accuracy of the CSHHO-LSSVM model was high.

Table 13. Training samples and test samples.

Input Sample		Output Sample	
Excitation Current/A	Exciting Voltage/V	Reactive Power/Mvar	System Voltage/kV
25.488	29.267	305.0	228.4
23.364	26.103	261.5	228.2
21.594	24.521	232.5	228.5
20.154	22.939	212.5	228.5
19.6824	22.148	197.5	228.6
18.8328	21.159	186.4	228.6
18.408	20.408	177.4	228.7
17.9124	19.9728	170.2	228.75
17.7	19.775	163.8	228.8
17.346	19.3795	158.5	228.8
16.992	18.984	154.2	228.8
16.7088	18.7863	150.5	228.8
16.638	18.5727	147.0	228.78
16.461	18.3987	144.2	228.85
16.1424	17.9715	137.4	228.84
15.6822	17.402	131.1	228.9
15.222	17.0065	125.0	228.9
14.868	16.611	118.35	229
14.2308	15.9782	108.75	229
12.9564	14.5346	88.5	229.1
12.39	13.8425	78.3	229.1
11.682	12.953	67.8	229.2
10.974	12.458	57.5	229.3
10.3368	11.4695	47.5	229.3
9.912	11.074	37.0	229.4
8.9916	10.283	27.5	229.4
8.4252	9.1756	17.2	229.48
7.8588	8.8592	7.5	229.5
7.2924	8.3055	1.75	229.57
6.9738	7.91	−2.5	229.6
6.372	6.9213	−11.7	229.6
5.8056	6.7235	−21.0	229.62
5.664	6.4467	−30.5	229.7
4.956	5.1415	−39.6	229.71
4.248	4.5483	−48.5	229.8
3.54	4.351	−57.5	229.875
2.832	3.164	−57.5	229.875
2.124	2.4521	−66.5	229.9

From the reactive power and system voltage simulation results of the synchronous condenser in Tables 14–16, we can see that the maximum absolute error of the reactive power simulation result of CSHHO-LSSVM model was 0.989 Mvar, and the maximum absolute error of the system voltage simulation result was 0.0415 kV, which were smaller than the simulation results of LSSVM model, indicating that the CSHHO-LSSVM model had higher accuracy and better regression fitting performance. The LSSVM model was more accurate and had better regression fitting performance.

**Table 14.** CSHHO-LSSV Model generalization ability verification.

		Prediction Value	Sample Value	Absolute Error	Relative Error
9	Reactive Power/MVar	163.0515	163.8	−0.7485	−0.4570
	System Voltage/kV	228.7877	228.8	−0.0123	−0.0054
14	Reactive Power/MVar	144.5989	144.2	0.3989	0.2766
	System Voltage/kV	228.8882	228.85	0.0382	0.0167
26	Reactive Power/MVar	28.1322	27.6	0.5322	1.9282
	System Voltage/kV	229.4014	229.45	−0.0486	−0.0212
35	Reactive Power/MVar	−40.5890	−39.6	−0.989	2.4975
	System Voltage/kV	229.7915	229.75	0.0415	0.0181

**Table 15.** Reactive power simulation results comparison.

		Prediction Value	Sample Value	Absolute Error	Relative Error
LSSVM	9	162.9631	163.8	−0.8369	−0.5128
	35	−40.8531	−39.6	−1.2531	3.1649
CSHHO–LSSVM	9	163.0515	163.8	−0.7485	−0.4570
	35	−40.5890	−39.6	−0.989	2.4975

**Table 16.** System voltage simulation results comparison.

		Prediction Value	Sample Value	Absolute Error	Relative Error
LSSVM	9	228.6717	228.8	−0.1283	−0.0561
	35	229.8859	229.75	0.1359	0.0591
CSHHO–LSSVM	9	228.7877	228.8	−0.0123	−0.0054
	35	229.7915	229.75	0.0415	0.0181

## 6. Conclusions

Here, we analyzed the shortcomings of the basic HHO algorithm and applied the chaotic mapping population initialization, adaptive weighting, variable spiral position update and optimal neighborhood disturbance mechanisms to the classical HHO algorithm, in which the Gauss chaotic mapping population initialization increased the coverage of the solution space by the initial solution of the algorithm, the adaptive weighting mechanism sped up the movement of Harris hawk populations to the optimal solution, and the variable spiral position update increased the ability of Harris hawk populations. The optimal neighborhood disturbance mechanism helped the improved algorithm to increase the algorithm's global search capability and avoided premature maturity. To verify the optimal performance of the four strategies, the experiments were separated into two groups.

First, seven commonly used chaotic mappings were selected for the population initialization of the HHO algorithm, including Sinusoidal, Tent, Kent, Cubic, Logistic, Gauss, and Circle mappings. The HHO algorithm's performance after population initialization of each of these seven mappings was evaluated. The HHO algorithm's performance after population initialization of Gauss mapping was significantly better than that of the HHO algorithm after population initialization of other mappings in terms of solution accuracy. Second, based on the results of the first set of experiments, the Gauss mapping was used for population initialization, and adaptive weights, variable spiral position update, and optimal neighborhood disturbance mechanisms were introduced into the algorithm after population initialization. Next, CSHHO was compared with other classical algorithms including WOA, SCA, CSO and advanced algorithms including GCHHO and DEPSSOASS and Improved GSA and DGOBLFOA based on 23 classical test functions and the means and standard deviations of all algorithms were analyzed. Subsequently, each algorithm's

performance was evaluated comprehensively using Friedman's test and the Bonferroni–Holm corrected Wilcoxon signed-rank test with 5% confidence level, where numerical analysis concluded that CSHHO outperformed the other algorithms. In detail, analyzing the experimental results of this work, in the population initialization phase, Gauss chaos mapping had the best results in F2, F6, F12, F17, and F23 test functions, and comparing the results of the remaining six chaotic mappings, Gauss chaos mapping obtained the most optimal solutions; CSHHO algorithm outperformed HHO in 17 benchmark functions out of 23 classical test functions, outperformed WOA in 21 results, SCA in 23 results, and CSO in 22 experiments. It outperformed GCHHO in 9 results. Meanwhile, in the statistical experiments of advanced meta-heuristic and classical meta-heuristic, the ARVs obtained by CSHHO were 1.93 and 2.57, respectively, which were lower than the values obtained by other pairwise meta-heuristics in the same group of experiments. Additionally, dimensional scalability tests were conducted for CSHHO on the IEEECEC2017 dataset, including 50 and 100 dimensions, and the results showed that the improved optimizer effectively handled high-dimensional data with good stability. Meanwhile, in the statistical experiments of advanced meta-heuristic and classical meta-heuristic, CSHHO obtained ARVs of 1.70 and 2.57, respectively, which were lower than the values obtained by other meta-heuristic algorithms in the same set of experiments. Furthermore, dimensional scalability tests were conducted for CSHHO on the IEEE CEC 2017 dataset, including 50 and 100 dimensions, and the results showed that the improved optimizer effectively handled high-dimensional data with excellent stability.

Here, the CSHHO algorithm was also applied to the engineering problem of reactive power output modeling of the synchronous condenser. In view of the defects of the many calculations and low accuracy of the traditional reactive power output modeling method of the synchronous condenser, CSHHO-LSSVM was used to model the reactive power output of the synchronous condenser based on the advantages of LSSVM, which was not easy to fall into local minimum and had strong generalization ability, and CSHHO had high search accuracy and strong global search ability. The excitation current and excitation voltage of the synchronous condenser were used as the input of the LSSVM model, and the reactive power and system voltage were used as the LSSVM model's output. CSHHO was used to find the optimal values of the penalty parameter, kernel function parameter, and loss function parameter of LSSVM. The experiment showed that the CSHHO-LSSVM model had better accuracy and better regression fitting performance compared with LSSVM.

In future work, we will try to improve the convergence speed and search accuracy of the algorithm and balance the exploration and exploitation phases of the algorithm to obtain better search performance. Additionally, the next step will be to investigate how CSHHO can be used to solve multi-objective optimization problems. In addition, CSHHO can also be used for evolutionary ML, such as extreme learning machines and parameter tuning of convolution neural networks. Other problems include grid scheduling and 3D multi-objective tracking.

**Author Contributions:** Methodology, C.W.; resources, S.J.; funding acquisition, S.J., R.G., Y.L. and Q.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (No. 61871318), the Shaanxi Provincial Key Research and Development Project (No. 2019GY-099) and Open project of Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing (No. 2020CP10).

**Institutional Review Board Statement:** The study did not involve human and animals.

**Informed Consent Statement:** The study did not involve human and animals.

**Data Availability Statement:** In the paper, all the data generation information has been given in detail in the related chapter.

**Acknowledgments:** The author thanks the referees for detailed and constructive criticism of the original manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Appendix A**

**Table A1.** The classical 23 test functions.

Function	Dimensions	Range	$f_{\min}$
unimodal benchmark functions			
$f_1(x) = \sum_{i=1}^n x_i^2$	30,100, 500, 1000	[100, 100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30,100, 500, 1000	[10, 10]	0
$f_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	30,100, 500, 1000	[100, 100]	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30,100, 500, 1000	[100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30,100, 500, 1000	[30, 30]	0
$f_6(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$	30,100, 500, 1000	[100, 100]	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30,100, 500, 1000	[128, 128]	0
multimodal benchmark functions			
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30,100, 500, 1000	[500, 500]	$-418.9829 \times n$
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30,100, 500, 1000	[5.12, 5.12]	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30,100, 500, 1000	[32, 32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,100, 500, 1000	[600, 600]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30,100, 500, 1000	[50, 50]	0
$y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30,100, 500, 1000	[50, 50]	0
fixed-dimension multimodal benchmark functions			
$f_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^j (x_i - a_{ij})^6} \right)^{-1}$	2	[-65, 65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5, 5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[-5, 5]	0.398
$f_{18}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times \left[ 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	2	[-2, 2]	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3	[1, 3]	-3.86
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	6	[0, 1]	-3.32
$f_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

## References

1. Kundu, T.; Garg, H. A Hybrid ITLHHO Algorithm for Numerical and Engineering Optimization Problems. *Int. J. Intell. Syst.* **2021**, *36*, 1–81. [[CrossRef](#)]
2. Garg, H. A Hybrid GSA-GA Algorithm for Constrained Optimization Problems. *Inf. Sci.* **2019**, *478*, 499–523. [[CrossRef](#)]
3. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [[CrossRef](#)]
4. Wu, Y. A Survey on Population-Based Meta-Heuristic Algorithms for Motion Planning of Aircraft. *Swarm Evol. Comput.* **2021**, *62*, 100844. [[CrossRef](#)]
5. Alabool, H.M.; Alarabiat, D.; Abualigah, L.; Heidari, A.A. Harris hawks optimization: A comprehensive review of recent variants and applications. *Neural Comput. Appl.* **2021**, *33*, 8939–8980. [[CrossRef](#)]
6. Vasant, P. *Handbook of Research on Artificial Intelligence Techniques and Algorithms*; IGI Global: Hershey, PA, USA, 2015. [[CrossRef](#)]
7. Simon, D. *Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2013; ISBN 978-0-470-93741-9.
8. Dréo, J. (Ed.) *Metaheuristics for Hard Optimization: Methods and Case Studies*; Springer: Berlin, Germany, 2006; ISBN 978-3-540-23022-9.
9. Salcedo-Sanz, S. Modern Meta-Heuristics Based on Nonlinear Physics Processes: A Review of Models and Design Procedures. *Phys. Rep.* **2016**, *655*, 1–70. [[CrossRef](#)]
10. Adeyanju, O.M.; Canha, L.N. Decentralized Multi-Area Multi-Agent Economic Dispatch Model Using Select Meta-Heuristic Optimization Algorithms. *Electr. Power Syst. Res.* **2021**, *195*, 107128. [[CrossRef](#)]
11. Zhu, M.; Chu, S.-C.; Yang, Q.; Li, W.; Pan, J.-S. Compact Sine Cosine Algorithm with Multigroup and Multistrategy for Dispatching System of Public Transit Vehicles. *J. Adv. Transp.* **2021**, *2021*, 1–16. [[CrossRef](#)]
12. Fu, X.; Fortino, G.; Li, W.; Pace, P.; Yang, Y. WSNs-Assisted Opportunistic Network for Low-Latency Message Forwarding in Sparse Settings. *Future Gener. Comput. Syst.* **2019**, *91*, 223–237. [[CrossRef](#)]
13. Dhiman, G. SSC: A Hybrid Nature-Inspired Meta-Heuristic Optimization Algorithm for Engineering Applications. *Knowl.-Based Syst.* **2021**, *222*, 106926. [[CrossRef](#)]
14. Han, Y.; Gu, X. Improved Multipopulation Discrete Differential Evolution Algorithm for the Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.* **2021**, *60*, 5530–5547. [[CrossRef](#)]
15. Loukil, T.; Teghem, J.; Tuytens, D. Solving multi-objective production scheduling problems using meta-heuristics. *Eur. J. Oper. Res.* **2005**, *161*, 42–61. [[CrossRef](#)]
16. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, 1–15. [[CrossRef](#)]
17. Li, Y.; Liu, J.; Tang, Z.; Lei, B. Deep Spatial-Temporal Feature Fusion From Adaptive Dynamic Functional Connectivity for MCI Identification. *IEEE Trans. Med. Imaging.* **2020**, *39*, 2818–2830. [[CrossRef](#)] [[PubMed](#)]
18. Corazza, M.; di Tollo, G.; Fasano, G.; Pesenti, R. A Novel Hybrid PSO-Based Metaheuristic for Costly Portfolio Selection Problems. *Ann. Oper. Res.* **2021**, *304*, 109–137. [[CrossRef](#)]
19. Gaspero, L.D.; Tollo, G.D.; Roli, A.; Schaerf, A. Hybrid Metaheuristics for Constrained Portfolio Selection Problems. *Quant. Financ.* **2011**, *11*, 1473–1487. [[CrossRef](#)]
20. Shen, L.; Chen, H.; Yu, Z.; Kang, W.; Zhang, B.; Li, H.; Yang, B.; Liu, D. Evolving Support Vector Machines Using Fruit Fly Optimization for Medical Data Classification. *Knowl.-Based Syst.* **2016**, *96*, 61–75. [[CrossRef](#)]
21. Wang, M.; Chen, H.; Yang, B.; Zhao, X.; Hu, L.; Cai, Z.; Huang, H.; Tong, C. Toward an Optimal Kernel Extreme Learning Machine Using a Chaotic Moth-Flame Optimization Strategy with Applications in Medical Diagnoses. *Neurocomputing* **2017**, *267*, 69–84. [[CrossRef](#)]
22. Song, J.; Zheng, W.X.; Niu, Y. Self-Triggered Sliding Mode Control for Networked PMSM Speed Regulation System: A PSO-Optimized Super-Twisting Algorithm. *IEEE Trans. Ind. Electron.* **2021**, *69*, 763–773. [[CrossRef](#)]
23. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
24. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gen. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
25. Dong, W.; Zhou, M. A Supervised Learning and Control Method to Improve Particle Swarm Optimization Algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1135–1148. [[CrossRef](#)]
26. Mareda, T.; Gaudard, L.; Romerio, F. A Parametric Genetic Algorithm Approach to Assess Complementary Options of Large Scale Windsolar Coupling. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 260–272. [[CrossRef](#)]
27. Jian, Z.; Liu, S.; Zhou, M. Modified cuckoo search algorithm to solve economic power dispatch optimization problems. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 794–806.
28. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science.* **1983**, *220*, 671–680. [[CrossRef](#)]
29. Xing, B.; Gao, W.-J. Gravitational Search Algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*; Intelligent Systems Reference Library; Springer International Publishing: Cham, Germany, 2014; Volume 62, pp. 355–364. ISBN 978-3-319-03403-4.
30. Alatas, B. ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **2011**, *38*, 13170–13180. [[CrossRef](#)]
31. Patel, V.K.; Savsani, V.J. Heat transfer search (HTS): A novel optimization algorithm. *Inf. Sci.* **2015**, *324*, 217–246. [[CrossRef](#)]

32. Abdechiri, M.; Meybodi, M.R.; Bahrami, H. Gases brownian motion optimization: An algorithm for optimization (GBMO). *Appl. Soft Comput.* **2013**, *13*, 2932–2946. [[CrossRef](#)]
33. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [[CrossRef](#)]
34. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–72. [[CrossRef](#)]
35. Schneider, B.; Ranft, U. *Simulationsmethoden in der Medizin und Biologie*; Springer: Berlin/Heidelberg, Germany, 1978.
36. Yang, X.-S. Differential evolution. In *Nature-Inspired Optimization Algorithms. Algorithms*; Elsevier: London UK, 2021; Volume 6, pp. 101–109.
37. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
38. Simon, D. Biogeography-Based Optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [[CrossRef](#)]
39. Karaboga, D.; Akay, B. A comparative study of Artificial Bee Colony algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132. [[CrossRef](#)]
40. Yang, X.S. Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78. [[CrossRef](#)]
41. Jiang, X.; Li, S. BAS: Beetle antennae search algorithm for optimization problems. *Int. J. Robot. Control* **2018**, *1*, 1. [[CrossRef](#)]
42. Okwu, M.O. (Ed.) *Grey Wolf Optimizer, Metaheuristic Optimization, Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*; Springer International Publishing: Cham, Switzerland, 2021. [[CrossRef](#)]
43. Li, M.D.; Zhao, H.; Weng, X.W.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* **2016**, *92*, 65–88. [[CrossRef](#)]
44. Glover, F. Tabu search—Part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [[CrossRef](#)]
45. Kumar, M.; Kulkarni, A.J.; Satapathy, S.C. Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. *Future Gener. Comput. Syst.* **2018**, *81*, 252–272. [[CrossRef](#)]
46. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems. *Inf. Sci.* **2012**, *183*, 1–15. [[CrossRef](#)]
47. Hosseini, S.; Al Khaled, A. A Survey on the Imperialist Competitive Algorithm Metaheuristic: Implementation in Engineering Domain and Directions for Future Research. *Appl. Soft. Comput.* **2014**, *24*, 1078–1094. [[CrossRef](#)]
48. Jia, H.; Peng, X.; Kang, L.; Li, Y.; Jiang, Z.; Sun, K. Pulse coupled neural network based on Harris hawks optimization algorithm for image segmentation. *Multimed Tools Appl.* **2020**, *79*, 28369–28392. [[CrossRef](#)]
49. Fan, C.; Zhou, Y.; Tang, Z. Neighborhood centroid opposite-based learning harris hawks optimization for training neural networks. *Evol. Intell.* **2020**, *14*, 1847–1867. [[CrossRef](#)]
50. Saravanan, G.; Ibrahim, A.M.; Kumar, D.S.; Vanitha, U.; Chandrika, V.S. Iot Based Speed Control of BLDC Motor with Harris Hawks Optimization Controller. *Int. J. Grid Distrib. Comput.* **2020**, *13*, 1902–1919.
51. Qu, C.; He, W.; Peng, X.; Peng, X. Harris hawks optimization with information exchange. *Appl. Math. Model.* **2020**, *84*, 52–75. [[CrossRef](#)]
52. Devarapalli, R.; Bhattacharyya, B. Application of modified harris hawks Optimization in power system oscillations damping controller design. In Proceedings of the 2019 8th International Conference on Power Systems (ICPS), Jaipur, India, 20–22 December 2019.
53. Elgamal, Z.M.; Yasin, N.B.M.; Tubishat, M.; Alswaiti, M.; Mirjalili, S. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the Medical Field. *IEEE Access.* **2020**, *8*, 186638–186652. [[CrossRef](#)]
54. Song, S.; Wang, P.; Heidari, A.A.; Wang, M.; Zhao, X.; Chen, H.; He, W.; Xu, S. Dimension decided harris hawks optimization with gaussian mutation: Balance analysis and diversity patterns. *Knowl.-Based Syst.* **2021**, *215*, 106425. [[CrossRef](#)]
55. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–83. [[CrossRef](#)]
56. Bednarz, J.C. Cooperative Hunting Harris’ Hawks (*Parabuteo unicinctus*). *Science* **1988**, *239*, 1525–1527. [[CrossRef](#)] [[PubMed](#)]
57. Lefebvre, L.; Whittle, P.; Lascaris, E.; Finkelstein, A. Feeding innovations and forebrain size in birds. *Anim. Behav.* **1997**, *53*, 549–560. [[CrossRef](#)]
58. Sol, D.; Duncan, R.P.; Blackburn, T.M.; Cassey, P.; Lefebvre, L. Big brains, Enhanced Cognition, and Response of birds to Novel environments. *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 5460–5465. [[CrossRef](#)] [[PubMed](#)]
59. Kazimipour, B.; Li, X.; Qin, A.K. A review of population initialization techniques for evolutionary algorithms. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 2585–2592. [[CrossRef](#)]
60. Migallón, H.; Jimeno-Morenilla, A.; Rico, H.; Sánchez-Romero, J.L.; Belazi, A. Multi-Level Parallel Chaotic Jaya Optimization Algorithms for Solving Constrained Engineering Design Problems. *J. Supercomput.* **2021**, *77*, 12280–12319. [[CrossRef](#)]
61. Alatas, B.; Akin, E. Multi-Objective Rule Mining Using a Chaotic Particle Swarm Optimization Algorithm. *Knowl.-Based Syst.* **2009**, *22*, 455–460. [[CrossRef](#)]
62. Arora, S.; Anand, P. Chaotic Grasshopper Optimization Algorithm for Global Optimization. *Neural Comput. Appl.* **2019**, *31*, 4385–4405. [[CrossRef](#)]
63. Wang, G.-G.; Guo, L.; Gandomi, A.H.; Hao, G.-S.; Wang, H. Chaotic Krill Herd Algorithm. *Inf. Sci.* **2014**, *274*, 17–34. [[CrossRef](#)]
64. Mitić, M.; Vuković, N.; Petrović, M.; Miljković, Z. Chaotic Fruit Fly Optimization Algorithm. *Knowl.-Based Syst.* **2015**, *89*, 446–458. [[CrossRef](#)]

65. Kumar, Y.; Singh, P.K. A Chaotic Teaching Learning Based Optimization Algorithm for Clustering Problems. *Appl. Intell.* **2019**, *49*, 1036–1062. [[CrossRef](#)]
66. Pierezan, J.; dos Santos Coelho, L.; Mariani, V.C.; de Vasconcelos Segundo, E.H.; Prayogo, D. Chaotic Coyote Algorithm Applied to Truss Optimization Problems. *Comput. Struct.* **2021**, *242*, 106353. [[CrossRef](#)]
67. Sayed, G.I.; Tharwat, A.; Hassanien, A.E. Chaotic Dragonfly Algorithm: An Improved Metaheuristic Algorithm for Feature Selection. *Appl. Intell.* **2019**, *49*, 188–205. [[CrossRef](#)]
68. Chen, K.; Zhou, F.; Liu, A. Chaotic Dynamic Weight Particle Swarm Optimization for Numerical Function Optimization. *Knowl.-Based Syst.* **2018**, *139*, 23–40. [[CrossRef](#)]
69. Anand, P.; Arora, S. A Novel Chaotic Selfish Herd Optimizer for Global Optimization and Feature Selection. *Artif. Intell. Rev.* **2020**, *53*, 1441–1486. [[CrossRef](#)]
70. Yao, X.; Liu, Y.; Lin, G. Evolutionary Programming Made Faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102. [[CrossRef](#)]
71. Digalakis, J.G.; Margaritis, K.G. On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* **2001**, *77*, 481–506. [[CrossRef](#)]
72. Chen, H.; Zhang, Q.; Luo, J. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft. Comput.* **2020**, *86*, 105–884. [[CrossRef](#)]
73. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
74. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In *Advances in Swarm Intelligence*; Tan, Y., Shi, Y., Coello, C.A.C., Eds.; Springer International Publishing: Cham, Germany, 2014; pp. 86–94. [[CrossRef](#)]
75. Zhang, J.; Chen, J.; Che, L. Hybrid PSO Algorithm with Adaptive Step Search in Noisy and Noise-Free Environments. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
76. Jordehi, A.R. Gravitational Search Algorithm with Linearly Decreasing Gravitational Constant for Parameter Estimation of Photovoltaic Cells. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 5–8 June 2017; pp. 37–42.
77. Feng, X.; Liu, A.; Sun, W.; Yue, X.; Liu, B. A Dynamic Generalized Opposition-Based Learning Fruit Fly Algorithm for Function Optimization. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7.
78. Demsar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
79. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **2010**, *180*, 2044–2064. [[CrossRef](#)]
80. Groppe, D.M.; Urbach, T.P.; Kutas, M. Mass univariate analysis of event-related brain potentials/fields I: A critical tutorial review. *Psychophysiology* **2011**, *48*, 1711–1725. [[CrossRef](#)] [[PubMed](#)]
81. Deliparaschos, K.M.; Nenedakis, F.I.; Tzafestas, S.G. Design and implementation of a fast digital fuzzy logic controller using FPGA Technology. *J. Intell. Robot. Syst.* **2006**, *45*, 77–96. [[CrossRef](#)]
82. Sapkota, B.; Vittal, V. Dynamic VAR planning in a large power system using trajectory sensitivities. *IEEE Trans. Power Syst.* **2010**, *25*, 461–469. [[CrossRef](#)]
83. Huang, H.; Xu, Z.; Lin, X. Improving performance of Multi-infeed HVDC systems using grid dynamic segmentation technique based on fault current limiters. *IEEE Trans. Power Syst.* **2012**, *27*, 1664–1672. [[CrossRef](#)]
84. Yong, T. A discussion about standard parameter models of synchronous machine. *Power Syst. Technol.* **2007**, *12*, 47.
85. Grigsby, L.L. (Ed.) *Power system Stability and Control*, 3rd ed.; Taylor & Francis: Boca Raton, FL, USA, 2012.
86. Tian, Z. Backtracking search optimization algorithm-based least square support vector machine and its applications. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103801. [[CrossRef](#)]
87. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [[CrossRef](#)]
88. Zendejboudi, A. Implementation of GA-LSSVM modelling approach for estimating the performance of solid desiccant wheels. *Energy Convers. Manag.* **2016**, *127*, 245–255. [[CrossRef](#)]
89. Chamkalani, A.; Zendejboudi, S.; Bahadori, A.; Kharrat, R.; Chamkalani, R.; James, L.; Chatzis, I. Integration of LSSVM technique with PSO to determine asphaltene deposition. *J. Pet. Sci. Eng.* **2014**, *124*, 243–253. [[CrossRef](#)]
90. Zhongda, T.; Shujiang, L.; Yanhong, W.; Yi, S. A prediction method based on wavelet transform and multiple models fusion for chaotic time series. *Chaos Solitons Fractals* **2017**, *98*, 158–172. [[CrossRef](#)]
91. Liu, W.; Zhang, X. Research on the supply chain risk assessment based on the improved LSSVM algorithm. *Int. J. U-Serv. Sci. Technol.* **2016**, *9*, 297–306. [[CrossRef](#)]
92. Jain, S.; Bajaj, V.; Kumar, A. Efficient algorithm for classification of electrocardiogram beats based on artificial bee colony-based least-squares support vector machines classifier. *Electron. Lett.* **2016**, *52*, 1198–1200. [[CrossRef](#)]
93. Yang, A.; Li, W.; Yang, X. Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines. *Knowl.-Based Syst.* **2019**, *163*, 159–173. [[CrossRef](#)]
94. Adankon, M.M.; Cheriet, M. Support vector machine. In *Encyclopedia of Biometrics*; Springer: Boston, MA, USA, 2014; Volume 3, pp. 1–9. [[CrossRef](#)]