

An Improved Sparrow Search Algorithm Based on Complementary Inertia Weight and LSTM Optimization for Timer Serial Forecasting

Jiaqi Ma

m13333282441@163.com

GBA Branch of Aerospace Information Research Institute, Chinese Academy of Sciences

Article

Keywords: Sparrow Search Algorithm, complementary inertia weight, mutation, LSTM, hyperparameter, wind speed forecasting

Posted Date: October 3rd, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-4933325/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

An Improved Sparrow Search Algorithm Based on Complementary Inertia Weight and LSTM Optimization for Timer Serial Forecasting

Jiaqi Ma^{1*}

¹ GBA Branch of Aerospace Information Research Institute, Chinese Academy of Sciences, Guangzhou 510530, China

* Corresponding author

E-mail address: 13333282441@163.com

Abstract: When dealing with optimization problems that require to achieve optimal result within fewer iterations, the basic Sparrow Search Algorithm (SSA) shows insufficient solution accuracy and poor stability. In view of the above shortcomings, an improved adaptive Sparrow Search Algorithm based on complementary inertia weight (CIW-SSA) is proposed. Firstly, considering the lack of population diversity, chaos mapping is used to generate all individuals. Secondly, in order to obtain high-precision optimization results within short-term iterations, this paper created and implemented a pair of complementary inertia weights to improve search efficiency. The cooperation of this pair of complementary inertia weights not only adjusts the inertia of each individual, but also adjusts the step size of movement, which greatly improves the speed and accuracy of convergence. Thirdly, to response to the alert value of basic SSA and improve the capacity of exploration, Gaussian mutation and Cauchy mutation are organically added together to process of position update of producer. Simulation experiments were carried out through 18 classical benchmark functions with different characteristic, and the result show that the improved algorithm has faster convergence speed, more stable convergence properties and higher convergence accuracy. In addition, to prove the superiority of CIW-SSA within a few iterations, CIW-SSA is employed to optimize the hyperparameter of Long Short-Term Memory (LSTM) model for wind speed forecasting, and the test results prove the effectiveness of the improvement strategies.

Keywords: Sparrow Search Algorithm, complementary inertia weight, mutation, LSTM, hyperparameter, wind speed forecasting

1. Introduction

Playing an important role in optimization algorithm, meta heuristic swarm intelligent algorithm simulates the behavioral rules and strategies of group of animals when searching for food, avoiding danger, or performing other cooperative behaviors, such as flocks of bird, schools of fish and colonies of ants. Current mainstream swarm optimization algorithms include the genetic algorithm [1], [2], gray wolf algorithm [3], [4], chicken swarm algorithm [5], [6], particle swarm algorithm [7], [8], and so on. These swarm intelligent algorithms provide a new perspective in the fields of structural design [9], electronic circuit design [10], communication networks [11] and finance [12].

Sparrow search algorithm (SSA) [13], proposed by Xue, and Shen in 2020, is one of these swarm intelligent algorithms with superior performance. SSA simulates the hierarchy of the sparrow population and the foraging process of sparrows with different roles cooperating with each other. According to fitness values, all individuals are divided into the roles of producers and scroungers, which act in different ways. Among them, the first certain proportion with high fitness value are considered as producers, providing foraging directions for all scroungers with lower fitness values. In addition, there is a group of watchers in the sparrow swarm who are aware of dangers, and they will randomly update their position toward the safe area. In the process of sparrow population evolution, each sparrow receives a diverse foraging procedure according to its role in the population.

Developing rapidly in recent years, SSA has been widely applied in many fields due to its searching precision and convergence speed. For example, Ma et al. [14] utilized the SSA to detect malicious URLs more effectively. Liu et al. [15] applied SSA improved with multi-strategy to optimize the Convolutional Neural Network (CNN), replacing original gradient descent algorithm. Ouyang et al. [16] applied improved SSA to the optimization of support vector machine parameter. Zhang et al. [17] used SSA to optimize the weights and bias of Back Propagation Neural Network for the prediction of material grinding particle size. Chen et al. [18] applied SSA to feature selection for intrusion detection, in which SSA is utilized to search for the best subset of features. Yao et al. [19] utilized SSA to optimize the hyperparameter of CNN-LSTM for forecasting meteorological runoff utilizing verifiable data as input. While SSA has many advantages, it still has some inherent weaknesses that hinder its further development, such as poor population diversity, premature convergence, weak global searching ability. In SSA, a search strategy based on exponential function and Gaussian distribution is adopted for the starving scrounger, which need to escape from the current barren location. However, this strategy actually strengthens the local search ability and makes it fall into local position, which is unreasonable. Besides, in the process of position update, the lack of adjustment of inertia weight not only prevents efficient convergence, but also greatly inhibits solution precision and stability.

To improve the global search capability of SSA, Zhang et al. [20] applied Levy flight strategy and reversing learning to perturb position update of some sparrows for randomness and globality. Yue et al. [21] introduced an intelligent weighting factor, varying with the number of iterations, into the process of position update, which can balance the capability of exploratory and exploitative. To control the solution process of algorithm, the logarithmic spiral strategy and adaptive step strategy are introduced by Tang et al. [22], which improved the sparrow search mechanism and accelerated the escape

from local optima.

In order to prevent premature convergence, Gao et al. [23] combined mutation operation and greedy strategy, which not only speeded up the solution time but also improved the search ability for global optimal solution. Furthermore, due to the search efficiency of Particle Swarm Optimization (PSO), Fan et al. [24] introduced the concept of velocity vector from PSO into SSA to guide the movement of individual more accurately, making SSA easier to converge to higher accuracy with faster speed.

Although overall performance of the above mentioned SSA variants is reinforced to a certain extent, they still have some shortcomings in exploration and exploitation. Yue et al. [21] applied the nonlinear adaptive weight to the process of the producer updating, which enhanced the global search ability. However, there are only slight improvements on the aspect of solution accuracy. Fan et al. [24] only improved the search efficiency by introducing velocity vector. Although the speed of convergence is accelerated, the SSA still suffers from poor global search ability. Aiming to the above shortcomings, this paper proposes an improved Sparrow Search Algorithm based on complementary inertia weight (CIW-SSA), which significantly improves the optimization performance of the SSA in terms of convergence speed, convergence accuracy and stability. Firstly, the chaos mapping is utilized to generate sparrow swarm, which enriches the diversity of population and strengthens the ability of exploration. Secondly, a pair of complementary inertial weight related to fitness value is introduced to position update of producers and scroungers, which significantly improves the accuracy of convergence and the stability of iteration. Thirdly, the clever combination of Gaussian mutation and Cauchy mutation is introduced to response to the alarm value, which further enhances the capability to escape from local optima. Simulation experiment results shows that CIW-SSA has obvious advantages in terms of convergence speed, solution precision and robustness, while the hyperparameters optimization experiment of LSTM network indicates that the proposed CIW-SSA has very practical value.

The rest of the paper is divided into the following sections. In section 2, the rules and steps of the original SSA are briefly outlined. Section 3 proposes an improved CIW-SSA and describes its improvement strategies in detail. Section 4 is responsible for conducting a series of simulation experiments and analyzing the experimental result. In section 5, the CIW-SSA is employed to improve LSTM for time series forecasting. In summary, section 6 summarizes this paper and anticipates future applications of the improved SSA.

2. Sparrow Search Algorithm

SSA is the excellent representative of heuristic swarm intelligent algorithm imitating the behaviors of sparrows during they forage process [25]. According to the fitness value, all individuals are mapping to producers and scroungers in the process of population evolution. For simplicity, the assignment of the individual roles and division of labor among individuals take after the following rules:

- 1) The sparrows with relatively better fitness value would be appointed to the producers, while individuals with poor fitness values are set as scroungers. After each iteration is completed, each sparrow will be reassigned to a producer or scrounger, but the proportions of the two roles are fixed across all iterations [26].
- 2) The producers are responsible for identifying areas where rich food sources can be found. Alarm value is introduced in SSA, once the producer detects that the alarm value surpasses security edge, the producers need to guide the scroungers to secure areas.
- 3) The scroungers follow the producer who can provide the richest area for foraging. In order to increasing the predation rate, part of scroungers continuously monitor producers and compete for food with producers. Among scroungers, several starving individuals, who have extremely poor fitness value, are more randomly to move toward other areas for food to obtain better fitness value.
- 4) In addition, the SSA assumes that a certain percentage of sparrows sensitive to the danger are assigned as watchers. Some of them at the edge of the swarm would rapidly move to the secure area, while the other sparrows randomly walk to close to others.

Suppose a population of N individuals searches in D -dimensional solution space, and the position of all sparrows can be represented in the Eq.(1):

$$X = \begin{bmatrix} X_{1,1} & \cdots & X_{1,D} \\ \vdots & \ddots & \vdots \\ X_{N,1} & \cdots & X_{N,D} \end{bmatrix} \quad (1)$$

The position update equation of sparrows varies based on different identifies. The producers play a leadership role in the foraging process and provide directions for all other sparrows. The position update equation of producers is described as the Eq.(2):

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp(\frac{-i}{\alpha \cdot iter_{max}}) & \text{if } R_2 < ST \\ X_{i,j}^t + Q \cdot L & \text{if } R_2 > ST \end{cases} \quad (2)$$

where t indicates the current iteration, and $X_{i,j}^t$ represents the value of the j th dimension of the i th sparrow at iteration t . $iter_{max}$ is the maximum number of iterations. $\alpha \in (0, 1]$ is a random number. $R_2 \in [0, 1]$ and $ST \in [0.5, 1.0]$ represent the alarm value and the safety threshold respectively. Q is a random number which obeys normal distribution. L represents a matrix of shape $1 \times D$ with all elements being 1.

The position update equation of scroungers is described as the Eq.(3):

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp(\frac{X_{worst}^t - X_{i,j}^t}{i^2}) & \text{if } i > n/2 \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (3)$$

where X_p is the optimal location occupied by the producer. X_{worst} denotes the current global worst location. A represents a matrix of $1 \times D$ for which each element inside is randomly assigned 1 or -1, and $A^+ = A^T(AA^T)^{-1}$.

The watchers aware of dangers update their position using the Eq.(4).

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot (\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon}) & \text{if } f_i = f_g \end{cases} \quad (4)$$

where X_{best} is the global optimal position in the current iteration. β , obeying the normal distribution with a mean value of 0 and a variance of 1, is used to control step size. $K \in [-1, 1]$ is a random number. Here f_i is the fitness value of this sparrow, while f_g and f_w represent the current global best and worst fitness values, respectively.

3. Improved Sparrow Search Algorithm Based on Complementary Inertia Weight

3.1 Chaotic map

SSA is an algorithm with relatively excellent performance, few control parameters and robustness, and its initial positions of swarm are obtained through the method of random initialization [27]. The method guarantees the randomness of initial position, but blindly random production is prone to the overlapping aggregation of initial position, which greatly reduce the global search capability and search efficiency of SSA [28].

Chaos mapping is a complex nonlinear motion type ubiquitous in nature [29]. Thanks to its ergodicity, randomness and regularity, chaotic map has been widely applied to optimization algorithm. Chaos mapping is an ideal pseudorandom number generator, which can generate not only stochastic but also ergodic number [30]. To sum up, the chaotic map can be used to generate the initial position of sparrow swarm, which can significantly enrich the population diversity. Among many chaotic mappings, iterative map has the optimal solution in large number of experiments on chaotic mapping initialization population of SSA [31]. Therefore, this paper chooses iterative map as the approach of population initialization, and its standard form is shown as Eq.(5):

$$Z_{k+1} = \sin(\frac{a \cdot \pi}{Z_k}) \quad (5)$$

where $a \in (0, 1)$ is the iterative map control parameter.

The CIW-SSA uses the corrected iterative mapping to initial position, and the specific expression as show in Eq.(6).

$$X_{n+1} = R_l + (R_u - R_l) \cdot \sin(\frac{a \cdot \pi}{X_n}) \quad (6)$$

where R_u , R_l represent the upper and lower limits of the search range respectively.

3.2 Complementary inertia weight

In order to obtain high-precision optimization results within short-term iterations, this paper considers introducing a

pair of complementary inertia weights to improve search efficiency. And, since the inertia weight has the ability to balance the exploitation and exploration of SSA, many scholars [32], [33] prefer to introduce various inertia weight to enhance the performance of algorithm. These adaptive inertia weights change according to the iterations, and they gradually decrease as the number of iterations increase. Early in the iterations, the changing mechanism of such inertia weights allows individual to maintain strong global search capability, while the mechanism pays more attention to local search capability in the later stage.

However, original SSA has taken the number of iterations into account for the impact on producers' position updates in the Eq.(2), so again using the number of iterations as the driver for the inertia weight adjustment is somewhat redundant and ineffective. Besides, these inertia weights [32], [34] ignore the meaning of inertia itself and the impact of fitness value on inertia. To be precise, inertia is the ability of an object to maintain a state of motion, and mass is a measure of the inertia of an object. In contrast, the fitness value can be viewed as a measure of the inertia of a position update. Based on the above analogy with object inertia, this paper proposes a position update mathematical model, and the formula is shown as Eq.(7):

$$P_{n+1} = \omega(f) \cdot P_n + \omega_{inverted}(f) \cdot \Delta P \quad (7)$$

where P_n , P_{n+1} respectively represent the position of the individual at the current moment and the next moment. ΔP represents the distance that needs to be moved. $\omega(f)$, $\omega_{inverted}(f)$ are a pair of complementary inertia weight, which change as the individual fitness value changes. To enable individuals to conduct regular self-learning to balance the capability of exploration and exploitation, $\omega(f)$ and $\omega_{inverted}(f)$ are expected to have the following characteristics:

- 1) The inertia weight $\omega(f)$ should increase as the fitness value becomes better. Because the better the fitness value is, the more motivated the individual is to stay in the current location.
- 2) The inverted inertia weight $\omega_{inverted}(f)$ should increase as the fitness value gets worse. In the similar way, the worse the fitness value is, the stronger the individual's motivation is to leave the current location.
- 3) To enhance the ability of exploration, the impact of complementary inertia weight should be appropriately enlarged. Simply put, the values of inertia weight should be more concentrated in larger and smaller areas.

According to the above analysis, this paper proposes a pair of complementary inertia weight related to fitness value, which is expressed as Eq.(8) and Eq.(9):

$$\omega(i) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot (\sin(\pi \cdot (\frac{f_i - f_{\text{best}}}{f_{\text{worst}} - f_{\text{best}}}) + \pi/2) + 1)/2 \quad (8)$$

$$\omega_{inverted}(i) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot (\cos(\pi \cdot (\frac{f_i - f_{\text{best}}}{f_{\text{worst}} - f_{\text{best}}}) - \pi) + 1)/2 \quad (9)$$

here $\omega_{\min}$, $\omega_{\max}$ represent the maximum and minimum values of inertia weight. f_{worst} , f_{best} represent the best and worst fitness values among all individuals, while f_i represent the fitness value of the i_{th} individual. Many experiments would be conducted in Section 4 to explore the most suitable range of inertia weight ($\omega_{\min} = 0$ and $\omega_{\max} = 2$).

In order to intuitively express the characteristics of this pair of complementary inertia weights, their curve image is given in Fig 1,

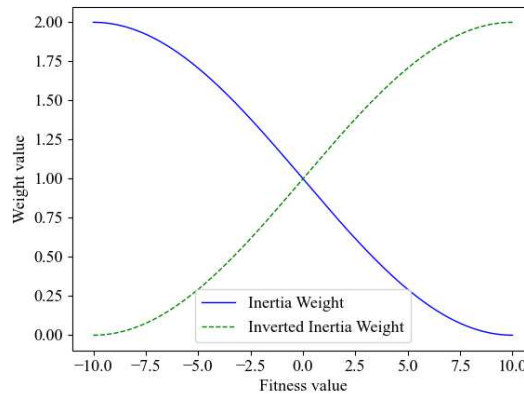


Fig 1. The curve of inertia weight and inverted inertia weight
($\omega_{\min} = 0$, $\omega_{\max} = 2$, $f_{\text{best}} = -10$ and $f_{\text{worst}} = 10$).

where we assume that the maximum and minimum of the inertia weight are 0 and 2, and the best and the worst of the fitness value are -10 and 10. It can be seen that both the values of inertia weight and inverted inertia weight are more likely to concentrate in larger and smaller values, which can amplify the effect of the complementary inertia weight.

3.3 Gaussian mutation and Cauchy mutation

Among the many improvement strategies of for optimization algorithm, Gaussian and Cauchy mutation are two popular techniques. It is widely known that Gaussian mutation is excellent in local convergence, while Cauchy mutation improves the capability of escaping local optima [35]. Therefore, Gaussian mutation and Cauchy mutation has been widely introduced into the improvements of Sparrow Search Algorithm [36], [37].

In original SSA, the producers forage through a nonlinear exponential function in safe conditions, while in dangerous situation they perform random strategy based on random variables obeying normal distribution [38], as described by Eq.(2). The normal distribution has the characteristics of relatively concentrated density, which suppresses individual diversity and weakens the global search capability. Besides, when the alarm value exceeds the safety threshold, the position update formula Eq.(2) of producers ignores the impact of the size of the alarm value.

Aiming to the above problems, this paper proposes an adaptive mutation factor that cleverly combines Gaussian and Cauchy mutation based on the alarm value. The adaptive mutation factor is as Eq.(10):

$$Factor_{mutation} = 1 + R_2 \cdot Cauchy(0, 1) + (1 - R_2) \cdot Gaussian(0, 1) \quad (10)$$

where R_2 represents the alarm value, and $Cauchy(0, 1)$, $Gaussian(0, 1)$ represent the Cauchy mutation and Gaussian mutation respectively. Due to infinite variance of the probability density function, the Cauchy mutation could be easier to escape dangerous place, so the proportion of Cauchy variation in the $Factor_{mutation}$ should increase as the alarm value increases. In contrast, because Gaussian mutation is excellent in local convergence, the proportion of the Gaussian mutation should increase as the alarm value decreases, making the sparrows further exploit the local area.

3.4 The procedure of CIW-SSA

The basic steps of the CIW_SSA can be summarize as follows:

Step 1: Initialization.

Set the parameter of algorithm, such as the size of population, maximum number of iterations and the proportion of producers, scroungers and watchers. In addition, initialize individual positions using Eq.(6).

Step 2: Assessment

Calculate the fitness value for each individual and record the best individual and worst individual in the current iteration.

Step 3: Establishment

Rank the sparrows according to their fitness value, establish the hierarchy of swarm, and assign roles to every individual.

Step 4: Position Update

Update the position of producers through the following formula:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{max}}\right) & \text{if } R_2 < ST \\ \omega(i) \cdot X_{i,j}^t + Q(j) \cdot L \cdot Factor_{mutation} & \text{if } R_2 > ST \end{cases} \quad (11)$$

Update the position of scroungers through the following formula:

$$X_{i,j}^{t+1} = \begin{cases} \omega(i) \cdot Q(j) \cdot \exp\left(\frac{X_{worst}^t - X_{i,j}^t}{i^2}\right) & \text{if } i > n/2 \\ X_P^{t+1} + \omega_{inverted}(i) \cdot |X_{i,j}^t - X_P^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (12)$$

Update the position of sparrows aware of danger through the following formula,

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \omega_{inverted}(i) \cdot \beta \cdot |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_g \\ \omega(i) \cdot X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g \end{cases} \quad (13)$$

and obtain the global optimal solution.

Step 5: Termination

If the program does not reach the maximum number of iterations, go back to Step 2; otherwise, the optimal solution and location are returned, and the algorithm ends.

4. Simulation Experiment and Result Analysis

In this part, this paper carries out three main sets of experiments to demonstrate the effectiveness of the CIW-SSA comprehensively. The first one is used to search the appropriate range of the complementary inertia weight, the second one is used to test the effectiveness of the inertia weight strategy, and the third experiment is used to compare the comprehensive performance of CIW-SSA, which combines all strategies in this paper, with several other closely related optimization algorithms.

4.1 Experimental setup

All these experiments were conducted on a person computer with Intel(R) Core (TM) i7-9700 CPU, NVIDIA GPU RTX3060, 16GB RAM and Windows 11 OS. All the algorithms were implemented with python 3.10.9.

In order to evaluate the convergence speed ,optimization accuracy and stability of the proposed algorithm, this paper selects 18 standard benchmark functions [13], [39],[40] as evolution criteria, which are listed in **Table 1**.

Table 1. The benchmark functions.

Name	Function	Range	Optimal Solution
Sphere	$f_1 = \sum_{i=1}^n x_i^2$	[-100, 100]	0
Sum of powers	$f_2 = \sum_{i=1}^n x_i ^{i+1}$	[-1, 1]	0
Sum squares	$f_3 = \sum_{i=1}^n ix_i^2$	[-10, 10]	0
SchwefelsP2.22	$f_4 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-10, 10]	0
Rotated hyper-ellipsoid	$f_5 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	[-100, 100]	0
SchwefelsP2.21	$f_6 = \max\{ x_i , 1 \leq i \leq n\}$	[-100, 100]	0
Zakharov	$f_7 = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$	[-10, 10]	0
Quartic	$f_8 = \sum_{i=1}^n ix^4 + rand[0, 1)$	[-1.28, 1.28]	0
Step	$f_9 = \sum_{i=1}^n (x_i + 0.5)^2$	[-100, 100]	0
Millhugue	$f_{10} = 10^6 x_1^2 \sum_{i=1}^n x_i^2$	[-100, 100]	0
Rosenbrock	$f_{11} = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]	0
Restrigin	$f_{12} = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12, 5.12]	0

Ackley	$f_{13} = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	[-32, 32]	0
Alpine	$f_{14} = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	[-10, 10]	0
Tesima	$f_{15} = \left[1 - \left \frac{\sin[\pi(x_1 - 2)] \sin[\pi(x_2 - 2)]}{\pi^2(x_1 - 2)(x_2 - 2)} \right ^5 \right] [2 + (x_1 - 7)^2 + 2(x_2 - 7)^2]$	[0, 14]	0
Shekel	$f_{16} = - \sum_{i=1}^{10} \left(\sum_{j=1}^4 (x_j - C_{ji})^2 + \beta_i \right)^{-1}$ $\beta = \frac{1}{10} [1, 2, 2, 4, 4, 6, 3, 7, 5, 5]$ $C = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \end{bmatrix}$	[-10, 10]	-10.5364
Maytas	$f_{17} = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	[-10, 10]	0
Three Hump	$f_{18} = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2$	[-10, 10]	0

Among the above objective functions, function $f_1 \sim f_{11}$ are unimodal functions, which are extremely hard to converge to optima, so they are utilized to test the solution accuracy of algorithm. Multimodal function $f_{12} \sim f_{14}$, with several local optima, are used to evaluation the global search capability. Besides, to more evaluate the performance of the algorithm, multimodal and fixed low-dimensional function $f_{16} \sim f_{18}$ are introduced into the evaluation system. To obtain unbiased result, in all comparative experiments, the size of population and the dimension of the functions is set to 100 and 30 respectively. Depending on the usage scenario, the maximum number of iterations are set to 1000 or 200. In addition, all algorithms are run independently 30 times to obtain the average results.

4.2 The selection of complementary inertia weight in CIW-SSA

To search the appropriate range of the pair of complementary inertia weight, this paper conducted a series of experiments on the benchmark functions from **Table 1** and obtained basically consistent results. Based on the inspiration of the object inertia, that the inertial weight determines the individual's motivation to stay in or leave the original position, the inertia weight and the inverted inertia weight should float up and down around 1. The several prepared sets of inertia weight ranges are listed in the **Table 2**,

Table 2. The solution results of CIW-SSA with different ranges of inertia weight (D = 30)

Ranges	f_1	f_2	f_{12}	f_{15}
$\omega_{min} = 0.0, \omega_{max} = 2.0$	0	0	0	8.2564E-320
$\omega_{min} = 0.1, \omega_{max} = 1.9$	8.3160E-273	8.2267E-245	0	9.5423E-73
$\omega_{min} = 0.2, \omega_{max} = 1.8$	8.8012E-248	6.5580E-306	0	3.3287E-10
$\omega_{min} = 0.3, \omega_{max} = 1.7$	8.6617E-284	1.1459E-284	0	6.0487E-09
$\omega_{min} = 0.4, \omega_{max} = 1.6$	1.1307E-293	5.8549E-278	0	5.1255E-09
$\omega_{min} = 0.5, \omega_{max} = 1.5$	1.1345E-261	7.2197E-292	0	3.3732E-08
$\omega_{min} = 0.6, \omega_{max} = 1.4$	2.4044E-297	2.9625E-294	0	8.4255E-09
$\omega_{min} = 0.7, \omega_{max} = 1.3$	3.4086E-265	1.0239E-292	0	4.4384E-08
$\omega_{min} = 0.8, \omega_{max} = 1.2$	6.4146E-296	5.5663E-281	0	4.8068E-08
$\omega_{min} = 0.9, \omega_{max} = 1.1$	1.1476E-263	3.0251E-259	0	2.9730E-08

where the values of ω_{min} increases from 0 to 1 with an increment of 0.1; while the ω_{max} decreases from 2 to 1 with a decrement of 0.1, until the ω_{min} and ω_{max} meet at 1. For comprehensiveness, unimodal, multimodal, and fixed low-dimensional objective functions are selected to find the best weight range, and the solution accuracy of CIW-SSA with different ranges of inertia weights are presented in **Table 2**, which indicates that the CIW-SSA achieves the highest solution accuracy when the $\omega_{min} = 0.0$ and $\omega_{max} = 2.0$.

Furthermore, to intuitively show the influence of different ranges of inertia weight on the convergence speed, the convergence curves of the experimental result for f_1 and f_2 are shown in Fig 2. It can be observed that when the $\omega_{min} = 0.0$ and $\omega_{max} = 2.0$, the CIW-SSA has faster convergence speed and more accurate convergence results, and this set of boundaries is helpful for CIW-SSA to converge to the optimal value in the short-term iterations. This paper chooses 0.0 and 2.0 as the minimum value and maximum value of inertia weight, respectively.

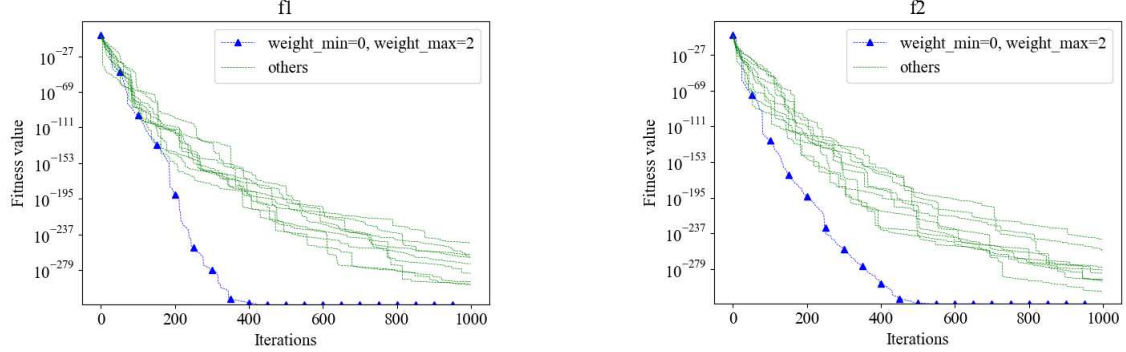


Fig 2. The convergence curve of CIW-SSA with different inertia weight ranges

4.3 The effectiveness of complementary inertia weight

To demonstrate the superiority of the strategy of complementary inertia weight related to fitness value, two different types of inertia weights, which are related to the number of iterations, from other improved SSA^{[21][20]} are compared with our inertia weight. The expressions of the two inertia weights are described as Eq.(14) and Eq.

(15):

$$\omega(iter) = \left[\omega_{end} + (\omega_{star} - \omega_{end}) \left(1 - \frac{e^{\frac{iter}{iter_{max}}} - 1}{e - 1} \right) \right] \quad (14)$$

where ω_{star} is the initial intelligence weight, ω_{end} is the ending intelligent weight, $iter$ is the number of current iterations. The original paper [14] says that the inertia weight has the best effect when $\omega_{star} = 0.8$ and $\omega_{end} = 0.4$.

$$\omega = \begin{cases} \omega_0 & t \leq t_0 \\ \left(\frac{1}{t}\right)^{0.9} & t > t_0 \end{cases} \quad (15)$$

where ω_0 is the given positive real number, t is the current number of iterations, and t_0 is the number of current iterations. When $\omega_0 = 1.5$, the inertia weight can give the greatest improvements to SSA [31].

To ensure fairness of the comparison, the above two inertia weight and the complementary inertia weight of CIW-SSA are used as the only strategy for SSA improvement, and they are donated SSA-W1, SSA-W2 and CIW-SSA-W in sequence. The performance of the two algorithms with the only weight strategy is compared with that of CIW-SSA-W, and the experimental results after 1000 iterations are presented in Table 3, where the optimal results are bold font.

Table 3. The solution accuracy of three types of inertia weight

Functions	Dimension	Result	SSA-W1	SSA-W2	CIW-SSA-W
f_1	30	Mean	2.7717E-223	4.7355E-81	0
		Std	0	2.5501E-80	0
f_2	30	Mean	1.1095E-191	1.2475E-86	0
		Std	0	6.7182E-86	0
f_3	30	Mean	2.2419E-203	8.3360E-69	0
		Std	0	4.2903E-68	0
f_4	30	Mean	9.7066E-111	5.6971E-35	0
		Std	5.2271E-110	3.0680E-34	0
f_5	30	Mean	1.8966E-223	2.8288E-84	0
		Std	0	1.5233E-83	0
f_6	30	Mean	9.0954E-90	6.0268E-36	2.9437E-301
		Std	4.8980E-89	3.2455E-35	0
f_7	30	Mean	5.9549E-217	4.1239E-84	0
		Std	0	2.2111E-83	0
f_8	30	Mean	0	1.9723E-157	0

		Std	0	1.0621E-156	0
f_9	30	Mean	3.3553E-09	1.6960E-09	7.5159E-11
		Std	6.8558E-09	6.3986E-09	1.9315E-10
f_{10}	30	Mean	0	1.8730E-168	0
		Std	0	0	0
f_{11}	30	Mean	1.5843E-06	5.1232E-07	7.5050E-08
		Std	2.5278E-06	9.1293E-07	3.9885E-07
f_{12}	30	Mean	0	0	0
		Std	0	0	0
f_{13}	30	Mean	0	0	0
		Std	0	0	0
f_{14}	30	Mean	1.4097E-94	1.0072E-42	2.6414E-313
		Std	7.5916E-94	5.4239E-42	0
f_{15}	30	Mean	2.5729E-05	2.6155E-05	2.6542E-06
		Std	4.5915E-05	3.9727E-05	1.4293E-05
f_{16}	30	Mean	-1.0513E+01	-1.0532E+01	-1.0532E+01
		Std	8.3678E-02	7.2597E-02	1.3097E-02
f_{17}	30	Mean	0	1.0233E-248	0
		Std	0	0	0
f_{18}	30	Mean	0	4.0626E-140	0
		Std	0	2.1878E-139	0

Table 3 shows that for almost function, CIW-SSA-W always obtain the better mean value and standard deviation than SSA-W1 and SSA-W2. In particular, for f_1 , f_2 , f_3 , f_4 and f_7 , CIW-SSA-W converges to the theoretical optimum, and its solution is on average 100 orders of magnitude higher than SSA-W1 and 200 orders of magnitude higher than SSA-W2. Even for fixed multimodal function f_{16} , f_{17} and f_{18} , which are regarded as hard to escape from local optimum, CIW-SSA-W still maintains its lead.

In addition, in this paper, we studied the convergence behaviors of the CIW-SSA-W, compared to the SSA-W1 and SSA-W2. For the sake of brevity and comprehensiveness, only the convergence curves for the unimodal f_1 and multimodal f_{15} are plotted in Fig 3.

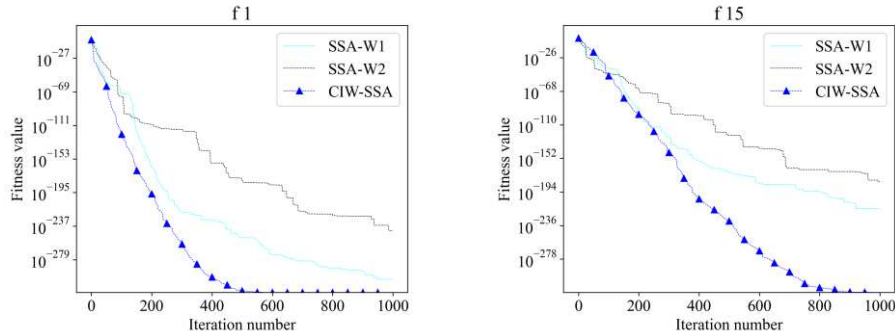


Fig 3. The convergence diagrams of the three improved SSA variants with different inertia weights

Fig 3 shows that the CIW-SSA-W uses fewer iterations but achieves higher solution accuracy compared to the other two SSA variants. To sum up, the CIW-SSA-W, using the inertia weight proposed by this paper as the only improvement strategy, is superior to SSA-W1 and SSA-W2 with only inertia weight strategy, in terms of solution precision and robustness. Especially, for unimodal function, CIW-SSA-W could converge to the theoretical optimum quickly. Those sufficiently indicate that the complementary inertia weight related to fitness value could enable SSA achieve even more outstanding performance.

4.4 Performance comparison between CIW-SSA, SSA, GWO, PSO

To investigate the performance of CIW-SSA, this paper selects several excellent intelligent swarm optimization algorithms for comparison, including original SSA, GWO and PSO. To facilitate the follow-up comparison experiments, the parameter settings of SSA, GWO, PSO and CIW-SSA are summarized in Table 4, where the $P_{producer}$ and P_{aware} means the proportion of the producers and the sparrows aware of danger respectively.

Table 4. Setting of the parameter value

Algorithm	Parameter settings
SSA	$ST = 0.8$, $P_{producer} = 0.2$, $P_{aware} = 0.1$
GWO	$r_1 = rand(0,1)$, $r_2 = rand(0,1)$

PSO	$c_1 = 1.5, c_2 = 1.5, \omega = 0.8$
HSSA	$c_1 = 1.5, c_2 = 1.5, \omega = 0.8, ST = 0.8$ $P_{producer} = 0.2, P_{aware} = 0.1$
SCA-CSSA	$a = 2$ $P_{producer} = 0.2, P_{aware} = 0.1$
ICSSOA	$ST = 0.8, \omega_0 = 1.5, t_0 = 125, \rho = 2.595, \beta = 1.5$ $P_{producer} = 0.2, P_{aware} = 0.1$
WMR-SSA	$ST = 0.8, a_{star} = 0.8, a_{end} = 0.4$ $P_{producer} = 0.2, P_{aware} = 0.1$
CIW-SSA	$ST = 0.8, \omega_{min} = 0, \omega_{max} = 2$ $P_{producer} = 0.2, P_{aware} = 0.1$

The optimization precision and stability of these algorithms is evaluated by the mean value and standard deviation, respectively. The statistical result of the 4 algorithms on 30-dimensional objective function are presented in Table 5, where the best results on each function are marked in bold font.

Table 5. The comparison results with original SSA, GWO and PSO, Dn = 30

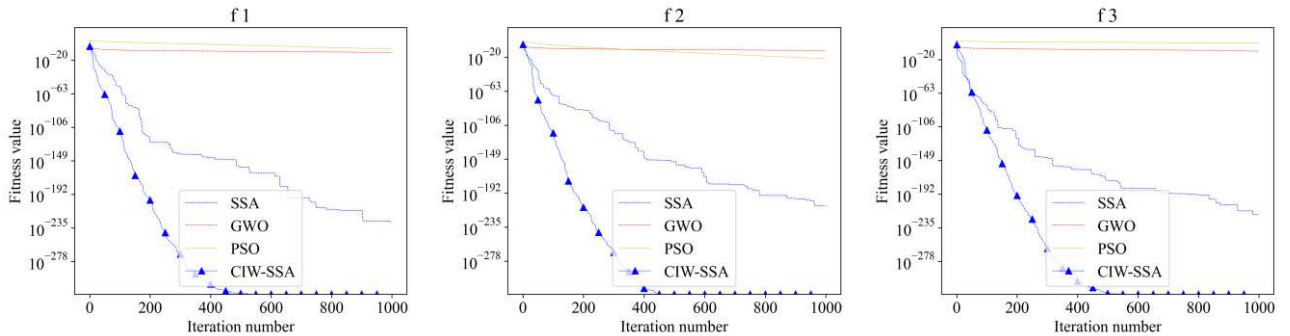
Function	Iterations	Result	SSA	GWO	PSO	CIW-SSA
f_1	200	Mean	1.7848E-32	1.3817E-07	1.4604E+02	9.8919E-112
		Std	9.5581E-32	2.6615E-07	6.2383E+01	5.2631E-111
	1000	Mean	7.6794E-78	1.3263E-10	4.1477E-06	0
		Std	4.1355E-77	2.2395E-10	6.1530E-06	0
f_2	200	Mean	1.7027E-34	2.9618E-09	2.4539E-07	1.1368E-99
		Std	9.1563E-34	4.9373E-09	2.8512E-07	6.1218E-99
	1000	Mean	2.4853E-69	1.8421E-12	3.4147E-21	0
		Std	1.3384E-68	3.0866E-12	6.5479E-21	0
f_3	200	Mean	2.0786E-42	1.0412E-06	6.6795E+02	5.7608E-104
		Std	9.7539E-42	1.8193E-06	5.1983E+02	2.1615E-103
	1000	Mean	4.1394E-71	2.4262E-09	6.2667E+02	0
		Std	2.2291E-70	3.6482E-09	5.3412E+02	0
f_4	200	Mean	7.3312E-18	1.7976E-03	4.4528E+01	9.2903E-44
		Std	3.9317E-17	1.9304E-03	1.8441E+01	5.0030E-43
	1000	Mean	4.0314E-34	3.5513E-05	3.1667E+01	2.0943E-310
		Std	2.1710E-33	3.7226E-05	1.6750E+01	0
f_5	200	Mean	6.1644E-32	1.8302E-05	1.1297E+06	5.8931E-107
		Std	2.7169E-31	2.2241E-05	8.3868E+05	3.1736E-106
	1000	Mean	3.9901E-73	1.4995E-08	1.0973E+06	0
		Std	2.1488E-72	2.1955E-08	8.5591E+05	0
f_6	200	Mean	1.5751E-19	3.6412E-05	3.2899E+01	1.9472E-56
		Std	8.0760E-19	3.3310E-05	6.2774E+00	7.5138E-56
	1000	Mean	7.1439E-25	9.9328E-07	9.0301E+00	6.4853E-315
		Std	3.8471E-24	6.4078E-07	2.7610E+00	0
f_7	200	Mean	1.1846E-35	1.9206E-04	5.2430E+02	4.1724E-91
		Std	4.5628E-35	3.1519E-04	7.3439E+01	2.2469E-90
	1000	Mean	1.7266E-65	1.7069E-07	1.2926E+02	0
		Std	8.9988E-65	2.9324E-07	5.4952E+01	0
f_8	200	Mean	2.4455E-79	8.8024E-10	1.6125E-02	1.4066E-140
		Std	1.3170E-78	2.4426E-09	1.2878E-02	7.5748E-140
	1000	Mean	1.1391E-137	4.1622E-14	4.7923E-03	0
		Std	6.1343E-137	7.2692E-14	3.8751E-03	0
f_9	200	Mean	9.4650E-11	3.7243E-07	1.8330E+03	1.6387E-10
		Std	1.2358E-10	8.2675E-07	3.6812E+03	3.8319E-10
	1000	Mean	3.4887E-12	3.7525E-10	1.6634E+03	1.3670E-10
		Std	8.2683E-12	7.4162E-10	3.7196E+03	3.4117E-10
f_{10}	200	Mean	4.5533E-63	3.4008E-10	6.1216E-05	8.3486E-173
		Std	2.4520E-62	7.5515E-10	1.2415E-04	0
	1000	Mean	6.4725E-138	2.8976E-16	6.1183E-51	0
		Std	3.4856E-137	8.5755E-16	1.9957E-50	0
f_{11}	200	Mean	7.6934E-06	5.8139E-06	1.7343E+02	1.9599E-08
		Std	2.8939E-05	7.7537E-06	2.1836E+02	1.0391E-07
	1000	Mean	4.0033E-11	7.9011E-09	1.2454E+02	2.9354E-10
		Std	8.4235E-11	1.5475E-08	2.1316E+02	1.1135E-09
f_{12}	200	Mean	0	8.3401E-06	1.8921E+02	0
		Std	0	1.4033E-05	3.0080E+01	0

	1000	Mean	0	2.4116E-08	9.7148E+01	0
		Std	0	3.2501E-08	3.1102E+01	0
f_{13}	200	Mean	1.1842E-16	1.9900E-04	1.3962E+01	0
		Std	6.3773E-16	1.8587E-04	7.3317E+00	0
	1000	Mean	0	5.9486E-06	1.2251E+01	0
		Std	0	5.6099E-06	9.4304E+00	0
f_{14}	200	Mean	1.0079E-07	8.2531E-05	1.1142E+01	3.4694E-18
		Std	5.4279E-07	7.0874E-05	4.0598E+00	1.8684E-17
	1000	Mean	3.6671E-37	5.6039E-06	5.5433E+00	1.6927E-122
		Std	1.9748E-36	7.9718E-06	3.8241E+00	9.1153E-122
f_{15}	200	Mean	1.6743E-06	3.0300E-03	1.9333E+00	1.7846E-05
		Std	2.8138E-06	6.0888E-03	3.5901E-01	8.3448E-05
	1000	Mean	3.0245E-08	2.0720E-04	1.9333E+00	2.8789E-11
		Std	6.2579E-08	2.4360E-04	3.5901E-01	1.5495E-10
f_{16}	200	Mean	-1.0536E+01	-1.0536E+01	-7.9168E+00	-1.0496E+01
		Std	7.9958E-04	8.1151E-05	3.1958E+00	1.6523E-01
	1000	Mean	-1.0536E+01	-1.0536E+01	-9.0586E+00	-1.0526E+01
		Std	6.6344E-04	1.4811E-0s7	2.7033E+00	3.6664E-02
f_{17}	200	Mean	9.6941E-57	1.6117E-10	2.9387E-14	1.8984E-84
		Std	5.2203E-56	4.3389E-10	8.9752E-14	1.0223E-83
	1000	Mean	1.5395E-207	1.8741E-13	1.1731E-49	0
		Std	0	4.9791E-13	5.2228E-49	0
f_{18}	200	Mean	1.3619E-44	1.9556E-08	5.1299E-14	7.5761E-95
		Std	7.3342E-44	3.2709E-08	8.8839E-14	4.0799E-94
	1000	Mean	3.9514E-181	2.2437E-11	1.7348E-53	0
		Std	0	3.2784E-11	5.1967E-53	0

According to the Table 5, this paper divides the experimental results into unimodal groups ($f_1 \sim f_{11}$) and multimodal groups ($f_{12} \sim f_{18}$) to analyse the local search ability and global search ability, respectively. First, this article analyzes the experimental results of 1000 iterations. In the unimodal group, for f_1 to f_{11} , CIW-SSA can always obtain the best solution, compared with all other original algorithm. For f_1 , f_2 , f_3 , f_5 , f_7 , f_8 and f_{10} , not only can CIW-SSA converge the theoretical optimal value, but also far surpass other algorithms, in which the CIW-SSA obtains the optimization space that are about 200 orders of magnitude higher than other algorithms. For f_4 and f_6 , it has not converged to the theoretical optimum, but it still far exceeded others; and as for f_9 and f_{11} , although CIW-SSA is slightly better than others, it still maintains its advantages in solution accuracy and stability. In the multimodal group, except for f_{18} , CIW-SSA achieved the best results. For f_{12} and f_{13} , the theoretical solutions are obtained by CIW-SSA and original SSA, and the mean value of CIW-SSA are 85, 105, 139 orders of magnitude higher than SSA, the second-best algorithm, for f_{14} , f_{17} and f_{18} respectively.

The optimization results after 200 iterations illustrate that CIW-SSA can achieve excellent results in a smaller number of iterations. For the most benchmark functions, CIW-SSA can obtain solutions that are 50 to 100 orders of magnitude more accurate than other algorithms within 200 iterations. Especially for f_{11} , f_{12} and f_{13} , when iteration reaches 200 times, CIW-SSA has established a leading advantage in accuracy and stability, but when iteration reaches 1000 times, CIW-SSA loses the advantages of accuracy and stability. In summary, CIW-SSA performs remarkable superiority whether it is 1000 iterations or 200 iterations. And, if the number of iterations is limited by the time and computing power, CIW-SSA can still obtain higher-precision results within a smaller number of iterations.

To demonstrate the advantages of CIW-SSA more comprehensively, we plotted the convergence curves of all function's fitness value in Fig 4.



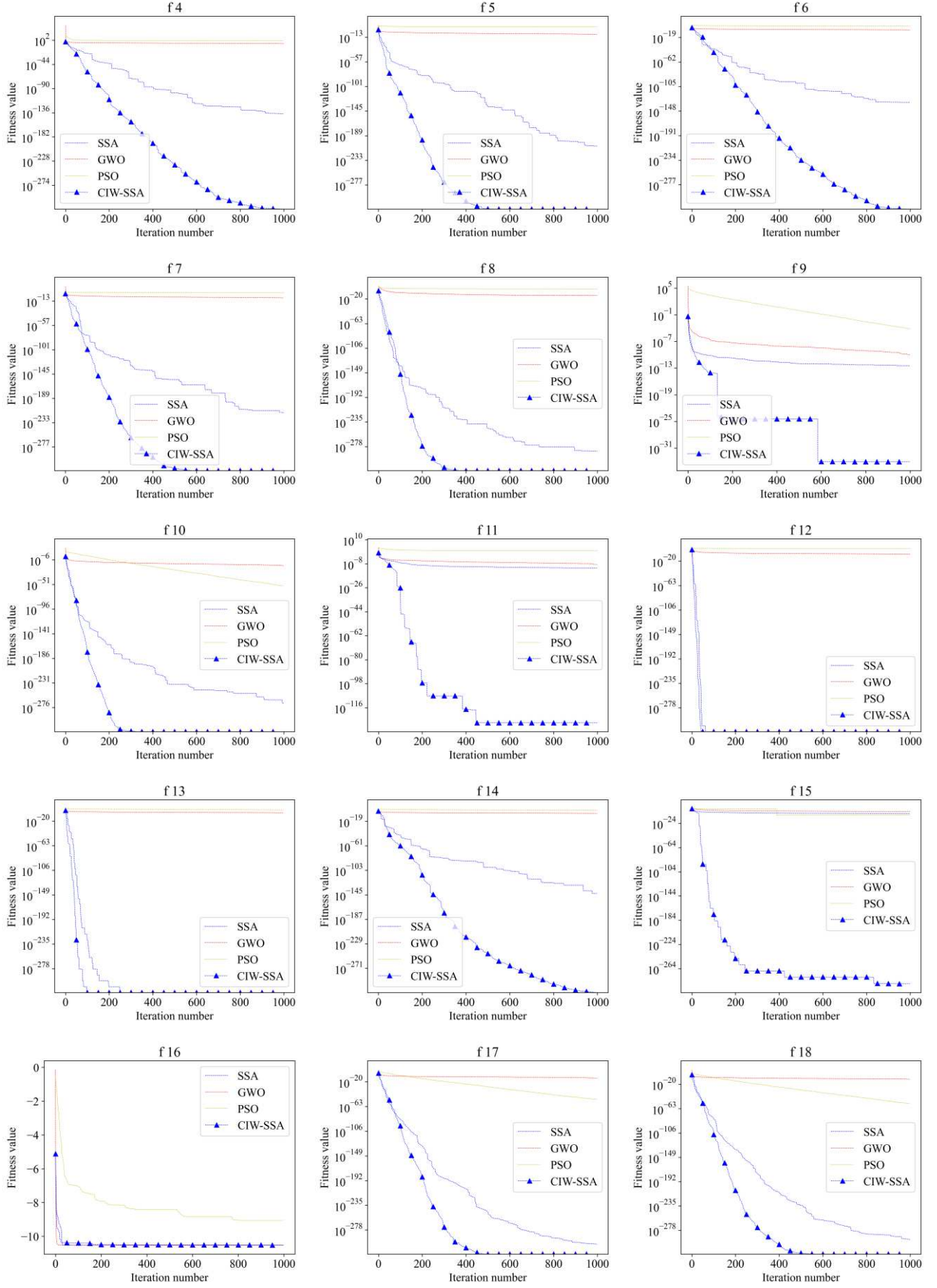


Fig 4. The convergence diagrams of SSA, GWO, PSO and CIW-SSA

Fig 4 shows that CIW-SSA and original SSA has huge advantage in convergence speed compared to the GWO and PSO for the $f_1 \sim f_8$, $f_{10} \sim f_{14}$, f_{17} and f_{18} . Moreover, for these objective function, CIW-SSA in turn has a significant improvement over original SSA in terms of convergence rate. As for the 2-dimensional multimodal f_{16} with large search range $[-10, 10]$, these algorithms except PSO are good at searching in a large range. For almost all the objective functions, the CIW-SSA has faster convergence speed and higher convergence accuracy. In addition, the convergence curves of CIW-SSA are much smoother, which implies more stable convergence performance.

In conclusion, CIW-SSA shows remarkable advantages in terms of solution precision, search speed and stability compared with GWO, PSO and the original SSA. This fully validates the effectiveness of the improvement method proposed by the paper for the SSA.

4.5 Performance comparison between CIW-SSA and several improved SSA

To further demonstrate the performance of the CIW-SSA, in this subsection HSSA [24] and SCA-CSSA [41] are selected to compared with CIW-SSA. The selected algorithms using hybrid strategy have excellent performance, and they are the outstanding representatives of many improved SSA. To fair, the four improved SSA use the same base parameters, such as the proportion of producer and sparrow aware of danger, and Table 4 lists the all the parameters used by those improved SSA. Similar to the previous subsection, we use the mean value and standard deviation of 30 independent experiments to measure they convergence performance, and the experimental results are presented in Table 6.

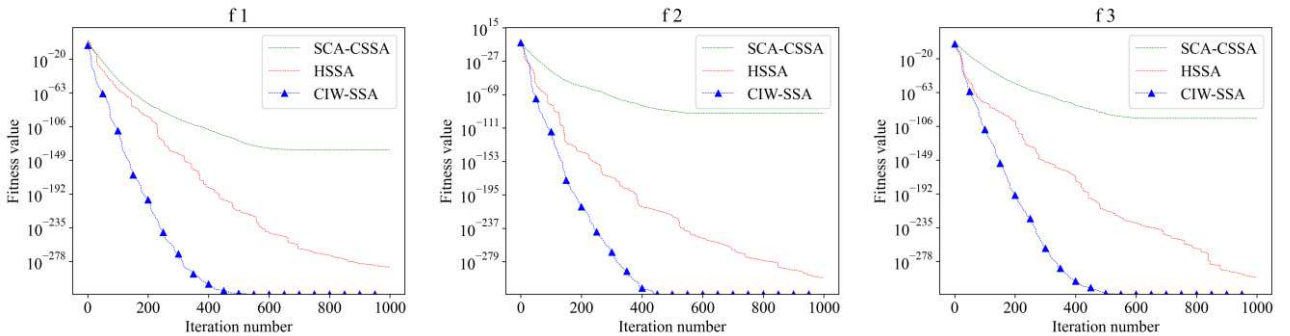
Table 6. Comparison results with several SSA variants

Function	Iterations	Result	HSSA	SCA-CSSA	CIW-SSA
f_1	200	Mean	6.8200E-48	4.6990E-59	9.8919E-112
		Std	3.6630E-47	2.5303E-58	5.2631E-111
	1000	Mean	1.0448E-187	4.6371E-107	0
		Std	0	2.0075E-106	0
f_2	200	Mean	4.2934E-39	6.1386E-46	1.1368E-99
		Std	2.1743E-38	2.4945E-45	6.1218E-99
	1000	Mean	2.0494E-179	5.8828E-74	0
		Std	0	3.1278E-73	0
f_3	200	Mean	5.3749E-51	5.2241E-42	5.7608E-104
		Std	2.8945E-50	2.7883E-41	2.1615E-103
	1000	Mean	1.1407E-178	2.6866E-74	0
		Std	0	1.4467E-73	0
f_4	200	Mean	1.3124E-29	5.4471E-21	9.2903E-44
		Std	4.9370E-29	2.9207E-20	5.0030E-43
	1000	Mean	5.1892E-89	3.5237E-37	2.0943E-310
		Std	2.7945E-88	1.8707E-36	0
f_5	200	Mean	5.7483E-46	9.4446E-45	5.8931E-107
		Std	3.0956E-45	3.2755E-44	3.1736E-106
	1000	Mean	6.0957E-170	5.1352E-71	0
		Std	0	2.7652E-70	0
f_6	200	Mean	1.6957E-25	5.4253E-19	1.9472E-56
		Std	7.9198E-25	2.6933E-18	7.5138E-56
	1000	Mean	5.1328E-92	9.0019E-41	6.4853E-315
		Std	2.7641E-91	4.7779E-40	0
f_7	200	Mean	2.2886E-37	3.0219E+02	4.1724E-91
		Std	1.2261E-36	2.9966E+02	2.2469E-90
	1000	Mean	3.3775E-135	1.0735E+01	0
		Std	1.8189E-134	4.4354E+01	0
f_8	200	Mean	4.3459E-87	2.7687E-90	1.4066E-140
		Std	2.3403E-86	1.0878E-89	7.5748E-140
	1000	Mean	0	6.2637E-203	0
		Std	0	0	0
f_9	200	Mean	2.4175E-03	8.8097E-03	1.6387E-10
		Std	3.1935E-03	4.7034E-03	3.8319E-10
	1000	Mean	5.0548E-05	2.5608E-06	1.3670E-10
		Std	7.1997E-05	1.3916E-06	3.4117E-10
f_{10}	200	Mean	8.8787E-44	4.4380E-78	8.3486E-173
		Std	4.7813E-43	2.3866E-77	0
	1000	Mean	0	2.8704E-160	0
		Std	0	1.5451E-159	0
f_{11}	200	Mean	4.6556E-03	2.6760E+01	1.9599E-08
		Std	9.5923E-03	3.9054E-01	1.0391E-07
	1000	Mean	1.1705E-04	2.3147E+01	2.9354E-10

		Std	2.2131E-04	9.4107E-01	1.1135E-09
f_{12}	200	Mean	0	0	0
		Std	0	0	0
	1000	Mean	0	0	0
		Std	0	0	0
f_{13}	200	Mean	0	0	0
		Std	0	0	0
	1000	Mean	0	0	0
		Std	0	0	0
f_{14}	200	Mean	1.5714E-30	2.7386E-20	3.4694E-18
		Std	6.2384E-30	1.0476E-19	1.8684E-17
	1000	Mean	9.6278E-96	3.6329E-39	1.6927E-122
		Std	5.1847E-95	1.9487E-38	9.1153E-122
f_{15}	200	Mean	2.0174E-01	5.5703E-11	1.7846E-05
		Std	5.9943E-01	1.1065E-10	8.3448E-05
	1000	Mean	1.6845E-04	3.0761E-12	2.8789E-11
		Std	5.1336E-04	9.3423E-12	1.5495E-10
f_{16}	200	Mean	-7.7220E+00	-9.5293E+00	-1.0496E+01
		Std	3.4352E+00	2.2621E+00	1.6523E-01
	1000	Mean	-8.3021E+00	-9.8154E+00	-1.0526E+01
		Std	3.1650E+00	1.8384E+00	3.6664E-02
f_{17}	200	Mean	1.3594E-72	7.9568E-47	1.8984E-84
		Std	7.3204E-72	3.6250E-46	1.0223E-83
	1000	Mean	0	2.6773E-74	0
		Std	0	1.4417E-73	0
f_{18}	200	Mean	4.5193E-75	2.2936E-41	7.5761E-95
		Std	2.4337E-74	1.2333E-40	4.0799E-94
	1000	Mean	0	1.0444E-80	0
		Std	0	4.5652E-80	0

As described in the 1000 iterations from Table 6, for the unimodal function f_1 , f_2 , f_3 , f_5 , f_7 and f_{10} , although HSSA and ICSSOA exhibit the same level of stability as CIW-SSA, CIW-SSA converges to an accuracy that is approximately 150 orders of magnitude higher than the HSSA, the second best algorithm. For f_4 , f_6 , f_9 and f_{11} , CIW-SSA not only achieved higher solution accuracy, but also showed obvious stability. For the three SSA variants, multimodal f_{12} and f_{13} are easy to converge to the theoretical optimal value, so they all performed well. The results of f_{16} , f_{17} and f_{18} show the superiority of CIW-SSA in the situation with multiple optimal values. From the experimental results of 200 iterations shown in Table 6, it can be seen that CIW-SSA still obtained solutions with higher accuracy than the other two SSA variants within 200 iterations, for the most benchmark functions. Particularly, for f_9 , f_{10} and f_{16} , the solution accuracy of CIW-SSA after 200 iterations is almost the same as that of 1000 iterations. To sum up, CIW-SSA has better local and global search capability and efficiency than HSSA and SCA-CSSA.

Convergence speed is an important criterion for optimization algorithm, which determines the application value and feasibility of algorithm, and this section plots convergence curves of the three SSA variants against each objective function in Fig 5.



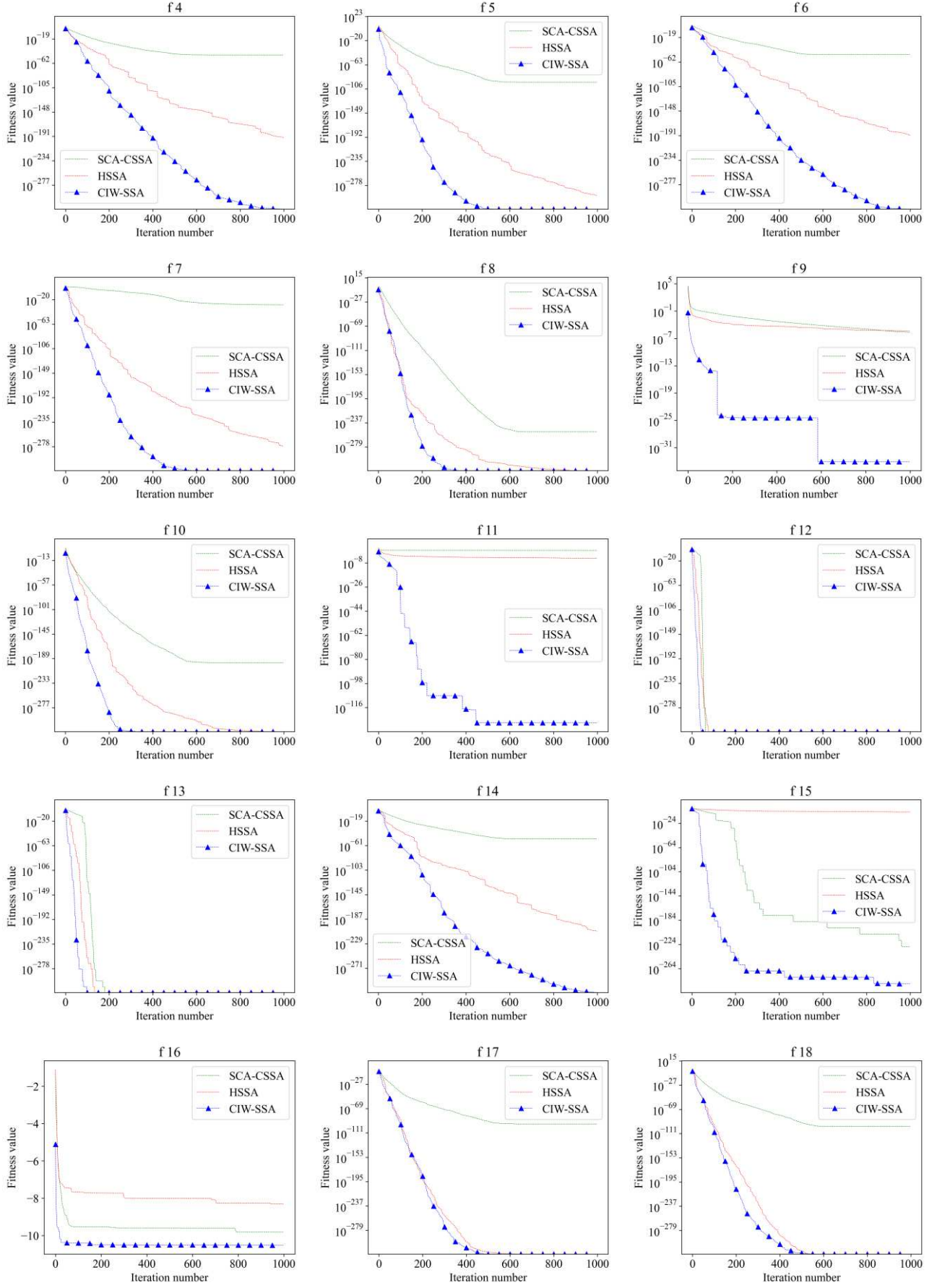


Fig 5. The convergence diagrams of HSSA, SCA-CSSA and CIW-SSA

Fig 5 shows that CIW-SSA has faster convergence speed, especially for f_1 , f_2 , f_3 , f_5 , f_7 and f_8 , with about 400 iterations earlier than other two algorithms to get the theoretical optimal value. For the most objective functions, CIW-SSA can achieve a clear lead within 200 iterations in terms of solution accuracy and convergence speed. In addition, compared with the HSSA, the convergence curve of CIW-SSA is smoother, which once again confirms that the CIW-SSA has stronger stability. As for f_9 and f_{11} , although the convergence curve of CIW-SSA is relatively unsmooth, CIW-SSA relies on its powerful exploration capabilities to achieve solution results that are much more accurate than the other two. In addition, for $f_1 \sim f_3$, $f_5 \sim f_8$ and $f_{10} \sim f_{12}$, compared with the results of 1000 iterations, CIW-SSA achieves 50% ~ 80% accuracy using about a quarter of the iterations. In contrast, the accuracy of the optimization results of HSSA after 200 iterations is far less than the results of 1000 iterations. To sum up, CIW-SSA shows obvious accuracy advantages in both long-term and short-term iterations, and when focusing on the optimization efficiency, CIW-SSA can achieve higher accuracy solutions than HSSA and SCA-CSSA.

To facilitate the observation of the solution set distributions of the three algorithms, f_1 , f_9 , f_{15} and f_{16} are taken as examples, and we plot boxplots of the 30 independent experimental result in Fig 6.

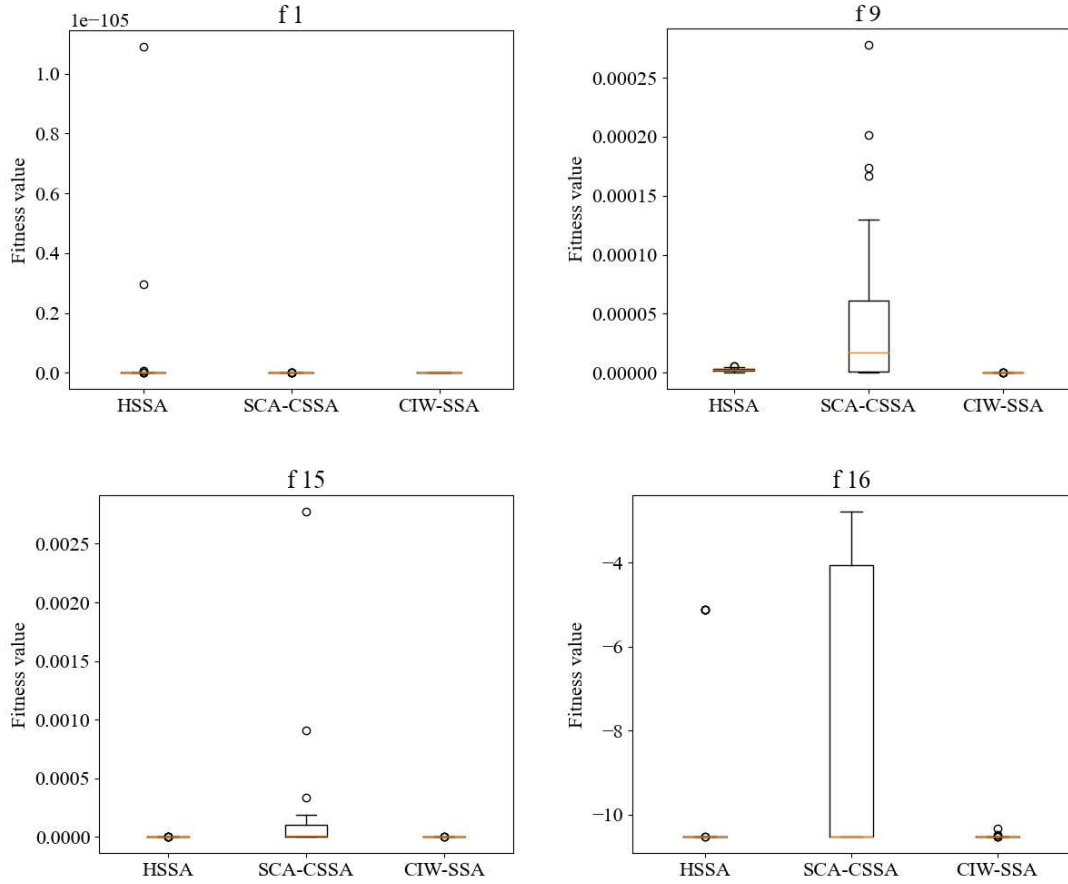


Fig 6. The boxplot of the three SSA variants

It shows that the CIW-SSA solution results have fewer outliers and a narrower data range than HSSA and SCA-CSSA. In addition, for the four objective functions, there is no doubt that the performances of CIW-SSA are all very excellent, while the other two algorithms are sometimes good, and they cannot cope well with all types of objective functions. Compared with the other two SSA variants, CIW-SSA is superior in terms of convergence speed, solution accuracy and stability.

4.6 Signed rank sum test of CIW-SSA

To further demonstrate the significant improvements of CIW-SSA over the comparison SSA variants, Wilcoxon test is applied to this paper, which is a classical non-parametric test to determine whether the distribution of the two datasets are significantly different [42]. The Wilcoxon test does not have requirements on the distribution of the data, due to the efficiency and applicability, the Wilcoxon test is used by many scholars [43][44] to evaluate the performance of algorithm. In order to ensure that the difference in algorithm performance is statistically significant, we calculated the Wilcoxon statistic and p-value of the optimization results of each algorithm for 20 objective functions, which are shown in Table 7.

Table 7. The result of Wilcoxon test

Algorithm	Wilcoxon statistic	p-value
CIW-SSA versus SSA	113.0	4.7497E-03
CIW-SSA versus HSSA	22.0	6.9217E-04
CIW-SSA versus SCA-CSSA	64.0	1.0956E-04

It can be seen that the p-values of CIW-SSA with SSA, HSSA and CIW-SSA are all little than the commonly significance level (0.05), which means that the CIW-SSA does have a significant improvement in the term of solution accuracy. In addition, the average rank of the three improved SSA is listed in Table 8, which further indicates that CIW-SSA is indeed superior to the other two SSA variants.

Table 8. Average rank

Algorithm	Average rank	Total rank
HSSA	2.39	2
SCA-CSSA	3.05	3
CIW-SSA	1.59	1

4.7 Scalability test of CIW-SSA

In view of the widespread existence of high-dimensional optimization problems in practical engineering applications, this subsection evaluates the scalability of CIW-SSA in solving high-dimensional optimization problems. Except for the fixed-dimensional function $f_{16} \sim f_{20}$, $f_1 \sim f_{15}$ are chosen as the objective function, whose dimensions are set to 100 and 500, to test the CIW-SSA's performance in high-dimensional problems. The test results are presented in Table 9.

Table 9. The experimental result of CIW-SSA for high-dimensional function

Objective Functions	Result	CIW-SSA		
		$D = 30$	$D = 100$	$D = 500$
f_1	Mean	0	0	0
	Std	0	0	0
f_2	Mean	0	0	0
	Std	0	0	0
f_3	Mean	0	0	0
	Std	0	0	0
f_4	Mean	2.0943E-310	0	0
	Std	0	0	0
f_5	Mean	0	0	0
	Std	0	0	0
f_6	Mean	6.4853E-315	2.0257E-322	0
	Std	0	0	0
f_7	Mean	0	0	0
	Std	0	0	0
f_8	Mean	0	0	0
	Std	0	0	0
f_9	Mean	1.3670E-35	6.5801E-10	1.8029E-09
	Std	3.4117E-9	1.0972E-09	5.6258E-09
f_{10}	Mean	0	0	0
	Std	0	0	0
f_{11}	Mean	2.9354E-11	1.8597E-10	5.7777E-10
	Std	1.1135E-09	6.8293E-10	3.0203E-09
f_{12}	Mean	0	0	0
	Std	0	0	0
f_{13}	Mean	0	0	0
	Std	0	0	0
f_{14}	Mean	0	0	0
	Std	0	0	0
f_{15}	Mean	1.6927E-122	3.3947E-250	1.0246E-318
	Std	9.1153E-122	0	0

The Table 9 shows that CIW-SSA has a high degree of accuracy in solving higher dimensional problems. For $f_1 \sim f_3$, f_5 and $f_{12} \sim f_{14}$, same as solving 30-dimensional functions, CIW-SSA obtains the theoretical optimum in solving high-dimension functions. By comparing side-by-side, for almost every objective function, regardless of whether the dimension of function is 30, 100 or 500, the CIW-SSA can always achieve consistent solution results with high accuracy, which indicates that the accuracy of CIW-SSA is not sensitive to changes in dimensions. Therefore, we can speculate that as the dimensions continue to increase, the performance of CIW-SSA will at least not decrease.

5. Hyperparameter optimization of LSTM model

In this section, CIW-SSA is used to optimize the hyperparameters of neural network model for time series forecasting. In each iteration process of the hyperparameters optimization of the neural network model, each fitness value calculation requires the neural network model to go through N epochs. In order to avoid increasing the computational burden and prolonging the calculation time, the optimization process should not carry out thousands of iterations. However, CIW-SSA shows the advantage of solving optimization problems in a small number of iterations, and it often convergence to higher accuracy within about 200 iterations than other above algorithms. Therefore, this paper selects this optimization problem of hyperparameters to evaluate the CIW-SSA's advantage and practical application value in dealing with optimization problems that require to achieve optimal result within a few iterations.

5.1 Construction of CIW-SSA-LSTM model

Long short-Term Memory, as an improved recurrent neural network, is regarded as the state-of-the-art network for time series forecasting owing to its ability of learning long-term temporal dependencies through the design of gated [45]. Besides that, an appropriate set of hyperparameters related to the network topology and learning properties can make LSTM model perform better. However, the identification of the optimal hyperparameter settings for LSTM model remains a challenging task owing to the complexity of the problems at hand and the requirement of profound domain knowledge.

In the process of optimization hyperparameters of LSTM using optimization algorithms, each fitness value calculation requires training the LSTM network for many epochs, so thousands of iterations to obtain the optimal hyperparameters of LSTM are time-consuming and intolerable. The CIW-SSA has a faster convergence speed, and it often be the first to obtain the optimal value after hundreds of iterations, so it is used in the scenario that requires rapid convergence.

In this subsection, CIW-SSA is applied to optimize the hyperparameters and topologies of LSTM to undertake the tasks of time series forecasting. To be specific, the structural parameters of the LSTM modal we optimized include the number of LSTM layers and hidden node, and the hyperparameters we optimized include the size of batch, the length of sequence, the learning rate and the dropout rate. Table 10 lists the search range of each of above parameters to be optimized. As such, the confounding impacts of changes in the different kinds of hyperparameters could be thoroughly explored through the evolving process of the proposed CIW-SSA.

Table 10. The search range of the hyperparameters

Hyperparameters	Search range
Number of LSTM layers	[1, 3]
Number of hidden nodes	[5, 25]
Batch size	[20, 50]
Sequence length	[7, 20]
Learning rate	$[10^{-1}, 10^{-3}]$
Dropout rate	[0, 0.5]

The process of CIW-SSA optimizing these hyperparameters is described as follows. Firstly, the sparrow population is initialized though the iterative chaotic map according to the Eq.(6) and the search range Table 10. The position of each individual represents the possible setting of hyperparameter of the LSTM modal. Secondly, we choose the mean squared error (MSE) as the loss function of the LSTM, which is also the objective function of CIW-SSA. Then, based on the training dataset, the LSTM models under each setting of hyperparameters are trained, and based on the validation set, we calculate the loss of each LSTM model, which is also the fitness value of each individual. Thirdly, based on the fitness value, each individual is assigned as the producer or scrounger to update its position according to the Eq.(11) ~ Eq.(13). After an artificially set number of iterations, the global optimal location is selected as optimal hyperparameter of the LSTM model. The flow diagram of the CIW-SSA with LSTM network is shown in Fig 7.

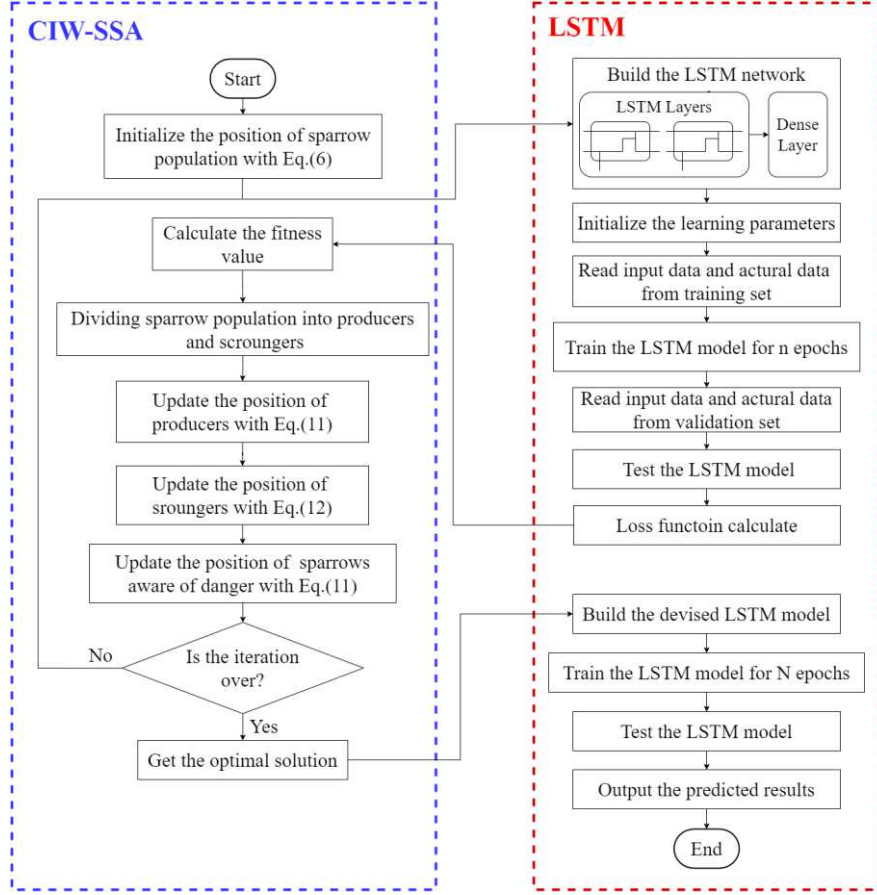


Fig 7. CIW-SSA_LSTM algorithm flow

5.2 The experiment and result analysis

This subsection applies LSTM optimized by CIW-SSA to solve time series forecasting problems, wind speed forecasting for India. The dataset is taken from the Daily Climate time series data, which provides wind speed data from 1st January 2013 to 24th April 2017 in the city of Delhi, India. From this dataset, we select the last 1000 sets for the experiments, 90% of which are used for training and 10% of which are used for validation.

This paper utilizes the Pytorch framework to build LSTM network and deploys the model on the GPU (RTX3060). To verify the practicality and superiority of CIW-SSA, the original SSA and GWO are compared with CIW-SSA to optimize the hyperparameters. To balance performance and computing cost, the LSTM model is trained for 20 epochs when the fitness value needs to be calculated, and the three algorithms were added with greedy strategy in the evolution stage. Besides that, the Adam optimizer is applied and the MSE is used as the loss function. To take advantage of the fast convergence CIW-SSA, the number of iterations is set to 200. After 200 iterations, the optimal solution of sparrow population is used as the hyperparameters of LSTM. The manually randomly set hyperparameters and the hyperparameters optimized by original SSA, GWO and CIW-SSA are listed in Table 11.

Table 11. The manually set hyperparameters and optimized hyperparameters.

Method of obtained	Hyperparameters
Manually set	Layers of LSTM = 3 Hidden node = 17 Sequence length = 18 Batch size = 42 Learning rate = 0.01496 Dropout rate = 0.0227
Optimized by SSA	Layers of LSTM = 1 Hidden node = 19 Sequence length = 17 Batch size = 27

	Learning rate = 0.09048 drop out = 0.0
Optimized by GWO	Layers of LSTM = 1 Hidden node = 14 Sequence length = 19 Batch size = 38 Learning rate = 0.04566 drop out = 0.0
Optimized by CIW-GWO	Layers of LSTM = 2 Hidden node = 14 Sequence length = 14 Batch size = 41 Learning rate = 0.00569 Dropout rate = 0.3862

Using the four sets of hyperparameters in Table 11, then the four LSTM models are retrained 50 epochs to forecast the wind speed. In order to evaluate the performance of these four devised LSTM models, five commonly error indexes are selected, and the formulas of the five indicators are shown from Eq.(16) to Eq.(20),

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (16)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|^2 \quad (17)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (18)$$

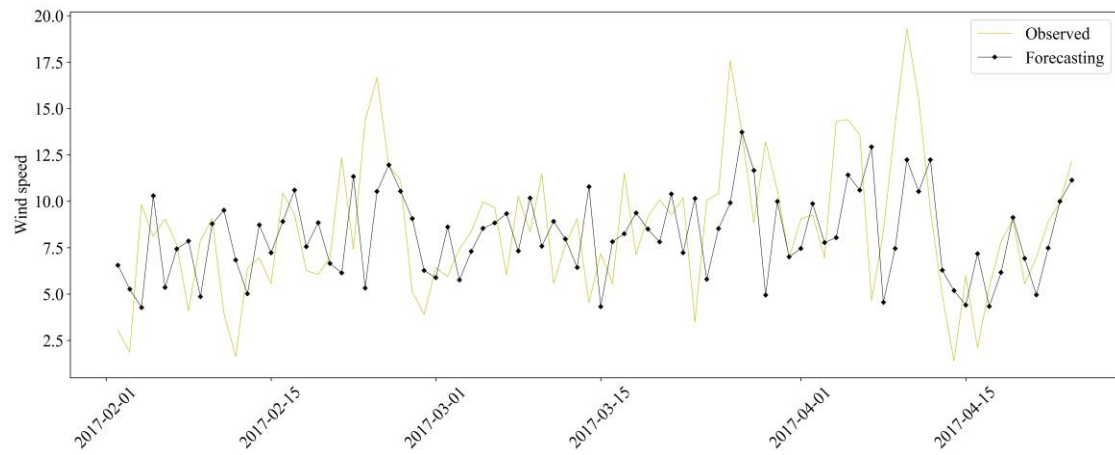
$$SSE = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (19)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (20)$$

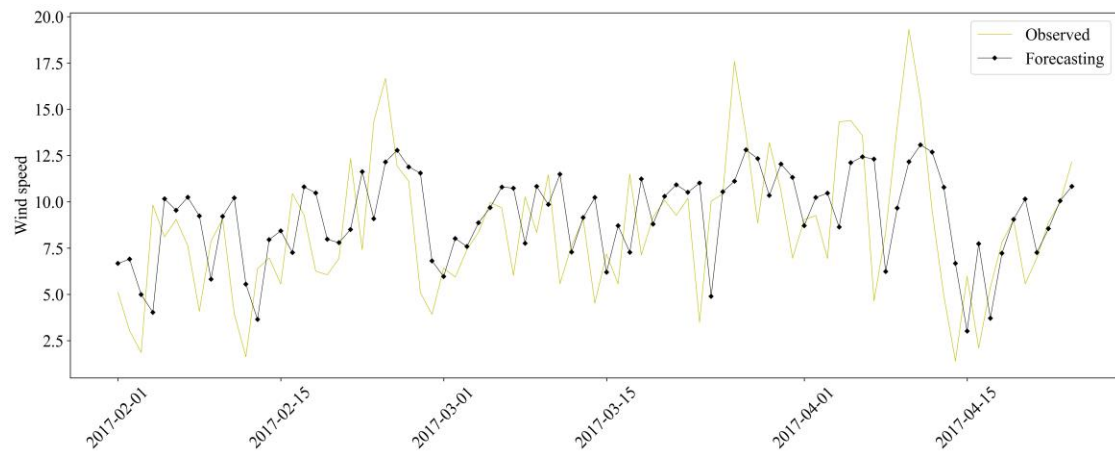
where n is the number of test data, and the $\hat{y}_i$ and y_i represent the actual value and predicted value at i_{th} moment respectively, and $\bar{y}$ is the average of actual values. The experimental results of the four LSTM models are presented in Table 12, and Fig 8 shows the wind speed predication curve.

Table 12. The five error indexes of LSEM models

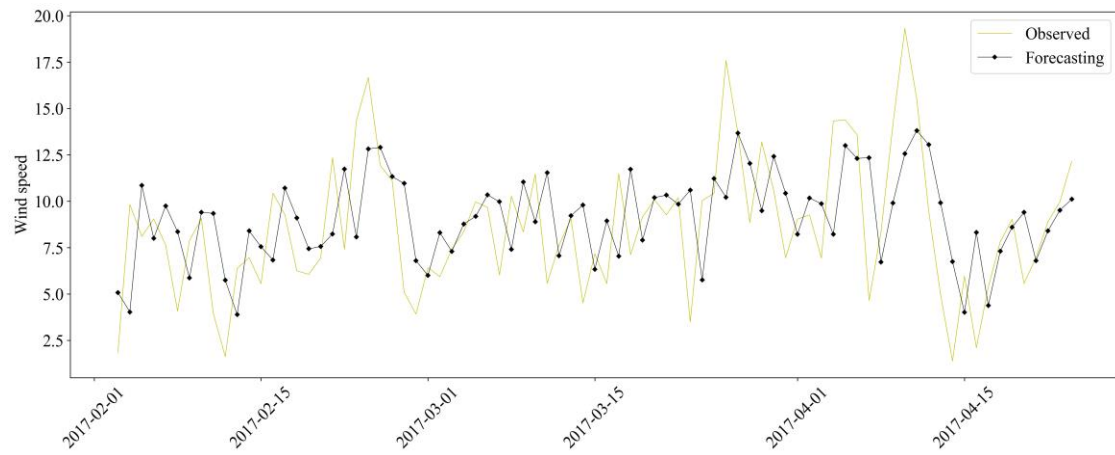
Model	MAE	MSE	RMSE	SSE	R^2
LSTM	2.94006	13.10499	3.62008	1074.60082	0.00636
SSA-LSTM	2.75935	12.03236	3.46877	998.68553	0.08651
GWO-LSTM	2.69669	11.51715	3.39369	932.88899	0.11203
CIW-SSA-LSTM	2.57337	10.33590	3.21495	888.88773	0.21241



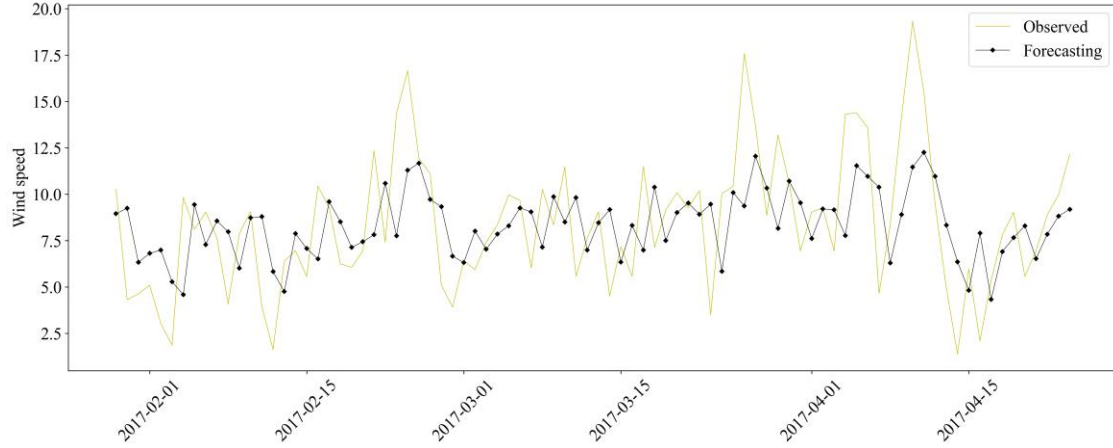
(a) LSTM model with manually set hyperparameters.



(b) LSTM model with hyperparameters optimized by SSA.



(c) LSTM model with hyperparameters optimized by GWO.



(d) LSTM model with hyperparameters optimized by CIW-SSA

Fig 8. The wind speed forecasting curve

Table 12 shows the five types of quantitative error of the three LSTM models. It is apparent that all the optimized LSTM models outperform than the non-optimized LSTM model, and the forecasting results of LSTM model optimized by CIW-SSA achieves relatively minimal error, which indicates that CIW-SSA has higher solution accuracy to the other two optimization algorithms in solving the problem of hyperparameters optimization of LSTM. It can be seen from Fig 8 that, although these four LSTM models cannot predict wind speed very accurately, they can predict the overall trend of wind speed changes, which still has strong practical significance. Among the four LSTM models, the LSTM model optimized by CIW-SSA obtained the most accurate prediction result. In conclusion, CIW-SSA has relatively higher application value in optimizing the hyperparameters of LSTM model to improve prediction accuracy.

6. Conclusions and Feature Direction

Aiming to overcome the shortcomings of original sparrow search algorithm, this paper made improvements to the original SSA, and the based on complementary inertia weight improved CIW-SSA has been proposed. The CIW-SSA employs iterate chaotic map to enhance the diversity of population, a pair of complementary inertia weight for the local and global search efficiency, as well as the Gaussian and Cauchy mutation to escape from local optimum.

Though extensive simulation experiments, this paper demonstrated the feasibility and effectiveness of the CIW-SSA and improvement strategy. Firstly, this paper determined the best range of the inertia weight by controlled experiments. Secondly, the comparison with other two different types of inertia weight illustrates the superiority of the inertia weight proposed by this paper. Secondly, by comparing with original SSA, GWO, PSO and two SSA variants, the experimental results consistently show that the CIW-SSA is more excellent than several other swarm intelligent optimization algorithms. Finally, the CIW-SSA is applied to the hyperparameter optimization of LSTM for time series forecasting, which further demonstrates the feasibility and practicality of CIW-SSA.

Although this article gets some achievements, a few works have to be attempted in the future. Judging from the performance of the CIW-SSA on some benchmark functions, we speculate that sparrow search algorithm still has room for optimization. The next step could be planned to change hierarchy of the sparrow population and foraging mechanism, which has the potential to fundamentally overcome the shortcomings of the SSA. In addition, there are some aspects worthing improving about original SSA, but the SSA variant proposed by this paper must choose to ignore them due to the space constraints. For example, the alarm value R_2 is randomly generated. However, if the value of the R_2 could quantify the exploration value of the current position, which would provide guidance for the position update and further improve the search efficiency.

As for application value, this paper does not test CIW-SSA in too many application scenarios. In the future research, we consider applying the CIW-SSA to more engineering scenarios, such as the optimization of electronic component parameters in circuit design, network topology optimization in communication technology and the parameters optimization of PID algorithm in motion control field. In fact, every practical problem has its own characteristics, and any existing general optimization algorithm may not be suitable for a certain problem. For the next research, we plan to design specific optimization algorithms for specific problems to overcome specific practical problems. In future, the specialized optimization algorithms applied to specialized fields will get more attention.

Data Availability

The datasets generated and analysed during the current study are available in the CIW-SSA-for-LSTM repository, <https://github.com/JiaQiMark/CIW-SSA-for-LSTM>.

References

- [1] Mirjalili, Seyedali, and Seyedali Mirjalili. "Genetic algorithm." *Evolutionary Algorithms and Neural Networks: Theory and Applications* (2019): 43-55.
- [2] Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." *Multimedia tools and applications* 80 (2021): 8091-8126.
- [3] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." *Advances in engineering software* 69 (2014): 46-61.
- [4] Faris, Hossam, et al. "Grey wolf optimizer: a review of recent variants and applications." *Neural computing and applications* 30 (2018): 413-435.
- [5] Meng, Xianbing, et al. "A new bio-inspired algorithm: chicken swarm optimization." *Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17-20, 2014, Proceedings, Part I* 5. Springer International Publishing, 2014.
- [6] Deb, Sanchari, et al. "Recent studies on chicken swarm optimization algorithm: a review (2014–2018)." *Artificial Intelligence Review* 53 (2020): 1737-1765.
- [7] Kennedy, James, and Russell C. Eberhart. "A discrete binary version of the particle swarm algorithm." 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation. Vol. 5. IEEE, 1997.
- [8] Shi, Yuhui. "Particle swarm optimization: developments, applications and resources." *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*. Vol. 1. IEEE, 2001.
- [9] Xie, Yi, et al. "Enhanced optimization algorithm for the structural design of an air-cooled battery pack considering battery lifespan and consistency." *International Journal of Energy Research* 46.15 (2022): 24021-24044.
- [10] Adair, Alec, and Armin Tajalli. "Computational efficiency of circuit design and optimization algorithms: a comparative study." 2023 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2023.
- [11] Shehadeh, Hisham A., et al. "Optimal topology planning of electromagnetic waves communication network for underwater sensors using multi-objective optimization algorithms (MOOAs)." *Automatika* 64.2 (2023): 315-326.
- [12] Hu, Hongwen, et al. "Stock prediction model based on mixed fractional Brownian motion and improved fractional-order particle swarm optimization algorithm." *Fractal and Fractional* 6.10 (2022): 560.
- [13] Xue, Jiankai, and Bo Shen. "A novel swarm intelligence optimization approach: sparrow search algorithm." *Systems science & control engineering* 8.1 (2020): 22-34.
- [14] Ma, Yiran, et al. "Malicious URL classification model based on improved sparrow search algorithm." 2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC).
- [15] Liu, Tingting, et al. "An optimal brain tumor detection by convolutional neural network and enhanced sparrow search algorithm." *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 235.4 (2021): 459-469.
- [16] Ouyang, Chengtian, Donglin Zhu, and Fengqi Wang. "Application of improved sparrow search algorithm in SVM optimization." *Journal of Physics: Conference Series*. Vol. 1966. No. 1. IOP Publishing, 2021.
- [17] Zhang, Shida, et al. "Regression prediction of material grinding particle size based on improved sparrow search algorithm to optimize BP neural network." 2021 2nd international symposium on computer engineering and intelligent communications (ISCEIC). IEEE, 2021.
- [18] Chen, Hongwei, Xin Ma, and Song Huang. "A feature selection method for intrusion detection based on parallel sparrow search algorithm." 2021 16th International Conference on Computer Science & Education (ICCSE). IEEE, 2021.
- [19] Yao, Zhiyuan, et al. "An ensemble CNN-LSTM and GRU adaptive weighting model based improved sparrow search algorithm for predicting runoff using historical meteorological and runoff data as input." *Journal of Hydrology* 625 (2023): 129977.
- [20] Zhang, Xuan-Yu, et al. "An improved chaos sparrow search optimization algorithm using adaptive weight modification and hybrid strategies." *Ieee Access* 10 (2022): 96159-96179.
- [21] Yue, Xiaofeng, et al. "A sparrow search algorithm with intelligent weight factors and mutation operators and applications to image classification." *Journal of Intelligent & Fuzzy Systems* 43.3 (2022): 3627-3653.
- [22] Tang, Andi, et al. "A Chaos Sparrow Search Algorithm with Logarithmic Spiral and Adaptive Step for Engineering

- Problems." CMES-Computer Modeling in Engineering & Sciences 130.1 (2022).
- [23] Gao, Bingwei, et al. "Research on multistrategy improved evolutionary sparrow search algorithm and its application." IEEE Access 10 (2022): 62520-62534.
 - [24] Fan, Yanyan, et al. "A hybrid sparrow search algorithm of the hyperparameter optimization in deep learning." Mathematics 10.16 (2022): 3019.
 - [25] Gharehchopogh, Farhad Soleimanian, et al. "Advances in sparrow search algorithm: a comprehensive survey." Archives of Computational Methods in Engineering 30.1 (2023): 427-455.
 - [26] Gad, Ahmed G., et al. "An improved binary sparrow search algorithm for feature selection in data classification." Neural Computing and Applications 34.18 (2022): 15705-15752.
 - [27] Zhang, Xuan-Yu, et al. "An improved chaos sparrow search optimization algorithm using adaptive weight modification and hybrid strategies." IEEE Access 10 (2022): 96159-96179.
 - [28] Fathy, Ahmed, et al. "Optimal energy management of micro-grid using sparrow search algorithm." Energy Reports 8 (2022): 758-773.
 - [29] Khokhar, Bhuvnesh, Surender Dahiya, and KP Singh Parmar. "Load frequency control of a microgrid employing a 2D Sine Logistic map based chaotic sine cosine algorithm." Applied Soft Computing 109 (2021): 107564.
 - [30] Cao, Weijia, Hang Cai, and Zhongyun Hua. "n-Dimensional Chaotic Map with application in secure communication." Chaos, Solitons & Fractals 163 (2022): 112519.
 - [31] Wang, Yanfeng, et al. "Multistrategy improved sparrow search algorithm optimized deep neural network for esophageal cancer." Computational Intelligence and Neuroscience 2022 (2022).
 - [32] Sharma, Shailja, Pooja Jain, and Akash Saxena. "Adaptive inertia-weighted firefly algorithm." Intelligent Computing Techniques for Smart Energy Systems: Proceedings of ICTSES 2018. Springer Singapore, 2020.
 - [33] Pawan, YVR Naga, and Bhanu Prakash Kolla. "Improved PSO Performance using LSTM based Inertia Weight Estimation." International Journal of Advanced Computer Science and Applications 11.11 (2020).
 - [34] Gu, Yanchun, et al. "Adaptive Simplified Chicken Swarm Optimization Based on Inverted S-Shaped Inertia Weight." Chinese Journal of Electronics 31.2 (2022): 367-386.
 - [35] Lan, Kuo-Torng, and Chun-Hsiung Lan. "Notes on the distinction of Gaussian and Cauchy mutations." 2008 Eighth International Conference on Intelligent Systems Design and Applications. Vol. 1. IEEE, 2008.
 - [36] Ouyang, Mingsan, et al. "Continuous Reactor Temperature Control with Optimized PID Parameters Based on Improved Sparrow Algorithm." Processes 11.5 (2023): 1302.
 - [37] Li, Chunyan, et al. "Time-Optimal trajectory Generation for Industrial robots based on Elite Mutation Sparrow Search Algorithm, 126-135." Int. J. Robotics Autom. 38.2 (2023).
 - [38] Ma, Jie, Zhiyuan Hao, and Wenjing Sun. "Enhancing sparrow search algorithm via multi-strategies for continuous optimization problems." Information Processing & Management 59.2 (2022): 102854.
 - [39] Jamil, Momin, and Xin-She Yang. "A literature survey of benchmark functions for global optimisation problems." International Journal of Mathematical Modelling and Numerical Optimisation 4.2 (2013): 150-194.
 - [40] Kudela, Jakub. "Novel zigzag-based benchmark functions for bound constrained single objective optimization." 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2021.
 - [41] Zhang, Jiangnan, et al. "Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection." Mathematical Problems in Engineering 2021 (2021): 1-18.
 - [42] Derrac, Joaquín, et al. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." Swarm and Evolutionary Computation 1.1 (2011): 3-18.
 - [43] Zhang, Xinming, et al. "Improved GWO for large-scale function optimization and MLP optimization in cancer identification." Neural computing and applications 32 (2020): 1305-1325.
 - [44] Xue, F. "Research and application of heuristic intelligent optimization based on bat algorithm." Beijing University of Technology (2016).
 - [45] Xie, Hailun, Li Zhang, and Chee Peng Lim. "Evolving CNN-LSTM models for time series prediction using enhanced grey wolf optimizer." IEEE access 8 (2020): 161519-161541.