Journal of Computers Vol. 33 No. 4, August 2022, pp. 181-193
doi: 10.53106/199115992022083304015

An Improved Cuckoo Search Algorithm Based on Inertial Weight and Scaling Factor

Yang Yang, Maosheng Fu, Chaochuan Jia*, Xiancun Zhou

Electronics and Information Engineering, West Anhui University, Lu'an, People's Republic of China
yangyy@wxc.edu.cn, 49593838@qq.com, ccjia@hfcas.ac.cn, 382453126@qq.com

Received 26 January 2022; Revised 8 February 2022; Accepted 8 February 2022

Abstract. This paper presents a novel and an improved cuckoo search algorithm (ISCS), different from other techniques, which sets nonlinear decreasing inertia weight and adaptive scaling factor. As the iteration goes on, these two parameters are controlled by two functions to change the iterations dynamically. At the beginning of the iteration, the values of the two parameters are favorable for global search, and at the end of the iteration, their values are more favorable for local optimization. In this work, 23 classical benchmark functions are selected to improve the accuracy and convergence speed by conducting the simulation experiments on CS, ISCS and other three algorithms. The results show that the improved cuckoo algorithm enhances the accuracy of understanding and speeds up the convergence of the curves. Finally, ISCS is applied to probabilistic neural network (PPN) neural network classification and recognition technology, and the results show that the optimization of ISCS can effectively improve the classification accuracy of the test sets, and has diverse applications.

Keywords: Cuckoo search algorithm, inertia weight, scaling factor, PPN neural network

1 Introduction

In 2009, Yang from Cambridge University and Deb from Raman Engineering University proposed a global optimization algorithm called CS algorithm to simulate the nesting behavior of the cuckoo [1-2]. It was based on studying the ecology over a long time and used the way the cuckoo lays eggs, fosters care and levy flight characteristics. Its virtues includes adaptability, few arguments, stochastic exploration, and optimization ability [3]. However, in the later stage of the algorithm, the algorithm has drawbacks like, slow search speed, less accuracy, etc., so CS needs further improvement in its performance [4-6].

At present, the CS algorithm method is used to solve various engineering optimization problems [7] and has potential research value. Although the CS algorithm has few control parameters, like simple structure, easy to implement, and a slow convergence rate later, it may give optimal local solutions like other biological heuristic algorithms. Therefore, many domestic and foreign scholars studied these problems and proposed suitable improved algorithms [8].

Thanh Cuong-Le et al. [9] proposed a new cuckoo algorithm (NMP-CS) where, the step size is changed to a scalar number of direction parameters making random walking more flexible, better convergence speed and accuracy. A. Jaballah et al. [10] presented a new adaptive cuckoo search (SACS) algorithm, which dynamically adjusts the control parameters. The adaptive algorithm increases the flexibility of the algorithm, and is applied to THE RFID network planning problem. T. T. Nguyen et al. [11] suggested an improved cuckoo search algorithm (ICSA), which adds a local search mechanism, uses the candidate solution of the optimal solution to find better the global optimal solution earlier and better, and applies it to eElectric Distribution Network Reconfiguration (EDNR). P. Ong et al. [12] formulated a new initialization strategy for cuckoo search algorithm (MCSA), which mimics cuckoo reproduction mechanism. The refined solution is used as the initial translation vector of the Wavelet nNeural Networks (WNN) to optimize the prediction accuracy.

X. Li et al. [13] appended neighborhood parameters in order to enhance population diversity. New strategies were suggested to achieve a balance in development and discovery. A. F. Ali [14] used number of iterations to search and then pass the optimal solution to the Nelder-Mead algorithm as a strengthening parameter so that the search speed is significantly improved. Y. Tian [15] took the fitness as a waiver measure, and showed that the better solution was more likely to survive, and the property of that algorithm was also enhanced. H. Zheng [16] proposed a walking strategy, a swap reversal strategy, and a greedy strategy to get a smaller average iteration number. X. Zhou [17] presented the bird's nest position update strategy and discard operator resulting in better convergence speed and accuracy of the algorithm.

* Corresponding Author

J. B. Liao [18] adjusted the inertia weight by changing the iteration and current position. Y. Peng [19] used two parameters of step size and probability with a dynamic change. Y. W. Zhang [20] used the strategy of keeping the improvement rate at 1/5 to control the step factor and discovery probability. L. Lu [21] adaptively adjusted the step size according to the change from the optimal nest position in different stages. S. Zhao [22] sorted the fitness values of bird's nest positions to form the dynamic fitness value of the inertial weight and improved the CS algorithm. X. Q. Yang [23] replaced one of the random locations in the local search with a historical optimal individual location to improve the local exploration performance of the population. T. Liu [24] added a coefficient of dynamic change in front of the step size factor to control the magnitude of the step size reduction. H. Zhu [25] added the scaling method in local search to improve the local search effect. L. Chen [26] introduced a random local search algorithm and a dynamic inertia weight strategy with a nonlinear decrease. X. Hu [27] used the individual fitness value to adjust the scaling factor and discovery probability for improving search capability and convergence speed.

In order to improve the convergence speed and increasing the accuracy of the algorithm, the first part proposed an improved cuckoo algorithm (ISCS) based on the inertia weight and scaling factor. The second part introduced the standard CS, the third part described the improved strategy of ISCS, and the fourth part compared the solution and convergence of the improved algorithm with other algorithms through simulation experiments. The fifth part used ISCS to optimize the Probabilistic Neural Network (PNN) classification method.

2 The Standard CS Algorithm

Entomologists have found that gulfs raise their young ones by depositing in other birds' nests rather than building their nests [28]. However, these exotic birds' eggs may be discovered by the host, which may abandon the young birds or rebuild the new nest [29].

Yang and Deb proposed three perfect states: (a) cuckoos produce one egg at a time and select stochastic nests for incubation; (b) the best nests are carried on to the next generation; and (c) the owner of the nest may find it with a probability Pa. If the nest is found, the host can discard it and establish a new one.

The nest position is updated, as shown in Equation (1):

$$\chi_i^{(t+1)} = \chi_i^{(t)} + \alpha \oplus levy(\lambda) \ . \tag{1}$$

Where, $\chi_i^{(t)}$ represents the nest position of t, α represents the step size control quantity, $\oplus$ is dot-to-dot multiplication, and $levy(\lambda)$ is the stochastic exploration strategy as shown in Equation (2):

$$levy(\lambda) \tilde{} \ \mu = t^{-\lambda} \left(1 < \lambda \leq 3\right). \tag{2}$$

In the process of nest hunting, cuckoo shows the essence of Levi's flight. In the optimization path, frequent short steps occasionally appear as long steps.

In literature [30], Equation (3) is used to represent Levi's flight:

$$levy(\lambda) \tilde{} \ \frac{\phi * \mu}{\left|\upsilon\right|^{\frac{1}{\beta}}} \ . \tag{3}$$

Where, υ and μ follow the standard normal distribution, β=1.5.

$$\phi = \left[\frac{\Gamma\ (1+\beta) \times \sin\left(\pi \frac{\beta}{2}\right)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}\right)} \right]^{\frac{1}{\beta}} . \tag{4}$$

To calculate the Levy flight, α in reference [2] is adopted:

$$\alpha = \alpha_0 * \left(\chi_{\mathrm{i}}^{(\mathrm{t})} - \chi_{\mathrm{best}}\right). \tag{5}$$

α_0 is 0.01, χ_{best} is the best position.

By combining formula (1) ~ (5), The CS algorithm adopts Equation (6) to update the new bird nest position:

$$\chi_i^{(\mathrm{t+1})} = \chi_i^{(t)} + \alpha_0 \frac{\phi * \mu}{\frac{1}{|\upsilon|^{\beta}}} * \phi * \left(\chi_{\mathrm{i}}^{(\mathrm{t})} - \chi_{\mathrm{best}}\right). \tag{6}$$

If the non-host bird egg is found, the compression factor γ is generated after the bird's nest position is updated through levy flight, and the interval is (0, 1). When γ > Pa, a new solution is generated by preference migration. $\chi_i^{(\mathrm{t+1})}$ as shown in (7):

$$\chi_i^{(\mathrm{t+1})} = \chi_i^{(t)} + \gamma * \left(\chi_{\mathrm{j}}^{(\mathrm{t})} - \chi_{\mathrm{k}}^{(\mathrm{t})}\right). \tag{7}$$

Where, $\chi_{\mathrm{j}}^{(\mathrm{t})}$ and $\chi_{\mathrm{k}}^{(\mathrm{t})}$ represent the two random solutions of generation t.

3 The ISCS Algorithm

The standard CS algorithm uses levy flight random walk and preference random walk strategies to update the nest locations. However, due to the uncertainty of the random walk, the search capability gets weaker, and the convergence speed also slows down [28]. In ISCS, a nonlinear decreasing inertia weight and the adaptive scaling factor are proposed to enhance convergence speed and solution accuracy.

3.1 Nonlinear Decreasing Inertial Weight

At the beginning of the iteration, in order to ensure the global exploration capability, the inertia weight should be set higher; however, with iteration, it should be set smaller to improve the local exploration capability. Therefore, it is necessary to adjust it dynamically to balance the whole and part search capabilities fully. The ISCS algorithm introduces the inertial weight ω:

$$\chi_i^{(\mathrm{t+1})} = \omega \times \chi_i^{(t)} + \alpha \oplus levy(\lambda) \ . \tag{8}$$

The inertia weight ω adopts a non-linear decreasing strategy, as shown in Formula (9):

$$\omega = \omega_{\max} - \left(\omega_{\max} - \omega_{\min}\right) * \left(2.0 / \mathrm{iter}\right)^{0.3} . \tag{9}$$

Where, iter is the number of current iterations, $\omega_{\max}$ and $\omega_{\min}$ are the original and terminal values and ω increases as iter increases. At the beginning of the iteration, the initial value is small, indicating that the influence of the optimal solution of previous generation is small, the diversity of the population can be maintained, and the global exploration ability is strong. As the iteration goes on, larger the current value, the greater is the influence of the previous generation solution, and stronger the local development ability, the faster is the convergence rate. Until the last iteration, the value is the largest and the local optimization ability is the strongest.

3.2 Adaptive Scaling Factor

Formula (7) produces a new solution, which is the variation operation of individuals, and to a certain extent, it makes the population maintain high diversity [27]. The scaling factor γ is used to control the scaling degree. In the beginning, the scaling factor is a big number, and the range of random walk is large, which is conducive to improve the population diversity. Later, the smaller the value of scaling factor, the smaller is the random walk range, and the local exploration ability is strong. Therefore, the ISCS algorithm proposes an adaptive scaling factor, as shown in Formula (10):

$$\gamma = \gamma_{\min} + \left(\gamma_{\max} - \gamma_{\min}\right) * \left(2.0 / \text{iter}\right)^{0.3}. \quad \textbf{(10)}$$

Where, $\gamma_{\max}$ and $\gamma_{\min}$ are the original and the terminal values and γ decreases as iter increases. At the beginning of iteration, the initial value is big, indicating that the influence of the random value of previous generation is high, the diversity of the population can be maintained, and the global exploration ability is strong. As the iteration goes on, the smaller the parameter value, the smaller is the influence of the previous generation of random solutions and the stronger the local development ability, the faster is the convergence speed. Until the late iteration, the value is the smallest and the local optimization ability is the strongest.

3.3 ISCS Algorithm Process

The ISCS algorithm process has five steps:

The first step chooses the objective function, initializes the bird's flock, and sets the algorithm parameters like dimension, number of nests, the initial position of nests, maximum iteration times, etc;

The second step counts a new combination of fitness values;

The third step uses levy flight of nonlinear decreasing inertia weight [(formula (8) and (9)] to renew the location of the nests and make corresponding changes;

The fourth step calculates the adaptive scaling factor [formula (10)] and compares with Pa. If $\gamma >$ Pa, then enter a random offset and get another value;

The fifth step determines whether the optimal nest location satisfies the maximum iteration times of the algorithm. If yes, output is the optimal value; otherwise, it returns to step 2 to continue iteration.

The ISCS algorithm process is shown in Fig. 1.

3.4 Time Complexity Analysis of the Algorithm

According to the CS process, if the time to generate uniformly distributed random numbers is b_1, the population size is N, the objective function is f(n), the maximum number of iterations is n, and the computation time complexity is $O(N(b_1n + f(n))) = O(n + f(n))$, The execution time of Formula (1) is b_2, and the comparison time between the new solution and the previous solution is b_3. If the new solution is better, the replacement time of the previous solution is b_4. In the generation stage of the first new solution, the time complexity is $O(N(b_2n + f(n) + b_3 + b_4n)) = O(N + f(n))$. In the second new solution stage, the time to generate a new solution is b_4 and the time complexity is $O(N(b_5n + f(n) + b_3 + b_4n)) = O(n + f(n))$. The recorded time complexity of the optimal solution is $O(N(b_3 + b_4n)) = O(n)$, so the total intertemporal complexity of each iteration is $T(n) = 3O(n + f(n)) + O(n) = O(n + f(n))$. [20].

Similarly, to analyze the process of ISCS, the inertia weight ω and the contraction factor γ are added, and the time of ω generation is set as b, and the time of γ generation is set as b'. In the first stage of the new solution, the time interval complex impurity is $O(N(bn + b'n + b_2n + f(n) + b_3 + b_4n)) = O(n + f(n))$. In the second stage of the new solution, the time complexity is $O(N(bn + b'n + b_5n + f(n) + b_3 + b_4n)) = O(n + f(n))$. Therefore, the time complexity of each iteration remains unchanged, $T(n) = O(n + f(n))$, which is consistent with CS without increasing the time complexity.

4 Experiment and Discussion

The experiment selected 23 functions [31] (see Table 2 to Table 4) to test the performance of ISCS and carried out simulation experiments, the test results were compared with CS, ICS [2], ACSDAM [32] and AIWCS [1] Io and analyzed the solving accuracy and convergence rate of the curves. Then CS and ISCS were used to optimize PNN classification and recognition, and the classification accuracy curve of the optimized test set was finally analyzed.

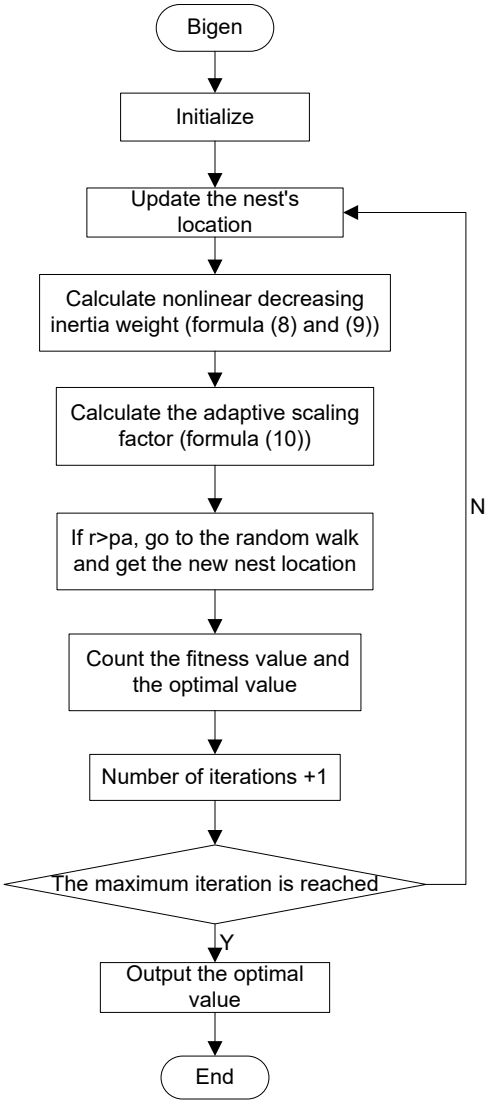


Fig. 1. Process of ISCS

4.1 Accuracy Analysis of the Algorithm

Algorithm parameters include bird's nest n=30 and the maximum number of iterations are 600. After many experimental tests, the range of ω is (0.4, 0.9), the range of γ is (0.1, 0.4), and the discovery probability is 0.25. Under these circumstances, the fitness value of the ISCS algorithm is the best.

Since the optimization algorithm is random, the algorithm is run 50 times in the same environment. Table 1 to Table 3 record the four groups of values.

Functions 1-7 are unimodal benchmark functions, and each function has only one optimal value. In f1-f5 and f7, the best, worst, average, and variance of the ISCS algorithm are all smaller than the other four algorithms, indicating that ISCS has higher solving accuracy and optimization capability.

Functions f8-f13 are multimodal benchmark functions. In f8-f11, the optimal global value, worst value, average value, and variance of the ISCS algorithm are all superior to the other four algorithms, indicating that the ISCS algorithm has strong stability and the ability to deal with the complex optimization processes.

Functions f14-f23 are fixed-dimension multimodal benchmark functions. In f15, the worst value of ISCS is the lowest. In other functions, ISCS and the other four algorithms can search for the optimal global value. In f14, f16, f17, f22, the perspective of variance of the ISCS algorithm is much smaller than the other algorithms, indicating that the solution result of ISCS is more stable than other algorithms.

Table 1. Simulation results of unimodal benchmark functions

f	Algorithm	Best	Worst	Mean	Std. Dev.
f1	CS	0.8361	3.7105	2.0768	1.0008
	ICS	0.0020	0.0125	0.0054	0.0036
	ACS-DAM	0.0016	0.0074	0.0042	0.0017
	AIWCS	5.0551e-83	6.4094e-55	6.4094e-56	2.0268e-55
	ISCS	4.3265e-136	5.0760e-135	1.8245e-135	1.3503e-135
f2	CS	3.5068	10.6319	5.7035	1.9012
	ICS	0.0889	1.1367	0.3868	0.3403
	ACS-DAM	0.0323	0.2886	0.1097	0.0713
	AIWCS	5.5856e-43	30.0000	8.2610	11.7310
	ISCS	1.1663e-68	2.4656e-68	1.8065e-68	4.7644e-69
f3	CS	1.0766e+03	2.1503e+03	1.4635e+03	376.0671
	ICS	505.1173	1.9933e+03	897.1253	437.4988
	ACS-DAM	941.1554	3.1924e+03	2.0317e+03	654.4582
	AIWCS	1.5776e+03	3.4878e+03	2.5548e+03	609.6305
	ISCS	8.8004e-135	5.0249e-134	1.7692e-134	1.3130e-134
f4	CS	4.8005	10.5037	7.4291	1.8677
	ICS	3.8949	6.5773	5.1417	1.0478
	ACS-DAM	2.6962	6.2897	4.2896	1.0539
	AIWCS	7.9689	12.4614	9.4647	1.4184
	ISCS	2.1346e-68	4.1004e-68	3.1058e-68	6.1597e-69
f5	CS	157.2571	445.7113	265.9129	87.3025
	ICS	32.8073	130.7288	63.1054	40.3580
	ACS-DAM	27.2029	64.4901	40.0813	12.9790
	AIWCS	27.3735	28.4305	28.0787	0.3175
	ISCS	25.4376	28.8092	28.6955	0.1173
f6	CS	0.5402	2.5344	1.3549	0.7177
	ICS	0.0018	0.0136	0.0069	0.0047
	ACS-DAM	0.0025	0.0052	0.0039	8.5925e-04
	AIWCS	0.8804	2.6582	1.2954	0.5142
	ISCS	1.6706	3.5672	2.8776	0.5431
f7	CS	0.0389	0.1344	0.0747	0.0291
	ICS	0.0276	0.0536	0.0334	0.0129
	ACS-DAM	0.0164	0.0453	0.0324	0.0094
	AIWCS	1.0797e-05	1.6187e-04	5.6250e-05	5.0628e-05
	ISCS	6.3108e-06	2.7575e-04	1.5113e-05	6.2293e-05

This shows that ISCS exhibits properties like better global detection capability, complex optimization processing capability, higher solution accuracy, and a more stable process.

Table 2. Simulation results of multimodal benchmark functions

f	Algorithm	Best	Worst	Mean	Std. Dev.
f8	CS	-8.6627e+03	-7.7840e+03	-8.2524e+03	275.2227
	ICS	-7.6853e+03	-6.3918e+03	-7.0744e+03	434.3226
	ACS-DAM	-8.6816e+03	-6.4654e+03	-7.2657e+03	637.8040
	AIWCS	-6.5397e+03	-5.7079e+03	-6.0734e+03	325.6517
	ISCS	-6.5477e+03	-4.7601e+03	-5.3878e+03	521.3770
f9	CS	71.9421	120.1477	95.8603	16.6470
	ICS	111.7343	204.3825	166.2980	35.3873
	ACS-DAM	101.7714	174.1430	142.9221	20.7705
	AIWCS	0	25.4409	6.7067	9.3890
	ISCS	0	0	0	0
f10	CS	3.4130	6.1485	4.9966	0.9182
	ICS	0.1099	1.9669	0.8055	0.7186
	ACS-DAM	0.0636	4.3734	0.9838	1.2906
	AIWCS	14.2795	18.4561	15.6434	1.4188
	ISCS	8.8818e-16	8.8818e-16	8.8818e-16	0
f11	CS	0.7342	0.9957	0.8919	0.0768
	ICS	0.0103	0.1790	0.0741	0.0560
	ACS-DAM	0.0190	0.0816	0.0439	0.0221
	AIWCS	1.0047	1.2032	1.0992	0.0631
	ISCS	0	0	0	0
f12	CS	1.7529	4.4834	2.7070	0.9832
	ICS	0.0088	0.9469	0.1690	0.2857
	ACS-DAM	0.0033	0.3809	0.0781	0.1164
	AIWCS	0.6897	3.5517	2.2895	0.8410
	ISCS	0.0749	0.2803	0.1886	0.0636
f13	CS	0.9467	3.5069	1.8451	0.5715
	ICS	0.0333	0.1151	0.0798	0.0267
	ACS-DAM	0.0070	0.0705	0.0376	0.0216
	AIWCS	0.4750	5.8514	1.4187	1.6236
	ISCS	0.9406	1.4354	1.2080	0.2829

Table 3. Simulation results of fixed-dimension multimodal benchmark functions

f	Algorithm	Best	Worst	Mean	Std. Dev.
f14	CS	0.9980	0.9980	0.9980	1.2820e-16
	ICS	0.9980	0.9980	0.9980	0
	ACS-DAM	0.9980	0.9980	0.9980	0
	AIWCS	0.9980	0.9980	0.9980	5.4641e-16
	ISCS	0.9980	0.9980	0.9980	0
f15	CS	3.0855e-04	4.4121e-04	3.6213e-04	4.4456e-05
	ICS	3.0749e-04	4.2793e-04	3.2149e-04	3.7799e-05
	ACS-DAM	3.0749e-04	3.6087e-04	3.1913e-04	1.8696e-05
	AIWCS	3.0749e-04	3.1027e-04	3.0779e-04	8.7418e-07
	ISCS	3.0749e-04	3.0749e-04	3.0749e-04	8.0301e-16
f16	CS	1.0316	1.0316	1.0316	1.4803e-16
	ICS	1.0316	1.0316	1.0316	7.4015e-17
	ACS-DAM	1.0316	1.0316	1.0316	0
	AIWCS	1.0316	1.0316	1.0316	2.6686e-16
	ISCS	1.0316	1.0316	1.0316	0

Table 4. Simulation results of fixed-dimension multimodal benchmark functions (continued)

f	Algorithm	Best	Worst	Mean	Std. Dev.
	CS	0.3979	0.3979	0.3979	0
	ICS	0.3979	0.3979	0.3979	0
f17	ACS-DAM	0.3979	0.3979	0.3979	0
	AIWCS	0.3979	0.3979	0.3979	0
	ISCS	0.3979	0.3979	0.3979	0
	CS	3.0000	3.0000	3.0000	1.2207e-15
	ICS	3.0000	3.0000	3.0000	4.4409e-16
f18	ACS-DAM	3.0000	3.0000	3.0000	5.3373e-16
	AIWCS	3.0000	3.0000	3.0000	1.7764e-15
	ISCS	3.0000	3.0000	3.0000	1.6244e-15
	CS	-3.8628	3.8628	-3.8628	9.0043e-16
	ICS	-3.8628	3.8628	-3.8628	9.3622e-16
f19	ACS-DAM	-3.8628	3.8628	-3.8628	9.3622e-16
	AIWCS	-3.8628	3.8628	-3.8628	5.9212e-16
	ISCS	-3.8628	3.8628	-3.8628	7.4015e-16
	CS	-3.3220	-3.3220	-3.3220	1.7382e-08
	ICS	-3.3220	-3.3220	-3.3220	4.6811e-16
f20	ACS-DAM	-3.3220	-3.3220	-3.3220	4.6811e-16
	AIWCS	-3.3220	-3.3220	-3.3220	1.3846e-08
	ISCS	-3.3220	-3.3220	-3.3220	3.9324e-09
	CS	-10.1532	-10.1532	-10.1532	1.0555e-09
	ICS	-10.1532	-10.1532	-10.1532	1.7764e-15
f21	ACS-DAM	-10.1532	-10.1532	-10.1532	1.1842e-15
	AIWCS	-10.1532	-10.1532	-10.1532	3.4410e-07
	ISCS	-10.1532	-10.1532	-10.1532	2.1495e-10
	CS	-10.4029	-10.4029	-10.4029	1.4222e-07
	ICS	-10.4029	-10.4029	-10.4029	1.7764e-10
f22	ACS-DAM	-10.4029	-10.4029	-10.4029	1.5666e-10
	AIWCS	-10.4029	-10.4029	-10.4029	3.4446e-15
	ISCS	-10.4029	-10.4029	-10.4029	2.2411e-15
	CS	-10.5364	10.5364	-10.5364	6.3881e-06
	ICS	-10.5364	10.5364	-10.5364	1.1842e-15
f23	ACS-DAM	-10.5364	10.5364	-10.5364	1.7764e-15
	AIWCS	-10.5364	10.5364	-10.5364	2.0900e-06
	ISCS	-10.5364	10.5364	-10.5364	2.6123e-10

4.2 Convergence Analysis of the Algorithm

Fig. 2 and Fig. 3 show several convergence curves (a–o) of CS, ICS, ACSDAM, AIWCS, and ISCS. ISCS performs very well when the test function searches for the optimal global value.

First of all, the ISCS algorithm has a great advantage in the convergence speed from the early stage of the iteration to the end of the iteration, and it converges to the optimal faster speed. This situation is reflected in f1-f4.

Secondly, it can be seen from f5, f7, f9-f13 that in the early stage of iteration, the curve of the ISCS algorithm drops faster, which has advantages in convergence speed. Later, the ISCS curve is more gently compared with other curves, ensuring the accuracy of the local search.

Finally, in f15, f18, and f20, the convergence rates of the five algorithms are very close in the early stage of the iteration. In the progress of iteration, the convergence speed of the ISCS rapidly accelerates, and the optimal solution is found earlier.

In general, the ISCS algorithm, in the entire iteration process, can balance the global exploitation and local precision exploration increase the convergence rate, and find the optimal global value at an earlier stage.

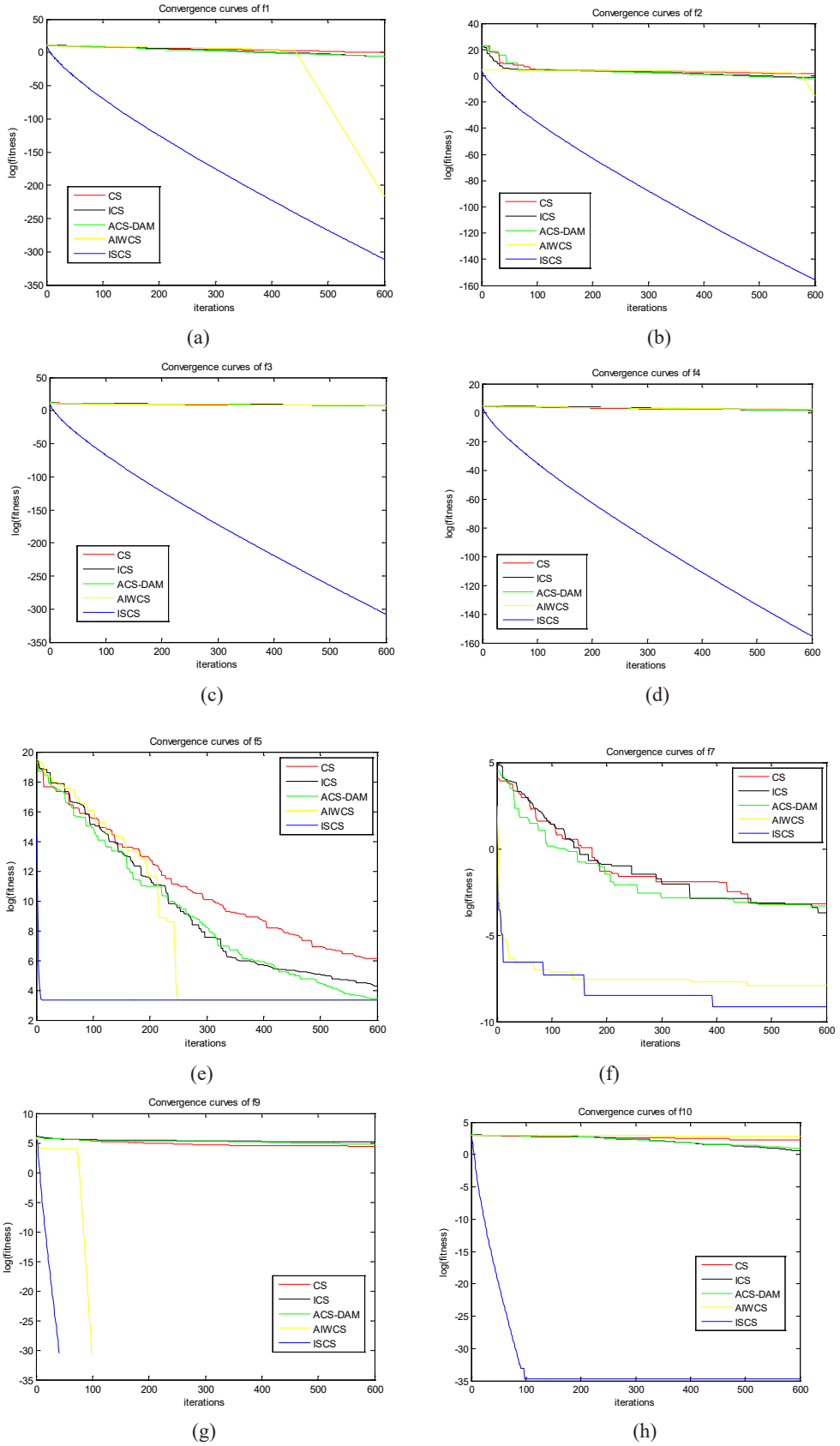
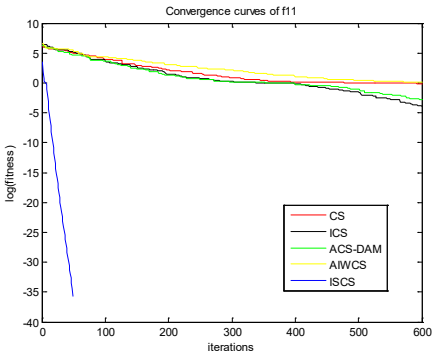
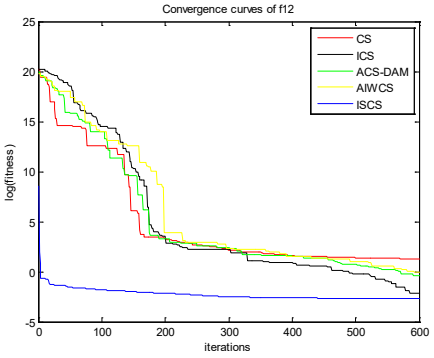


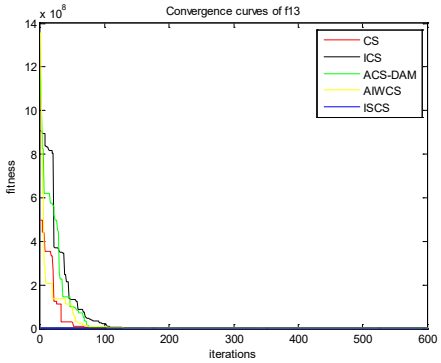
Fig. 2. Convergence curves of functions



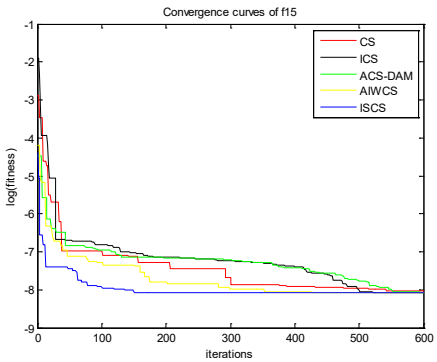
(a)



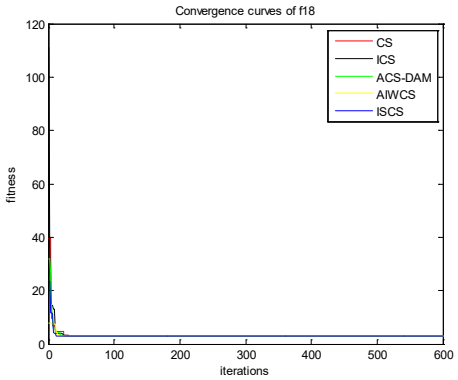
(b)



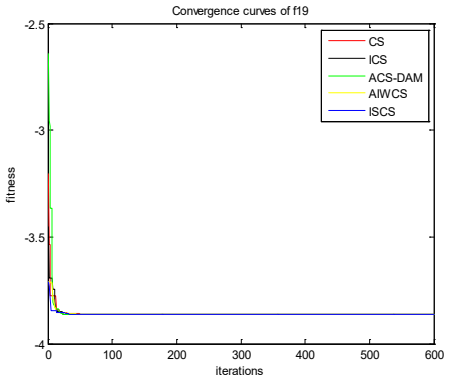
(c)



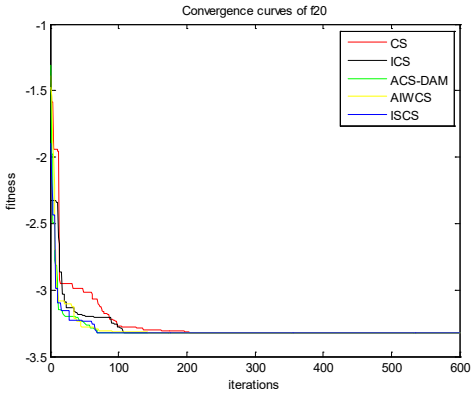
(d)



(e)



(f)



(g)

Fig. 3. Convergence curves of functions (continued)

5 ISCS Optimized PNN Classification Method

PNN is a probabilistic neural network algorithm based on the probability density function estimation of the Par-Zen window and Bayes classification rule [33]. Standard PNN neural network has a simple structure principle, concise training, and faster convergence speed [34]. However, since the value range of smooth factor spread is relatively larger, the selection of its value can influence the classification performance and running speed of PNN to a greater Extent. Therefore, to enhance the classification and recognition efficiency of PNN, it is usually necessary to optimize the spread.

The test data imported in the experiment were samples from three different types of iris, and 120 samples (40 samples*3 groups) were randomly selected to form the training set, and 30 samples (10 samples*3 groups) to form the test set. When the random value of smooth factor spread was selected, the prediction accuracy of the test set was 90%. When the spread is optimised with CS, the prediction accuracy of test sets is 93.33%. However, if the spread is optimised with ISCS, the prediction accuracy of the test sets reaches 96.667% as shown in Fig. 4. On the other hand, Fig. 5 shows that after the parameter optimization of the two algorithms, ISCS reached the highest prediction accuracy in the second iteration of the test set. In contrast, CS reached the highest accuracy in the 20th iteration which indicates that the optimization capability of ISCS is stronger, both the convergence rate and the efficiency are higher than CS.

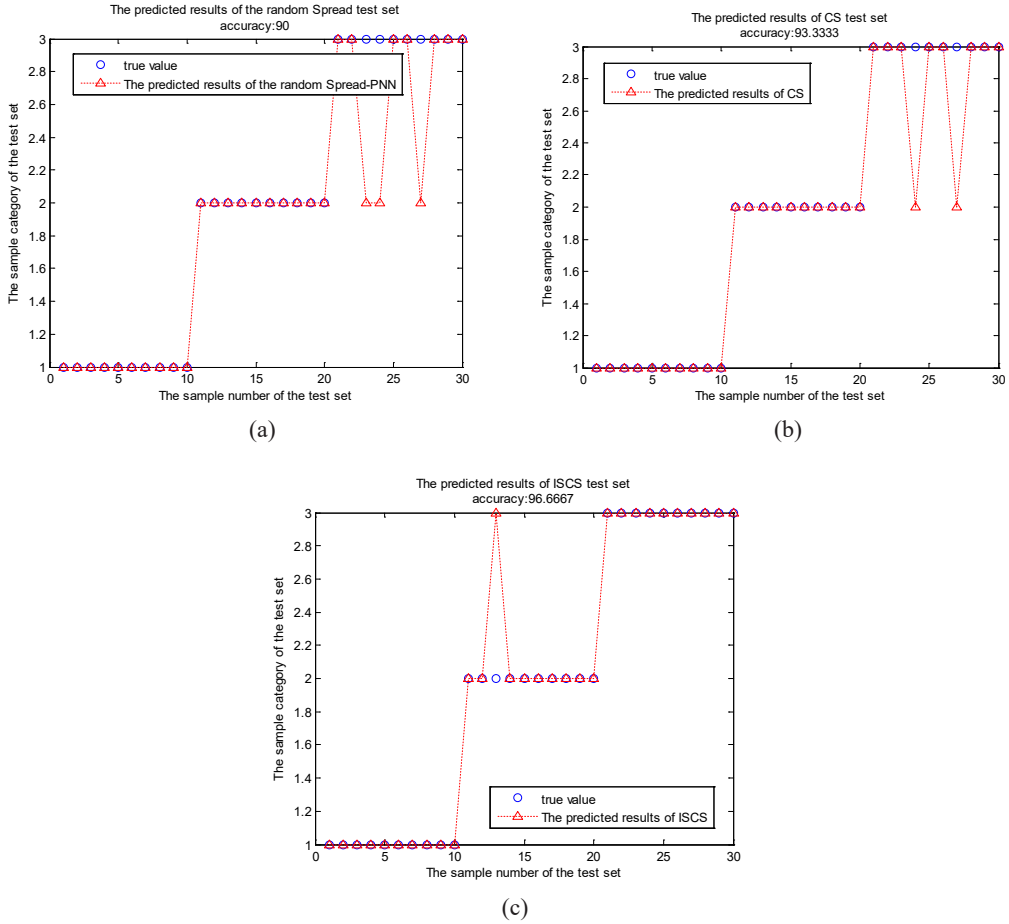


Fig. 4. Comparison of predicted results of test sets

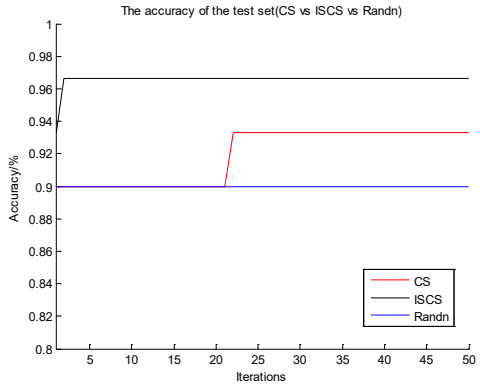


Fig. 5. Comparison of the accuracy of the test sets (CS vs ISCS vs Random)

6 Conclusion

This paper proposes a new cuckoo search algorithm, ISCS. The main innovations include nonlinear decreasing inertia weight and automatic adjustment of adaptive scaling factor. When the nest iterates continuously, these two parameters are adjusted dynamically to achieve a balance between global development and local exploration to find the optimal solution. Simulation results show that the convergence speed and optimization accuracy of the proposed algorithm are better than those of the standard CS algorithm and other three improved algorithms. Finally, the classification and recognition optimization problem of PNN are studied, and the ISCS algorithm shows better advantages, which not only improves the prediction accuracy, but also helps to achieve the accuracy at an earlier stage. Therefore, ISCS algorithm is a new and effective cuckoo algorithm, and has important application value in the optimization of classification recognition. In the future, the ISCS algorithm will be improved and combined with the classification and recognition technology of characteristic plant digital images.

7 Acknowledgment

This study is funded by the Youth Program of West Anhui University (WXZR201903), Academic funding project for top-notch academic talents in universities (gxbjZD2020084), the Natural Science Research Project of West Anhui University (0041021002, 0041021003). The authors thank all the agencies for extending their help to conduct this work.

References

[1] X.S. Yang, S. Deb, Cuckoo Search via Levy Flights, in: Proc. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009.
[2] X.S. Yang, S. Deb, Engineering Optimization by Cuckoo Search, International Journal of Mathematical Modelling and Numerical Optimization 1(4)(2010) 330-343.
[3] P.R. Srivastava, R. Khandelwal, S. Khandelwal, S. Kumar, S.S. Ranganath, Automated test data generation using cuckoo search and tabu search (CSTS) algorithm, Journal of Intelligent Systems 21(2)(2012) 195-224.
[4] K.T. Lan, C.H. Lan, Notes on the distinction of Gaussian and Cauchy mutations, in: Proc. 2008 Eighth International Conference on Intelligent Systems Design and Applications, 2008.
[5] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics 19(1)(2005) 43-53.
[6] Y. Li, J. Zhou, J. Yang, L. Liu, Modified Shuffled Frog Leaping Algorithm based on threshold selection strategy, Computer engineering and applications 43(35)(2007) 19-21.
[7] X. Zhang, X. Wang, Survey of cuckoo search algorithm, Computer Engineering and Applications 54(18)(2018) 8-16.
[8] H. Zhang, X. You, S. Liu, Z. Liu, Interactive learning cuckoo search algorithm, Computer Engineering and Applications 56(7)(2020) 147-154.
[9] T. Cuong-Le, H.-L. Minh, S. Khatir, M.A. Wahab, M.T. Tran, S. Mirjalili, A novel version of Cuckoo search Algorithm for solving Optimization problems, Expert Systems with Applications 186(2021) 115669.
[10]A, Jaballah, A. Meddeb, A new variant of cuckoo search algorithm with self adaptive parameters to solve complex RFID network planning problem, Wireless Networks 25(4)(2017) 1585-1604.

[11]T.T. Nguyen, T.T. Nguyen, An improved cuckoo search algorithm for the problem of electric distribution network reconfiguration, Applied Soft Computing 84(2019) 105720.
[12]P. Ong, Z. Zainuddin, Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction, Applied Soft Computing 80(2019) 374-386.
[13]X. Li, M. Yin, A particle swarm inspired cuckoo search algorithm for real parameter optimization, Soft Computing 20(4) (2016) 1389-1413.
[14]A.F. Ali, M.A. Tawhid, A hybrid cuckoo search algorithm with Nelder Mead method for solving global optimization problems, Springer Plus 5(1)(2016) 473.
[15]Y. Tian, M. Fang, An Improved Cuckoo Search Algorithm, Journal of Changchun University of Science and Technology (Natural Science Edition) 40(1)(2017) 115-118.
[16]Y. Zhou, H. Zheng, An improved Cuckoo Search Algorithm for Solving Planar Graph Coloring Problem, Applied Mathematics & Information Sciences 7(2)(2013) 785-792.
[17]X. Zhou, Y. Liu, B. Li, A multi-objective discrete cuckoo search algorithm with local search for community detection in complex networks, Modern Physics Letters B 30(7)(2016) 1650080.
[18]J. Liao, Z. Pa, Z. Zhi, Application of Optimized SVM and AIWCS in Fault Diagnosis of Grid-connected Inverter, Process Automation Instrumentation 36(3)(2015) 13-16.
[19]Y. Peng, H. Zheng, Improved Cuckoo Search Algorithm for No-idle Flow Shop Scheduling Problems, Modern Information Technology 3(24)(2019) 20-22.
[20]Y. Zhang, L. Wang, Q. Wu, Dynamic adaptation cuckoo search algorithm, Control and Decision 29(4)(2014) 617-622.
[21]L. Lu, L. Cheng, Network Traffic Prediction Model Based on Optimising Svm with Improved Cuckoo Search Algorithm, Computer Applications and Software 32(1)(2015) 124-127.
[22]S. Zhao, C. Qu, An Improved Cuckoo Algorithm for Solving the Problem of Logistics Distribution Center Location, Mathematics in Practice and Theory 47(3)(2017) 206-213.
[23]X. Yang, Research of DV-Hop Localization Algorithm Based on Modified Cuckoo Search Algorithm, Computer & Digital Engineering 46(9)(2018) 1819-1823.
[24]T. Liu, C. Ye, TFT-LCD Manufacturing Scheduling Method Based on Improved Cuckoo Search Algorithm, Computer Systems & Applications 29(3)(2020) 47-54.
[25]H. Zhu, G. Li, Image segmentation algorithm based on improved cuckoo search algorithm, Computer Engineering and Design 39(5)(2018) 1428-1432.
[26]L. Chen, W. Long, Modified cuckoo search algorithm for solving engineering structural optimization problem, Application Research of Computers 31(3)(2014) 679-683.
[27]X. Hu, Improved cuckoo search algorithm for function optimization problems, Computer Engineering and Design 34(10) (2013) 3639-3642.
[28]Y. Li, Z.Y. Shang, J. Liu, Improved Cuckoo Search Algorithm for Function Optimization Problems, Computer Science 47(1)(2020) 219-230.
[29]Z. Wang, C. Jia, Y. Sun, Parasitized breeding and nestlings growth in oriental cuckoo, Chinese Journal of Zoology 39(1) (2004) 103-105.
[30]F. Wang, X. He, Y. Wang, The cuckoo search algorithm based on Gaussian disturbance, Journal of Xi'an Polytechnic University 25(4)(2011) 566-569.
[31]S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software 95(2016) 51-67.
[32]Y. Yang, X. Zuo, M. Fu, S. Yu, C. Jia, Adaptive Cuckoo Search Algorithm based on Dynamic Adjustment Mechanism, Journal of Computers 32(5)(2021) 171-183.
[33]D. Zhang, Neural network application design based on Matlab, Beijing: China Machine Press, 2011.
[34]Q. Zhang, Y. Wang, C. Wang, Research on identification of pick wear degree of roadheader based on PNN neural network, Coal Science and Technology 47(6)(2019) 37-44.