

Received January 13, 2020, accepted January 31, 2020, date of publication February 17, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974498

An Improved Chicken Swarm Optimization Algorithm and Its Application in Robot Path Planning

XIMING LIANG¹, DECHANG KOU¹, AND LONG WEN²

¹School of Science, Beijing University of Civil Engineering and Architecture, Beijing 102600, China

²Key Laboratory of Economics System Simulation, Guizhou University of Finance and Economics, Guiyang 550025, China

Corresponding author: Dechang Kou (1048658032@qq.com)

This work was supported in part by the National Key Research Program of China under Grant 2016YFC0700601, in part by the National Natural Science Foundation of China under Grant 61463009, in part by the Joint Program for the Science and Technology Top Talents of Higher Learning Institutions of Guizhou under Grant KY[2017]070, in part by the Education Department of Guizhou Province Projects under Grant KY[2017]004, in part by the Central Support Local Projects under Grant PXM 2013_014210_000173, and in part by the Fundamental Research Funds for Beijing University of Civil Engineering and Architecture under Grant X18193.

ABSTRACT Chicken swarm optimization (CSO) algorithm is one of very effective intelligence optimization algorithms, which has good performance in solving global optimization problems (GOPs). However, the CSO algorithm performs relatively poorly in complex GOPs for some weaknesses, which results the iteration easily fall into a local minimum. An improved chicken swarm optimization algorithm (ICSO) is proposed and applied in robot path planning. Firstly, an improved search strategy with Levy flight characteristics is introduced in the hen's location update formula, which helps to increase the perturbation of the proposed algorithm and the diversity of the population. Secondly, a nonlinear weight reduction strategy is added in the chicken's position update formula, which may enhance the chicken's self-learning ability. Finally, multiple sets of unconstrained functions are used and a robot simulation experimental environment is established to test the ICSO algorithm. The numerical results show that, comparing to particle swarm optimization (PSO) and basic chicken swarm optimization (CSO), the ICSO algorithm has better convergence accuracy and stability for unconstrained optimization, and has stronger search capability in the robot path planning.

INDEX TERMS Chicken swarm optimization algorithm, Levy flight, nonlinear weight reduction strategy, numerical experiments, robot path planning.

I. INTRODUCTION

The swarm intelligent optimization algorithm, such as genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], bat algorithm (BA) [3], artificial bee colony algorithm (ABC) [4] *et al.*, is a stochastic optimization algorithm constructed by simulating the swarm behavior of natural organisms. These algorithms search for the optimal solution of an optimization problem by simulating the physical laws of nature phenomena, the living habits and behavioral characteristics of various biological populations in nature. The swarm intelligent algorithms provide a new way to solve global optimization problems in the fields of computational science, engineering science, management science and so on. The swarm intelligent optimization algorithms have become a research hotspot and are particularly important.

The associate editor coordinating the review of this manuscript and approving it for publication was Hongwei Du.

The chicken swarm optimization (CSO) algorithm [5] is a stochastic search method based on chicken swarm search behavior, which was proposed by Meng, *et al.* in 2014. In CSO, the whole chicken swarm is divided into several groups, each of which includes a rooster, a couple of hens and several chicks. Different chickens follow different laws of motions. There exist mutual learning and competitions between different chickens, and the hierarchy of the chicken group is updated again after several generations of evolution. The CSO algorithm has great research potential because of its good convergence speed and convergence accuracy. However, like other swarm intelligent optimization algorithms, the basic chicken swarm optimization algorithm has the disadvantages of premature convergence, whose iteration is easy to fall into a local minimum, in solving the large scale optimization problem with more complexity. Therefore, scholars have conducted in-depth research and proposed some improved chicken swarm optimization algorithm,

which have achieved good numerical results. For example, Fei and Dinghui [6] modified the position update formula of the chicks to avoid the iteration falling into some local optimums, but kept the position update formula of the other two groups unchanged, which only improves the CSO partially. Chen et al. [7] updated the hen's position formula to improve the accuracy and effectiveness of the CSO algorithm, but their algorithm needs more running time to reach the optimal solution. Chiwen et al. [8] substituted Gaussian distribution with an adaptive t-distribution in the rooster position update formula, and introduced an elite opposition learning strategy in the hen position update formula. These algorithms achieved good global search ability. The chicken swarm optimization algorithm has been applied in some practical areas. For example, Tiana et al. [9] used the chicken swarm algorithm to solve the problem of deadlock-free migration for virtual machine consolidation. Compared with the other deadlock-free migration algorithms, the chicken swarm algorithm have higher convergence rate. Shaolong et al. [10] solved the parameter estimation problem of nonlinear system using chicken swarm algorithm. Their numerical experiment results show that the chicken swarm algorithm is feasible for parameter estimation of nonlinear systems.

An improved chicken swarm algorithm is proposed, where the Levy flight strategy is added to the hen's position update formula to make the population distributed evenly. The nonlinear weight reduction strategy is employed into chick's position update formula to prevent the iteration from premature convergent and promote the convergence precision of the proposed algorithm. The better convergence accuracy and higher convergence speed of the improved chicken swarm algorithm has been verified by the numerical experiments on 8 benchmark GOPs. The robot path planning experiment results show that, compare to CSO, the proposed improved chicken swarm algorithm is more effective in improving search speed and quality of path.

II. CHICKEN SWARM ALGORITHM

The unconstrained continuous optimization problems can be expressed as follows.

$$\min f(X), \quad X \in R^D \quad (1)$$

If $X^* \in R^D$ satisfies that:

$$f(X^*) \leq f(X), \quad \forall X \in R^D$$

X^* is called the global minimum point of $f(X)$ in the whole space R^D .

The chicken swarm optimization algorithm mimics the hierarchal order and behaviors of searching food in the chicken swarm. Under specific hierarchal order, different chickens follow different laws of motions. Chicken swarm optimization algorithm uses the following four rules to idealize the behavior of chickens:

1) There are several subgroups in the chicken swarm. Each subgroup consists of a dominant rooster, a couple of hens and chicks.

2) How to divide the whole chicken swarm into several groups and how to determine the species of chicken depend on the fitness values of the chicken themselves. In the whole chicken swarm, several individuals with the best fitness values are identified as roosters; The chickens with worst several fitness values would be acted as chicks, the others would be hens. The hen chooses its subgroup randomly and the mother-child relationship between the hen and the chick is also formed randomly.

3) The hierarchal order, dominance relationship and mother-child relationship in a group keep unchanged during several generations until the role are reassigned.

4) The hens follow their rooster-mate to search food, and the chicks search for food around their mothers. The dominant individuals have advantage in searching food.

The whole chicken swarm is divided into three types of chickens. When the CSO algorithm solves the optimization problem (1), each chicken represents a feasible solution to problem (1) and different chickens follow different optimization strategies. In the CSO algorithm, the number of chickens are assumed to be N and chickens are arranged in rising order according to their fitness values. The N_R chickens in front are defined as roosters, the N_C chickens at last are called chicks and the remaining $N_H (= N - N_R - N_C)$ chickens in the middle are hens. Let $x_i^j(t)$ denote the position of the i^{th} chicken in the j^{th} dimension searching space in the t^{th} iteration, where $i \in (1, 2, \dots, N), j \in (1, 2, \dots, D), t \in (1, 2, \dots, M), M$ is the maximal iterative number. The position update formula of different type of chickens are as follows:

(a) The rooster's position update formula are:

$$x_i^j(t+1) = x_i^j(t) * (1 + Randn(0, \sigma^2)) \quad (2)$$

$$\sigma^2 = \begin{cases} 1, & \text{if } f_i \leq f_k \\ \exp\left(\frac{f_k - f_i}{|f_i| + \varepsilon}\right), & \text{else} \end{cases} \quad (3)$$

where $i \in [1, \dots, N_R], j \in [1, \dots, D], Randn(0, \sigma^2)$ is a Gaussian distribution with mean 0 and standard deviation σ , ε is a small constant to avoid the denominator being 0. k is a rooster's index, which is selected randomly between 1 and N_R except i, f_i and f_k is the fitness value of the i^{th} and k^{th} roosters.

(b) The hen's position update formula are:

$$x_i^j(t+1) = x_i^j(t) + S_1 * rand * (x_{r_1}^j(t) - x_i^j(t)) + S_2 * rand * (x_{r_2}^j(t) - x_i^j(t)) \quad (4)$$

$$S_1 = \exp\left(\frac{f_i - f_{r_1}}{|f_i| + \varepsilon}\right) \quad (5)$$

$$S_2 = \exp(f_{r_2} - f_i) \quad (6)$$

where $i \in [N_R + 1, \dots, N_H], j \in [1, \dots, D], rand$ is a uniform random number between 0 and 1, r_1 is an index of the rooster, which is in the i^{th} hen's group-mate, r_2 is an index of a chicken in all roosters and hens, and let $r_1 \neq r_2$.

(c) The chick's position update formula are:

$$x_i^j(t+1) = x_i^j(t) + FL * (x_m^j(t) - x_i^j(t)) \quad (7)$$

where $i \in [N_R + N_H + 1, \dots, N], j \in [1, \dots, D], m$ is an index of the mother hen corresponding to i^{th} chick, FL is a parameter in the range $[0, 2]$, which keeps the chick to forage for food around its mother.

III. IMPROVED CHICKEN SWARM ALGORITHM

A. IMPROVED SEARCH STRATEGY BASED ON LEVY FLIGHT

Given that the entire flock looks for food in an unpredictable environment, the ‘‘Levy flight’’ search strategy is characterized by short-range deep local search and occasional longer distance walks [11], [12], which can help to improve the search efficiency and increase disturbance to make the chickens distribute more evenly. In the chicken swarm optimization algorithm, the number of hens is the largest, so hens play an important role in the entire flock. Inspired by this, the ICSO algorithm introduces the Levy flight search strategy into hen’s position update formula, which can avoid the iterations falling into a local minimum and enhance the global search capability of the ICSO algorithm. The improved hen’s position update formula are as follows:

$$x_i^j(t+1) = x_i^j(t) + S_1 * rand * (x_{r_1}^j(t) - x_i^j(t)) + S_2 * rand * L_{evy}(\lambda) \otimes (x_{r_2}^j(t) - x_i^j(t)) \quad (8)$$

where $L_{evy}(\lambda)$ is the jump path of a random search whose step size obeys the Levy distribution, λ is a scaling parameter in range $[1, 3]$, \otimes is a vector operator representing point multiplication.

B. NONLINEAR STRATEGIES OF DECREASING INERTIA WEIGHT

In the basic CSO algorithm, the chicks only learn from their own mother. Once a mother falls into a local minimum, the chicks following this mother will also fall into the local minimum. In the proposed ICSO algorithm, the nonlinear strategies of decreasing inertia weight is employed to update chick’s position, which helps the chicks not only learn from their mother, but also learn from themselves. It will be verified by the numerical experiments that coupling the nonlinear decreasing inertia weight in chick’s position formula can avoid the ICSO algorithm falling into local minimum as soon as possible. The nonlinear decreasing inertia weights [13] ω update as follows:

$$\omega = \omega_{min} (\omega_{max} / \omega_{min})^{(1+ct/M)} \quad (9)$$

The position updating formula of chicks coupling the nonlinear decreasing inertia weights are as follows:

$$x_i^j(t+1) = \omega * x_i^j(t) + FL * (x_m^j(t) - x_i^j(t)) \quad (10)$$

where ω_{min} is the minimum inertia weight, $\omega_{min} = 0.4$, ω_{max} is the maximum inertial weight, $\omega_{max} = 0.95$, c represents an acceleration factor, $c = 10$.

C. THE PROCESS OF ICSO ALGORITHM

The procedure of the ICSO algorithm is as follows:

Step 1: Initialize chicken swarm and set the parameters N, N_R, N_H, N_c, N_m, G and M

Step 2: For each chicken, calculate its fitness value and initialize its current and global optimal position, let $t=1$.

Step 3: If $\text{mod}(t, G) = 1$, sort all chickens in rising order according to their fitness values, the N_R chickens in front would be acted as roosters, each of roosters represent a group. The N_c chickens at last would be designated as chicks, the others in the middle would be hens. The hens randomly choose a group to live in, the mother-child relationship between the hens and chicks is also randomly established.

Step 4: Update the positions of rooster, hen, chick by (2), (8), (10).

Step 5: Recalculate the fitness value of each chicken and update the global optimal position and current optimal position for each chicken.

Step 6: Judge whether the iteration termination condition is satisfied, if so, output the global optimal position and stop the iteration; otherwise, let $t=t+1$ and return to Step 3.

The flowchart of the ICSO algorithm is as follows:

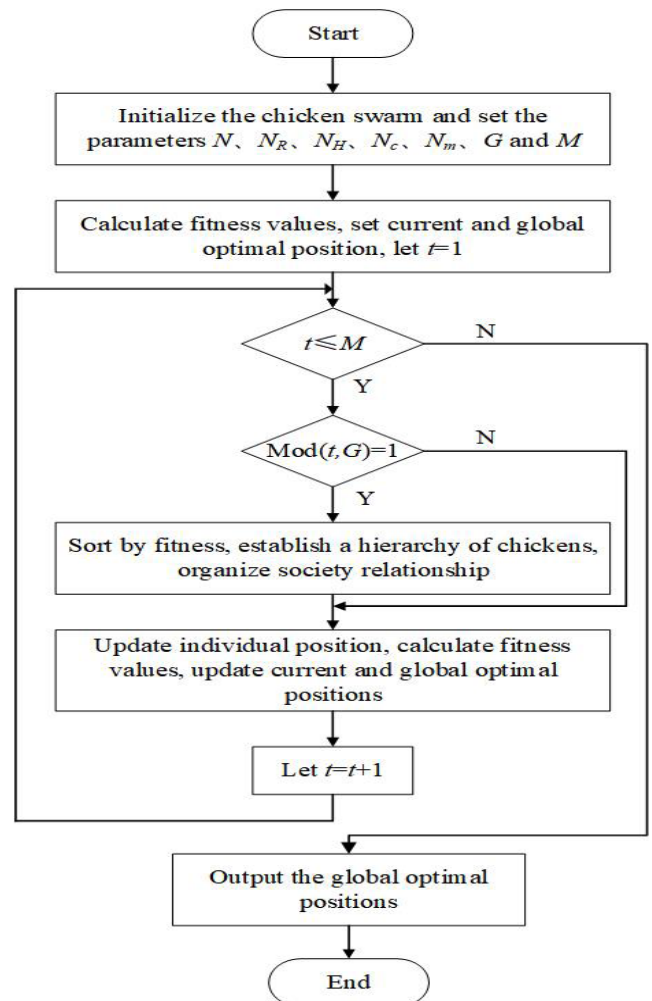


FIGURE 1. flowchart of ICSO algorithm.

TABLE 1. The benchmark problems used in the experiments.

function	ID	formulation	Minimum	Range
Sphere	F1	$f(x) = \sum_{i=1}^n x_i^2$	0	[-100,100]
Schwefel 1.2	F2	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	0	[-100,100]
Schwefel 2.21	F3	$f(x) = \max_{i=1}^n (x_i)$	0	[-100,100]
Step	F4t	$f(x) = \sum_{i=1}^n (x_i + 0.5)^2$	0	[-100,100]
Rastrigin	F5	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0	[-5.12,5.12]
Griewank	F6	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	0	[-600,600]
Bent cigar	F7	$f(x) = x_1^2 + 10^6 \sum_{i=2}^n x_i^2$	0	[-100,100]
Discuss	F8	$f(x) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$	0	[-100,100]

TABLE 2. The related parameter values.

Algorithm	Parameters
PSO	$c1 = c2 = 1.5; \omega = 0.7$
CSO	$G = 10, N_R = 0.2N, N_H = 0.6N, N_M = 0.1N_H, N_C = N - N_R - N_H, FL \in [0.4,1]$
	$G = 10, N_R = 0.2N, N_H = 0.6N, N_M = 0.1N_H, N_C = N - N_R - N_H, FL \in [0.4,1]$
ICSO	$c = 10, \lambda = 1.5$

IV. NUMERICAL EXPERIMENT OF ICSO ALGORITHM

A. PARAMETERS SETTING

The following eight typical unconstrained optimization problem are used to test the performance of the proposed ICSO algorithm, by comparing among PSO algorithm, CSO algorithm and ICSO algorithm. The objective functions [14] of these test problems are shown in Table 1. In three algorithms, the maximum number M of iterations is set to be a same value as 600, the size N of the population are all set to be 50 and the other parameter settings for three algorithm are shown in Table 2. The dimensions of all test problems are set to be 30.

B. EXPERIMENTAL RESULTS

To reduce the influence of contingency, for each of the three comparison algorithms, the experiment on each benchmark

TABLE 3. Statistical experimental results on three algorithms.

Problem	Algorithm	Best	Worst	Mean	Std.
F1	PSO	7.3404e-14	4.4807e-12	1.1001e-12	1.6380e-13
	CSO	7.5095e-32	7.5499e-25	5.4944e-26	1.6393e-25
	ICSO	8.0950e-45	1.4032e-40	1.1355e-42	2.6354e-42
F2	PSO	1.1490e+3	1.2614e+6	2.5697e+5	4.6706e+4
	CSO	412.1944	9.0970e+3	3.5042e+3	2.2085e+3
	ICSO	2.6872e-23	3.1442e-18	1.6810e-19	5.8434e-20
F3	PSO	4.1609	25.7548	11.5368	0.3654
	CSO	0.0879	33.9246	9.9903	10.1370
	ICSO	1.9631e-17	3.9461e-16	8.9889e-17	8.8235e-17
F4	PSO	8.9094e-10	3.6060e+5	1.2020e+4	2.1945e+3
	CSO	2.2884	4.0716	3.1028	0.5044
	ICSO	1.4159e-5	2.6548e-4	2.1898e-4	3.3596e-4
F5	PSO	64.6721	2.4077e+2	1.0858e+2	17.4140
	CSO	0	26.0365	0.8679	4.7536
	ICSO	0	0	0	0
F6	PSO	7.8048e-14	90.0071	8.3145	0.5466
	CSO	0	0.2019	0.0107	0.0396
	ICSO	0	0	0	0
F7	PSO	1.2534e-7	1.000e+4	3.666e+3	6.6943e+2
	CSO	4.7417e-26	2.0101e-20	1.5876e-21	4.0959e-21
	ICSO	1.9837e-40	5.3030e-35	2.1290e-35	9.6930e-35
F8	PSO	7.5554e-9	4.3781e-5	3.5024e-6	5.2523e-7
	CSO	8.9349e-32	5.6349e-23	1.8891e-24	1.1852e-39
	ICSO	3.6794e-45	5.9591e-40	4.0745e-41	1.1852e-40

problem is repeated 30 times independently, and the average values of the relative results of the experiment are used to compare. The means, standard deviations, the worst and the best of objective function values obtained from the experimental data are shown in Table 3.

Table 3 shows that, compared to the PSO algorithm and CSO algorithm, the ICSO algorithm performs well in the accuracy of the objective functions values and the stability. For problems F1 and F8, the basic chicken swarm optimization (CSO) algorithm is better than PSO algorithm in the accuracy of objective function value, and the ICSO algorithm has found the optimal objective function values with the highest precision. For problem F2 and F3, the accuracy of objective function values obtained by the PSO algorithm and CSO algorithm is very poor, and the resulting objective function values are far from the theoretical values. The objective functions values obtained by ICSO algorithm have significantly higher accuracy than those obtained by the PSO and CSO algorithm. For problem F4, although the best objective function values obtained by the PSO algorithm is 1e-10, its worst solution, mean value and variance are all very poor. The results obtained by the CSO algorithm are also poor and can't meet the accuracy requirements. Compared to the PSO algorithm and CSO algorithm, the ICSO algorithm have a great improvement in the objective function value accuracy and the results are more stable than those of PSO and CSO algorithm. For the complex multimodal problems F5 and F6 with a large number of local minimum values, the ICSO algorithm succeeds in finding their theoretical optimal objective function value 0 in all 30 independent experiments. The PSO algorithm and CSO algorithm perform bad obviously. For problem F7, the accuracy of the objective function value obtained by the ICSO algorithm is significantly better than that of the CSO

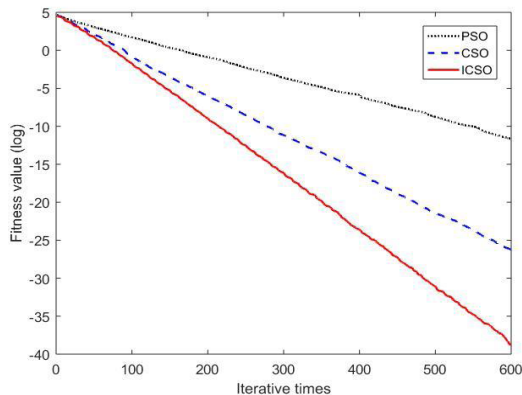


FIGURE 2. Evolution curve of F1.

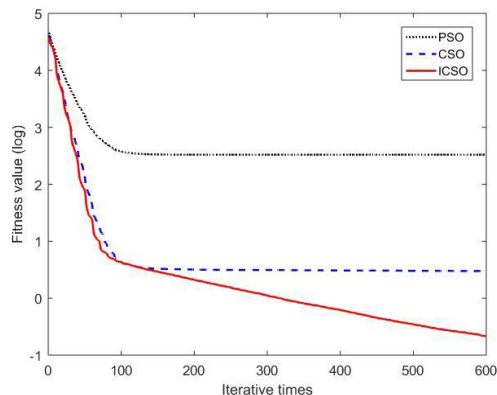


FIGURE 5. Evolution curve of F4.

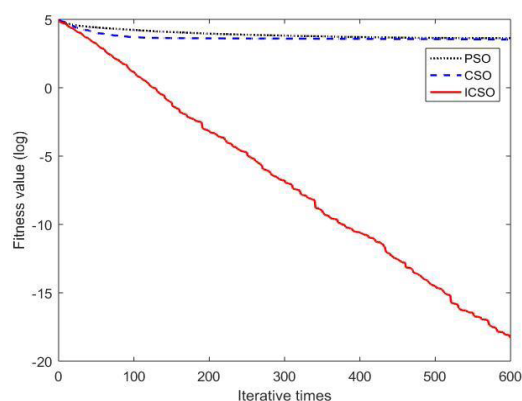


FIGURE 3. Evolution curve of F2.

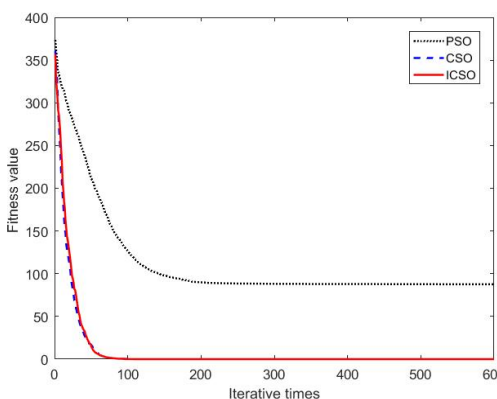


FIGURE 6. Evolution curve of F5.

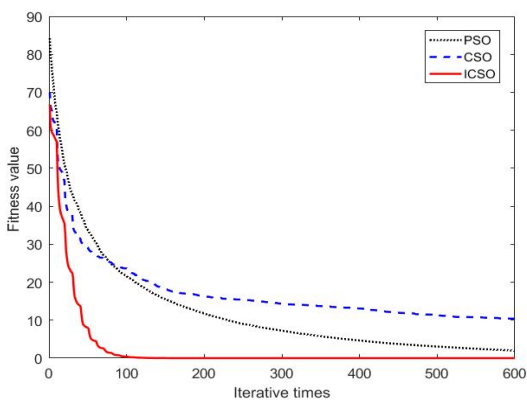


FIGURE 4. Evolution curve of F3.

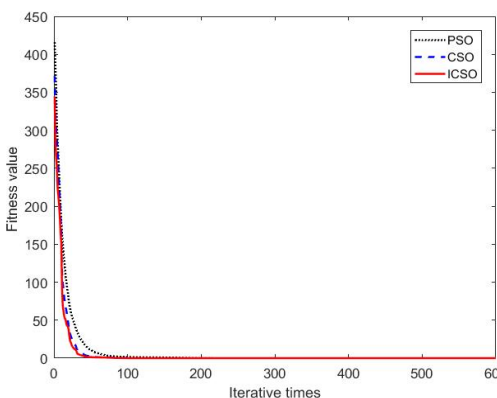


FIGURE 7. Evolution curve of F6.

algorithm, and the PSO algorithm performs bad and unstable, which shows that the ICSO algorithm performs far better than the PSO and CSO algorithm for this multimodal problem. The above experiments show that the ICSO algorithm performs much better than the PSO and CSO algorithms.

In order to further compare the performance of the three algorithms, the evolution curve of thirty-time average optimal objective function values for eight test problems are shown as Fig. 2-9. The vertical coordinates of the figures are taken

as logarithm of the average optimal objective function values except Fig. 4 and Fig. 6-7.

Fig. 2-9 show that the proposed ICSO algorithm is well performed among three algorithms. For the benchmark problem F1-F8, the number of iteration that the ICSO algorithm needed to obtain the optimal solution are much smaller than those in other two algorithms. In Fig. 2, the descending speed and accuracy of objective function value in the ICSO algorithm are significantly better than those in the PSO and CSO algorithms. The ICSO algorithm iterates about 170 times to

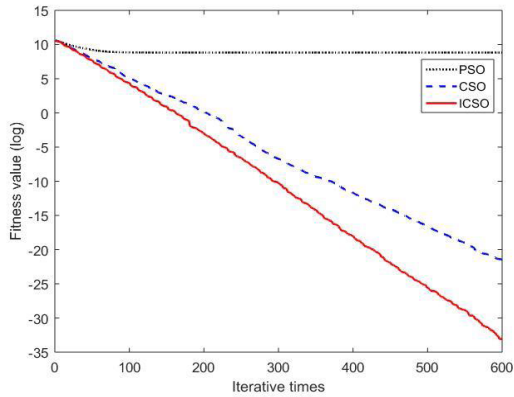


FIGURE 8. Evolution curve of F7.

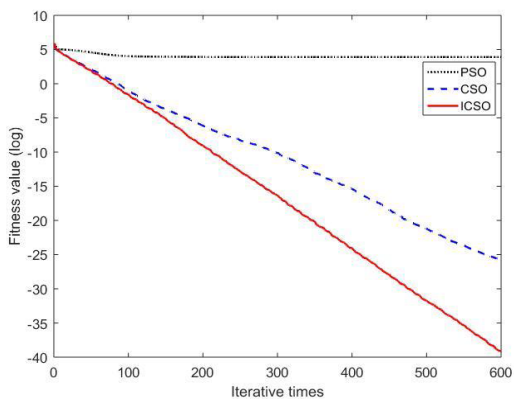


FIGURE 9. Evolution curve of F8.

achieve the accuracy of $1e-6$. The CSO algorithm requires about 200 times iterations, and the PSO algorithm iterates about 400 times to achieve this accuracy. In Fig. 3, the PSO and CSO algorithm have similar performance and their objective function value almost keep unchanged after 400 times iterations. The ICSO algorithm makes the objective function values go or come down in all iterations. In Fig. 4, the descending speed of objective function values in CSO algorithm is faster than that in PSO algorithm in the front 100 iterations and then it reverses. Their objective function values are far from zero. However, the ICSO algorithm iterates about 100 times to make the objective function value close to the theoretical optimal value 0. In Fig. 5, after iterating 100 times, the PSO and CSO algorithm no longer make the objective function values down, which means these two algorithms both fall into local minimum. The ICSO algorithm performs significantly better than any of these two algorithms, and its descending rate of objective function value is very obvious. In Fig. 6, the ICSO algorithm and CSO algorithm make their objective function values drop quickly to the theoretical optimal value 0 in about 100 times iterations. The PSO algorithm performs bad obviously. In Fig. 7, the evaluation curves of the three algorithms are almost the same, but ICSO algorithm is slightly better than CSO and PSO algorithm. For Fig. 8 and Fig. 9, the PSO algorithm can not make

the objective function values descended after about 100 iterations. The ICSO algorithm and CSO algorithm both make the objective function values keep down in all iterations and the descending rate in ICSO algorithm is bigger than that in CSO algorithm.

V. THE APPLICATION OF IMPROVED CHICKEN SWARM ALGORITHM IN ROBOT PATH PLANNING

A. ENVIRONMENTAL MODELING

In recent years, more and more people pay attention to the intelligent optimization algorithm in solving practical application problems. In the field of artificial intelligence, many scholars have made in-depth research on the problem of robot path planning [15]. The chicken swarm optimization algorithm has obvious advantages in fast convergence speed and high convergence accuracy compared to other algorithms, however, its iteration is easy to fall into a local minimum. The proposed improved chicken swarm optimization (ICSO) algorithm is combined with a traditional grid method and a constrained optimization problem is established to search the optimal robot path. The simulation results show that the good global search ability of the ICSO algorithm accelerates the search speed of the robot and improves the quality of the search path.

The working environment of the mobile robot needs to be modeled and preprocessed before the mobile robot plans a path. Grid method is a traditional method used to model the work environment of the mobile robot path planning [16], which is widely used and is employed here. The grid map mainly includes two states: free grid and obstacle grid. MatlabR2016a is hired to do the simulation experiments. The obstacles in the robot working environment are processed and projected into the grid map, as shown in Fig. 10.

In the grid map, assuming that the number of columns is N , the coordinate origin is the grid vertices in the lower left corner, the relationship between grid coordinates and grid numbers may be written as follows:

$$n = y * N + x \tag{11}$$

where x is the horizontal coordinates of the grid under the coordinate system, y is the ordinate of grid in the coordinate system, n is the grid number.

In robot path planning, the most effective and direct way is to take the path length of the robot as the fitness function for each path. The length of the collision-free path of the robot in the course of travel is calculated as follows.

$$f = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{12}$$

where n represents the number of path nodes, x_i and y_i represent the horizontal and vertical coordinates of the path i .

B. EXPERIMENTAL SIMULATION

The simulation platform is built with Matlab and the grid map of working environment of the mobile robot is as shown in Fig. 10. The grid map is a 10×10 grid matrix, where

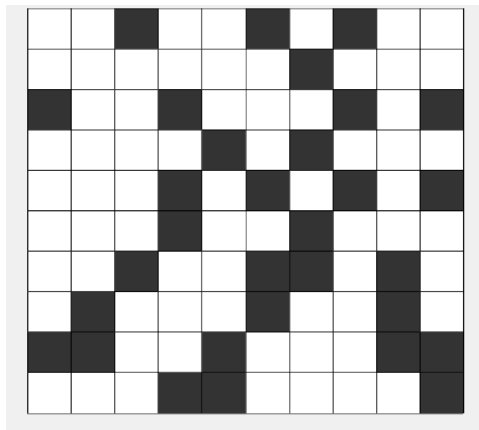


FIGURE 10. Grid map.

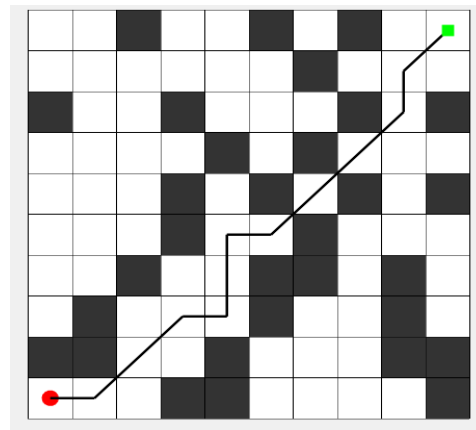


FIGURE 12. Path obtained by ICSO.

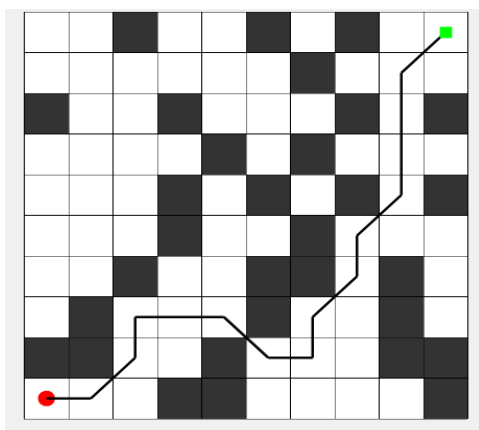


FIGURE 11. Path obtained by CSO.

TABLE 4. Comparison of experimental results.

Algorithm	Path length	Number of iterations
CSO	17.0711	34
ICSO	14.4853	21

the barrier grid is represented in black and the free grid is represented in white. The robot needs to start from the lower left corner of the grid map and end at the top right corner of the map. The parameters in the proposed improved chicken swarm optimization (ICSO) algorithm are initialized as follows: number of chickens $N = 50$, maximum number of iterations $M = 100$, $N_R = 0.2N$, $N_H = 0.6N$, $N_M = 0.1N_H$. The robot path planning are carried out using the basic chicken swarm optimization (CSO) algorithm and the improved chicken swarm optimization (ICSO) algorithm. The results are shown in Fig. 11-12 and Table 4.

It is shown that, from Fig. 11-12 and Table 4, the basic chicken swarm optimization (CSO) algorithm can not find the global optimal path using more iterations and longer search time and has “detour” phenomenon in the process of robot travel. The improved chicken swarm optimization (ICSO) algorithm can jump out of the local optimal both and avoid the

“detour” phenomenon. The length of the robot path obtained by ICSO algorithm is shorter than that by CSO algorithm, and the number of iterations for ICSO to search the robot path is less than that for CSO algorithm. Compared to the CSO algorithm, the ICSO algorithm shortens the path length 15.15% and reduces the number of iterations by 38.24%, which means that the ICSO algorithm can solve the robot path planning problem more effectively.

VI. CONCLUSION

It is well known that the chicken swarm optimization (CSO) algorithm is easy to fall into a local minimum. An improved chicken swarm optimization (ICSO) algorithm based on Levy flight strategy and nonlinear weight reduction strategy is proposed. Compare to CSO algorithm, the ICSO algorithm overcomes the blindness searches and has the significant search efficiency and the high convergence rate. The numerical results on eight benchmark problems show that, compare to CSO algorithm and PSO algorithm, the proposed ICSO algorithm performs well in convergence speed and precision. In solving the robot path planning problem, the ICSO algorithm shortens the path length and needs less iterations than the CSO algorithm does, which means that the ICSO algorithm is feasible in the application of robot path planning. In the future, the CSO algorithm applications in different fields should be studied deeply, especially for high dimensional problems, such as the unmanned aerial vehicle (UAV) route planning, the location issues and so on.

REFERENCES

- [1] J. H. Holland, “Adaptation in natural and artificial systems,” *Quart. Rev. Biol.*, vol. 6, no. 2, pp. 126–137, 1975.
- [2] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Int. Conf. Neural Netw.* Perth, PW, Australia: IEEE Service Centre, Nov./Dec. 1995, pp. 1942–1948.
- [3] X. S. Yang, “A new metaheuristic bat-inspired algorithm,” *Comput. Knowl. Technol.*, vol. 284, pp. 65–74, 2010.
- [4] D. Karaboga, B. Gorkemli, N. Karaboga, and C. Ozturk, “A comprehensive survey: Artificial bee colony (ABC) algorithm and applications,” *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.
- [5] X. Meng, Y. Liu, H. Zhang, and X. Gao, “A new bio-inspired algorithm: Chicken swarm optimization,” in *Advances in Swarm Intelligence*. New York, NY, USA: Springer, 2014, pp. 86–94.

- [6] K. Fei and W. Dinghui, "An improved chicken swarm optimization algorithm," (in Chinese), *J. Jiangnan Univ. (Natural Sci. Ed.)*, vol. 14, no. 6, pp. 681–688, 2015.
- [7] Y. L. Chen, P. L. He, and Y. H. Zhang, "Combining penalty function with modified chicken swarm optimization for constrained optimization," in *Proc. 1st Int. Conf. Inf. Sci., Machinery, Mater. Energy*, 2015, pp. 1899–1907.
- [8] C. Qu, S. Zhao, Y. Fu, and W. He, "Chicken swarm optimization based on elite opposition-based learning," *Math. Problems Eng.*, vol. 2017, no. 11, pp. 1–20, 2017.
- [9] F. Tian, R. Zhang, J. Lewandowski, K.-M. Chao, L. Li, and B. Dong, "Deadlock-free migration for virtual machine consolidation using chicken swarm optimization algorithm," *J. Intell. Fuzzy Syst.*, vol. 32, no. 2, pp. 1389–1400, Jan. 2017.
- [10] S. Chen, R. Yang, R. Yang, L. Yang, X. Yang, C. Xu, B. Xu, H. Zhang, Y. Lu, and W. Liu, "A parameter estimation method for nonlinear systems based on improved boundary chicken swarm optimization," *Discrete Dyn. Nature Soc.*, vol. 2016, pp. 1–11, Dec. 2016.
- [11] M. Yahya and M. P. Saka, "Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights," *Autom. Construct.*, vol. 38, no. 2, pp. 14–29, Mar. 2014.
- [12] A. M. Reynolds, "Cooperative random Lévy flight searches and the flight patterns of honeybees," *Phys. Lett. A*, vol. 354, nos. 5–6, pp. 384–388, Jun. 2006.
- [13] C. Guimin, J. Jianyuan, and H. Qi, "Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm," (in Chinese), *J. Xi'an Jiaotong Univ.*, vol. 40, no. 1, pp. 53–56 and 61, 2006.
- [14] Y. Tan, J. Z. Li, and Z. Y. Zheng. (2014). *Competition on Single Objective Optimization*. ICSI. [Online]. Available: <http://www.Icsi.Org/competition/ICSI.Pdf>
- [15] Z. Daqi and Y. Mingzhong, "Survey on technology of mobile robot path planning," (in Chinese), *Control Decis.*, vol. 25, no. 7, pp. 961–967, 2010.
- [16] L. Juntao, D. Tingting, L. Yuanyuan, and H. Yan, "Study on robot path collision avoidance planning based on the improved ant colony algorithm," in *Proc. 8th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*. Washington, DC, USA: IEEE, Aug. 2016, pp. 540–544.



XIMING LIANG received the M.S. degree in applied mathematics from Yunnan University, Kunming, China, in 1992, and the Ph.D. degree in computational mathematics from Xi'an Jiaotong University, Xi'an, China, in 1998. He is currently a Professor with the School of Science, Beijing University of Civil Engineering and Architecture. His current research interests include constrained optimization, evolutionary computation, large-scale numerical optimization, and their practical applications.



DECHANG KOU was born in Jinan, Shandong, China, in 1994. He is currently pursuing the master's degree in Beijing, China. His research interest includes optimization method and application.



LONG WEN received the M.S. degree in system science from Guangxi Normal University, Guilin, China, in 2008, and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2011. He is currently a Professor with the Key Laboratory of Economics System Simulation of Guizhou, Guizhou University of Finance and Economics. His current research interests include evolutionary computation, nonlinear optimization, constrained single- and multi-optimizations, machine learning, data mining, and their practical applications.

• • •