BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 4, 2019
DOI: 10.24425/bpasts.2019.130177

Improved dolphin swarm optimization algorithm based on information entropy

Y. LI* and X. WANG

Hebei University of Engineering, Handan056038, China

Abstract. In order to overcome the shortcomings of the dolphin algorithm, which is prone to falling into local optimum and premature convergence, an improved dolphin swarm algorithm, based on the standard dolphin algorithm, was proposed. As a measure of uncertainty, information entropy was used to measure the search stage in the dolphin swarm algorithm. Adaptive step size parameters and dynamic balance factors were introduced to correlate the search step size with the number of iterations and fitness, and to perform adaptive adjustment of the algorithm. Simulation experiments show that, comparing with the basic algorithm and other algorithms, the improved dolphin swarm algorithm is feasible and effective.

Key words: dolphin swarm optimization. information entropy, convergence, self-adaptive, combinational optimization.

1. Introduction

Swarm intelligence optimization results from spontaneous selection of the shortest path in the foraging process by biological groups. Swarm intelligence provides a train of thought for solving complex distributed problems without centralized control and at a lack of local information and models [1]. With the characteristics of simplicity, distribution, robustness, scalability and wide applicability, it has gradually become a research hotspot in the field of intelligent optimization. The representative ant colony algorithm [2–3], genetic algorithm [4–5], particle swarm optimization algorithm [6–8], firefly algorithm [9] and artificial bee colony algorithm [10–12] have been widely used in many fields and have been studied by a large number of scholars [13].

The dolphin swarm optimization algorithm (DSA) is a new swarm intelligence optimization algorithm proposed by Professor Wu Tianqi in 2016 [14]. This algorithm simulates the biological characteristics and life habits of dolphin echo location, information exchange, division of work and cooperation, and generates optimization of the dolphin swarm algorithm through four key stages of search, call, reception and predation. Compared with the traditional swarm intelligence optimization algorithm, it only uses the echolocation strategy to get the optimum solution.

As a new algorithm, it also has some defects, such as proneness to falling into local optimum and premature convergence. With the above in mind, Li Zhipeng and Li Weizhong introduced the chaotic search strategy into the dolphin swarm algorithm, and improved the global optimization ability through chaotic initialization, dynamic grouping and precocious mechanism [15]. Other scholars have improved it further [16–17]. Here, aiming at the defect of the algorithm being prone to falling into local optimum and premature convergence, an information entropy strategy was proposed to improve the algorithm. Information entropy is used to represent uncertainty in the algorithm search process, and the selection probability of the path is controlled.

*e-mail: liyancang@hebeu.edu.cn

Manuscript submitted 2018-12-27, revised 2019-02-22 and 2019-03-30, initially accepted for publication 2019-05-12, published in August 2019

2. Standard dolphin swarm algorithm

The mechanism of the dolphin algorithm stems from the simulation of the dolphin population's hunting process. In this algorithm, dolphins accomplish predation by means of four key stages: search stage, call stage, reception stage and predation stage. The corresponding search links are designed by using the behavior of four joint stages, and the optimum solution of the problem is obtained through continuous iteration.

2.1. Initialization. In the optimization problem, each dolphin represents a feasible solution. The dolphin in this study is defined as $Dol_i = [x_1, x_2, \ldots, x_D]^T (i = 1, 2, \ldots, N)$, namely, a feasible D-dimensional solution, where N is the number of dolphins and $x_j (j = 1, 2, \ldots, D)$ is the component of each dimension to be optimized. The individual optimum solution (denoted as L) and neighborhood optimum solution (denoted as K) are two variables associated with the dolphin. For each $Dol_i (i = 1, 2, \ldots, N)$, there are two corresponding variables $L_i (i = 1, 2, \ldots, N)$, and $K_i (i = 1, 2, \ldots, N)$, where L_i represents the optimum solution that Dol_i finds in a single time and K_i stands for the optimum solution of what Dol_i finds by itself or gets from others. *Fitness E* is the basis for judging whether the solution is better. In DSA, E is calculated by a fitness function, and the closer it is to zero, the better it proves. In DSA, three types of distances are used in total. The first is the distance between Dol_i and Dol_j, named $DD_{i,j}$, and $DD_{i,j} = \| Dol_i - Dol_j \|$, $i, j = 1, 2, \ldots, N$, $i \neq j$. The second is the distance between Dol_i and K_i, named DK_i, and $DK_i = \| Dol_i - K_i \|$, $i = 1, 2, \ldots, N$. The third is the distance between L_i and K_i, named DKL_i, and $DKL_i = \| L_i - K_i \|$, $i = 1, 2, \ldots, N$.

2.2. Pivotal stages.

2.2.1. Search stage. During the search stage, each dolphin searches the neighborhood using the sound wave. The sound is defined as $V_i = [v_1, v_2, \ldots, v_D]^T (i = 1, 2, \ldots, M)$, where M is the number of sounds and $V_j = (j = 1, 2, \ldots, D)$ is the component of each dimension, namely the direction attribute of the sound. In addition, sound satisfies the $\| V_i \| = \text{speed}$ $(i = 1, 2, \ldots, M)$ characteristic, where "speed" is a constant representing the speed attribute of sound. The maximum search time is T_1. Within the maximum search time T_1, the sound V_j that $Dol_i = (i = 1, 2, \ldots, N)$ makes at time t will search for a new solution X_{ijt}, which can be formulated as:

$$X_{ijt} = Dol_i + V_j t \tag{1}$$

For the new solution X_{ijt} that Dol_i gets, its fitness E_{ijt} is: $E_{ijt} = Fitness(X_{ijt})$. The individual optimum solution L_i of Dol_i is determined as $L_i = X_{iab}$. If $Fitness(L_i) < Fitness(K_i)$, K_i is replaced by L_i; otherwise, K_i does not change.

2.2.2. Call stage. At the call stage, each dolphin sends out a sound, notifying other dolphins of its search results, including finding a better solution.

2.2.3. Reception stage. Other dolphins will compare the optimum information received with their own optimum solutions, choosing the best solution as the best solution K_i.

In DSA, the exchange process, including the call stage and the reception stage, is implemented by an $N \times N$ order matrix named "transmission time matrix" TS. In TS, $TS_{i,j}$ represents the rest of voice from Dol_j to Dol_i. For K_i, K_j and $TS_{i,j}$, if $Fitness(K_i) <$ $< Fitness(K_j)$ and $TS_{i,j} > \left[\frac{DD_{i,j}}{A \cdot speed}\right]$ then $TS_{i,j} = \frac{DD_{i,j}}{A \cdot speed}$ where A is an acceleration constant that can make sound faster. At this point, $TS_{i,j}$ is re-labeled as maximum contact time T_2. If $Fitness(K_i) > Fitness(K_j)$, K_i is replaced with K_j; otherwise, K_i remains the same.

2.2.4. Predation stage. At the predation stage, each dolphin needs to calculate the surround radius R and determine the distance between the dolphin neighborhood optimum solution and the position after the predation stage based on the known information. It will then get a new location.

3. Algorithm improvement

Because of the uncertainty of dolphin selection in the search stage of the dolphin algorithm, the dolphin algorithm tends to fall into local optimum and premature convergence in the iteration process. Aiming to correct the shortcomings of the algorithm, a strategy of introducing information entropy into the algorithm search stage is proposed.

3.1. Information entropy. Information entropy is a physical concept. It was first proposed by Clausius in thermodynamics to describe the state of the system. Information entropy was proposed by Shannon to introduce thermodynamic entropy into information theory [18–21].

Because the search stage of the dolphin algorithm is uncertain, and entropy itself can be used as a measure of uncertainty, entropy is introduced into the algorithm to control the selection probability of the dolphin swarm search phase by controlling the value of information entropy to realize the goal of adaptive adjustment of the algorithm.

3.2. Algorithm improvement. At the beginning of the algorithm, the search probability of each location is the same, and the information entropy value is the largest. With the increase of the search probability of a location, information entropy decreases gradually.

Define the information entropy value as:

$$H(Fit) = -\Sigma_{i=1}^{N} p_i \ln p_i \tag{2}$$

$$p_i = Fit_i / \Sigma_{i=1}^{N} Fit_i \tag{3}$$

The dolphins' group performed an improved search according to the formula. And then introduced:

$$\alpha = \frac{H_{max} - H}{H_{max}} \tag{4}$$

$$\beta = 1 - \frac{H_{max} - H}{2H_{max}}. \tag{5}$$

Where, α is the proportion of dolphins that can search for prey in a suitable small range, and β is the probability that dolphins can keep their best prey by echolocation. This way, we can make use of change in the entropy value to ensure that α is larger at the early stage of the algorithm, so as to search the solution space as much as possible. For β, the initial stage of operation is larger, which ensures that the best prey can be found through echolocation as far as possible, and at the later stage, it reduces appropriately, increases the role of random operation and avoids premature convergence. H_{max} is the maximum entropy value.

Introducing α and β into (1) and (4), we can get

$$p_i = \alpha \cdot Fit_i / \Sigma_{i=1}^{N} Fit_i \tag{6}$$

$$X_{ijt} = Dol_i + \frac{V_j t}{\beta} \tag{7}$$

In order to balance convergence speed and precision of the algorithm in the optimization process, the step size adaptive parameter γ and dynamic balance factor δ are introduced.

The definition is as follows:

$$\gamma_s = \gamma_{min} + \delta\left(1 - \frac{s}{m}\right) + (1-\delta)\frac{fit_{min}}{fit_{max}} \tag{8}$$

The formula for finding the new solution of the improved algorithm is as follows:

$$X_{ijt} = Dol_i + \frac{\gamma_s}{\beta} V_j t. \tag{9}$$

Among them, γ is a step size adaptive parameter and δ is a dynamic balance factor, with value range of [0,1]. s is the current number of iterations, m is the maximum number of iterations. fit_{min} is the current optimum fitness, and fit_{max} is the initial fitness. The step size adaptive parameter γ can be achieved by correlating with the ratio of iteration times and ratio of fitness. The dynamic balance factor δ is introduced to balance convergence speed and accuracy. As the algorithm approaches top quality, fitness fit_{min} decreases and step size decreases. The dynamic balance factor is introduced to balance $\frac{\text{s}}{\text{m}}$ and $\frac{\text{Fit}_{\text{min}}}{\text{Fit}_{\text{max}}}$, to achieve the goal of balancing convergence speed and accuracy.

The basic flow of the improved dolphin algorithm is as follows:

Step 1: Initialization. Determine the number of dolphins and D-dimensional space dolphin individuals.
Step 2: Search stage. Dolphins randomly emit sound waves in M directions. Entropy and the values of α, β and γ_s of each dolphin are calculated during that period.
Step 3: Call stage. The dolphins use the call to spread individual optimum solutions.
Step 4: Reception stage. Other dolphins compare the optimum solution information received with the individual optimum solution they are looking for, and choose the better of the two as the neighborhood optimum solution K.
Step 5: Predation stage. Following completion of the reception stage, dolphins prey on their own location according to the neighborhood optimum solution K.

According to the size of R_1 value and the position relationship among X_i, L_i and K_i, the specific process is discussed in three cases. The following three cases are illustrated by the example of $Dol_i (i = 1, 2, \cdots, N)$:

1) If $DK_i \ll \text{R}_1$, the neighborhood optimum solution K_i of Dol_i is within the scope of the search, as shown in Fig. 1. For simplicity, L_i is considered as K_i.

$$R_2 = \left(1 - \frac{2}{e}\right) DK_i, \qquad e > 2 \tag{10}$$

Among them, e is a constant, called the "radius reduction factor", greater than 2, usually set at 3 or 4. It can be seen that R_2 gradually converges to zero.

$$\text{newDol}_\text{i} = \text{K}_\text{i} + \frac{\text{Dol}_\text{i} - \text{K}_\text{i}}{\text{DK}_\text{i}} \text{R}_2 \tag{11}$$

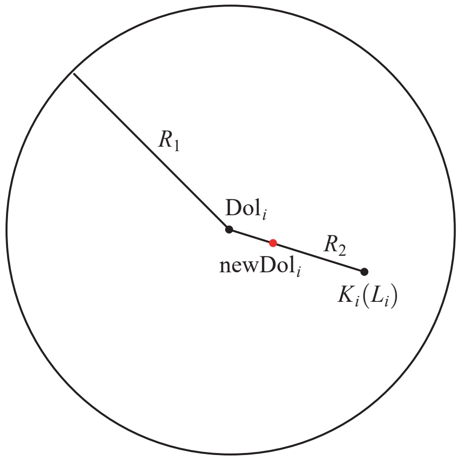

Fig. 1. Results of Dol_i movement in case (1)

That is, Dol_i moves in the K_i direction and stops at a position where the distance from the K_i direction is R_2.

2) If $\text{DK}_\text{i} > \text{R}_1$, and $\text{DK}_\text{i} \gg \text{DKL}_\text{i}$, the neighborhood optimum solution K_i is outside the search range, and L_i is closer to K_i than Dol_i, as shown in Fig. 2.

$$\text{R}_2 = \left(1 - \frac{\frac{\text{DK i}}{\text{Fitness}(\text{K}i)} + \frac{\text{DK i} - \text{DKL i}}{\text{Fitness}(\text{L}i)}}{\text{e} \cdot \text{DK i} \frac{1}{\text{Fitness}(\text{K}i)}}\right), \quad e > 2 \tag{12}$$

$$\text{newDol}_\text{i} = \text{K}_\text{i} + \frac{\text{Random}}{\|\text{Random}\|} \text{R}_2 \tag{13}$$

That is, Dol_i moves to a random position where the distance from K_i is R_2.

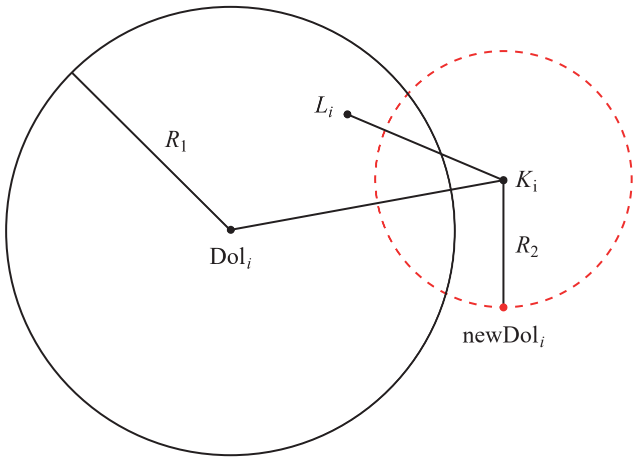

Fig. 2. Results of Dol_i movement in case (2)

3) If $\text{DK}_\text{i} > \text{R}_1$, and $\text{DK}_\text{i} \gg \text{DKL}_\text{i}$, the neighborhood optimum solution K_i is outside the search range, and Dol_i is closer to K_i than L_i, as shown in Fig. 3.

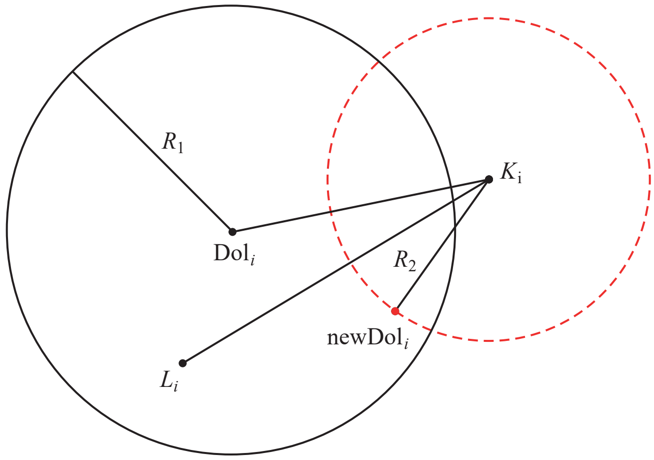

Fig. 3. Results of Dol_i movement in case (3)

$$R_2 = \left(1 - \frac{\frac{\text{DKi}}{\text{Fitness}(Ki)} - \frac{\text{DKLi} - \text{DKi}}{\text{Fitness}(Li)}}{\text{e} \cdot \text{DK i} \frac{1}{\text{Fitness}(Ki)}}\right)\text{DKi}, \quad e > 2 \quad (14)$$

$$\text{newDol}_i = K_i + \frac{\text{Random}}{\|\text{Random}\|} R_2 \quad (15)$$

Dol_i moves to a random position where the distance from K_i is R_2.

After Dol_i moves to the location of newDol_i, compare the fitness of newDol_i and K_i, and if:

$$\text{Fitness}(\text{newDol}_i) < \text{Fitness}(K_i) \quad (16)$$

then replace K_i with newDol_i, $K_i = \text{newDol}_i$. Otherwise, K_i does not change.

The dolphins keep entering new rounds of searches until the termination condition is met.

Figure 4 shows the flow chart of the improved algorithm.

4. Simulation experiment

To verify the effectiveness of the improved algorithm, three standard test functions were chosen to test the performance of the improved dolphin swarm algorithm. In addition, the TSP problem was selected for simulation experiments and compared with the dolphin swarm algorithm and the ant colony algorithm.

4.1. Function test. Test Rosenbrock function, the sphere function and Rastrigin function.

Rosenbrock function is the first to be tested:

$$f_1(x) = 100(x_2 - x_1)^2 + (x_1 - 1)^2 \qquad -5.12 \ll x_i \ll 5.12. \quad (17)$$

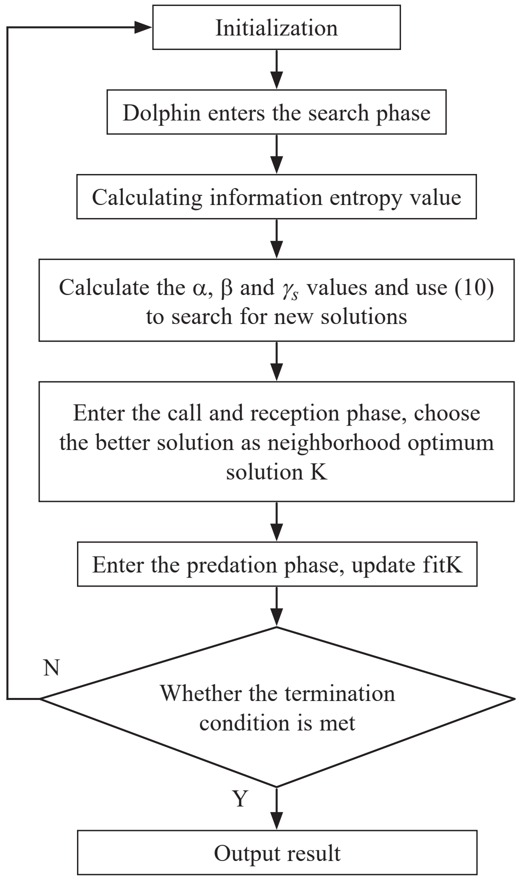

Fig. 4. Flow chart of improved dolphin swarm algorithm

The sphere function is tested second:

$$F_2(x) = \sum_{i=1}^{n} x_i^2 \qquad -2.048 \ll x_i \ll 2.048. \quad (18)$$

Rastrigin function comes third:

$$f_3(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \qquad -5.12 \ll x_i \ll 5.12. \quad (19)$$

It is followed by testing Schwefel function:

$$f_4(x) = \Sigma_{i=1}^{n}\left(\Sigma_{j=1}^{i} x_j\right)^2 \qquad -100 \ll x_i \ll 100. \quad (20)$$

And, finally, Griewank function:

$$f_5 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(x_i/\sqrt{i}\right) + 1 \qquad -600 \ll x_i \ll 600. \quad (21)$$

Basic control parameters of the dolphin swarm algorithm are as follows: $T_1 = 3$, $T_2 = 100$, speed $= 1$, $M = 3$, $e = 4$ and $A = 5$. Following random initialization, when the information entropy value $H \ll 0.0001$, the algorithm ends. After 30 runs, the average is taken. The improved dolphin swarm optimization algorithm (IDSA), DSA, PSO, and the improved artificial bee colony algorithm (IABC), which is an algorithm with good convergence speed and accuracy [23], are compared with each other under 3 conditions and 6 benchmark functions: 1) 10 dimensions, 10 individuals and 1000 calls of benchmark functions; 2) 30 dimensions, 10 individuals and 1000 calls of benchmark functions; 3) 30 dimensions, 10 individuals and 2000 calls of benchmark functions.

The test results are shown in Table 1, Table 2 and Table 3.

From Table 1 to Table 3, it can be seen that in most cases, the average and standard deviation of IDSA calculation results under each benchmark function are smaller than those of other algorithms. It shows that IDSA is superior to the other three algorithms in convergence accuracy in most cases, and its stability is better than those of the other three algorithms, which indicates strong robustness.

Among them, in the iterative process, when the dimension is 30 and the number of iterations is 1000, except for the DSA algorithm which can successfully converge several times, most of the other algorithms cannot achieve the established convergence precision, but the final convergence result of IDSA is better than that of other algorithms. This is because IDSA uses information entropy and adaptive iterative parameters to control iteration altogether, avoiding the algorithm being premature in the optimization process, and avoiding random fluctuations caused by step size uncertainty and slow search, thus improving convergence speed.

4.2. TSP problems. This paper calculates the TSP Chn31 problem, bayes29 problem, d1291 problem, d2103w problem, fl3795 problem and fnl14461 problem [24], and compares it

Table 1
Results of 10 dimensions, 10 individuals and 1000 calls of benchmark functions

Function	IDSA		DSA		IABC		PSO	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Sphere	3.08e-16	4.12e-15	6.87e-14	8.05e-13	6.08e-14	3.66e-13	5.77e-12	7.07e-10
Rosenbrock	4.70e-03	5.22e-03	8.41e-03	9.33e-03	8.23e-03	4.47e-02	4.70e-02	7.11e-02
Rastrigin	7.02e-12	9.35e-12	4.98e-10	6.58e-10	4.98e-10	3.09e-9	6.93e-7	5.50e-7
Schwefel	5.28e-11	5.01e-11	7.33e-10	8.11e-10	7.06e-9	3.08e-8	7.08e-9	5.08e-9
Griewank	6.22e-13	3.28e-12	6.13e-10	6.68e-10	4.08e-9	8.27e-9	9.11e-10	7.57e-9

Table 2
Results of 30 dimensions, 10 individuals and 1000 calls of benchmark functions

Function	IDSA		DSA		IABC		PSO	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Sphere	3.69e-19	5.36e-19	7.21e-15	7.85e-15	6.67e-14	3.97e-14	5.31e-13	7.15e-12
Rosenbrock	4.16e-03	5.01e-03	7.70e-02	8.37e-02	8.65e-03	4.10e-02	4.01e-02	7.20e-02
Rastrigin	6.91e-13	9.10e-12	4.07e-12	6.23e-11	4.18e-10	4.02e-9	3.07e-7	5.41e-7
Schwefel	6.78e-12	6.61e-12	7.17e-10	7.97e-10	6.76e-9	3.01e-8	6.68e-9	5.36e-9
Griewank	7.01e-14	3.97e-14	7.21e-11	6.01e-10	3.82e-9	7.08e-9	9.06e-10	7.22e-9

Table 3
Results of 30 dimensions, 10 individuals and 2000 calls of benchmark functions

Function	IDSA		DSA		IABC		PSO	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Sphere	4.11e-18	5.07e-18	6.61e-15	7.03e-14	6.13e-13	3.02e-13	5.08e-12	7.21e-12
Rosenbrock	3.70e-03	4.91e-03	6.77e-02	8.16e-02	8.23e-03	4.09e-02	3.93e-01	7.05e-01
Rastrigin	6.10e-14	8.83e-13	4.22e-13	6.10e-12	4.09e-11	4.02e-10	3.21e-6	5.77e-6
Schwefel	5.91e-13	6.27e-13	7.09e-11	8.2e-11	6.87e-10	4.261e-9	6.22e-10	5.01e-9
Griewank	3.31e-14	3.22e-14	7.06e-11	5.01e-10	3.52e-9	7.21e-9	8.81e-10	6.24e-9

with the basic dolphin swarm algorithm and ant colony algorithm. Parameter setting is the same as above. This also involves random initialization, population size SN = 100 and maximum iteration number MCN = 1000. Each performance indicator value is the average of 30 runs of the algorithm. The calculation results are shown in Table 4.

Table 4
Results of algorithm performance comparison experiment

Calculation type	Known optimum solution	Improved dolphin swarm algorithm	Basic dolphin swarm algorithm	Ant colony algorithm
Tsp Chn31	15377	15438	15681	15896
bayes 29	2020	2039	2143	2190
d1291	50801	50851	50893	50989
d2103	80450	80511	80597	80783
fl3795	28772	28781	28796	28907
fnl4461	182566	182601	182797	183254

It can be seen from Table 4 that the improved dolphin swarm algorithm is better than other algorithms, and that it has a solution that is very close to the optimum solution.

4.3. Iterative curve comparison. To further test the performance of the improved algorithm, Figs. 5–7 present the iterative curve of the basic dolphin swarm algorithm (DSA), IDSA, PSO, and IABC. Under the four benchmark functions mentioned above, the algorithm takes the same parameter settings as above, i.e. 500 iterations and 30 runs, and then uses the average of the final results.

The figures show that IDSA converges faster, achieves the optimum value faster in less time, and has higher convergence accuracy. It can thus be closer to the optimum solution than the other algorithms.

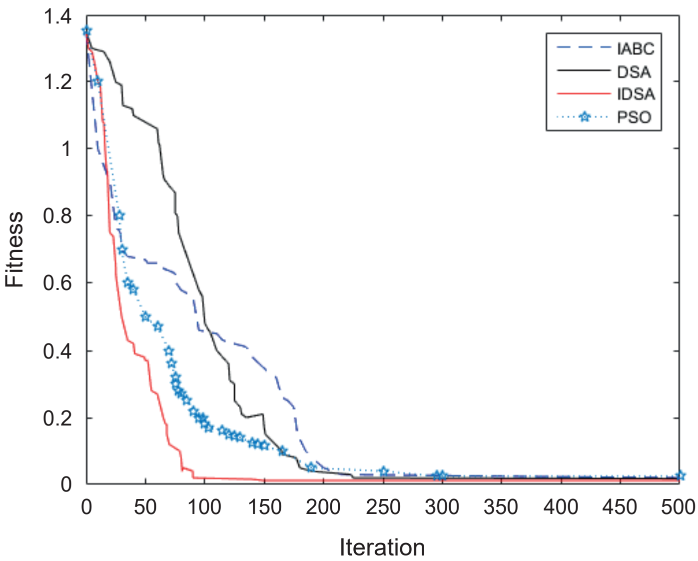

Fig. 5. Iteration curves under sphere function

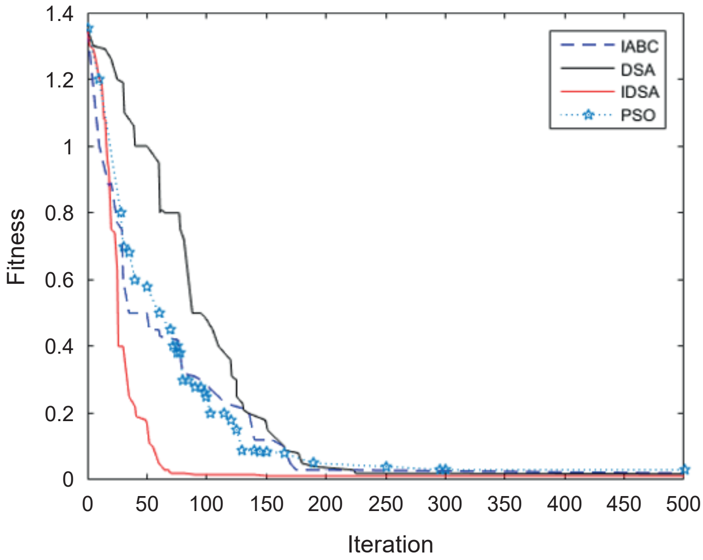

Fig. 6. Iteration curves under Rosenbrock function

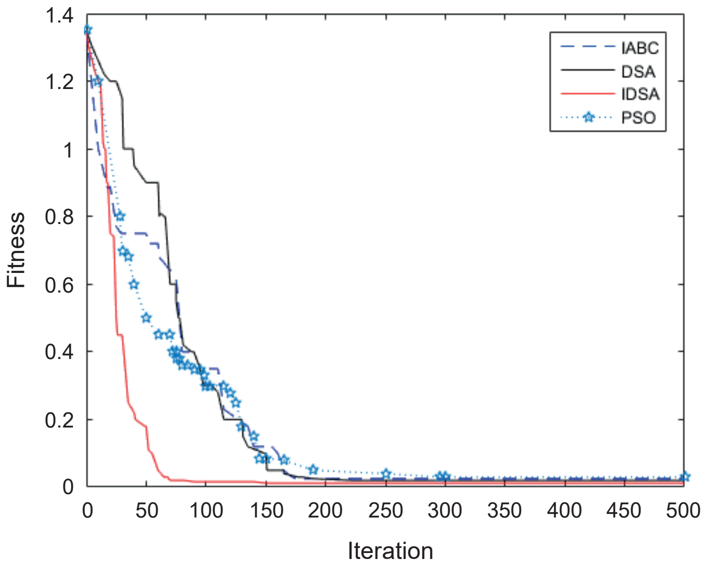

Fig. 7. Iteration curves under Rastrigin function

It can also be seen that the final optimization result of the IDSA is improved, and that convergence speed and accuracy are higher than for the other two algorithms. The improved dolphin swarm algorithm, based on information entropy and adaptive step size parameters and proposed in this paper, thus proves effective.

5. Conclusion

In this paper, an improved dolphin swarm algorithm, based on information entropy and adaptive step size parameters, is proposed for the basic dolphin swarm algorithm, which is prone to falling into local optimum and premature convergence. In the iterative process, the algorithm introduces information entropy and variable α and β to control the dolphin search stage in the search phase. The adaptive step size parameter and dynamic balance factor are introduced to adaptively adjust the search step size to the iteration number and fitness, and balance the search speed and convergence accuracy. The improved algorithm compares the convergence accuracy, search speed and robustness with other algorithms under different conditions. The simulation experiment and algorithm comparison show that the search speed and convergence precision of the algorithm are effectively improved, and the algorithm is prevented from falling into local optimum and premature convergence. The improved dolphin swarm algorithm proposed in this paper is feasible and effective, and it constitutes a better method for solving complex combinatorial optimization problems.

Acknowledgements. The work was supported by the Project of Scientific Research Program of Colleges and Universities in Hebei Province (No. ZD 2019114).

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Systems*, Oxford University Press, 1999.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents", *IEEE Trans Syst Man Cybern B*, 26(1) 29–41 (1996).

[3] O. Deepa and A. Senthilkumar, "Swarm intelligence from natural to artificial systems: ant colony optimization", *International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Network*, 8(1) 9–17 (2016).

[4] C.R. Reid and T. Latty, "Collective behaviour and swarm intelligence in slime moulds", *Fems Microbiology Reviews*, 40(6) 798–806 (2016).

[5] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: algorithms and applications", *Swarm & Evolutionary Computation*, 33, 1–17 (2017).

[6] S. Strasser, R. Goodman, J. Sheppard et al., "A new discrete particle swarm optimization algorithm", Genetic and Evolutionary Computation Conference, Denver, 53–60 (2016).

[7] Y. Rahmat-Samii, D. Gies, and J. Robinson, "Particle swarm optimization (PSO): a novel paradigm for antenna designs", *Ursi Radio Science Bulletin*, 76(3) 14–22 (2017).

[8] G.G. Wang, A.H. Gandomi, A.H. Alavi et al., "A hybrid method based on krill herd and quantum-behaved particle swarm optimization", *Neural Computing & Applications*, 27(4) 989–1006 (2016).

[9] J. Kwiecień and B. Filipowicz, "Firefly algorithm in optimization of queueing systems", *Bull. Pol. Ac.: Tech.*, 60(2) 363–368 (2012).

[10] H. Renaudineau, F. Donatantonio, J. Fontchastagner et al., "A PSO-based global MPPT technique for distributed PV power generation", *IEEE Trans Ind Electron*, 62(2) 1047–1058 (2015).

[11] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", *Information Sciences*, 192(1) 120–142 (2012).

[12] D. Karaboga, B. Gorkemli, C. Ozturk et al., "A comprehensive survey: artificial bee colony (ABC) algorithm and applications", *Artificial Intelligence Review*, 42(1) 21–57 (2014).

[13] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: algorithms and applications", *Swarm & Evolutionary Computation*, 33, 1–17 (2017).

[14] W. Tian-Qi, Y. Min, and Y. Jian-hua, "Dolphin swarm algorithm", *Frontiers of Information Technology & Electronic Engineering* (8) 717–729 (2016).

[15] A. Kaveh and N. Farhoudi, "A new optimization method: dolphin echolocation", *Advances in Engineering Software*, 59(5) 53–70 (2013).

[16] Y. Wang, T. Wang, C.Z. Zhang et al., "A new stochastic optimization approach—dolphin swarm optimization algorithm", *International Journal of Computational Intelligence & Applications*, 15(02) 1650011 (2016,).

[17] Z. Li, W. Li et al., "Air-targets Threat Assessment Using Grey Neural Network Optimized by Chaotic Dolphin Swarm Algorithm", *Control and Decision*, 1–7 (2018)

[18] C.E. Shannon, "Prediction and entropy of printed English", *Bell Labs Technical Journal*, 30(1) 50–64 (1951).

[19] A. Ben-Naim, "Entropy, Shannon's measure of information and Boltzmann's H-theorem", *Entropy*, 19(2) 48 (2017).

[20] L. Nemzer, "Shannon information entropy in the genetic code", *Journal of Theoretical Biology*, 415, 158–170 (2017).

[21] H. Wang and X. Yao, "Objective reduction based on nonlinear correlation information entropy", *Soft Computing*, 20(6) 2393–2407 (2016,).

[22] G. Ye, C. Pan, X. Huang et al., "A chaotic image encryption algorithm based on information entropy", *International Journal of Bifurcation & Chaos*, 28(1) 1850010 (2018).

[23] L. Yan-Cang and P. Yang, "Improved artificial bee colony algorithm based on information entropy", *Control and decision*, 30(6) 1121–1125 (2015).

[24] L.D. Applegate et al., The Traveling Salesman Problem: A Computational Study, Princeton University Press, 2006.