Research Article

Li Songyang*, Yu Haipeng, and Wang Miao

Cat swarm optimization algorithm based on the information interaction of subgroup and the top-N learning strategy

https://doi.org/10.1515/jisys-2022-0018
received October 20, 2021; accepted January 09, 2022

Abstract: Because of the lack of interaction between seeking mode cats and tracking mode cats in cat swarm optimization (CSO), its convergence speed and convergence accuracy are affected. An information interaction strategy is designed between seeking mode cats and tracking mode cats to improve the convergence speed of the CSO. To increase the diversity of each cat, a top-N learning strategy is proposed during the tracking process of tracking mode cats to improve the convergence accuracy of the CSO. On ten standard test functions, the average values, standard deviations, and optimal values of the proposed algorithm with different N values are compared with the original CSO algorithm and the adaptive cat swarm algorithm based on dynamic search (ADSCSO). Experimental results show that the global search ability and the convergence speed of the proposed algorithm are significantly improved on all test functions. The proposed two strategies will improve the convergence accuracy and convergence speed of CSO greatly.

Keywords: cat swarm optimization, information interaction, top N learning, swarm intelligence

1 Introduction

Cat swarm optimization (CSO) is a new swarm intelligence algorithm, which was designed and proposed by Chu et al. to solve continuous and single objective optimization problems based on the observation of cat swarm behavior in 2006 [1]. CSO algorithm adopted multiple iterations to approach target, which had the advantages of easy implementation, global search, and fast convergence speed.

CSO had attracted the attention of many researchers and was successfully applied in various practical optimization problems. Pradhan and Panda extended the CSO for multiobjective optimization problems [2]. Tsai et al. proposed a parallel CSO. It achieved faster convergence and better accuracy under the condition of fewer numbers of cats and iterations [3]. Then Taguchi method was utilized to further improve the convergence speed and global search ability in the process of cat swarm tracking mode [4]. After the CSO was proposed, there were many variants of CSO. Sharafi et al. proposed a binary discrete version based on CSO. It greatly improved the accuracy of 0–1 knapsack problem [5]. Then Siqueira et al. simplified the binary discrete version by replacing the continues operators with Boolean operator to apply CSO with limited memory and less-computation power [6]. Zhao proposed a compact version by replacing the original mutation approach of seeking model with a differential operator [7]. Nie et al. combined CSO with quantum mechanics and a tent map technique to improve accuracy and avoid local optima trapping [8].

* **Corresponding author: Li Songyang,** Henan University of Engineering, No. 1, Xianghe Road, Xinzheng, P. R. China, e-mail: lsy_cqu@cqu.edu.cn
Yu Haipeng: Henan University of Engineering, No. 1, Xianghe Road, Xinzheng, P. R. China, e-mail: 1187874@qq.com
Wang Miao: Henan University of Engineering, No. 1, Xianghe Road, Xinzheng, P. R. China, e-mail: 506872858@qq.com

CSO was also hybridized with other algorithms. Tsai et al. proposed a framework of hybrid optimization algorithm by combining CSO algorithm and artificial bee colony algorithm [9]. Vivek and Reddy proposed a hybrid algorithm combining CSO algorithm, particle swarm optimization (PSO) algorithm, and genetic algorithm to solve the problem of emotion recognition [10]. Nanda proposed a hybrid algorithm combining CSO algorithm and wavelet neural network to predict chaotic and nonlinear time series [11]. Sarswat et al. proposed a hybrid algorithm combining CSO algorithm, genetic algorithm, and simulated annealing algorithm to solve social network problems [12]. These methods mainly involved multiobjective, hybrid optimization. There were other methods to optimize key parameters (mixture rate [MR], seeking range of the selected dimension (SRD), inertia value of velocity [ω], etc.) of CSO [13]. For example, Orouskhani et al. introduced an adjustable inertia of velocity (ω) [14]. An adjustable MR was introduced, which varied linearly according to the number of iterations [15]. Tao et al. adjusted the mixture rate (MR) according to the ratio of the current iteration times to the maximum iteration times to enhance the self adaptability of the algorithm [16]. Chen et al. proposed an adaptive method of MR, inertia value of velocity (ω), and SRD by logistic function (called ADSCSO) [17]. The method had significant effects on the convergence speed and solution accuracy of CSO [18,19].

Although the convergence speed and accuracy of CSO can been improved to a certain extent by optimizing key parameters, the algorithm has a significant property, which is there are tracing mode cats and seeking mode cats while these two mode cats are separated and independent. To improve the learning speed of each cat, an information interaction strategy is designed between seeking mode cats and tracking mode cats to solve the problem that two swarm do not interact. The algorithm have another significant property is that each cat has a fixed learning pattern and has a high chance of falling into local optima. To increase the diversity of each cat and expand the search space of each cat, a top-N learning strategy is proposed during the tracking process of tracking mode cats to improve convergence accuracy of the algorithm.

In summary, an improved cat swarm optimization (IITNCSO) was proposed. First, a subgroup information interaction method was constructed between tracing cats and seeking cats. Then, top-N cats are chosen to update the velocity and position of tracking cats instead of choosing only one best cat. At last, the two proposed strategies were combined into the CSO. Ten standard test functions with different shapes were used to verify that the proposed algorithm significantly improved the performance of the convergence speed and convergence accuracy.

2 Material and methods

2.1 CSO

CSO is an intelligent algorithm that imitates features of natural cat's behavior. The whole cats have two different behavior patterns: tracking mode and seeking mode. Cats of seeking mode spend most of time resting. Cats are also observing their surroundings and are always alert. Cats of tracking mode will move quickly to approach the target after finding the prey.

In CSO, each cat corresponds to one solution of the problem to be optimized. Each cat has three characteristics including fitness, flag, and position. Fitness is used to evaluate how well each cat is positioned. Flag is used to distinguish between tracking and seeking behavior patterns of each cat. For the ith cat, its position P_i is a D-dimensional vector defined in the solution space. p_{ij} represents the jth dimension of P_i. p_{ij} has its own velocity (v_{ij}). Then the D-dimensional vector of v_{ij} constitutes the velocity vector (V_i) of the ith cat (equations (1) and (2)). D defines solution space dimensions.

$$P_i=\{p_{ij}\}, \quad j = 1, 2, \dots ,D, \tag{1}$$

$$V_i = \{v_{ij}\}, \quad j = 1, 2, \dots ,D. \tag{2}$$

The flow of CSO is shown in Figure 1.

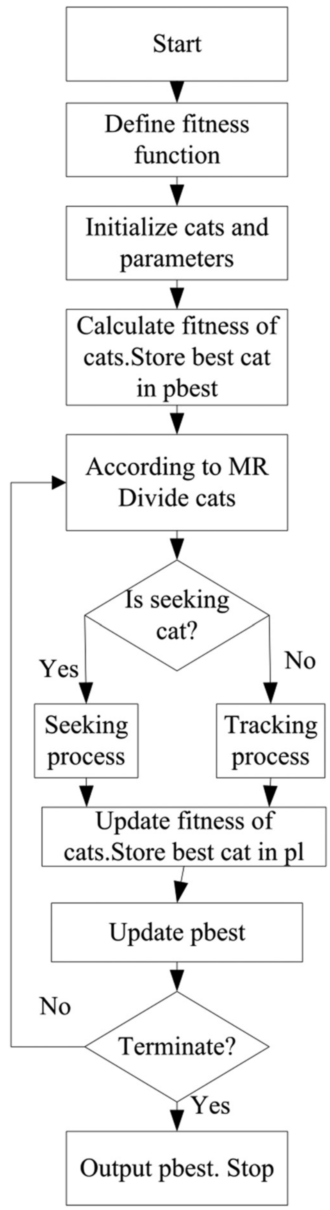


Figure 1: CSO flow.

Step 1: Initialize cats. In the D-dimensional solution space, the position, and speed of each cat are set in a random way to form a certain number of cats.

Step 2: Initialize parameters of CSO such as MR, inertia value of velocity (ω), and SRD.

Step 3: Calculate fitness of cats. The fitness for each cat is calculated, and the best fitness is recorded in *p*best.

Step 4: Divide cats. Cats are randomly divided into tracking and seeking modes based on the mixture rate. Based on the cat's flag, the tracking and seeking processes are performed.

Step 5: Update fitness of cats. After the tracking process and seeking process is completed, the fitness value of each cat is calculate again. The optimal fitness value is recorded in *p*l.

Step 6: Update *p*best. Compare *p*best and *p*l. Choose the better one to update *p*best.

Step 7: If the end condition is met, the algorithm is terminated; otherwise, repeat the steps from step 4 to 7.

2.2 Seeking mode

In the seeking mode, cats rest most of the time. However, cats are also observing their surroundings and are always alert. If the cat wants to move its position, it always moves to the new position carefully and slowly after observing the surrounding. Seeking process of CSO simulates the behavior of cats searching and looking for the next location. There are three key parameters in the seeking mode.

Seeking memory pool (SMP): it defines the seeking memory size of each cat, which is used to store all locations that can be remembered by cat.

SRD: it defines the range that can be changed in each dimension of cat position.

Counts of dimension to change (CDC): it defines the number of cat position dimensions that can mutate. The number cannot be greater than the solution space dimension (D).

The seeking process is as follows:

Step 1: Copy K copies of the ith cat in the seeking mode to the SMP (K = SMP).

Step 2: Among the K copies, keep one copy unchanged. Moreover, $K - 1$ copies undergo mutation. For each cat in $K - 1$ copies, the cat randomly adds or subtracts SRD percents of the present position value based on CDC (equation (3)).

$$p_{ij} = (1 \pm \mathrm{SRD}) \star p_{ij}, \quad p_{ij} \in P_i, \tag{3}$$

Step 3: Update and calculate the fitness of each copy in SMP.

Step 4: Choose the best fitness from the K copies, and apply the best copy to replace the position of the ith cat.

2.3 Tracking mode

In the tracking mode, the cat will move quickly to approach the target after finding prey. Each cat will adjust the moving speed to adjust the position so that the cat can continue to move toward the target.

The tracking process is as follows:

Step 1: Update speed of the ith cat in tracking mode (equation (4)). $p\mathrm{best}_j$ is the jth dimension of the best cat in whole cats. ω is the inertia value. rand is a random number with uniform distribution [0, 1], and c is a predetermined constant between 0 and 2.

$$v_{ij} = \omega \times v_{ij} + \mathrm{rand} \times c \times (p\mathrm{best}_j - p_{ij}), \quad j = 1, 2, \ldots, D. \tag{4}$$

Step 2: Check that the updated cat speed does not exceed the speed range.

Step 3: Update the position of the ith cat in the tracking mode (equation (5))

$$p_{ij} = p_{ij} + v_{ij}, \quad j = 1, 2, \ldots, D. \tag{5}$$

2.4 Subgroup information interaction

According to MR, the whole cats are divided into two modes. Therefore, information interaction between different mode cats also affects the effect of CSO. In the original CSO (Figure 1), the information interaction among cats depends on best cat ($p$best) of the whole cats. There is no other interaction between the two behavior modes. To solve the problem that two mode cats do not interact directly and improve the weakness, an information interaction strategy between subgroup is designed.

First, the best finesses of the two mode cats are compared. The cat swarm with better fitness is called excellent cats, and the other swarm is called ordinary cats. Then, one learning cat (P_l) is randomly selected from the excellent cats. All ordinary cats should learn from the learning cat (P_l).

The information interaction method is as following (equation (6)):

$$v_{ij} = \begin{cases} v_{ij} + \gamma \times (p_{ij} - p_{lj}), & f(P_i) > f(P_l), \\ v_{ij} - \gamma \times (p_{ij} - p_{lj}), & f(P_i) \le f(P_l). \end{cases} \tag{6}$$

The symbol $f(P_i)$ represents the fitness of the ith cat at position P_i. p_{lj} represents the jth dimension of position of the learning cat (P_l). p_{ij} represents the jth dimension of position of the ith cat (P_i). According to equation (6), it can be seen that when the fitness of the learning cat (P_l) is less than the fitness of the ith ordinary cat. The ith ordinary cat will move away from the learning cat (P_l). When the fitness of the learning cat (P_l) is greater than the fitness of the ith ordinary cat, the ith ordinary cat will approach to the learning cat (P_l). γ is the learning factor and is random value between [0, 1]. Then, the new position of the ith ordinary cat is calculated using equation (5). When the fitness of new position is better than the fitness of the old position, the ith ordinary cat will move to the new position; otherwise, it will not move.

2.5 Top-N learning strategy

To balance the convergence speed and convergence accuracy, the algorithm should increase the behavior diversity of each cat. So a top-N learning strategy is adopted to increase the diversity of cats and expand the search space of whole cats. The overall idea is that each cat should learn from the top-N cats with better fitness, not just the best one. Hence, the top-N cats with the better fitness are stored in the $p\text{best}_N$ after calculating the fitness of the whole cats. When executing the tracking process, the set $p\text{best}_N$ are used to update the speed and position of each tracking cat. The updated position is randomly selected to replace the original position of the tracking cat.

Tracking process can be adjusted as follows:

Step 1: Update the speed of the ith cat in the tracking mode based on $p\text{best}_N$ (equation (7)).

$$\begin{aligned} &v_{ij}^n = \omega \times v_{ij} + \text{rand} \times c \times (p\text{best}_{nj} - p_{ij}), \\ &p\text{best}_n \in p\text{best}_N, \quad n = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, D. \end{aligned} \tag{7}$$

Step 2: Check that the updated cat speed does not exceed the speed range.

Step 3: Calculate the expected positions (P_i^N) of the ith cat in tracking mode (equation (8))

$$p_{ij}^n = p_{ij} + v_{ij}^n n = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, D. \tag{8}$$

Step 4: Randomly select a position from P_i^N to update the position of the ith cat.

2.6 IITNCSO

The flow of IITNCSO is shown as follows.

Step 1: Initialize cats. In the D-dimensional solution space, the position and speed of each cat are set in a random way to form a certain number of cats.

Step 2: Initialize parameters of CSO such as MR, inertia value of velocity (ω), and SRD, etc.

Step 3: Calculate the fitness of cats. The fitness for each cat is calculated. Top-N cats having better fitness are recorded in $p\text{best}_N$.

Step 4: Divide cats. Cats are randomly divided into tracking and seeking modes based on the mixture rate. Based on the cat's flag, the tracking process with top-N learning strategy is done and the seeking process is completed.

Step 5: Do subgroup information interaction and update the fitness of cats. After the tracking process and seeking process is completed, the fitness of each cat is calculated again. The top-N better fitness cats are recorded in $p\text{l}_N$.

Step 6: Compare $p\text{best}_N$ and $p\text{l}_N$ to update $p\text{best}_N$.

Step 7: If the end condition is met, the algorithm is terminated. Select best $p$best from $p\text{best}_N$. Otherwise, repeat the steps from step 4 to 7.

3 Results

3.1 Algorithm settings

These experiments are done under Windows 10 operating system and Intel(R) Core(TM) i7-10700 CPU 2.90 GHz, 16 GB memory, Matlab R2021a simulation software. To fully compare the convergence accuracy and speed of CSO [1], ADSCSO [17], and IITNCSO, ten standard test functions are used for verification (Table 1). Algorithm parameter settings are determined in Table 2.

In Table 1, F1–F2 functions belong to the plate-shape function, and there is no local minimum except the global minimum. F3–F6 functions belong to many local minima shape. They have many local minima; hence, the algorithms have the risk of falling into local minima. F7 function is bowl shaped. There are D local minima except for the global one. F8 is valley-shaped. The function is unimodal, and the global minimum lies in a narrow, parabolic valley. However, even though this valley is easy to find, convergence to the minimum is difficult. F9 belongs to steep ridges shape. It is unimodal, and the global minimum has a small area relative to the search space. F10 is multimodal, with sharp peaks at the corners of the input domain.

The parameters of the three algorithms are shown in Table 2. The cat population size is set to 100, and the maximum number of iterations is 500. Each algorithm runs 20 times independently. Therefore, 20 values of $p$best for CSO, ADSCSO, and IITNCSO can be collected in three different sets.

$$\text{out}_{a\lg i} = \{p\text{best}^k\},\quad k = 1, 2, \ldots ,20,\quad a\lg i \in \{\text{CSO, ADSCSO, IITNCSO}\}. \tag{9}$$

Then the mean (Ave), standard deviation (Std), and optimal value (Optimal) of fitness of each algorithm are calculated (equations (10)–(12)).

$$\text{Ave} = \frac{1}{20}\sum_{k=1}^{20} f(p\text{best}^k), \tag{10}$$

$$\text{Std} = \frac{1}{20}\sum_{k=1}^{20} (f(p\text{best}^k) - \text{Ave})^2, \tag{11}$$

$$\text{Optimal} = \min\{f(p\text{best}^k)\},\quad k = 1, 2, \ldots ,20. \tag{12}$$

The mean value can reflect the convergence ability of the algorithm, and the standard deviation reflects the stability of the algorithm.

3.2 Result analysis

Comparisons of IITNCSO with CSO and ADSCSO algorithms are shown in Table 3. "Ave," "Std," and "Optimal" are the average value, standard deviation, and minimum of the fitness obtained after 20 independent runs, respectively. Boldface indicates that the value obtained by the proposed algorithm is the best in the corresponding test function. "N = 1," "N = 2," and "N = 3" defines the three different values of N applied in IITNCSO.

As we can see that "IITNCSO N = 1" has significant advantages than "CSO" and "ADSCSO" in solution accuracy on nine test functions except F2. CSO algorithm has more advantages than "IITNCSO N = 1" and "ADSCSO" on the low-dimensional test function F2 (solution space 2 dimensions).

Table 1: Mathematical descriptions of test function

Number	Name	Function formula	Range	Global minimum	Dimensions (D)
F1	Zakharov	$f(x) = \sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^4$	$[-5, 10]$	0	20
F2	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$[-10, 10]$	0	2
F3	Ackley	$f(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$	$[-32.768, 32.768]$	0	20
F4	Schaffe N.2	$f(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	$[-100, 100]$	0	2
F5	Rastrigin	$f(x) = 10d + \sum_{i=1}^{d}(x_i^2 - 10\ \cos(2\pi x_i))$	$[-5.12, 5.12]$	0	20
F6	Griewank	$f(x) = \sum_{i=1}^{d}\frac{x_i^2}{4,000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0	20
F7	Sphere	$f(x) = \sum_{i=1}^{d} x_i^2$	$[-5.12, 5.12]$	0	20
F8	Rosenbrock	$f(x) = \sum_{i=1}^{d-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-5, 10]$	0	20
F9	Easom	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	$[-100, 100]$	−1	2
F10	Beale	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	$[-4.5, 4.5]$	0	2

Table 2: Parameter settings

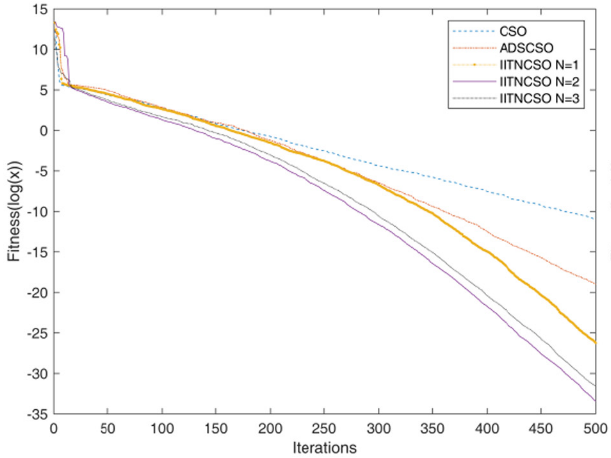
Parameter	CSO	ADSCSO	IITNCSO
SMP	5	5	5
SRD	0.2	[0.1, 0.5]	0.2
MR	0.2	[0.1, 0.4]	[0.1, 0.4]
CDC	0.8	0.8	0.8
Ω	1	[0.4, 0.9]	[0.4, 0.9]
c	2	2	2
V	0.2 * range	0.2 * range	0.2 * range
Cat population size	100	100	100
Iteration number	500	500	500
Algorithm runs number	20	20	20

Table 3: Comparisons of CSO, ADSCSO, and IITNCSO

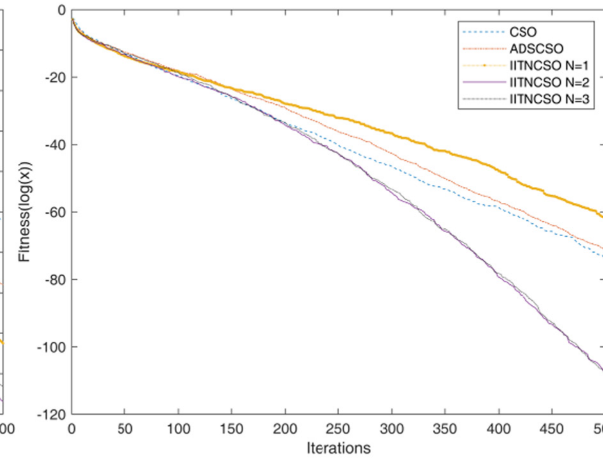
Function	Performance	CSO	ADSCSO	IITNCSO N = 1	IITNCSO N = 2	IITNCSO N = 3
F1	Ave	1.6347×10^{-05}	5.7390×10^{-09}	4.0490×10^{-12}	$\mathbf{3.0258 \times 10^{-15}}$	4.0490×10^{-12}
	Std	2.9639×10^{-05}	6.8147×10^{-09}	5.0979×10^{-12}	$\mathbf{3.7145 \times 10^{-15}}$	1.3542×10^{-14}
	Optimal	4.3212×10^{-07}	6.3890×10^{-10}	2.3621×10^{-13}	$\mathbf{2.0151 \times 10^{-16}}$	2.5161×10^{-15}
F2	Ave	1.3117×10^{-32}	1.3109×10^{-31}	1.2057×10^{-27}	2.0656×10^{-47}	$\mathbf{1.1123 \times 10^{-47}}$
	Std	4.9863×10^{-32}	3.1683×10^{-31}	2.8938×10^{-27}	3.0784×10^{-47}	$\mathbf{1.4133 \times 10^{-47}}$
	Optimal	8.0489×10^{-36}	2.5618×10^{-38}	4.3231×10^{-32}	3.2358×10^{-49}	$\mathbf{5.2822 \times 10^{-50}}$
F3	Ave	0.7262	0.2503	0.1508	$\mathbf{1.2405 \times 10^{-09}}$	2.8592×10^{-09}
	Std	1.3056	0.7831	0.6745	$\mathbf{7.1043 \times 10^{-10}}$	1.6670×10^{-09}
	Optimal	2.2728×10^{-07}	9.0203×10^{-08}	1.1742×10^{-09}	$\mathbf{3.4890 \times 10^{-10}}$	1.0721×10^{-09}
F4	Ave	0.0010	3.1086×10^{-16}	4.6629×10^{-16}	**0**	**0**
	Std	0.0020	1.1161×10^{-16}	1.1502×10^{-15}	**0**	**0**
	Optimal	2.2204×10^{-16}	$.2204 \times 10^{-16}$	0	**0**	**0**
F5	Ave	43.7423	39.3899	40.1614	21.6424	**21.6357**
	Std	14.4695	16.3469	15.2998	10.3648	**5.9307**
	Optimal	21.7625	14.3640	12.6009	**6.9736**	11.9637
F6	Ave	0.0118	0.0193	**0.0092**	0.0180	0.0199
	Std	0.01077	0.02736	**0.0135**	0.0190	0.0302
	Optimal	8.1112×10^{-13}	2.5235×10^{-13}	**0**	0	0
F7	Ave	1.6497×10^{-14}	6.0596×10^{-16}	1.0354×10^{-18}	$\mathbf{3.1708 \times 10^{-20}}$	1.9050×10^{-19}
	Std	2.0819×10^{-14}	4.3703×10^{-16}	1.32550×10^{-18}	$\mathbf{6.6799 \times 10^{-20}}$	2.2076×10^{-19}
	Optimal	1.7569×10^{-15}	5.3653×10^{-17}	4.9893×10^{-20}	$\mathbf{9.3708 \times 10^{-22}}$	2.5646×10^{-20}
F8	Ave	25.1348	27.7760	17.4183	**16.5550**	16.5924
	Std	31.0213	32.8583	15.5911	**15.5906**	13.6519
	Optimal	16.6310	**11.5716**	12.4806	11.9345	12.5009
F9	Ave	−0.9997	−0.9997	−0.9998	**−1**	**−1**
	Std	3.9864×10^{-04}	4.9837×10^{-04}	2.0160×10^{-04}	**0**	**0**
	Optimal	−1.0000	−1.0000	−1.0000	**−1.0000**	**−1.0000**
F10	Ave	4.2227×10^{-05}	1.0128×10^{-05}	3.6111×10^{-06}	**0**	**0**
	Std	9.2025×10^{-05}	1.2231×10^{-05}	5.5776×10^{-06}	**0**	**0**
	Optimal	5.7864×10^{-08}	4.4169×10^{-08}	7.8533×10^{-09}	**0**	**0**

Because the value of N is 1, it means that only subgroup information interaction strategy works, and the top-N learning strategy does not work. The results show that the subgroup information interaction strategy will improve the accuracy of the original CSO on most test functions.

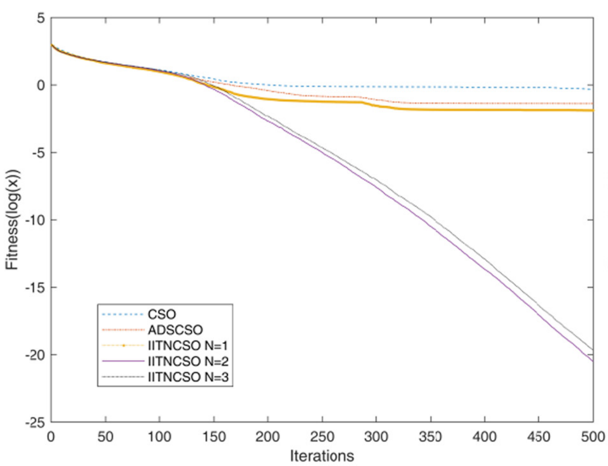
We also can see that "IITNCSO N = 2" and "IITNCSO N = 3" methods achieve the best average value (Ave), standard deviation (Std), and minimum (Optimal) of the fitness on nine test functions except F6. "N = 2" and "N = 3" define that the value of N is greater than 1 ("N = 1"). Therefore, the top-N learning


CSO
ADSCSO
IITNCSO N=1
IITNCSO N=2
IITNCSO N=3
Fitness(log(x))
Iterations

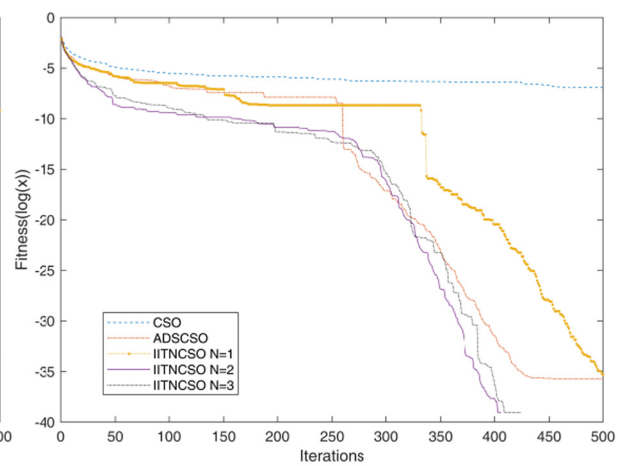
(1)


CSO
ADSCSO
IITNCSO N=1
IITNCSO N=2
IITNCSO N=3
Fitness(log(x))
Iterations

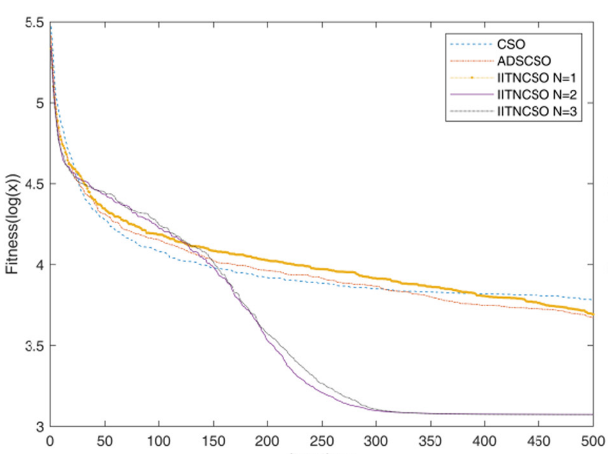
(2)


CSO
ADSCSO
IITNCSO N=1
IITNCSO N=2
IITNCSO N=3
Fitness(log(x))
Iterations

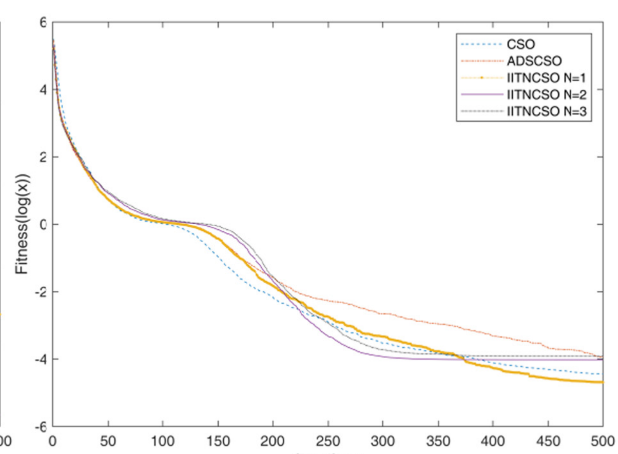
(3)


CSO
ADSCSO
IITNCSO N=1
IITNCSO N=2
IITNCSO N=3
Fitness(log(x))
Iterations

(4)


CSO
ADSCSO
IITNCSO N=1
IITNCSO N=2
IITNCSO N=3
Fitness(log(x))
Iterations

(5)


CSO
ADSCSO
IITNCSO N=1
IITNCSO N=2
IITNCSO N=3
Fitness(log(x))
Iterations

(6)

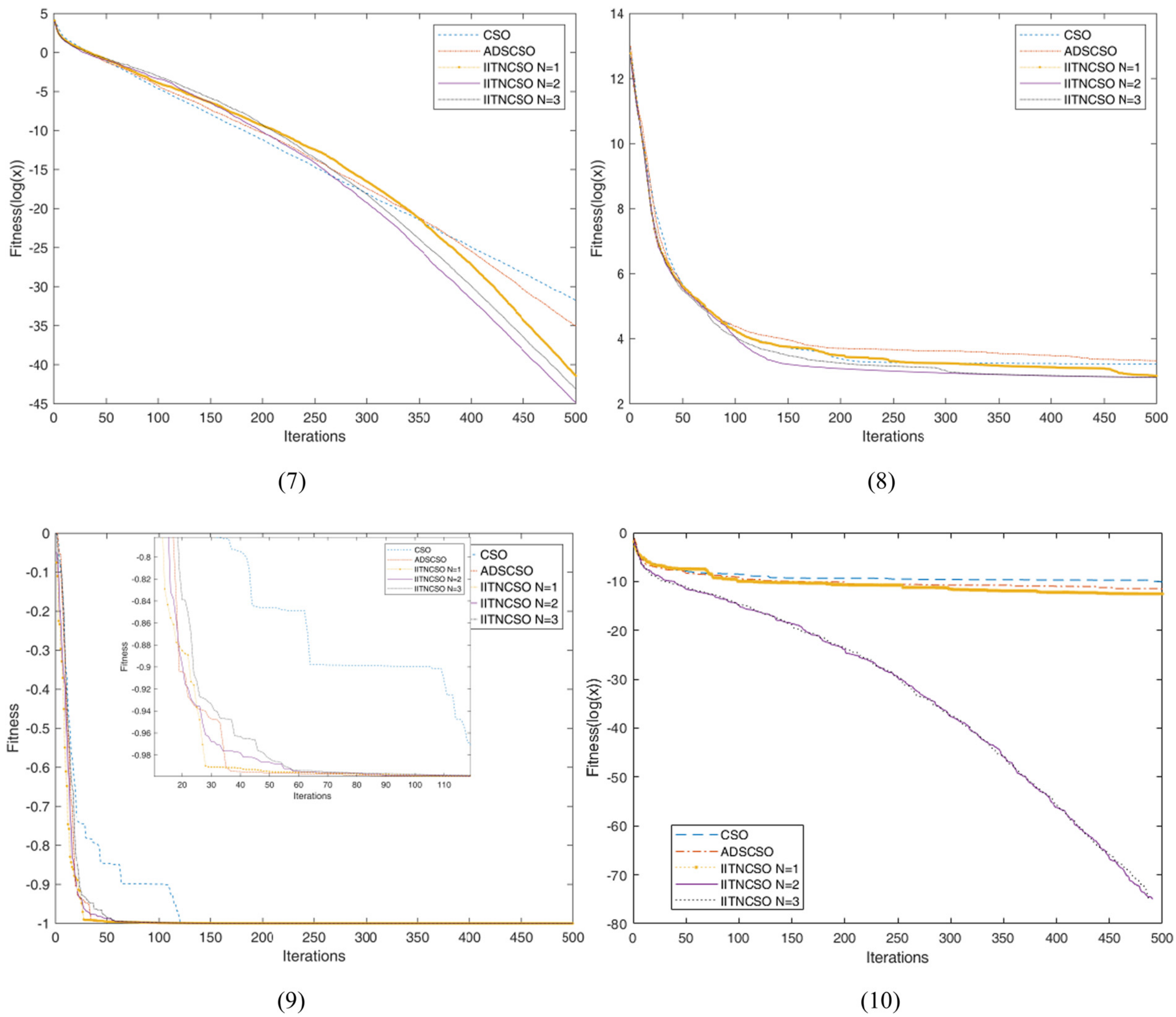


Figure 2: Comparisons of fitness change of F1–F10: (1) F1, (2) F2, (3) F3, (4) F4, (5) F5, (6) F6, (7) F7, (8) F8, (9) F9, and (10) F10.

strategy will work. The results show that the top-N mechanism to increase the diversity of cats plays an important role to expand the search space and improve search accuracy of CSO.

Changes of fitness of IITNCSO, CSO, and ADSCSO algorithms under ten different function tests are shown in Figure 2. To clearly show the comparisons of the algorithms, the fitness of vertical axis is done using the logarithmic transformation on other test functions except F9.

For the test functions F1, F3, F6–F10, the convergence speed and convergence accuracy of "IITNCSO $N = 1$" are faster than "CSO" and "ADSCSO." For the test functions F2, F4, and F5, although the convergence speed of "IITNCSO $N = 1$" is slower than that of "CSO" and "ADSCSO," it still has advantages in the optimal value search for F4 and F5.

In Figure 2, "IITNCSO $N = 2$" and "IITNCSO $N = 3$" methods achieve the fastest convergence speed of the fitness on nine test functions except F6. The results show that the subgroup information interaction strategy and the top-N learning strategy can greatly improve the convergence speed and convergence accuracy.

Although parameter N ($N > 1$) has an effect on the convergence speed and accuracy of IITNCSO, the effect is small.

4 Discussions and conclusions

On the ten standard test functions (F1–F10), the proposed methods with different top-N values are compared with the traditional CSO and adaptive CSO based on dynamic search (ADSCSO).

The convergence speed, mean value, standard deviation, and optimal fitness of "IITNCSO $N = 1$" have advantages than CSO and ADSCSO on the test functions except F2 function. F1 ($D = 2$) and F2 ($D = 20$) functions belong to the plate-shape function. However, the "IITNCSO $N = 1$" perform poorly in low-dimensional cases. The results show that the information interaction of subgroup plays a very important role on CSO. The strategy will improve the convergence accuracy of CSO greatly.

The convergence speed, mean value, standard deviation, and optimal fitness of "IITNCSO $N = 2$" and "IITNCSO $N = 3$" have advantages than CSO and ADSCSO on the test functions except F6 function. F6 is Griewank function, which has many widespread local minima, which are regularly distributed. From Figure 2(6), we can see that converge speeds of "IITNCSO $N = 2$" and "IITNCSO $N = 3$" are faster than other algorithms in the first 350 iterations. Then the convergence speed curves of "IITNCSO $N = 2$" and "IITNCSO $N = 3$" tend to be smooth. Moreover, the two algorithm fall into local optimization.

Results also show that the convergence speed, mean value, standard deviation, and optimal fitness of "IITNCSO $N = 2$" and "IITNCSO $N = 3$" have advantages than "IITNCSO $N = 1$." The results show that the top-N learning strategy plays a very important role on CSO. The strategy will improve the convergence accuracy and convergence speed of CSO greatly.

Comparing convergence speed, mean value, standard deviation, and optimal fitness between "IITNCSO $N = 2$" and "IITNCSO $N = 3$", it shows that the value of N chosen has an impact on the results, but the effects are slight.

Above all, an improved CSO with the information interaction strategy and the top-N learning strategy is proposed to solve the problem that two mode cats do not interact directly and each cat had a fixed learning pattern. On ten standard test functions, the average values, standard deviations, and optimal values of the proposed algorithm with different N values compared with CSO and ADSCSO are greatly improved. Further, proper nonparametric statistical analysis should been done to support the results presented. Moreover, comparisons will be done with other swarm intelligence algorithms including particle swarm optimization (PSO such as Chaotic HCLPSO, ensemble PSO, and others).

Funding information: This study was supported by the National Natural Science Foundation of China (61501174, U1704124), the Key scientific research projects of Henan Province (21A520005, 20A520010), the Scientific and Cultivation Fund of Henan University of Engineering (PYXM202020).

Conflict of interest: Authors state no conflict of interest.

References

[1] Chu S-C, Tsai PW, Pan JS. Cat swarm optimization. Paper Presented at the PRICAI 2006: Trends in Artificial Intelligence. Berlin, Heidelberg: 2006.
[2] Pradhan PM, Panda G. Solving multiobjective problems using cat swarm optimization. Expert Syst Appl. 2012;39(3):2956–64. doi: 10.1016/j.eswa.2011.08.157.
[3] Tsai P-W, Pan J-S, Chen S-M, Liao B-Y, Szu-Ping H. Parallel cat swarm optimization. Paper Presented at the 2008 International Conference on Machine Learning and Cybernetics; 2008, July 12–15.
[4] Tsai P-W, Pan J-S, Chen S-M, Liao B-Y. Enhanced parallel cat swarm optimization based on the Taguchi method. Expert Syst Appl. 2012;39(7):6309–19. doi: 10.1016/j.eswa.2011.11.117.
[5] Sharafi Y, Khanesar MA, Teshnehlab M. Discrete binary cat swarm optimization algorithm. Paper Presented at the 2013 3rd IEEE International Conference on Computer, Control and Communication (IC4); 2013, Sept 25–26.
[6] Siqueira H, Figueiredo E, Macedo M, Santana CJ, Bastos-Filho CJA, Gokhale AA. Boolean binary cat swarm optimization algorithm. Paper Presented at the 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI); 2018, Nov 7–9.

[7] Zhao M. A novel compact cat swarm optimization based on differential method. Enterp Inf Syst. 2020;14(2):196–220. doi: 10.1080/17517575.2018.1462405.
[8] Nie X, Wang W, Nie H. Chaos quantum-behaved cat swarm optimization algorithm and its application in the PV MPPT. Comput Intell Neurosci. 2017;2017:1583847. doi: 10.1155/2017/1583847.
[9] Tsai P-W, Pan J-S, Shi P, Liao B-Y. A new framework for optimization based-on hybrid swarm intelligence. In: Panigrahi BK, Shi Y, Lim M-H, editors. Handbook of swarm intelligence: concepts, principles and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 421–449.
[10] Vivek TV, Reddy GRM. A hybrid bioinspired algorithm for facial emotion recognition using CSO-GA-PSO-SVM. Paper Presented at the 2015 Fifth International Conference on Communication Systems and Network Technologies; 2015, April 4–6.
[11] Nanda SJ. A WNN-CSO model for accurate forecasting of chaotic and nonlinear time series. Paper Presented at the 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES); 2015, Feb 19–21.
[12] Sarswat A, Jami V, Guddeti RMR. A novel two-step approach for overlapping community detection in social networks. Soc Netw Anal Min. 2017;7(1):47. doi: 10.1007/s13278-017-0469-7.
[13] Ahmed AM, Rashid TA, Saeed SAM. Cat swarm optimization algorithm: a survey and performance evaluation. Comput Intell Neurosci. 2020;2020:4854895. doi: 10.1155/2020/4854895.
[14] Orouskhani M, Mansouri M, Teshnehlab M. Average-inertia weighted cat swarm optimization. Paper presented at the Advances in Swarm Intelligence, Berlin, Heidelberg; 2011.
[15] Yang J, Zheng Y, Ma L. Improved cat swarm optimization for solving traveling salesman problem. Appl Res Computers. 2017;12:3607–10.
[16] Tao Y, Zhang J, Wang Q. Routing optimizaton strategy of IoT perception layer based on improved cat swarm algorithm. Comput Eng. 2019;45(2):13–7.
[17] Chen C, Wang Y, Xie X. Adaptive cat swarm algorithm based on dynamic search. Comopter Eng degisn. 2021;42(7):7–14.
[18] Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. Comput Methods Appl Mech Eng. 2021;376:113609. doi: 10.1016/j.cma.2020.113609.
[19] Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH. Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng. 2021;157:107250. doi: 10.1016/j.cie.2021.107250.