IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2019, accepted May 5, 2019, date of publication July 22, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2916568

Multi-Strategy Adaptive Cuckoo Search Algorithm

SHUZHI GAO[1,2], YUE GAO[1,2], YIMIN ZHANG[1], AND LINTAO XU[1,2]
[1]Institute of Equipment Reliability, Shenyang University of Chemical Technology, Shenyang 110142, China
[2]College of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, China

Corresponding author: Yimin Zhang (zhangyimin@syuct.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1708254, in part by the Liaoning Province Higher School Climbing Scholar Project under Grant 2018]3807, in part by the Liaoning Natural Science Fund Project under Grant 20170540725, and in part by the Liaoning Province High-end Talent Construction Project-Distinguished Professor of Liaoning Province under Grant 20183533.

ABSTRACT Cuckoo search algorithm (CS) is a powerful biological-inspired search algorithm, which is widely used in continuous space optimization problems. However, a single search strategy in CS makes all cuckoos have similar search behavior, and it is liable to plunges into local optimal. In addition, whether CS can successfully solve a problem largely depends on the value of control parameters. Using the trial and error method to determine the value of parameters will cost a lot of computational expense frequently. In order to solve these problems, a multi-strategy adaptive cuckoo algorithm (MSACS) is proposed in this paper. Firstly, five search strategies are adapted to cooperate with each other, and the use of various previous strategies and control parameters are studied. The probability of each strategy being used and the value of control parameters are changed adaptively. Then, the performance of MSACS is tested and evaluated on 24 common benchmark functions. Finally, several advanced CS algorithms, particle swarm algorithm (PSO) and differential evolution algorithm (DE) variants will be compared with MSACS. The results show that the MSACS is better than the algorithms above.

INDEX TERMS Cuckoo search algorithm, multi-strategy, self-adaptive.

I. INTRODUCTION

In recent decades, people have proposed a series of new heuristic algorithms by observing the behavior of biological populations: swarm intelligence algorithm. Under the premise of no centralized control and no global model, swarm intelligence algorithm searches solutions by using the principle of 'trial-and-error', give full play to the advantages of the population, shows advanced and complex functions by using cooperation, competition, interaction, learning and other mechanisms, provides ideas for finding solutions to complex problems. Swarm intelligence algorithms do not rely on gradient information and have no requirements on continuity, derivable and other aspects for solving problems. It is suitable for both continuous numerical optimization and discrete combinatorial optimization.

Swarm intelligence optimization algorithm has great advantages in processing problems with large data volume due to its potential parallelism and distributed characteristics. At present, many swarm intelligence algorithms and their improved algorithm have been proposed, such as classical particle swarm optimization algorithm (PSO) [1], [2], ant colony optimization algorithm (ACO) [3], [4], artificial immune systems algorithm(AIS) [5], firefly algorithm(FA) [6], [7], Genetic algorithm(GA) [8], [9], differential evolution algorithm(DE) [10], [11]. These algorithms have been widely used in various practical problems.

Recently, Yang Xinshe and Deb Suash from Cambridge university proposed a novel meta-heuristic population optimization algorithm: cuckoo search algorithm [12], [13], inspired by the cuckoo's behavior in finding nests and laying eggs in nature. CS algorithm adopts Levy flight as its search strategy. By simulating the flight of fruit flies, and using a series of random walks characterized by mutant sequences, Levy's flight strategy is bound to find the optimal solution given enough time. There are only two control parameters in the Cuckoo search algorithm, which means that as long as one parameter is fixed, the change of the other parameter's value can be easily observed.

Cuckoo algorithm has been applied in various fields. Fateen and Bonilla-Petriciolett applied cuckoo search algorithm to the field of chemistry. Bonilla used gradient

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

based cuckoo search algorithm to solve phase equilibrium calculations in nonreactive systems [14]. Fateen used Cuckoo search algorithm to solve phase stability and phase equilibrium problems in applied thermodynamics [15]. Vazquez [16] used Cuckoo search algorithm to train spiking neural model. Galvez [17] used Cuckoo search to optimize weighted bayesian energy function. Deb and Mohamad et al. made a review of the progress of Cuckoo search algorithm, covering engineering, pattern recognition, software testing and other fields [18], [19].

However, scholars have found that Cuckoo search algorithm is not ideal in solving complex problems with multiple peaks. CS algorithm uses Levy flight strategy to conduct random search solely, which completely relies on random walk. Therefore, the search efficiency is low, it is difficult to find the optimal value, and it wastes computing efficiency. For some multimodal functions with many local minima, it takes many iterations to converge successfully. Levy's flight step size selection is also extremely important. Using a large step size causes it to hover around the real minimum, and making it difficult to achieve the desired accuracy. Using a smaller step size results in not finding all possible solutions, it is difficult to ensure the richness of the population. Therefore, many improved cuckoo algorithms have been proposed. In this paper, these improved cuckoo algorithms are divided into two categories:

(I)Improving the Control Parameters and Levy Flight Strategy: Walton et al proposed an improved Cuckoo search algorithm [20], which reduced the value of control parameters in a nonlinear way, and put the best part of the solution into a top-level set, randomly selected two cuckoos in each top-level sets. Then, connect them and create new individuals at the golden point of the line. Yongwei Zhang proposed an improved adaptive Cuckoo search algorithm [21], which changed the control parameters in a linear way adaptatively and grouped the cuckoo population for evaluation. This improves the survival rate of solution which is not the best one. Wang [22] changed the step size of Levy flight in a non-linear way adaptively, and changed the possibility of the nest being discovered linearly. Lijin Wang proposed a cuckoo algorithm using the strategy of Levy flight based on the nearest neighboring individuals [23]. The step size of Levy flight was changed from optimal-individual-based to the nearest-individual-based. Huang et al. [24] used chaos initialization strategy to enhance the ergodic type of search. Cheung et al. [25] adds quantum computing(QC) to the Cuckoo search algorithm, each individual can improve its search ability according to the expected potential. Considering the average position of individuals and the current optimal value, two new iterative schemes were generated. The original Levy flight scheme and the two new schemes were combined to optimize the cuckoo algorithm. Saida [26] applied Cuckoo search algorithm with quantum computing strategy to data clustering. Bilal [27] proposed 7 alternative schemes to replace Levy flight in iteration process and achieved good results.

(II)Hybridization of CS and Other Algorithms: Bhandari et al. [28] proposed a satellite image segmentation method combining CS algorithm and wind-driven optimization [29]. Li et al. [30] hybridized CS and PSO, realized the sharing of learning mechanism of the two methods. Liu et al. [31] proposed a hybridized CS based on leapfrog strategy, combined step-size strategy of individual grouping and Levy strategy. Xiangtao Li hybridized CS algorithm and DE algorithm [32], which used mutation strategy in DE algorithm to optimize the step size of CS algorithm, an self-adaptive way of adding markers was used to optimize the probability of being found. Uros Mlakar proposed a hybridized adaptive cuckoo algorithm [33], which had a more balanced exploration strategy and could automatically change the control variables of CS algorithm and reduced the size of population.

As can be seen from the literature review above, some scholars have proposed the improvement of CS algorithm, but it still needed to be improved. This paper proposes a multi-strategy adaptive cuckoo algorithm to improve the CS algorithm from the following two aspects: (i)Using multiple strategies to search with different step size. Make MSACS applicable to more problems. (ii) An adaptive strategy is added to CS to select the strategy adaptively and changes the value of parameters according to the algorithm. Let the problems themselves determine the strategy for solving them.

The structure of the paper is as follows: chapter II introduces the standard CS algorithm, chapter III elaborates the MSACS algorithm, chapter IV presents the numerical simulation results and analysis obtained after a large number of experiments, and chapter V is the summary.

II. CUCKOO SEARCH ALGORITHM

Actually, cuckoos find a nest which is suitable for them to lay eggs in a random way. To simulate this process, the following three idealized rules are introduced:

(a)Each cuckoo lays only one egg at a time, representing a solution to the problem and placing it in a randomly selected nest.

(b)Some of these nests have high-quality eggs, representing good solutions, and the nests will be preserved for the next generation.

(c)The number of available host nests is fixed, and the host finds the cuckoo's eggs with probability pa $\in$(0,1). In this case, the host can destroy the egg or abandon the old nest and build a new one.

In Cuckoo Search algorithm, the egg in each nest represents a solution, and the cuckoo egg represents a new solution. The goal is to use the new solution or potential superior solution to replace the inferior solution in the nest. Algorithms can be extended to be more complex, such as a nest with multiple eggs representing a group of solutions. In this article, the simplest approach is used. There is only one solution in a nest. In this case, eggs, nests, and cuckoos are essentially identical concepts which represent the solution in the algorithm.

According to the assumptions about cuckoo spawning behavior above, the path and position update formula of cuckoo algorithm is given as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\beta) \tag{1}$$

where $\alpha > 0$ is the step size scaling factor of Levy's flight, and its value is related to the selected problem, usually 0.01, 0.1 or 1. the product $\oplus$ denotes entry-wise multiplications.

$$Levy(\beta) = \frac{\varphi \cdot u}{|v|^{1/\beta}} \tag{2}$$

where Levy flight follows the following formula:

$$\varphi = \left(\frac{\Gamma(1+\beta) \cdot \sin(\pi \cdot \beta/2)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{(\beta-1)/2}\right)} \right)^{1/\beta} \tag{3}$$

where $v \sim$ N(0,1), $u \sim$ N(0,1). β is a constant on an interval [1,2], which is 1.5 in this paper. $\Gamma(\bullet)$ denotes gamma function.

III. MULTI-STRATEGY ADAPTIVE CUCKOO ALGORITHM

A. MULTI-STRATEGY SEARCH

The original cuckoo algorithm only uses Levy flight strategy, which results in all cuckoos having similar actions. When a cuckoo falls into a local minimum and cannot jump out of it on its own, it may not be able to jump out of the minimum by the cuckoo which is in the neighbor domain, because the cuckoo in the neighbor domain maybe under the same circumstance.

In MSACS, cuckoos search with five different strategies, instead of single Levy flight. In each iteration, each cuckoo randomly selects a strategy in an adaptive way to generate new cuckoos. We choose five strategies to make MSACS suitable for different problems. They have different characteristics, which are given below:

Strategy 1: Levy flight simulates the flight behavior of fruit flies in the original CS algorithm, and it is adopted to conduct the random search of small steps with a large probability and the long distance search with a large probability. Combined with other algorithms with different search lengths, Levy flight can give full play to its advantages. In this paper, Levy flight step size alpha is 1.

Strategy 2: Each cuckoo's search is related to the location of two other cuckoos which are randomly selected. As shown in equation (4), Cuckoo r1 which chooses this strategy will randomly select two cuckoos r2 and r3(r1, r2 and r3 are different individuals), and the nest of the next generation will be selected according to the location between them.

$$V_{i,G+1} = X_{r_1^i,G} + F \cdot \left(X_{r_2^i,G} - X_{r_3^i,G} \right) \tag{4}$$

where $X_{r_1^i,G}$, $X_{r_2^i,G}$, $X_{r_3^i,G}$ are three random and different individual vectors in generation G on [1,Np], which are also different from the i-th vector. F is the factor that determines how much to learn from other cuckoos.

Strategy 3: Similar to strategy 2, the search of each cuckoo is related to the locations of the other four cuckoos randomly. As shown in equation (5), Cuckoo r1 which chooses this strategy will randomly select two groups of cuckoos, with two cuckoos in each group, r2, r3, r4 and r5(r1, r2, r3, r4 and r5 are different individuals), will be randomly selected from the cuckoo r1 of this strategy, and the nest of the next generation will be selected according to the location among them. Compared with strategy 2, this strategy has smaller step size and is more suitable for small distance optimization.

$$V_{i,G+1} = X_{best,G} + F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G} + X_{r_3^i,G} - X_{r_4^i,G} \right) \tag{5}$$

where $X_{best,G}$ is the individual vector with the best fitness value in generation G. $X_{r_1^i,G}, X_{r_2^i,G}, X_{r_3^i,G}, X_{r_4^i,G}$ are four random and different individual vectors in generation G on [1,Np], which are also different from the i-th vector.

Strategy 4: Levy flight strategy has good search performance. MSACS adopts a strategy with memory to retain potential solutions which will be found through Levy flight strategy. For each cuckoo, a strategy is used to find the position with the best fitness value so far, and records it with a one-row, Np-column learning matrix LM. Where Np denotes the size of the population. In each generation, a random individual is selected from LM to generate a new individual according to equation (6) :

$$V_{i,G+1} = \xi \cdot \left(X_{L1,G} - X_{r_1^i,G} \right) + \xi \cdot \left(X_{L1,G} - X_{best,G} \right) \tag{6}$$

where ξ is a random number subject to Gaussian distribution with a variance of 0.3 and a mean of 0.5, $X_{L1,G}$ is a randomly selected individual in LM. This strategy makes MSACS memorable and prevents most cuckoos from getting stuck in a few local minima that are difficult to jump out.

Strategy 5: In recent decades, quantum computing (QC) has been applied in various optimization algorithms, shows good ability. MSACS learns from QC to improve their search ability. In search process, it is often the case that one cuckoo falls into a local minimum, and sometimes the cuckoo cannot jump out of this minimum only through its neighbors, because they follow the same search rules. Equation (7) adopts the strategy of QC method. Cuckoos jump out of the local minimum value by moving towards the average position of the whole population.

$$V_{i,G+1} = \overline{X_G} + \eta \cdot \left(\overline{X_G} - X_{i,G} \right) \cdot \lg(1/\gamma) \tag{7}$$

where η is a variable step size, its value is 1.6 in this paper, γ is a random number subject to uniform distribution on (0,1), $\overline{X_G}$ is the average position of all cuckoos in generation G, as shown in equation (8) :

$$\overline{X_G} = \frac{1}{Np} \sum_{i=1}^{Np} X_{i,G} \tag{8}$$

In MSACS, these five strategies work well together and thus produce better results. All cuckoos will search for a strategy at random according to the probability of each strategy being used in each generation.

At the beginning of iteration, strategy 1 relies on Levy flight and strategy 2 are used for medium-distance and long-distance searches, respectively. After finding a potential solution, Strategy 3 will be used for a small search. In order to maintain the diversity of the population, strategy 4, which is based on individual memory search and will be used to visit the optimal value once found. Strategy 5 can help jumping out of the local minimum region which cuckoos fall into. Each cuckoo selects only one strategy to search in each iteration, and adopts multiple strategies will not lead to slow operation of the algorithm.

B. ADAPTIVE STRATEGY

In the past, the adaptive algorithm made parameters change according to the pre-given rules. This adaptive algorithm is static and can not make parameters change completely according to the requirements of the algorithm. According to the theory of Smith J. and Fogarty T. (1997) [34], the adaption is divided into three categories: The first is a global adaptation. The adaptive operators are static, and the forms and parameters are fixed, and they act uniformly on the hole. The second is local adaption, each individual in the population are under consideration. The third is real-time adaption. Fitness parameters will change automatically in the process of iteration.

In MSACS each searching strategy will be used with a probability. Each cuckoo randomly selects a search strategy based on these probabilities to generate a new generation in each generation. These probabilities will generate depending on how they have played in previous generations before searching in each generation. It means that the more likely these search strategies are to find a better solution, the more likely they will be used in the future. However, no matter how the probability of each strategy being used changes, their sum is 1. That is, each cuckoo must choose one strategy to use in each iteration.

The probability of five search strategies being used is initialized to 0.2, that is, all strategies have the same probability of being selected at the beginning. A strategy will be selected for each cuckoo in the population in each generation. The ith strategy will be used with a probability of $Sp_{i,G}$ in generation G. In generation G, after calculating the fitness value of all generated cuckoos, the number of cuckoos generated by the ith strategy that can successfully enter the next generation was recorded as $n_{i,G}$. We introduced five memory libraries with length LM to store the memory data of each strategy separately, as shown in table 1. Where LM is the length of memory library, and $n_{i,G}$ is the number of success of the ith strategy in the G generation. Once the memories overflow after LM generations, the earliest records stored in the memories, $n_{i,G-LM+1}$, will be removed so that those numbers calculated in the current generation can be stored in memory libraries. That is, after calculating the fitness value in each generation, the last data in each memory store $n_{i,G-LM+1}$ will be deleted, and then each data in the memory library shift back one place. Finally, the newly generated success times $n_{i,G}$ are put into the memory library to become the first data.

TABLE 1. Memory bank of strageties.

Number of success	$n_{i,G}$	$n_{i,G-1}$	...	$n_{i,G-LM+2}$	$n_{i,G-LM+1}$
Index	1	2	...	LM-1	LM

After the initial LM generation, the probability that each policy which will be selected is generated based on their memory library. For example, the probability of the ith strategy used in the G generation is shown in equation (10) :

$$Sp_{i,G} = \frac{S_{i,G}}{\sum_{i=1}^{5} S_{i,G}} \tag{9}$$

$$S_{i,G} = \begin{cases} 0.01, & \sum_{j=G-LM}^{G-1} n_{i,j} = 0 \\ \dfrac{\sum_{j=G-LM}^{G-1} n_{i,j}}{\sum_{j=G-LM}^{G-1} m_{i,j}}, & else \end{cases} \tag{10}$$

where $S_{i,G}$ is the probability of the success of the ith strategy from generation G-LM to generation G, $m_{i,j}$ is the total number of times the ith strategy is used in generation j. Each strategy will calculate their probability of being used according to equation(9). Where $Sp_{i,G}$ is the mean probability of the ith strategy being used in generation G. It should be noted that, the usage probability of each strategy cannot be 0. Otherwise, it may lead to $m_{i,j}$ being 0 and $S_{i,G}$ not existing. So, when $S_{i,G} = 0$, we set its value equal to 0.01.

According to equations (9) and (10), it can be seen that the higher the ratio of all the success times in memory and the total number of times this strategy was used in these generations, namely, the higher the ratio of $\frac{\sum_{j=G-LM}^{G-1} n_{i,j}}{\sum_{j=G-LM}^{G-1} m_{i,j}}$, the higher the success rate of strategy i, and the more likely it will be used in the future.

C. DETERMINATION OF OTHER CONTROL PARAMETERS

The value of the control parameters depends largely on the problem that the algorithm deals with. In the traditional CS algorithm, the step size of Levy's flight and the probability of finding cuckoos are the controlled parameters of the algorithm. In MSACS, some of the parameters are given in sections 3.1 and 3.2. The scaling factor F in strategy 2 and strategy 3 is normally distributed with a mean value of 0.5 and a variance of 0.3, denoted by N(0.5,0.3).

An adaptive strategy similar to strategy selection is used to determine the probability of finding a cuckoo, pa. In the first

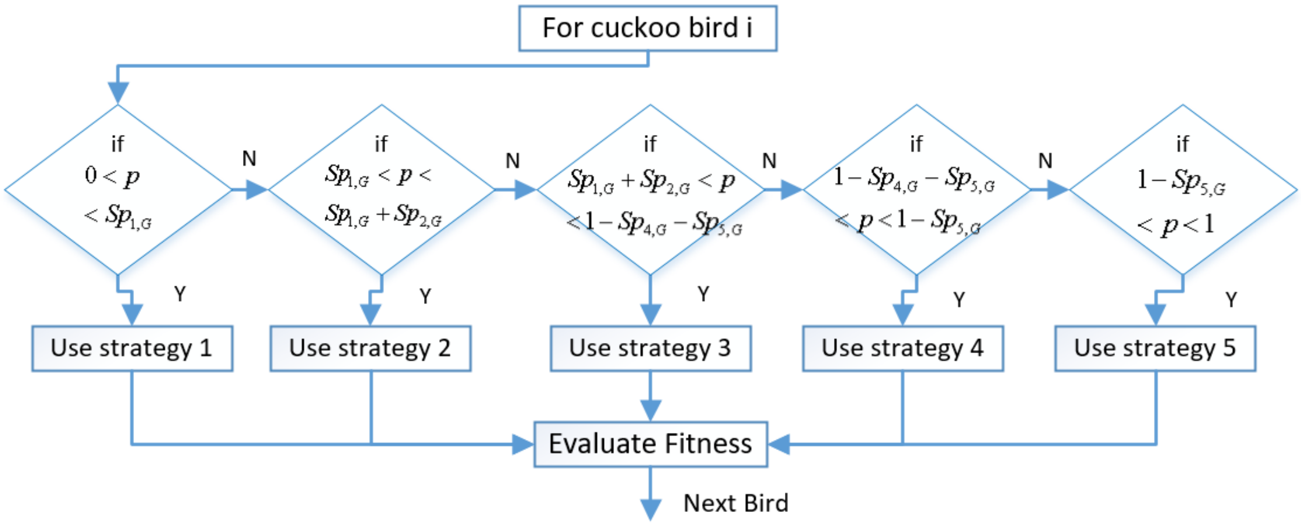

FIGURE 1. A schematic of how to select a strategy.

LM generation, pa is subject to a gaussian distribution with a mean of 0.5 and variance of 0.1. After the LM generation, record the pa value of the superior cuckoo in the previous LM generation in a memory matrix (p_{a_Memory}). A superior cuckoo is one with better fitness than before. At this point, pa is subject to a gaussian distribution with mean value of p_{a_mean} and variance of 0.1, denoted by N(p_{a_mean},0.1). Where p_{a_mean} is the average value of all pa in this memory bank. The value of pa should be between 0 and 1. If it is beyond this range, a random pa needs to be generated to participate in the iteration.

For the probability of finding a cuckoo in CS, an adaptive method is also used to automatically change its value. A number of experiments are used in chapter 4 to demonstrate the performance of MSACS in dealing with different types of problems. The pseudo-code for the MSACS algorithm is shown below.

MSACS uses adaptive strategies and multiple search strategies to optimize traditional CS algorithm, adaptively participates in the iteration with each cuckoo search result, so as to adaptively select more suitable strategies. It not only can be well applied to different adaptive functions, but also can automatically change the use probability of each strategy in different periods of iteration.

Algorithm 1 Pseudo-Code for MSACS

Begin
  Initialize G,the mean value of P_a (P_{a_mean}).
  Generate initial host nests x_i (i = 1,2,...,N).
  Evaluate the fifitness f of each nest x_i.
  Determine the global best nest, gbest.
  While the terminal condition is not reached, **Do**
    if G > LM
      for i = 1 to 5
        Update $Sp_{i,G}$ and $S_{i,G}$ by equation (9) and(10).
        Update the Memory Bank of Strageties.
      end
      Generate P_{a_mean} by P_{a_Memory}.
    end
    Generate F, ξ, γ, P_a.
    Choose stragety by $Sp_{i,G}$.
    Do Cuckoo Search by five strageties,generate new nests x_{i_new} (i = 1, 2, . . ., N).abandon Cuckoos by P_a randomly.
    if $f(x_i) \leq f(x_{i\_new})$
      $x_i = x_{i\_new}$, $n_{i,G-1} = n_{i,G-1} + 1$
    end
    G = G + 1
  End while
End

IV. SIMULATION AND COMPARISON

A. TEST FUNCTIONS

In order to examine the effectiveness of the proposed algorithm. In the following, we shift twenty-four benchmark functions which are commonly used in literatures. These benchmark functions are classified into three groups. Group1: Global optima of the benchmark function lies at the center of the search domain: f1-f3, f10, f12, f18, f23. Group2: The extremum of the function distributes on the coordinate axis: f4-f6, f9, f13, f14, f16, f21, f22, f24. Group3: The local optima of benchmark functions are independent of variables and dimensions: f7, f8, f11, f15, f17, f19, f20.

In these benchmark functions, f1, f4, f5, f8-f12 are shifted functions. f16 is plate-shaped. f1-f3, f6, f9, f21, f22 are bowl-shaped. f4, f15, f23, f24 are Valley-Shaped. f6-f8, f11, f12, f15, f17, f19, f20 are multimodal functions. f12, f23, f24 have a very sharp global optima in their searching range.

Function 1:Shifted Sphere function

$$f_1 = \sum_{i=1}^{n} z_i^2, \quad z = X - o$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 2:Generalized Penalized function 1

$$\begin{aligned} f_2 = &\frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \right.\\ &[1 + 10\sin^2(\pi y_{i+1})]\\ &\left. +(y_n - 1)^2\right\}\\ &+ \sum_{i=1}^{n} u(x_i, 10, 100, 4), |x_i| \leq 50\\ y_i = &1 + (x_i + 1)/4 \end{aligned}$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

Function 3:Generalized Penalized function 2

$$\begin{aligned} f_3 = &0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 \right.\\ &[1 + \sin^2(3\pi x_{i+1})]\\ &\left. +(x_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4) \end{aligned}$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

Function 4:Shifted Schwefel's problem 1.2

$$f_4 = \sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 5:Shifted Schwefel's Problem 1.2 with noise in fitness

$$f_5 = \left(\sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2\right)(1 + 0.4|N(0, 1)|)$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 6:Schwefel's Problem 2.22

$$f_6 = \sum_{i=1}^{n}|x_i| + \prod_{i=1}^{n}|x_i|$$

Function 7:Schwefel's Problem 2.26

$$f_7 = 418.9829 * n - \sum_{i=1}^{n}(x_i \sin(\sqrt{|x_i|}))$$

TABLE 2. Settings of 24 banchmark functions.

f	Demension	Iteration times	Global optimum f(x^*)	Search Range (Initialization Range)
f1	10	1000	0	[-5.12,5.12]D
	30	1000	0	[-5.12,5.12]D
f2	10	1000	0	[-50,50]D
	30	1000	0	[-50,50]D
f3	10	1000	0	[-50,50]D
	30	1000	0	[-50,50]D
f4	10	1000	0	[-100,100]D
	30	1000	0	[-100,100]D
f5	10	1000	0	[-32,32]D
	30	1000	0	[-32,32]D
f6	10	1000	0	[-10,10]D
	30	2000	0	[-10,10]D
f7	10	1000	0	[-500,500]D
	30	2000	0	[-500,500]D
f8	10	1000	0	[-5.12,5.12]D
	30	2000	0	[-5.12,5.12]D
f9	10	1000	0	[-600,600]D
	30	2000	0	[-600,600]D
f10	10	1000	0	[-10,10]D
	30	1000	0	[-10,10]D
f11	10	1000	0	[-100,100]D
f12	10	1000	0	[-32,32]D
f13	10	1000	0	[-0.5,0.5]D
f14	10	1000	0	[-10,10]D
f15	10	1000	0	[-4.5,4.5]D
f16	2	1000	0	[-10,10]D
f17	2	1000	0	[-100,100]D
f18	2	1000	0	[-100,100]D
f19	2	1000	0	[-2,2]D
f20	2	1000	0	[-100,100]D
f21	2	1000	0	[-100,100]D
f22	2	1000	0	[-100,100]D
f23	2	1000	0	[-100,100]D
f24	2	1000	0	[-30,30]D

Function 8:Shifted Rastrigin's function

$$f_8 = \sum_{i=1}^{n}\left[z_i^2 - 10\cos(2\pi z_i) + 10\right]$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 9:Shifted Griewank's Function

$$f_9 = \frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 10:Shifted Sum of different power function

$$f_{10} = \sum_{i=1}^{n}|z_i|^{i+1}$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 11:Shifted noncontinuous Rastrigin's function

$$f_{11} = \sum_{i=1}^{n}[u_i^2 - 10\cos(2\pi u_i) + 10]$$

$$u_i = \begin{cases} z_i, & z_i < 0.5 \\ round(2z_i)/2, & |z_i| >= 0.5 \end{cases}$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

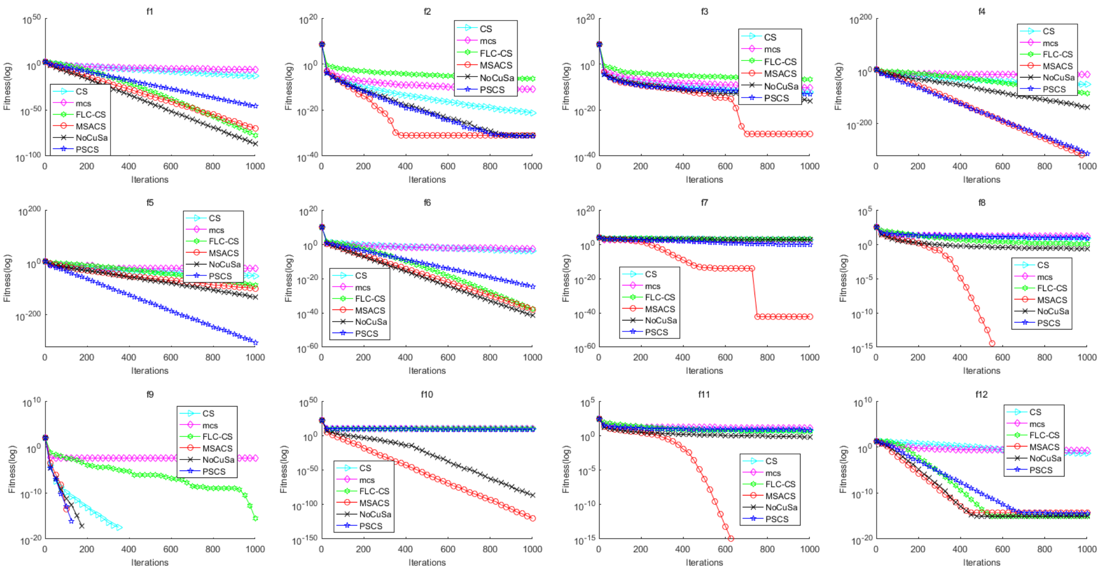

FIGURE 2. Comparison diagram of simulation results of CS-based algorithm on 24 benchmark functions(f1-f12).

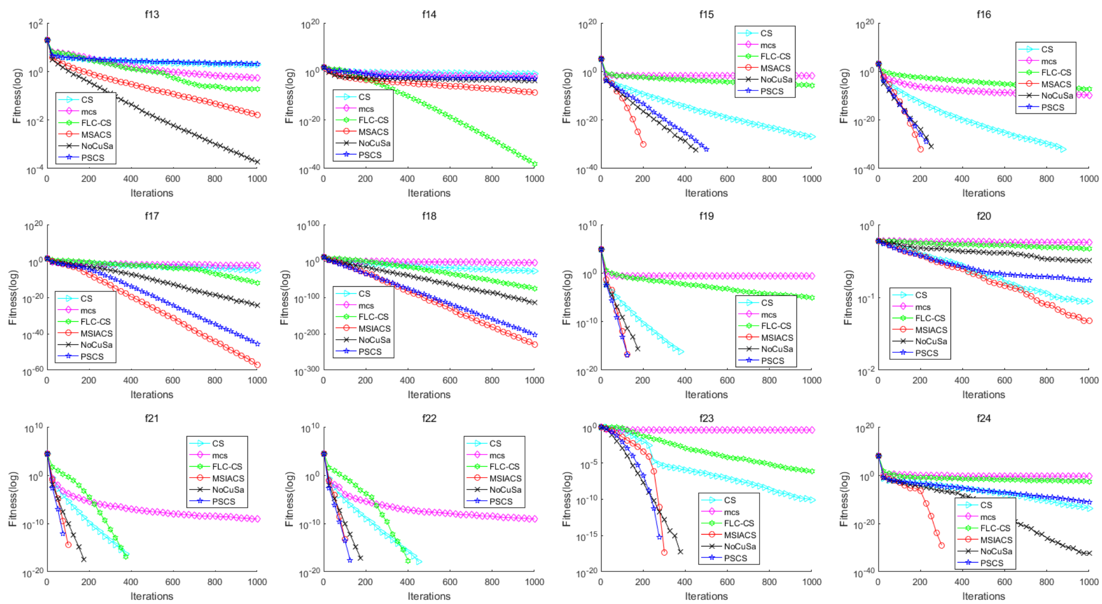

FIGURE 3. Comparison diagram of simulation results of CS-based algorithm on 24 benchmark functions(f13-f24).

Function 12:Shifted Ackley's function

$$f_{12} = -20\exp(-0.2\sqrt{\sum_{i=1}^{n} z_i^2/D}) - \exp(\sum_{i=1}^{n}\cos(2\pi z_i)/n) + 20 + e$$

$o = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

Function 13:Weierstrass function

$$f_{13} = \sum_{i=1}^{n}(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(x_i+0.5))]) - n\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k\cdot 0.5)] + 20$$

$$a = 0.5, \quad b = 3, \quad k_{\max} = 20$$

Function 14:Alpine function

$$f_{14} = \sum_{i=1}^{n}|x_i\sin(x_i) + 0.1x_i|$$

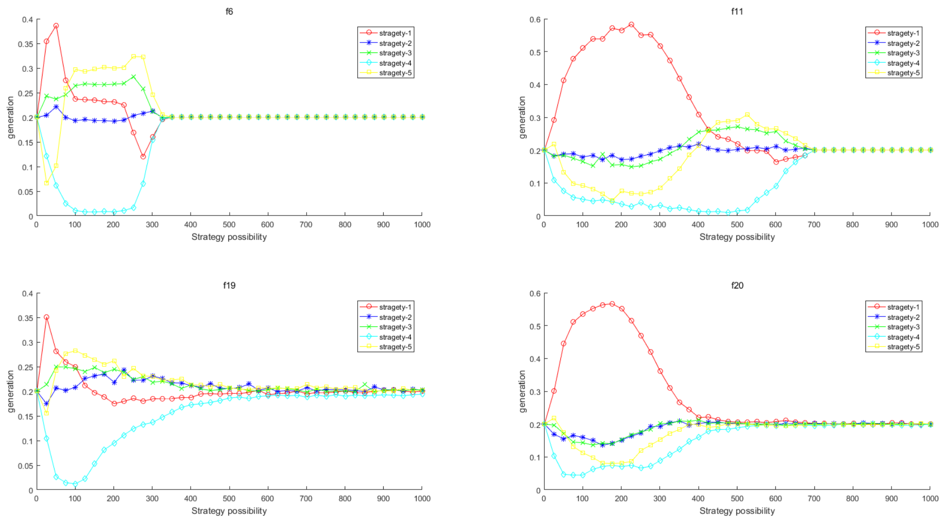

FIGURE 4. the probability of each strategy being used in an iteration.

Function 15:Beale's function

$$f_{15} = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1{x_2}^2)^2 + (2.625 - x_1 + x_1{x_2}^3)^2$$

Function 16:Booth's function

$$f_{16} = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

Function 17:Shaffer function

$$f_{17} = \left(x_1^2 + x_2^2\right)^{\frac{1}{4}} [\sin^2(50(x_1^2 + x_2^2)^{\frac{1}{10}}) + 1]$$

Function 18:Bent Cigar function

$$f_{18} = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$$

Function 19:Goldstein-Price function

$$f_{19} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] - 3$$

Function 20:J.D.Schaffer function

$$f_{20} = \frac{\sin^2\sqrt{x_1^2 + x_2^2 - 0.5}}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5$$

Function 21:Bohachevsky function 1

$$f_{21} = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

Function 22:Bohachevsky function 2

$$f_{22} = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$$

Function 23:Easom's 2D function

$$f_{23} = 1 - \cos(x_1)\cos(x_2)\exp[-(x_1 - \pi)^2 - (x_2 - \pi)^2]$$

Function 24:Rosenbrock's Function

$$f_{24} = \sum_{i=1}^{n-1} [100(x_2 - x_1^2)^2 + (1 - x_1)^2]$$

The dimension, global optimal value, search range and initial range of these test functions are given in table 2. The benchmark function f1-f10 is simulated in both 10 and 30 dimensions. For the 30-dimensional simulation of f6-f9, the number of iterations are set to 2000, and the rest are set to 1000.

B. ALGORITHMS OF COMPARISON

In this paper, by comparing the results of 12 optimization algorithms including MSACS on 24 objective functions, 5 of them are CS algorithm and its improved algorithm, and the remaining 6 are some classical improved algorithms for other optimization algorithms, including:Auto-enhanced population diversity differential evolution: AEPD-DE [35], DE with an evolution path: DEEP [36], Self-adaptive differential evolution: jDE [37], DE with strategy adaptation SaDE [38], PSO with Interswarm Interactive learning strategy:IILPSO [39], Quantum Behaved PSO:QPSO [40], CS[12], Modified CS: MCS [20], Particle swarm CS:PSCS [30], Frog leaping and chaotic CS:FLC-CS [31], A nonhomogeneous cuckoo search algorithm: NoCuSa [25]. Their parameters are set as follows:

AEPD-DE: $\tau_1 = \tau_2 = 0.1, F = 0.5$ and $CR = 0.9$

DEEP: $\tau_1 = \tau_2 = 0.1, F = 0.5, CR = 0.9, \lambda = 0.5, s = 20$

jDE: $\tau_1 = \tau_2 = 0.1, F = 0.5$ and $CR = 0.9$

SaDE: $LP = 10$, $CR_0 = 0.5$

IILPSO: $c_1{}' = c_2{}' = 1, c_3 = 2, N_1 = N_2 = 50$

QPSO: $\alpha = 0.54$

CS: $p_a = 0.25$, $\alpha = 0.01$, $\beta = 1.5$

MCS: $p_a = 0.75$, $\alpha = 100$

PSCS: $p_a = 0.25$, $\alpha = 0.01$

FLC-CS: $p_a = 0.25$, $\alpha = 0.01$, $\omega = (2/t)^{0.3}$

NoCuSa: $p_a = 0.3$, $\alpha = 0.1$, $\beta = 1.5$, $\delta = 1.6$

TABLE 3. Simulation results of the CS-based algorithm on 10-dimensional benchmark function f1-f10.

Algor-ithm	f1					f2					f3				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	7.20E-14	9.07E-15	2.82E-13	4.19E-14	**1**	5.19E-22	3.04E-24	7.15E-21	9.34E-22	**1**	4.99E-13	6.78E-17	5.55E-12	8.78E-13	**1**
MCS	9.38E-07	1.78E-07	2.56E-06	4.49E-07	**1**	9.15E-12	4.23E-17	9.90E-11	1.60E-11	**1**	5.15E-11	1.41E-17	4.69E-10	8.95E-11	**1**
PSCS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	1.33E-13	1.64E-19	3.08E-12	4.11E-13	**1**
FLC-CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	4.16E-07	3.50E-10	6.72E-06	7.82E-07	**1**	2.77E-07	3.57E-09	2.02E-06	3.32E-07	**1**
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	2.60E-16	**0.00E+00**	2.03E-14	2.10E-15	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f4					f5					f6				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	8.71E-05	3.79E-05	1.89E-04	2.73E-05	0
MCS	1.24E-16	**0.00E+00**	9.83E-15	9.94E-16	**1**	3.49E-24	**0.00E+00**	2.05E-22	2.41E-23	**1**	2.32E-03	1.09E-03	4.04E-03	5.67E-04	0
PSCS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	2.98E-25	5.56E-26	1.20E-24	2.14E-25	**1**
FLC-CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f7					f8					f9				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	4.74E+02	2.17E+02	7.66E+02	1.11E+02	0	7.50E+00	3.98E+00	1.13E+01	1.46E+00	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MCS	6.04E+02	2.75E-05	1.95E+03	4.90E+02	0	1.61E+01	4.97E+00	3.88E+01	8.13E+00	0	3.52E-03	**0.00E+00**	3.95E-02	5.85E-03	0.61
PSCS	5.61E-01	**0.00E+00**	3.65E+00	7.12E-01	0.05	8.22E+00	3.49E+00	1.29E+01	1.82E+00	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
FLC-CS	1.44E+03	6.79E+02	1.75E+03	1.88E+02	0	1.19E+00	**0.00E+00**	2.30E+01	4.53E+00	0.93	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
NoCuSa	5.14E+02	1.44E+02	7.44E+02	1.24E+02	**1**	1.73E-01	**0.00E+00**	2.42E+00	5.13E-01	0.78	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f10				
	Mean	Best	Worst	Std	Success
CS	5.33E+09	3.86E+01	1.00E+10	4.99E+09	0
MCS	3.72E+09	1.57E-10	1.00E+10	4.76E+09	0.14
PSCS	2.91E+08	3.56E+03	1.00E+10	1.49E+09	0
FLC-CS	5.14E+09	**0.00E+00**	1.00E+10	4.67E+09	0.26
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

TABLE 4. Simulation results of the cs-based algorithm on 30-dimensional benchmark function f1-f10.

Algor-ithm	f1					f2					f3				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	1.76E-07	5.87E-08	3.09E-07	5.56E-08	**1**	5.79E-22	4.72E-25	4.35E-21	7.74E-22	**1**	5.73E-13	2.14E-16	7.32E-12	1.30E-12	**1**
MCS	1.82E-06	7.69E-07	2.97E-06	4.04E-07	**1**	1.11E-11	5.15E-19	7.15E-11	1.58E-11	**1**	5.10E-11	2.45E-19	4.89E-10	8.46E-11	**1**
PSCS	2.83E-23	5.26E-24	1.19E-22	1.83E-23	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	1.15E-13	1.83E-18	2.42E-12	3.59E-13	**1**
FLC-CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	4.09E-07	7.41E-10	2.26E-06	4.56E-07	**1**	3.25E-07	2.74E-09	1.75E-06	3.65E-07	**1**
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	2.04E-16	**0.00E+00**	1.95E-14	1.95E-15	1
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f4					f5					f6				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	7.76E-02	4.83E-02	1.20E-01	1.53E-02	0
MCS	1.57E-15	**0.00E+00**	1.54E-13	1.54E-14	**1**	1.67E-18	**0.00E+00**	1.67E-16	1.67E-17	**1**	1.78E+00	4.37E-03	8.05E+01	1.08E+01	0
PSCS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	3.74E-13	1.95E-13	7.98E-13	1.19E-13	**1**
FLC-CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	2.75E-22	0.00E+00	1.14E-20	1.62E-21	**1**

Algor-ithm	f7					f8					f9				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	3.56E+03	2.72E+03	4.08E+03	2.79E+02	0	1.01E+02	6.73E+01	1.23E+02	1.10E+01	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MCS	2.67E+03	1.50E-05	7.74E+03	2.18E+03	0	6.65E+01	2.89E+01	1.44E+02	2.33E+01	0	3.37E-03	0.00E+00	3.95E-02	7.06E-03	0.68
PSCS	1.41E-01	7.54E-04	5.59E-01	1.35E-01	0	1.33E+02	9.98E+01	1.57E+02	1.01E+01	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
FLC-CS	7.22E+03	6.17E+03	7.80E+03	3.23E+02	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
NoCuSa	4.28E+03	3.10E+03	5.38E+03	5.10E+02	0	6.38E+00	0.00E+00	2.57E+01	5.58E+00	0.02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**4.09E-14**	**4.71E-15**	**1.81E-12**	**2.55E-13**	**1**	8.95E-01	**0.00E+00**	6.96E+00	1.42E+00	0.59	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f10				
	Mean	Best	Worst	Std	Success
CS	1.98E-07	1.11E-08	1.66E-06	2.74E-07	**1**
MCS	1.79E-02	3.06E-09	1.79E+00	1.79E-01	0.99
PSCS	2.49E-13	3.95E-15	1.54E-12	3.17E-13	**1**
FLC-CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

C. COMPARISON WITH CS-BASED ALGORITHMS

In this section, MSACS algorithm is compared with CS-based algorithms, including: CS, MCS, PSCS, FLC-CS, NoCuSa. The population size of these algorithms is set to 50. The simulation results of the CS-based algorithm on 24 benchmark functions are shown

TABLE 5. Simulation results of the cs-based algorithm on 10-dimensional benchmark function f11-f24.

Algor-ithm	f11					f12					f13				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	5.58E+00	4.06E+00	8.08E+00	7.81E-01	0	3.44E-02	6.92E-03	9.22E-02	1.87E-02	0	1.86E+00	1.24E+00	2.37E+00	2.12E-01	0
MCS	1.40E+01	4.00E+00	4.60E+01	8.55E+00	0	6.30E-01	6.54E-04	1.92E+01	2.70E+00	0	4.65E-01	1.80E-01	3.37E+00	4.37E-01	0
PSCS	6.78E+00	4.19E+00	8.08E+00	7.00E-01	**1**	4.01E-15	**8.88E-16**	4.44E-15	1.16E-15	**1**	2.08E+00	1.34E+00	2.61E+00	2.47E-01	0
FLC-CS	3.21E+00	**0.00E+00**	1.40E+01	4.73E+00	0.67	**8.88E-16**	**8.88E-16**	**8.88E-16**	0.00E+00	**1**	8.48E-02	**4.86E-10**	4.37E+00	5.97E-01	**0.95**
NoCuSa	7.55E-01	**0.00E+00**	3.53E+00	9.86E-01	0.41	**8.88E-16**	**8.88E-16**	**8.88E-16**	0.00E+00	**1**	**1.75E-04**	**4.86E-10**	**1.37E-03**	2.71E-04	0.6
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	4.44E-15	4.44E-15	4.44E-15	0.00E+00	**1**	1.77E-02	**4.86E-10**	7.55E-02	2.04E-02	0.23

Algor-ithm	f14					f15					f16				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	1.31E-01	3.24E-02	2.81E-01	5.00E-02	0	9.15E-28	2.37E-31	1.53E-26	2.24E-27	**1**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**1**
MCS	9.93E-03	2.86E-04	4.46E-02	1.00E-02	0	2.10E-02	9.34E-24	6.84E-01	1.14E-01	0.96	1.93E-10	6.56E-19	1.86E-09	3.41E-10	**1**
PSCS	1.44E-03	7.25E-04	2.07E-03	2.89E-04	0	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
FLC-CS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	2.78E-06	5.74E-09	3.37E-05	5.25E-06	0.95	4.43E-08	2.56E-10	3.10E-07	5.84E-08	**1**
NoCuSa	1.91E-04	5.12E-09	7.05E-04	1.83E-04	0.18	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	9.53E-09	**0.00E+00**	5.92E-07	6.90E-08	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f17					f18					f19				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	1.01E-05	4.11E-07	2.75E-05	5.89E-06	0.63	7.94E-30	**0.00E+00**	3.95E-28	3.97E-29	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MCS	4.18E-03	7.94E-04	5.42E-03	1.66E-03	0	7.63E+01	9.03E-17	3.09E+03	3.61E+02	0.86	1.69E-08	**0.00E+00**	1.28E-07	2.49E-08	**1**
PSCS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
FLC-CS	2.63E-13	2.66E-18	2.18E-11	2.18E-12	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	1.30E-05	9.23E-09	6.78E-05	1.40E-05	0.68
NoCuSa	4.34E-27	**0.00E+00**	2.40E-25	2.42E-26	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f20					f21					f22				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	8.84E-02	2.99E-10	**5.89E-01**	2.11E-01	0.74	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MCS	5.66E-01	4.44E-16	**5.89E-01**	1.16E-01	0.03	1.02E-09	**0.00E+00**	7.49E-09	1.62E-09	**1**	8.26E-10	**0.00E+00**	1.11E-08	1.55E-09	**1**
PSCS	1.72E-01	2.43E-06	**5.89E-01**	2.45E-01	0.03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
FLC-CS	4.67E-01	**0.00E+00**	**5.89E-01**	2.13E-01	0.09	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
NoCuSa	3.17E-01	**0.00E+00**	**5.89E-01**	2.91E-01	0.45	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**6.08E-02**	**0.00E+00**	**5.89E-01**	1.78E-01	**0.85**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

Algor-ithm	f23					f24				
	Mean	Best	Worst	Std	Success	Mean	Best	Worst	Std	Success
CS	3.86E-11	6.00E-15	7.64E-10	1.07E-10	**1**	2.37E-14	5.61E-18	5.76E-13	6.90E-14	**1**
MCS	4.85E-01	4.11E-15	1.00E+00	4.86E-01	0.48	1.75E-01	3.66E-20	1.19E+01	1.22E+00	0.93
PSCS	3.75E-08	**0.00E+00**	3.75E-06	3.75E-07	**1**	9.98E-12	4.91E-15	4.00E-10	4.13E-11	**1**
FLC-CS	8.91E-07	1.84E-09	2.77E-05	2.91E-06	0.99	2.65E-03	1.98E-06	5.88E-02	7.07E-03	0.02
NoCuSa	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**1**

in figure 2, figure 3, and table 3, table 4, table 5 in appendix.A.

Figure 2, 3 are the line graph of the change of the mean value of f(x)-f(x*) with the number of iterations after these 6 algorithms run 100 times on each benchmark function, where f1-f10 is 10-D. Table 3 and table 4 show their simulation results in f1-f10 10-d and 30-d respectively, and table 5 shows their simulation results in f11-f24, including mean value, optimal value, worst value, variance and success rate of successful operation. The success of an algorithm means that the algorithm has successfully reached the preset optimal value $f(x^*) = 1e\text{-}5$. The success rate refers to the proportion of the number of successful times in the 100 runs. In table 3, 4 and 5, $f(x^*) = 1e\text{-}30$ is the termination condition of iteration. The best solutions are in bold.

As can be seen from figure 2, figure 3, table 3 and table 5, NoCuSa is the best performing function except MSACS. In the first group of benchmark functions and the second group of benchmark functions except f3 and f9, it is far superior to CS. It is also significantly stronger than the CS algorithm in the third group of functions f15 and f17. In addition, it is slightly stronger than CS algorithm in f3, f8, f9, f11, f14 and f19. PSCS and FLC-CS slightly worse than NoCuSa, They are obviously superior to CS algorithms in f1, f6, f12 and f18. FLC-CS is much better in f13 f14. However, compared with CS algorithm, PSCS has greater improvement in f2, f4, f5, f9, f15, f16, f17, f18, f21, f22 and f23. Although MCS is not as good as CS algorithm in most functions, it exceeds CS algorithm in f10 and f14. MSACS is the best performing of all functions, with 100% convergence rate for all functions except f13 and f20. MSACS is trapped in local minimum value on f12, which is the same with FLC-CS and NoCuSa. It is better than CS algorithm in all the 24 benchmark problems. Among multimodal functions, MSACS is better than all other algorithms in f7, f8, f11, f15, f17 and f20, and almost the best in f6, f12 and f19, indicating that MSACS has a good global search ability.

As can be seen from table 4, in f1-f10 of 30-D, PSCS can converge well on f2, f4, f5 and f9. Although FLC-CS cannot converge on f7 and is weaker than NoCuSa on f2 and f3, but it can converge to 0 in other functions. NoCuSa cannot find the global optimal successfully in f7 and f8, and it is weaker than MSACS in f3. MSACS has a good effect in high dimension, with a 59% success rate in f8, which is weaker than 100%

TABLE 6. 10-D Comparison of simulation results of the improved algorithm based on DE and PSO on 24 benchmark functions.

10D	f1			f2			f3			f4			f5		
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success
AEPD-DE	2.62E-12	1.89E-11	**1**	**0.00E+00**	**0.00E+00**	**1**	1.04E-29	1.04E-28	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
DEEP	1.65E-06	1.65E-05	0.99	8.03E-04	8.26E-04	0.01	3.41E-04	3.54E-04	0.03	1.18E-06	2.01E-06	0.99	1.14E-07	3.17E-07	**1**
IILPSO	3.31E-02	9.20E-02	0.79	7.08E-04	1.80E-03	0.65	5.05E-04	2.18E-03	0.59	1.36E-05	1.12E-04	0.95	1.46E-06	9.43E-06	0.98
jDE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
QPSO	9.78E-06	9.64E-05	0.98	2.04E-10	1.01E-09	**1**	2.20E-10	7.93E-10	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
SaDE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	7.14E-17	5.08E-16	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
10D	f6			f7			f8			f9			f10		
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success
AEPD-DE	4.47E-06	1.71E-05	0.9	7.47E+02	4.02E+02	0	2.01E+01	3.63E+00	0	**0.00E+00**	**0.00E+00**	**1**	2.22E-30	1.57E-29	**1**
DEEP	1.43E+00	3.56E+00	0.76	1.56E+03	1.61E+02	0	5.12E+01	6.51E+00	0	2.47E-05	4.38E-05	0.55	1.02E+08	6.06E+08	0.02
IILPSO	3.97E-01	8.73E-01	0.35	7.82E+02	5.81E+02	0.09	1.95E+01	1.62E+01	0.12	1.06E-04	5.34E-04	0.84	8.67E+09	3.38E+09	0.11
jDE	2.63E-21	1.78E-21	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
QPSO	2.22E-04	8.85E-04	0.68	6.43E+02	3.05E+02	0.02	5.43E+00	2.51E+00	0	**0.00E+00**	**0.00E+00**	**1**	2.98E+01	2.79E+02	0
SaDE	2.96E-20	6.05E-20	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
10D	f11			f12			f13			f14			f15		
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success
AEPD-DE	1.21E+01	2.39E+00	0	3.12E-06	1.24E-05	0.93	3.82E+00	4.54E-01	0	2.15E-03	1.29E-03	0	**0.00E+00**	4.52E-30	**1**
DEEP	3.20E+01	7.31E+00	0	1.66E-01	8.60E-01	0.9	4.47E+00	3.13E-01	0	3.80E+00	1.32E+00	0	6.09E-08	2.79E-07	**1**
IILPSO	1.39E+01	1.26E+01	0.05	1.29E+00	2.07E+00	0.56	2.84E+00	1.18E+00	0	1.53E+00	1.85E+00	0.18	1.48E-04	9.24E-04	0.79
jDE	5.33E-17	5.33E-16	**1**	**4.44E-15**	0.00E+00	**1**	2.84E-01	4.95E-02	0	7.87E-06	5.72E-06	0.71	**0.00E+00**	**0.00E+00**	**1**
QPSO	5.07E+00	1.70E+00	0	1.37E+00	1.02E+00	0.01	4.44E-02	1.32E-01	0	2.93E-03	1.11E-02	0.51	**0.00E+00**	**0.00E+00**	**1**
SaDE	**0.00E+00**	**0.00E+00**	**1**	**4.26E-15**	0.00E+00	**1**	2.43E-04	1.30E-04	0	3.61E-10	4.05E-10	**1**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**1**	**4.44E-15**	0.00E+00	**1**	1.77E-02	2.04E-02	**0.23**	9.53E-09	6.90E-08	**1**	**0.00E+00**	**0.00E+00**	**1**
10D	f16			f17			f18			f19			f20		
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success
AEPD-DE	**0.00E+00**	**0.00E+00**	**1**	2.23E-09	1.10E-08	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**5.34E-06**	**5.34E-05**	**1**
DEEP	4.36E-06	2.91E-05	0.96	4.37E-04	1.12E-03	0.74	9.56E+01	1.05E+02	0	3.34E-05	2.88E-04	0.96	1.69E-01	1.41E-01	0
IILPSO	2.58E-04	1.01E-03	0.66	1.88E-01	2.64E-01	0.01	2.72E+01	1.10E+02	0.29	7.22E-03	5.78E-02	0.64	3.86E-01	2.68E-01	0.15
jDE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	1.74E-01	1.85E-01	0.04
QPSO	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	4.54E-01	2.48E-01	0.01
SaDE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	3.69E-02	1.09E-01	0.58
MSACS	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	6.08E-02	1.78E-01	0.85
10D	f21			f22			f23			f24					
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success			
AEPD-DE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	6.27E-10	2.57E-09	**1**	1.28E-26	9.58E-26	**1**			
DEEP	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	1.94E-01	1.57E-01	0	1.77E-03	1.76E-02	0.93			
IILPSO	6.01E-02	1.94E-01	0.4	6.31E-02	1.57E-01	0.41	1.48E-02	9.69E-02	0.52	5.15E-02	3.23E-01	0.65			
jDE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**			
QPSO	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	2.80E-01	4.51E-01	0.72	1.26E-29	1.26E-28	**1**			
SaDE	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**			
MSACS	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**			

of FLC-CS and stronger than other algorithms. It has a 100% success rate in the rest of the test functions and is superior to all other functions in f7.

D. COMPARISON WITH OTHER STATE-OF-THE-ART ALGORITHMS ALGORITHMS

In this section, MSACS is compared with other state-of-the-art algorithms, and the results of the comparison are recorded in table 6, table 7 in appendix.A.

As can be seen from table 6 and table 7, MSACS is the best of all algorithms in the first group of benchmark functions and can achieve 100% success in all functions. In the second group of basis functions, it ranks first in all functions except f14, which is only inferior to SaDE. In all the other functions, MSACS is in the first level. It surpasses all other algorithms particularly in f6 and f13. SaDE and jDE are at the second level, and they are trapped in the local minimum on f13 and cannot find the optimal value. In the third group of multimodal functions, MSACS and SaDE have shown good search performance. The optimal solution can be found successfully in f7, f8, f11, f15, f17 and f19. AEPD-DE has a 100% success rate in f20, which cannot be achieved by other algorithms. MSACS also has good performance in 30-D functions and is superior to all other algorithms in f1 and f6. SaDE is stronger than other functions in f7 and is the only algorithm with a 100% success rate in f8.

In disk-shaped function and bowl-shaped function. Most algorithms are able to find the optimal value. While in f12, f23, f24 and f6-f8, f11, f12 and f20, which have many local minima, only jDE, SaDE and MSACS can successfully converge, which indicates that they all have good optimization ability, and the results of MSACS are better. jDE is slightly worse when dealing with high-dimensional functions, and SaDE is successful in solving single-mode problems, but its convergence rate is slow.

TABLE 7. 30-D Comparison of simulation results of the improved algorithm based on DE and PSO on 24 benchmark functions.

30D	f1			f2			f3			f4			f5		
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success
AEPD-DE	1.50E-07	1.04E-06	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
DEEP	5.24E+00	1.04E+01	0.16	7.25E-04	7.34E-04	0.02	2.57E-04	2.77E-04	0.03	1.11E-06	2.02E-06	0.99	1.60E-07	3.63E-07	**1**
IILPSO	1.89E-01	9.43E-01	0.86	2.00E-03	8.65E-03	0.51	7.18E-04	2.31E-03	0.49	7.15E-06	3.26E-05	0.9	7.23E-06	5.87E-05	0.95
jDE	3.40E-26	2.53E-26	**1**	**0.00E+00**	**0.00E+00**	**1**	5.11E-29	5.10E-28	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
QPSO	5.07E-01	3.55E-01	0	9.46E-11	5.74E-10	**1**	4.69E-10	1.00E-09	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
SaDE	9.38E-26	2.68E-25	**1**	**0.00E+00**	**0.00E+00**	**1**	1.57E-17	1.20E-16	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
30D	f6			f7			f8			f9			f10		
	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success	Mean	Std	Success
AEPD-DE	5.78E-04	2.15E-03	0.27	6.81E+03	8.19E+02	0	1.78E+02	1.37E+01	0	**0.00E+00**	**0.00E+00**	**1**	2.79E-05	2.78E-04	0.99
DEEP	7.42E+01	6.14E+01	0	7.80E+03	2.65E+02	0	2.75E+02	1.82E+01	0	7.93E-06	1.73E-05	0.75	2.57E+12	1.97E+13	0
IILPSO	1.70E+01	1.39E+02	0.22	6.46E+03	1.58E+03	0	1.54E+02	8.12E+01	0	2.10E-04	1.00E-03	0.8	2.47E+11	2.10E+12	0.83
jDE	2.37E-14	8.51E-15	**1**	1.62E-12	5.97E-12	**1**	2.30E+01	3.28E+00	0	**0.00E+00**	**0.00E+00**	**1**	8.34E-29	4.12E-28	**1**
QPSO	3.92E+00	2.15E+00	0	4.98E+03	1.17E+03	0	4.72E+01	1.08E+01	0	**0.00E+00**	**0.00E+00**	**1**	3.14E+03	1.34E+04	0
SaDE	1.44E-12	1.86E-12	**1**	**4.71E-15**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**
MSACS	**2.75E-22**	**1.62E-21**	**1**	4.09E-14	2.55E-13	**1**	8.95E-01	1.42E+00	0.59	**0.00E+00**	**0.00E+00**	**1**	**0.00E+00**	**0.00E+00**	**1**

TABLE 8. 30-D Comparison of simulation results of the improved algorithm based on DE and PSO on 24 benchmark functions.

function	10D	15D	20D	25D	30D	35D	40D	45D	50D	55D	60D
f1	5.63E-72	3.60E-71	9.74E-73	2.74E-73	4.36E-72	1.89E-72	2.70E-71	8.23E-72	9.90E-72	1.31E-71	1.54E-71
f2	1.19E-31	9.74E-32	1.08E-31	1.12E-31	1.07E-31	9.74E-32	1.41E-31	9.11E-32	1.08E-31	1.23E-31	7.85E-32
f3	4.73E-31	2.76E-31	4.31E-31	1.19E-30	3.84E-31	3.42E-31	1.92E-31	8.03E-31	5.22E-31	4.35E-31	2.40E-31
f4	4.94E-324	4.98E-321	2.71E-316	3.10E-313	4.58E-308	8.80E-297	1.91E-297	4.24E-295	4.21E-290	8.45E-285	1.23E-281
f5	7.99E-112	3.50E-113	3.02E-126	2.88E-139	5.71E-149	2.27E-148	3.42E-158	3.92E-164	6.63E-162	3.32E-166	1.40E-159
f6	1.56E-38	4.67E-38	1.92E-38	7.73E-39	1.45E-38	2.50E-38	3.76E-38	1.70E-38	3.28E-38	5.70E-38	7.73E-38
f7	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43	4.81E-43
f8	0.00E+00	0.00E+00	0.00E+00	3.32E-02	0.00E+00	3.32E-02	0.00E+00	3.32E-02	3.32E-02	6.63E-02	6.63E-02
f9	0.00E+00	0.00E+00	0.00E+00	3.32E-02	0.00E+00	3.32E-02	0.00E+00	3.32E-02	3.32E-02	6.63E-02	6.63E-02
f10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f11	0.00E+00	0.00E+00	0.00E+00	3.33E-02	0.00E+00	3.33E-02	6.67E-02	0.00E+00	0.00E+00	0.00E+00	3.33E-02
f12	6.55E-14	5.74E-15	1.15E-14	4.01E-11	4.44E-15	4.32E-15	4.44E-15	4.44E-15	4.44E-15	4.44E-15	4.44E-15
f13	1.49E-02	2.00E-02	1.84E-02	1.51E-02	1.56E-02	1.41E-02	1.58E-02	1.11E-02	1.25E-02	1.40E-02	1.21E-02
f14	8.26E-10	5.99E-15	4.18E-11	1.21E-09	2.13E-11	3.31E-12	1.60E-17	6.85E-17	7.68E-12	9.44E-15	2.88E-15

MSACS makes each individual group into each strategy adaptively according to the performance of each strategy, and adjusts the probability of each strategy being used, so that MSACS has both good search ability and fast convergence, and shows good characteristics in many functions.

E. ANALYSIS OF THE STRATEGY POSSIBILITY

In order to study the adaptive selection of different strategies in MSACS algorithm, the selection probability of five strategies with the number of iterations was simulated, as shown in figure 4.

Figure 4 are the results of MSACS running on f7, f12, f19 and f24. Among them, strategy 1, Levy's flight strategy, is more likely to be used in f11 and f20. It also performs well in the first 100 generations of f6 and f19, but it is around 0.2 after 100 generations. Strategies 3 and 5 are more likely to be used on f6 and f19. Strategy 2, as a search strategy, remains around 0.2 in all functions. Strategy 4, as a search strategy, has a small probability in the process of rapid convergence of the function at the beginning, but after the function converges, the probability increases and gradually returns to 0.2, which makes the function have the ability to jump out of the local minimum after falling into it. To some extent, Strategy 2-5 make up for the shortcomings of Levy flight strategy in original CS algorithm and improve the performance of the algorithm.

F. ANALYSIS OF THE STRATEGY POSSIBILITY

Different length of memory bank will affect the results of MSACS algorithm. Using different memory lengths on different functions will yield different results. In this chapter, the memory bank length is changed from 10 to 60 with an interval of 5, and the experiment is repeated for 30 times for each memory bank length. The results of the experiment are recorded in table 8 in appendix.

It can be seen from table 8 that in f1-f3 and f6-f13, the changes of memory Banks have little influence on the results. In f4, a shorter memory bank can help MSACS find better solutions; In f5, a longer memory bank can help MSACS find better solutions; In f14, MSACS can find better solutions when the memory bank length is between 40 and 45.

This shows that, for most functions, the memory bank length has little influence on the final iteration result of the algorithm, but if the time required by the algorithm is taken into account, the minimum memory bank length should be selected to reduce the work of computer.

V. CONCLUSION

A MSACS algorithm is proposed in this paper. Five search strategies with different characteristics were used to replace Levy's flight strategy, so as to diversify individual search modes. The adaptive strategy is adopted. According to the play of each strategy in different functions and different stages of iteration. Selecting more appropriate strategies adaptively to make the use of strategies more reasonable. The adaptive strategy is used to determine the parameters of the algorithm, so that the size of the parameters is changed from pre-determined to according to their own needs. This adaptive strategy enhances the search ability of CS algorithm.

Through simulation experiments and surface analysis, compared with the other 11 algorithms, the multi-strategy adaptive algorithm proposed in this paper makes CS algorithm have strong optimization ability and can find effective solutions to the problem to be solved. Based on the results of running on 24 benchmark functions, it can be concluded that MSACS can effectively solve various optimization problems and significantly improve the performance of the original CS algorithm.

APPENDIX. A

See Table 3–8.

REFERENCES

[1] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, Dec. 2007, pp. 1942–1948.
[2] B. Cao *et al.*. "Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization," *IEEE Access*, vol. 5, pp. 8214–8221, 2017.
[3] S. Bououden, M. H. R. Chadli, and Karimi, "An ant colony optimization-based fuzzy predictive control approach for nonlinear processes," *Inf. Sci.*, vol. 299, pp. 143–158, Apr. 2015.
[4] W. Zhang *et al.*, "An improved ant colony algorithm for path planning in one scenic area with many spots," *IEEE Access*, vol. 5, pp. 13260–13269, 2017.
[5] D. Dagan, "Information processing in the immune system," in *New Ideas in Optimization*. New York, NY, USA: McGraw-Hill, 1999.
[6] I. Fister *et al.*, "A comprehensive review of firefly algorithms," *Swarm Evol. Comput.*, vol. 13, pp. 34–46, Dec. 2013.
[7] J. Huang, X. Chen, and D. Wu, "A switch-mode firefly algorithm for global optimization," *IEEE Access*, vol. 6, pp. 54177–54184, 2018.
[8] A. D. Coley, "An Introduction to Genetic Algorithms for Scientists and Engineers," *Bull. Volcanological Soc. Japan*, vol. 12, pp. 90–91, Jan. 2015.
[9] J. Meena, M. Kumar, and M. Vardham, "Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint," *IEEE Access*, vol. 4, pp. 5065–5082, 2016.
[10] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[11] Y. X. Lei *et al.*, "Improved differential evolution with a modified orthogonal learning strategy," *IEEE Access*, vol. 5, pp. 9699–9716, 2017.
[12] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.
[13] X. S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, 2013.
[14] X. S. Yang and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 169–174, Jan. 2014.
[15] A. B. Mohamad, A. M. Zain, and N. E. N. Bazin, "Cuckoo search algorithm for optimization problems–A literature review and its applications," *Appl. Artif. Intell.*, vol. 28, no. 5, pp. 419–448, 2014.
[16] S. E. K. Fateen and A. Bonilla-Petriciolet, "Unconstrained Gibbs free energy minimization for phase equilibrium calculations in nonreactive systems, using an improved cuckoo search algorithm," *Ind. Eng. Chem. Res.*, vol. 53, no. 26, pp. 10826–10834, 2014.
[17] S. E. K. Fateen and A. and Bonillapetriciolet, "On the effectiveness of nature-inspired metaheuristic algorithms for performing phase equilibrium thermodynamic calculations," *Sci. World J.*, vol. 1, Aug. 2014, Art. no. 374510.
[18] R. A. Vazquez, "Training spiking neural models using cuckoo search algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2011, pp. 4–5.
[19] G. Akemi, I. Andres, and C. Luis, "Cuckoo search with Lévy flights for weighted Bayesian energy functional optimization in global-support curve data fitting," *Sci. World J.*, vol. 2014, pp. 1–11, May 2014.
[20] S. Walton, O. Hassan, K. Morgan, and M. R. Brown, "Modified cuckoo search: A new gradient free optimisation algorithm," *Chaos Soliton Fract.*, vol. 44, no. 9, pp. 710–718, Aug. 2011.
[21] Y. Zhang, L. Wang, and Q. Wu, *Modified Adaptive Cuckoo Search (MACS) Algorithm and Formal Description for Global Optimisation*. Geneva, Switzerland: Inderscience, 2012.
[22] J. Wang and B. Zhou, "A hybrid adaptive cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1511–1517, May 2016.
[23] L. Wang, Y. Zhong, and Y. Yin, "Nearest neighbour cuckoo search algorithm with probabilistic mutation," *Appl. Soft Comput.*, vol. 49, pp. 498–509, Dec. 2016.
[24] L. Huang *et al.*, "Chaos-enhanced Cuckoo search optimization algorithms for global optimization," *Appl. Math. Model.*, vol. 40, no. 5, pp. 3860–3875, Mar. 2015.
[25] N. J. Cheung, X. M. H. B. Ding, and Shen, "A nonhomogeneous cuckoo search algorithm based on quantum mechanism for real parameter optimization," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 1–12, Feb. 2016.
[26] I. S. Boushaki and K. A. Nadjet, "A new quantum chaotic cuckoo search algorithm for data clustering," *Expert Syst. Appl.*, vol. 96, pp. 358–372, Aug. 2018.
[27] B. H. Abed-Alguni and F. Alkhateeb, "Novel selection schemes for cuckoo search," *Arabian J. Sci. Eng.*, vol. 42, no. 8, pp. 3635–3654, Aug. 2017.
[28] A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Singh, "Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3538–3560, 2014.
[29] Z. Bayraktar, M. Komurcu, J. A. Bossard, and D. H. Werner, "The wind driven optimization technique and its application in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 61, no. 5, pp. 2745–2757, May 2013.
[30] X. Li and M. Yin, "A particle swarm inspired cuckoo search algorithm for real parameter optimization," *Soft Comput.*, vol. 20, no. 4, pp. 1389–1413, 2016.
[31] X. Liu and M. Fu, "Cuckoo search algorithm based on frog leaping local search and chaos theory," *Appl. Math. Comput.*, vol. 266, pp. 1083–1092, Jul. 2015.
[32] X. Li and M. Yin, "Modified cuckoo search algorithm with self adaptive parameter method," *Inf. Sci.*, vol. 289, pp. 80–97, Mar. 2015.
[33] U. Mlakar and I. Fister, "Hybrid self-adaptive Cuckoo search for global optimization," *Swarm Evol. Comput.*, vol. 29, pp. 47–72, Aug. 2016.
[34] J. E. Smith and T. C. Fogarty, "Operator and parameter adaptation in genetic algorithms," *Soft Comput.*, vol. 1, no. 2, pp. 81–87, 1997.
[35] M. Yang *et al.*, "Differential evolution with auto-enhanced population diversity," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 302–315, Jul. 2017.
[36] Y. Long *et al.*, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1798–1810, Sep. 2015.
[37] J. Brest and E. al, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Jul. 2006.
[38] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
[39] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with Interswarm interactive learning strategy," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2238–2251, Oct. 2015.
[40] J. Sun, "Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection," *Evol. Comput.*, vol. 20, no. 3, pp. 349–393, 2014.

SHUZHI GAO was born in Henan, China, in 1968. She received the B.S. degree from the Shenyang University of Chemical Technology, in 1989, and the Ph.D. degree from Northeastern University, in 2012. She is currently a Professor with the Shenyang University of Chemical Technology and also a Distinguished Professor of Liaoning Province.

YUE GAO was born in Jiangsu, China, in 1996. He received the bachelor's degree from the School of International Education, Sanjiang University, in 2018. He is currently pursuing the master's degree in control engineering with the Shenyang University of Chemical Technology. His research interests include intelligent optimization algorithms, mechanical reliability, and optimized design of rolling bearings.

YIMIN ZHANG was born in Jilin, China, in 1958. He received the B.S., M.S., and Ph.D. degrees from the Department of Mechanics, Jilin University, Changchun, China, in 1982, 1989, and 1995, respectively. He was a Postdoctoral Researcher with Northeastern University, Shenyang, China, in 1998. He is currently a Professor with the Shenyang University of Chemical Technology, a Doctoral Tutor, a Distinguished Professor of the Changjiang Scholar, and a Climbing Scholar at the Liaoning Provincial Higher Education Institute.

LINTAO XU was born in Hebei, China, in 1994. He received the bachelor's degree in engineering from the Central University of Information Technology, Tianjin University of Technology, in 2017. He is currently pursuing the master's degree in control engineering with the Shenyang University of Chemical Technology. His research interests include intelligent optimization algorithms, mechanical reliability, and optimized design of rolling bearings.