

PAPER

Enhancing the Performance of Cuckoo Search Algorithm with Multi-Learning Strategies

Li HUANG^{†,††a}, Xiao ZHENG^{†††b}, Shuai DING^{†c}, Nonmembers, Zhi LIU^{††††d}, Member,
and Jun HUANG^{†††e}, Nonmember

SUMMARY The Cuckoo Search (CS) is apt to be trapped in local optimum relating to complex target functions. This drawback has been recognized as the bottleneck of its widespread use. This paper, with the purpose of improving CS, puts forward a Cuckoo Search algorithm featuring Multi-Learning Strategies (LSCS). In LSCS, the Converted Learning Module, which features the Comprehensive Learning Strategy and Optimal Learning Strategy, tries to make a coordinated cooperation between exploration and exploitation, and the switching in this part is decided by the transition probability P_c . When the nest fails to be renewed after m iterations, the Elite Learning Perturbation Module provides extra diversity for the current nest, and it can avoid stagnation. The Boundary Handling Approach adjusted by Gauss map is utilized to reset the location of nest beyond the boundary. The proposed algorithm is evaluated by two different tests: Test Group A (ten simple unimodal and multimodal functions) and Test Group B (the CEC2013 test suite). Experiments results show that LSCS demonstrates significant advantages in terms of convergence speed and optimization capability in solving complex problems.

key words: multi-learning strategy, cuckoo search, bound handling mechanism, elite learning

1. Introduction

In tackling objective functions which are distinguishable and continuous, especially those with large size characteristics, metaheuristics-based algorithms are usually justified as a robust and powerful solution, and this kind of algorithm is generally believed to be superior to the traditional optimization. Inspired by the unique characteristics of the cuckoos, the standard Cuckoo Search (CS) algorithm [1] is categorized as metaheuristics. Its rules state that each host has a probability of discovering an alien egg, and if the probability is greater than a switching parameter P_a , the host will find the alien egg and will build a new nest in a new location.

Different from such metaheuristic algorithms as the genetic algorithm (GA) [2], [3] and the particle swarm optimization (PSO) [4]–[6], the standard Cuckoo Search (CS) algorithm has distinguished itself with its reference to the Lévy Flight which features a heavy-tailed probability distribution, and which application in the global random walk makes the algorithm more effective. In solving a unimodal function, the global random walk of the CS algorithm will quicken speed at which each nest moving to the global optimal nest. However, in solving more complex functions, the global random walk is at the risk of excessive exploitation [7], which occurs when a nest loses its global exploration capacity within a few iterations, and thus makes the standard CS algorithm be prone to premature convergence, and makes nest baffled in pursuing local optimal [8]. Thus, the standard CS algorithm is prone to premature convergence, and makes nest baffled in pursuing local optima.

To improve the performance of the standard CS algorithm, various modified Cuckoo search algorithms have been proposed. With a new orthogonal learning strategy, Xi-angtao Li proposed the Orthogonal Learning Cuckoo Search Algorithm (OLCS) [9] in 2014, which has been proved to be able to enhance the exploitation ability of the basic cuckoo search. In 2015, he proposed that the self-adaptive cuckoo search algorithm (SACS) [10], which is based on the rand and best individuals among the entire population, turns to a linear decreasing probability rule to balance two new mutation rules. Li Huang [11] improved three parts of the standard CS algorithm, i.e., chaotic initial position, variable step size of Lévy Flight, and chaos transboundary treatment. Liu Xiaoying [12] introduced the inertia weight factor of the particle swarm algorithm into the Levy flight, and adopted the leapfrog algorithm in the local search mechanism of the standard CS algorithm. Seen in the light of No Free Lunch Theorem [13], any existing optimization strategy can hardly solve all kinds of problems. Inspired by this theoretical assumption, this paper attempts to study on the cuckoo search algorithm with Multi-learning Strategies (LSCS), which is expected to be more effective in finding the optimal value of complex functions, compared with the standard CS algorithm.

Manuscript received January 15, 2019.

Manuscript revised May 31, 2019.

Manuscript publicized July 9, 2019.

[†]The authors are with School of Management, Hefei University of Technology, Hefei 230009, Anhui, P.R. China.

^{††}The author is with School of Management Science and Engineering, Anhui University of Technology, Maanshan, P.R. China.

^{†††}The authors are with the School of Computer Science and Technology, Anhui University of Technology, China.

^{††††}The author is with the Department of Mathematical and Systems Engineering, Shizuoka University, Hamamatsu-shi, 432–8561 Japan.

a) E-mail: hli0902@126.com (Corresponding author)

b) E-mail: xzheng@ahut.edu.cn (Corresponding author)

c) E-mail: dingshuai@hfut.edu.cn

d) E-mail: liu@shizuoka.ac.jp

e) E-mail: huangjun.cs@163.com

DOI: 10.1587/transinf.2019EDP7013

2. The Standard Cuckoo Search Algorithm (CS)

By imitating the breeding behavior of Cuckoo in the D -dimensional search space, the standard CS algorithm, to im-

prove its effectiveness, adopts two moving strategies: the local random walk and the global random walk. The former is exploration, which is derived from the differential evolution [14]–[16] and is designed to ensure the diversity of the location of nests in the searching space. And the latter is exploitation, which is expected to accelerate the process of making all the location of nests near-optimal solutions. According to the standard CS algorithm, the action of moving the nest is conducted by acting up to the following two formulas.

The formula (1) is the local random walk, the j th dimension of host i is updated as follows:

$$nest(t + 1)_i^j = nest(t)_i^j + rand * (nest(t)_{r_1}^j - nest(t)_{r_2}^j), \quad rand < P_a \quad (1)$$

where $nest(t)_i^j$ is the j th dimension position of the i th nest, $nest(t)_i = (nest(t)_{r_1}^1, nest(t)_{r_1}^2, \dots, nest(t)_{r_1}^D)$, and they are real-valued vectors, that is n number of randomly generated D -dimensional; where $i = 1, 2, \dots, n, j = 1, 2, \dots, D$. $nest(t)_{r_1}^j$ and $nest(t)_{r_2}^j$ represent two randomly-selected nests. t is the current iteration number. To ensure a probability of 25% of crossover mutation for each dimension of each nesting, P_a is the switching parameter which is set to be 0.25. $rand$ is a uniformly distributed random number within the range of (0,1).

The formula (2) is the global random walk, the j th dimension of host i is updated as follows:

$$nest(t + 1)_i^j = nest(t)_i^j + \alpha \times (nest(t)_i^j - bestnest(t)^j) \oplus Levy(\lambda), \quad rand \geq P_a \quad (2)$$

where, $nest(t + 1)_i^j$ denotes the j th dimension position of the i th nest through the $t + 1$ iterations. $bestnest(t) = (bestnest(t)^1, bestnest(t)^2, \dots, bestnest(t)^D)$ is the history of the best location through the t iterations by the whole population, where $j = 1, 2, \dots, D$. \oplus is entry-wise multiplication. The step size coefficient α [17] is a constant over zero. This value varies in different cases, but $\alpha = 0.01$ in general. And $Lévy(\lambda)$ [17] is the Lévy distribution. The Lévy flight behavior has been applied to the optimum random search, and it shows a good performance [18].

3. The Cuckoo Search Algorithm with Multi-Learning Strategies (LSCS)

By turning to exploration and exploitation, the standard CS algorithm is used to solve optimization problems. Since these two ways are contradictory, a carefully-designed coordination is indispensable so as to avoid the possible "excessive exploitation". Therefore, a sound coordination between them will be expected to enhance the performance of the search algorithm.

In LSCS, our design includes: 1) the Converted Learning Module which aims at making a coordinated cooperation between exploration and exploitation; 2) the Elite Learning Perturbation Module which is supposed to alleviate pre-

mature convergence. 3) the new way of boundary setting. Set on Boundary [19], which is regarded as a drawback of the Bound Handling Mechanism [20], [21] of standard CS, stops the game violators and resets nest to boundary values. And this drawback will hopefully be overcome by the Boundary Handling Approach adjusted by Gauss map adopted in the LSCS. The pseudo codes of the proposed LSCS search procedure given in this section are listed as follow:

LSCS optimization algorithm:

```

1 counter  $N$ , switching parameter  $P_a = 0.25$ ,
2 dimension number  $D$ , the population  $n$ ; refreshing gap  $m$ 
3 Initialize host nest location;
4 For  $i = 1:n$  do
5     evaluating all new solutions;
6     find the current  $bestnest$ ;
7 End for

8 While termination condition is not satisfied ( $N$ ) do,
9     generating chaotic sequence  $cc$ ;
Converting learning strategies module
10    generating the transition probability  $P_c$  using formula (5);
11    If  $rand < P_c$ ,
12        goto tournament selection procedure;
13        choose the  $nest(t)_i^j$ ;
14        update  $nest_{t+1}(i, j)$  using formula (3);
15    Else if  $rand \geq P_c$ ,
16        update  $nest_{t+1}(i, j)$  using formula (4);
17    End if
18    evaluating all new solutions of the nest location;
19    find the current  $bestnest$ ;
20    Boundary Handling Approach adjusted by Gauss map;

```

Local random walk

```

21 If  $rand < P_a$ ,
22    update  $nest_{t+1}(i, j)$  using formula (1);
23 End if
24 evaluating all new solutions of the nest location;
25 find the current  $bestnest$ ;
26 Boundary Handling Approach adjusted by Gauss map;

```

```

27 If the fitness of  $bestnest$  is not improved for  $m$  iterations;

```

Elite Learning Perturbation Module

```

28    update  $nest_{t+1}(i, j)$  using formula (6);
29    End if
30    evaluating all new solutions of the nest location;
31    find the current  $bestnest$ ;
32 End While

```

3.1 Convert Learning Module

Controlled by transition probability P_c [22], the Converted Learning Module of the LSCS turns to a balanced combination of Comprehensive Learning Strategy and Optimal Learning Strategy. Comprehensive Learning Strategy encourages the current nest to obtain information about location from other individuals so as to keep the diversity of nests. The Optimal Learning Strategy encourages current nest to learn the location of the best nest in each iteration. It can quickly find a local optimum and exploit the promising area.

The Comprehensive Learning Strategy is defined as:

$$\begin{aligned} nest(t+1)_i^j = nest(t)_i^j + cc * (nest(t)_i^j - nest(t)_r^j) \\ \oplus Lévy(\lambda), rand < P_c \end{aligned} \quad (3)$$

where $nest(t)_i^j$ denotes the j th dimension position of the i th nest through the t iterations. The step length coefficient cc is chaotic sequence generated by Gauss map. With reference to the literature [11], cc as [0.01, 0.3]. $rand$ is a uniformly distributed random number within the range of (0,1). P_c is the transition probability. Following the Tournament Selection Procedure [22], two randomly-chosen nests are compared in terms of the fitness, and the undesirable one will be selected as the exemplar ($nest(t)_r$) denoting the position of the r th nest through the t iterations, which is $nest(t)_r^j$ here.

We employ the tournament selection procedure [22] when the nests dimension learns from the exemplar ($nest(t)_r^j$) as follows. 1) we randomly single out two nests from the population which excludes the nest whose position is updated. 2) We compare the fitness values of these two nests and choose the r th nest with a larger fitness value as an exemplar. 3) The nest's j dimension learns from the exemplar ($nest(t)_r^j$). If all exemplars of a nest are its own, we will randomly choose one dimension to learn from another nest's exemplar's corresponding dimension.

The Optimal Learning Strategy is defined as:

$$\begin{aligned} nest(t+1)_i^j = nest(t)_i^j + cc * (nest(t)_i^j - bestnest(t)_i^j) \\ \oplus Lévy(\lambda), rand \geq P_c \end{aligned} \quad (4)$$

where $bestnest(t)_i^j$ denotes the history of the best location through the t iterations by the whole population, where $j = 1, 2, \dots, D$.

3.2 Transition Probability P_c

As explained in [23], different P_c values yielded different solutions on the same function if the same P_c value was used for all the nests. Thus, we developed transition probability P_c by referring to CLPSO [22]. We propose to set such that each nest has a different P_c value. Therefore, nests have different levels of exploration and exploitation ability in the population and are able to solve diverse problems. Formula governing the transition probability P_c of the Converted Learning Module :

$$P_c = 0.05 + 0.45 \times \frac{\exp(10(i-1)/n-1) - 1}{\exp(10) - 1} \quad (5)$$

where n denotes the number of nest, and i denotes the serial number of the nest, i th.

3.3 Elite Learning Perturbation Module

If the fitness of best nest fails to be improved after m iterations, nest falls into stagnation, and the algorithm will enter

the state of premature convergence. The Elite Learning Perturbation Module provides extra diversity to the current nest, so it may get rid of the stagnation. Taking the const and convergence of the algorithm into consideration, m , the value of the refreshing gap should be set moderately. Here it is set at 7 by referring to CLPSO [22] and ATLPSO-ELS [24].

The formula for Elite Learning Perturbation Module is expressed as follow:

$$bestnest(t+1)^j = \begin{cases} bestnest(t)^j + (Ub^j - Lb^j) \\ \oplus Lévy(\lambda), rand < P_i \\ bestnest(t)^j + (Lb^j - Ub^j) \\ \oplus Lévy(\lambda), rand \geq P_i \end{cases} \quad (6)$$

where $bestnest(t+1)^j$ denotes the history of the best location through the $t+1$ iterations by the whole population, where $j = 1, 2, \dots, D$. P_i stands for the direction probability, and it guides nest to fly to the direction of the optimal nest, $P_i = 0.5$. And $rand$ is a uniform random number ranging among [0, 1]. Lb^j and Ub^j are the maximum and minimum boundary values of the searching space respectively.

4. Experiments and Results

We test LSCS algorithm and the other meta-heuristics with two group functions: Test Group A (ten simple unimodal and multimodal functions, from f_1 and f_{10}) and Test Group B (the CEC2013 test suite, from f_{11} and f_{38}) [25]. The complexity of the functions from Test Group A to B increases gradually to examine the performance of the LSCS algorithm.

4.1 Test Group A: Ten Simple Problems

As is shown in Table 1, the unimodal function (f_1, f_2, f_3 and f_4) contains only one optimum, and their properties are scalable and separable. The multimodal functions (f_5, f_6, f_7, f_8, f_9 and f_{10}) have a lot of local optimums, but there is only one global optimum. The global optimum of these ten benchmark functions is $f(x) = 0$, and x is the location of the global optimal solution.

4.1.1 Test Group A: Experimental Setup

Test Group A is used in comparing the performance of the standard CS and that of LSCS, and the dimensions of functions are set at 30 and 50 respectively, and each algorithm runs 30 times. Besides, the population of each algorithm is set at 40. The parameters of the two algorithms are set as follows: 1) Standard CS: The probability switching parameter $P_a = 0.25$; 2) LSCS: The probability switching parameter $P_a = 0.25$; Refreshing gap $m = 7$.

4.1.2 The Average (Mean) and Standard Deviation (SD) of Test Group A

Table 2 shows the experimental results of CS and LSCS in

Table 1 Test Group A: simple unimodal and multimodal functions

Formula	Domain
$f_1(x) = \sum_{i=1}^n x_i^2$	[-100,100]
$f_2(x) = \sum_{i=1}^n (ix^2)$	[-100,100]
$f_3(x) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$	[-100,100]
$f_4(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-10,10]
$f_5(x) = -20 \exp \left[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right] + (20 + e)$	[-32,32]
$f_6(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]
$f_7(x) = \sum_{i=1}^n x_i^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i\right)^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i\right)^4$	[-100,100]
$f_8(x) = \sum_{i=1}^n \left[\sum_{k=0}^{\lfloor k \max \right]} [a^k \cos(2\pi b^k (x_i + 0.5))] \right]$	
$-n \sum_{k=0}^{\lfloor k \max \rfloor} [a^k \cos(2\pi b^k * 0.5)]$	[-0.5,0.5]
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]
$f_{10}(x) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i-1})^2 + (x_i - 1)^2)$	[-30,30]

terms of the Average(Mean) and Standard Deviation (SD) of different solutions. The best results are shown in bold face. The results suggest that the LSCS solves the function with 30 dimensions much easier than it solves the function with 50 dimensions. The reason is that the more the dimension is, the more complex the function will be, and the more difficult the seeking of the global optimum will be. Since the step size of the random walk of the original CS algorithm is fixed, from the function f_1 to f_4 , the optimization ability of the original CS is worse than that of the LSCS algorithm. In solving multimodal functions (From the function f_5 to f_{10}), the optimizing capacity of the LSCS is better than that of the CS algorithm. Noticeably, to Rastrigin's function(f_9), the LSCS optimization algorithm can obtain the global optimal solution,0.Compared with the original algorithm, the Converted learning module of the LSCS optimization algorithm can randomly traverse to a larger search space to find the global optimal solution to the function.

4.1.3 Test Group A: The Convergence Tests

Figure 1 presents the convergence progress of two algorithms (CS and LSCS) on a single run of 10000 iterations, and the dimension of nest (D) is 50. As for the results of the unimodal Problems, Fig. 1(from subfigure (a) to subfigure (d)) suggests that the LSCS algorithm with Converted learning module performs better than the standard algorithm in terms of search and converge capability. Especially, in solving Schwefels P2.22(Fig. 1-subfigure (d)), the performance of LSCS algorithm is particularly prominent. The results of the multimodal problems are presented in Fig. 1 (Fig. 1 or Fig. 1) (from sub-figure(e) to subfigure(j)) which suggests (suggest) a better reliability and a more desirable stability of LSCS over CS. Nevertheless, the result of the Rosebrock function(f_{10}) shown in subfigure (j) of Fig. 1, after the last 10000 iterations, the convergence rate of the LSCS is lower

Table 2 The mean and SD of Test Group A

30D		f_1	f_2	f_3
CS	Mean	2.91E-16	3.18E-15	2.90E-13
	SD	3.08E-16	3.45E-15	3.20E-13
LSCS	Mean	3.47E-35	2.20E-34	2.26E-32
	SD	5.82E-35	4.33E-34	5.57E-32
30D		f_4	f_5	f_6
CS	Mean	3.09E-07	1.08E-01	2.43E-07
	SD	1.37E-07	4.68E-07	9.35E-07
LSCS	Mean	3.51E-21	5.51E-15	7.38E-03
	SD	3.90E-21	1.66E-15	8.56E-03
30D		f_7	f_8	f_9
CS	Mean	6.58E-13	4.79E+01	6.39E-01
	SD	8.21E-13	7.99E+00	2.41E-01
LSCS	Mean	5.35E-31	9.95E-02	0.00E+00
	SD	7.78E-31	4.01E-01	0.00E+00
30D		f_{10}		
CS	Mean	1.67E+01		
	SD	4.09E+00		
LSCS	Mean	3.22E+01		
	SD	2.94E+01		
50D		f_1	f_2	f_3
CS	Mean	2.65E-17	4.18E-16	4.35E-14
	SD	2.34E-17	3.03E-16	2.72E-14
LSCS	Mean	1.09E-27	2.44E-26	4.90E-25
	SD	1.38E-27	3.87E-26	4.63E-25
50D		f_4	f_5	f_6
CS	Mean	1.24E-08	6.79E-01	2.43E-07
	SD	5.17E-09	2.22E+00	9.35E-07
LSCS	Mean	4.48E-17	1.65E-14	2.13E-03
	SD	2.90374E-17	5.94E-15	5.08E-03
50D		f_7	f_8	f_9
CS	Mean	1.09E-11	8.05E+01	1.06E+00
	SD	1.33E-11	9.73E+00	0.182
LSCS	Mean	1.62E-22	5.18E+01	0.00E+00
	SD	1.9817E-22	3.45E+01	0.00E+00
50D		f_{10}		
CS	Mean	4.60E+01		
	SD	1.98E+01		
LSCS	Mean	6.68E+01		
	SD	3.76E+01		

than that of the standard CS. To find the optimal solution to the Rosebrock, the LSCS requires a certain number of iterations for the sake of particle diversity. In solving Ackley(f_5), Griewangk(f_6), Weierstrass(f_8) and Rastrigin(f_9) functions, the LSCS optimization algorithm quickly finds the global optimal solution($f(x) = 0$), because the application of the Multi-Learning Strategies of the LSCS optimization algorithm has improved the searching capability of the nests.

4.2 Test Group B: The CEC2013 Test Suite

The previous analysis of Test Group A shows that the performance of the proposed LSCS is significantly better than that of the standard CS algorithm.In the following study, we use more complex benchmark functions (Group B: CEC2013 test suite) to examine the performance of the LSCS.The CEC2013 test suite consists of five Unimodal functions(from f_{11} to f_{15}), fifteen Basic Multimodal functions(from f_{16} to f_{30}) and eight Compositions functions(from f_{31} to f_{38}), and they are more complex with rotation and displacement characteristics of the functions.

4.2.1 Test Group B: Experimental Setup

We compare the proposed algorithm LSCS with five approaches, i.e., CS, TCP SO [26], SACS [10], CCS3 [11] and

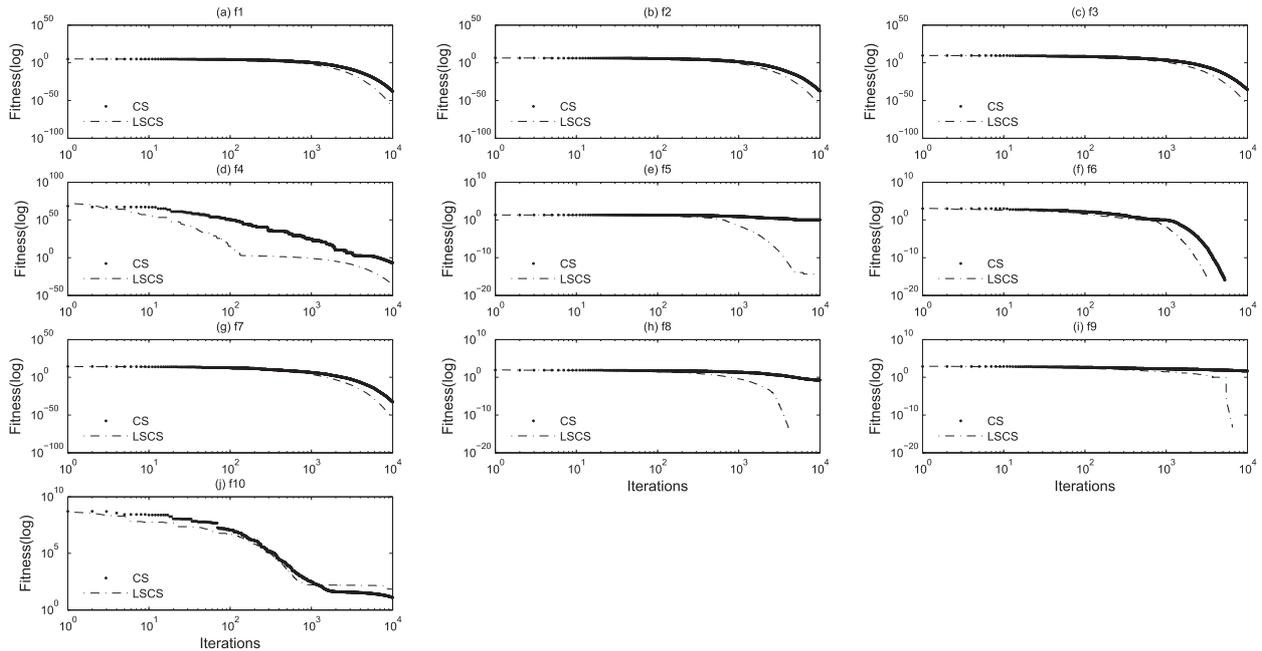


Fig. 1 Convergence performance of the CS and LSCS on Test Group A.

CLPSO [22]. Comparative experiments are conducted within Test Group B. The functions of Test Group B are set at 30 and 50 respectively, and the iterations of the corresponding algorithm are set at 30000 and 50000 accordingly. The population of each algorithm is 40 and they run 30 times respectively. Parameter configurations of the comparing algorithms are listed below: 1) Standard CS: The probability switching parameter $P_a = 0.25$; 2) TCPSO: $l = 2, w^M = 0.9, c_1^s = c_2^s = c_1^M = c_2^M = c_3^M = 1.6$; 3) SACS: The probability switching parameter $P_a = 0.25$; 4) CCS3: The probability switching parameter $P_a = 0.25$; 5) CLPSO: $w_0 = 0.9, w_1 = 0.4, c = 1.49445$, Refreshing gap $m = 7$; 6) LSCS: The probability switching parameter $P_a = 0.25$; Refreshing gap $m = 7$;

4.2.2 Performance Evaluation of Group B

The means and standard deviations of the algorithms on the CEC2013 test suite with 30D problems are shown in Table 3. The CCS3 algorithm and LSCS algorithm have demonstrated a better performance on five Unimodal functions (from f_{11} and f_{15}). For f_{11} and f_{15} , the CCS3 and LSCS algorithms can find the optimal solution(0). Among the 15 basic multimodal functions, LSCS algorithm performs the best within 8 functions (namely, $f_{16}, f_{18}, f_{19}, f_{21}, f_{23}, f_{24}, f_{25}$ and f_{27}). Among the 8 compositions functions (from f_{31} to f_{38}), the LSCS algorithm has performed the best. Because the Elite Learning Perturbation Module prevents the *bestnest* from being trapped in the local optima, if the fitness of the *bestnest* is not improved for m successive fitness. And different the transition probability P_c values yield the best performance for different 8 compositions functions. In order to compare the performance of the six algorithm, we also make

another set of experiments which is the CEC2013 test suite with 50D. According to the results of the 50D problems shown in Table 4, our proposed method LSCS still achieves the best or better performance against other comparing algorithms over most of the test functions.

4.2.3 Test Group B: The Non-Parametric Wilcoxon's Rank Sum Test

Wilcoxon's rank sum test returns p -Value and z . The p -Value represents the minimal significance level for detecting differences. If the p -Value is less than 0.05, it means that, the better result achieved by the best algorithm is statistically significant in each case, and it is not obtained incidentally. Table 4 shows the non-parametric Wilcoxon's rank sum test of the algorithms on the CEC2013 test suite with 30D problems. For the 30D problems, in function(s) f_{11} and f_{15} (Unimodal), function f_{30} (Multimodal) and functions f_{33}, f_{36} and f_{38} (Composition), p -Value obtained through Wilcoxon's rank sum test are bigger than 0.05. About Unimodal Functions (f_{11} and f_{15}) and Composition Functions (f_{38}), optimal solutions of the LSCS and CS are two independent samples, which have the same p -value, 1. On the whole, in the CEC2013 test suite, LSCS achieves significant better performance against CS.

5. Conclusion

The global random walk of the standard CS algorithm is at the risk of excessive exploitation, and when it occurs nests lose the global exploration capacity and thus makes the algorithm prone to premature convergence. In this paper, we propose a cuckoo search algorithm with Multi-learning

Table 3 The result of CEC2013 functions with 30D, MaxFES:30000

Algo		f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}
CS	Mean	0.00E+00	4.87E+05	5.47E+05	3.05E+02	0.00E+00	4.45E-02	9.37E+01
	SD	0.00E+00	1.27E+05	8.82E+05	5.57E+01	0.00E+00	5.82E-02	1.38E+01
TCPSO	Mean	1.14E-12	2.65E+06	4.44E+08	1.87E+03	7.84E-13	3.33E+01	1.16E+02
	SD	1.36E-12	1.45E+06	4.38E+08	8.02E+02	6.92E-13	2.18E+01	2.89E+01
SACS	Mean	0.00E+00	3.46E+06	4.81E+07	1.36E+03	1.14E-13	1.32E+01	4.98E+01
	SD	0.00E+00	1.21E+06	2.77E+07	3.45E+02	0.00E+00	4.01E+00	1.06E+01
CCS3	Mean	0.00E+00	3.85E+05	1.05E+05	2.83E+00	0.00E+00	3.65E-02	6.13E+01
	SD	0.00E+00	1.43E+05	2.11E+05	2.59E+00	0.00E+00	3.83E-02	3.40E+01
CLPSO	Mean	2.05E-13	6.11E+06	1.06E+08	5.55E+03	1.67E-13	1.50E+01	6.26E+01
	SD	6.94E-14	1.78E+06	4.71E+07	1.19E+03	5.77E-14	4.18E+00	1.16E+01
LSCS	Mean	0.00E+00	3.40E+05	8.05E+05	4.35E+04	0.00E+00	3.35E-04	6.59E+01
	SD	0.00E+00	9.27E+05	1.61E+06	3.33E+04	0.00E+00	3.12E-04	2.48E+01
Algo		f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}
CS	Mean	2.08E+01	2.80E+01	5.35E-04	8.95E-01	1.17E+02	1.56E+02	6.02E+02
	SD	6.61E-02	1.92E+00	2.23E-03	3.04E-01	2.38E+01	3.10E+01	1.79E+02
TCPSO	Mean	2.09E+01	2.73E+01	4.45E-01	8.34E+01	2.17E+02	2.29E+02	1.37E+03
	SD	7.23E-02	3.11E+00	6.41E-01	2.49E+01	5.50E+01	2.57E+01	4.12E+02
SACS	Mean	2.09E+01	2.76E+01	1.50E+00	0.00E+00	1.01E+02	1.36E+02	7.07E-01
	SD	5.67E-02	2.05E+00	2.52E-01	0.00E+00	1.68E+01	1.39E+01	6.97E-01
CSS3	Mean	2.09E+01	2.42E+01	5.01E-03	0.00E+00	7.75E+01	1.20E+02	2.22E+02
	SD	4.75E-02	1.94E+00	4.54E-03	0.00E+00	1.20E+01	1.67E+01	1.52E+02
CLPSO	Mean	2.09E+01	2.25E+01	8.85E-01	1.25E-13	7.28E+01	1.23E+02	4.87E+02
	SD	5.55E-02	1.72E+00	2.81E-01	9.49E-14	1.22E+01	2.23E+01	1.18E+02
LSCS	Mean	2.07E+01	2.21E+01	6.45E-02	0.00E+00	8.90E+01	1.18E+02	1.81E-01
	SD	9.88E-02	5.167398	4.02E-02	0.00E+00	2.99E+01	26.536110	5.05E-02
Algo		f_{25}	f_{26}	f_{27}	f_{28}	f_{29}	f_{30}	f_{31}
CS	Mean	4.03E+03	2.07E+00	5.63E+01	1.67E+02	4.79E+00	1.19E+01	2.04E+02
	SD	2.75E+02	2.74E+00	7.28E+00	2.24E+01	5.73E-01	2.55E-01	4.62E+01
TCPSO	Mean	5.38E+03	1.64E+00	9.93E+01	2.42E+02	8.09E+00	1.50E+01	3.34E+02
	SD	1.13E+03	4.03E-01	2.17E+01	4.90E+01	3.12E+00	0.00E+00	1.02E+02
SACS	Mean	3.64E+03	1.45E+00	3.04E+01	1.86E+02	2.31E+00	1.14E+01	2.50E+02
	SD	2.53E+02	2.17E-01	7.42E-03	1.23E+01	3.17E-01	3.43E-01	4.95E+01
CSS3	Mean	3.81E+03	1.01E+00	3.89E+01	1.25E+02	3.31E+00	1.29E+01	2.33E+02
	SD	3.00E+02	1.36E-01	3.43E+00	1.44E+01	6.38E-01	1.85E+00	4.79E+01
CLPSO	Mean	3.18E+03	6.03E-01	3.21E+01	1.03E+02	7.64E-01	1.41E+01	1.62E+02
	SD	3.12E+02	1.39E-01	3.10E+00	9.73E+00	2.51E-01	8.00E-01	4.76E+01
LSCS	Mean	4.43E-01	1.74E+00	3.01E+01	1.36E+02	9.01E-01	1.21E+01	3.89E+02
	SD	2.68E-01	5.77E-01	1.19E-01	3.41E+01	2.73E-01	9.22E-01	8.19E+01
Algo		f_{32}	f_{33}	f_{34}	f_{35}	f_{36}	f_{37}	f_{38}
CS	Mean	8.42E+02	4.61E+03	2.78E+02	2.96E+02	2.00E+02	1.02E+03	3.00E+02
	SD	2.62E+02	4.80E+02	4.97E+00	3.37E+00	7.07E-03	2.08E+02	0.00E+00
TCPSO	Mean	1.55E+03	5.99E+03	2.70E+02	2.72E+02	3.30E+02	1.00E+03	5.19E+02
	SD	4.69E+02	9.66E+02	9.82E+00	9.77E+00	7.34E+01	7.69E+01	5.62E+02
SACS	Mean	1.76E+02	4.25E+03	2.68E+02	2.88E+02	2.00E+02	1.00E+03	3.00E+02
	SD	2.95E+01	4.41E+02	8.34E+00	5.77E+00	4.93E-02	5.36E+01	0.00E+00
CCS3	Mean	2.82E+02	4.36E+03	2.44E+02	2.80E+02	2.00E+02	9.24E+02	2.93E+02
	SD	1.11E+02	1.16E+03	5.37E+01	2.32E+01	5.53E-03	1.37E+03	3.65E+01
CLPSO	Mean	4.30E+02	4.07E+03	2.38E+02	2.95E+02	1.97E+02	4.86E+02	1.61E+02
	SD	8.54E+01	5.11E+02	2.79E+01	5.59E+00	1.02E+01	1.98E+02	8.74E+01
LSCS	Mean	1.31E+02	4.40E+03	2.26E+02	2.77E+02	1.93E+02	7.92E+02	3.00E+02
	SD	6.93E+01	8.00E+02	19.70212	1.19E+01	17.41992	1.48E+02	0.00E+00

strategies (LSCS). In LSCS, the converted learning module tries to make a coordinated cooperation between exploration and exploitation, and the Elite Learning Perturbation Module alleviates the premature convergence.

The Boundary Handling Approach adjusted by Gauss map overcomes the drawback of Set on Boundary. The proposed algorithm is evaluated on Test Group A (ten simple unimodal and multimodal functions) and Test Group B (the CEC2013 test suite). And it is compared with the standard CS and some other well-known existing optimization algorithms. The experimental results indicate that LSCS substantially outperforms the standard CS algorithm in terms of convergence speed. Besides, LSCS can always find global optimal in the benchmark functions.

Future work may include the ability to add social learning to the nests of the LSCS algorithm, enabling information transfer between particles so as to find optimal solutions in complex environments. By using multi-population parti-

tioning, the nests of LSCS algorithm will be more heterogeneous, so the exploration ability will be more desirable. Nevertheless the more important question is how to apply the LSCS optimization algorithm in solving more complex optimization problems within the engineering domain.

Acknowledgments

This work is partially supported by the Anhui Province Philosophy and social sciences planning project under Grant No. AHSKZ2015D49, the University Humanities and social science Research Projects at the provincial level under Grant No. SK2018A0069, the Natural Science Foundation of the Educational Commission of Anhui Province of China under Grant No. KJ2018A0050, and the Key Research and Development Program of Anhui, China under Grant No. 201904a05020071.

Table 4 The result of CEC2013 functions with 50D, MaxFES:50000

Algo		f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}
CS	Mean	0.00E+00	2.10E+06	7.83E+06	1.58E+02	4.32E-14	4.34E+01	1.10E+02
	SD	0.00E+00	4.02E+05	5.91E+06	7.14E+01	5.63E-14	7.23E-15	8.91E+00
TCPSO	Mean	2.36E-05	1.01E+07	1.85E+09	6.29E+03	5.48E-07	7.14E+01	1.19E+02
	SD	8.37E-05	6.61E+06	1.42E+09	2.86E+03	2.99E-06	2.67E+01	2.56E+01
SACS	Mean	2.20E-13	9.19E+06	1.69E+09	2.92E+03	1.14E-13	4.40E+01	1.17E+02
	SD	4.15E-14	2.24E+06	8.11E+08	6.61E+02	0.00E+00	2.56E+00	1.13E+01
CCS3	Mean	0.00E+00	5.90E+06	3.53E+05	1.88E-02	9.85E-14	4.34E+01	9.09E+01
	SD	0.00E+00	7.02E+06	5.04E+05	2.76E-02	3.94E-14	1.44E-14	2.04E+01
CLPSO	Mean	2.27E-13	1.09E+07	3.86E+08	3.89E+03	2.43E-13	4.10E+01	8.88E+01
	SD	0.00E+00	2.32E+06	1.55E+08	6.83E+02	3.93E-14	6.18E+00	1.03E+01
LSCS	Mean	0.00E+00	1.52E+06	4.47E+06	1.43E+03	0.00E+00	2.05E+01	9.53E+01
	SD	0.00E+00	2.90E+06	3.70E+06	3.84E+02	0.00E+00	2.22E+01	1.88E+01

Algo		f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}
CS	Mean	2.11E+01	5.58E+01	1.87E-02	8.03E+00	3.27E+02	3.61E+02	1.09E+03
	SD	4.50E-02	1.99E+00	1.01E-02	6.82E+00	4.29E+01	2.97E+01	2.91E+02
TCPSO	Mean	2.11E+01	5.42E+01	2.44E+01	2.33E+02	4.23E+02	4.86E+02	3.51E+03
	SD	5.03E-02	4.49E+00	2.41E+01	5.15E+01	8.61E+01	7.28E+01	5.89E+02
SACS	Mean	2.11E+01	5.49E+01	8.06E+00	5.68E-14	3.41E+02	3.78E+02	2.24E+02
	SD	4.23E-02	3.15E+00	2.11E+00	0.00E+00	4.43E+01	4.50E+01	2.79E+01
CCS3	Mean	2.11E+01	5.48E+01	1.15E-01	9.49E-14	1.99E+02	3.27E+02	8.67E+02
	SD	3.07E-02	2.92E+00	2.14E-01	3.12E-14	5.67E+01	7.15E+01	3.15E+02
CLPSO	Mean	2.11E+01	4.59E+01	1.27E+00	6.15E-06	1.68E+02	2.81E+02	1.01E+03
	SD	1.32E-02	2.68E+00	2.10E-01	1.62E-05	2.16E+01	2.86E+01	1.19E+02
LSCS	Mean	2.10E+01	5.20E+01	8.36E-02	0.00E+00	1.87E+02	3.16E+02	3.75E-01
	SD	3.79E-02	1.11E+01	4.71E-02	0.00E+00	4.81E+01	4.49E+01	2.88E-01

Algo		f_{25}	f_{26}	f_{27}	f_{28}	f_{29}	f_{30}	f_{31}
CS	Mean	8.17E+03	1.79E+00	1.11E+02	3.99E+02	1.54E+01	2.14E+01	2.92E+02
	SD	4.32E+02	2.08E-01	8.29E+00	4.52E+01	2.96E+00	5.63E-01	2.81E+02
TCPSO	Mean	1.24E+04	2.24E+00	2.71E+02	5.74E+02	2.61E+01	2.38E+01	8.83E+02
	SD	1.05E+03	5.11E-01	5.25E+01	9.33E+01	7.34E+00	1.33E+00	3.36E+02
SACS	Mean	7.78E+03	2.44E+00	6.08E+01	4.53E+02	7.26E+00	2.15E+01	3.25E+02
	SD	5.31E+02	2.58E-01	1.61E+00	2.29E+01	6.37E-01	2.92E-01	2.71E+02
CCS3	Mean	7.61E+03	1.90E+00	7.53E+01	2.31E+02	6.10E+00	2.03E+01	3.46E+02
	SD	6.27E+02	4.95E-01	5.03E+00	2.96E+01	2.19E+00	1.91E+00	2.35E+02
CLPSO	Mean	6.52E+03	1.02E+00	6.20E+01	2.24E+02	2.69E+00	2.37E+01	1.99E+02
	SD	5.15E+02	2.11E-01	1.46E+00	2.15E+01	2.92E-01	7.76E-01	2.51E+01
LSCS	Mean	3.96E-01	2.07E+00	5.10E+01	3.36E+02	1.50E+00	2.24E+01	5.12E+02
	SD	2.99E-01	8.93E-01	7.08E-02	6.50E+01	3.34E-01	1.29E+00	3.99E+02

Algo		f_{32}	f_{33}	f_{34}	f_{35}	f_{36}	f_{37}	f_{38}
CS	Mean	1.46E+03	9.97E+03	3.59E+02	3.91E+02	2.00E+02	1.84E+03	4.00E+02
	SD	4.90E+02	3.41E+02	7.94E+00	5.27E+00	4.93E-02	3.69E+01	0.00E+00
TCPSO	Mean	3.54E+03	1.29E+04	3.43E+02	3.39E+02	4.20E+02	1.74E+03	1.89E+03
	SD	5.37E+02	1.49E+03	1.13E+01	9.59E+00	6.05E+01	8.71E+01	2.01E+03
SACS	Mean	6.70E+02	9.23E+03	3.43E+02	3.76E+02	2.40E+02	1.74E+03	4.00E+02
	SD	1.96E+02	8.28E+02	6.85E+00	4.70E+00	8.87E+01	5.34E+01	0.00E+00
CCS3	Mean	1.23E+03	8.56E+03	2.42E+02	3.98E+02	2.06E+02	1.39E+03	4.00E+02
	SD	4.74E+02	6.76E+02	1.38E+01	3.92E+01	1.45E+01	2.29E+02	1.58E-13
CLPSO	Mean	1.25E+03	8.63E+03	3.30E+02	3.94E+02	2.01E+02	1.52E+03	4.00E+02
	SD	2.34E+02	6.28E+02	9.94E+00	9.77E+00	2.49E-01	3.12E+02	5.50E-02
LSCS	Mean	1.10E+02	8.94E+03	3.08E+02	3.26E+02	2.62E+02	1.21E+03	4.00E+02
	SD	1.09E+02	1.31E+03	1.18E+01	4.06E+01	9.61E+01	6.26E+01	1.48E-01

Table 5 Comparison LSCS with CS on Wilcoxon rank sum test

	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}
Wilcoxon W	915	615	695	465	915	685	583
Z	0	-4.439	-3.267	-6.657	0	-3.405	-4.913
p-value	1	9.045E-6	0.0011	2.784E-11	1	6.598E-4	8.944E-7

	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}
Wilcoxon W	639.5	602	465	510	687	618.5	465
Z	-4.2859	-4.633	-6.675	-6.947	-3.376	-4.38	-6.659
p-value	1.82E-5	3.587E-6	2.464E-11	3.708E-12	7.352E-4	1.133E-5	2.746E-11

	f_{25}	f_{26}	f_{27}	f_{28}	f_{29}	f_{30}	f_{31}
Wilcoxon W	528	753	465	666	465	852	513
Z	-6.645	-2.397	-6.6574	-3.683	-6.659	-0.934	-6.197
p-value	3.016E-11	0.0165	2.786E-11	2.298E-4	2.753E-11	0.3504	5.734E-10

	f_{32}	f_{33}	f_{34}	f_{35}	f_{36}	f_{37}	f_{38}
Wilcoxon W	465	849	465	495	840	591	915
Z	-6.656	-0.976	-6.661	-6.229	-1.989	-4.798	0
p-value	2.801E-11	0.328	2.724E-11	4.685E-10	0.0467	1.601E-6	1

References

- [1] C. Cobos, H. Muñoz-Collazos, R. Urbano-Muñoz, M. Mendozaab, E. Leonc, and E. Herrera-Viedmade, "Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion," *Information Sciences*, vol.281, no.2, pp.248–264, Oct. 2014.
- [2] L. Liu, H. Mu, and J. Yang, "Generic constraints handling techniques in constrained multi-criteria optimization and its application," *European Journal of Operational Research*, vol.244, no.2, pp.576–591, July 2015.
- [3] R. Ruiz and C. Maroto, "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility," *European Journal of Operational Research*, vol.169, no.3, pp.781–800, 2006.
- [4] C. Kea, Z. Fengyua, and L. Alingb, "Chaotic Dynamic Weight Particle Swarm Optimization for Numerical Function Optimization," *Knowledge-Based Systems*, vol.139, no.3, pp.23–40, Jan. 2018.
- [5] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H.S.-H. Chung, Y. Li, and Y.-H. Shi, "Particle Swarm Optimization With an Aging Leader and Challengers," *IEEE Trans. Evol. Comput.*, vol.17, no.2, pp.241–258, April 2013.
- [6] W.H. Lim and N.A.M. Isa, "Bidirectional teaching and peer-learning particle swarm optimization," *Information Sciences*, vol.280, no.4, pp.111–134, Oct. 2014.
- [7] M. Nasirac, S. Das, D. Maity, S. Sengupta, U. Halder, and P.N. Suganthan, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Information Sciences*, vol.209, no.5, pp.16–36, Nov. 2012.
- [8] L. Liu, H. Mu, and J. Yang, "Generic constraints handling techniques in constrained multi-criteria optimization and its application," *European Journal of Operational Research*, vol.244, no.2, pp.576–591, July 2015.
- [9] X. Li, J. Wang, and M. Yin, "Enhancing the performance of cuckoo search algorithm using orthogonal learning method," *Neural Computing & Applications*, vol.24, no.6, pp.1233–1247, May 2014.
- [10] X. Li and M. Yin, "Modified cuckoo search algorithm with self adaptive parameter method," *Information Sciences*, vol.298, pp.80–97 March 2015. (DOI:10.1016/j.ins.2014.11.042)
- [11] L. Huang, S. Ding, S. Yu, J. Wang, and K. Lu "Chaos-enhanced Cuckoo search optimization algorithms for global optimization," *Applied Mathematical Modelling*, vol.40, no.5, pp.3860–3875, March 2016.
- [12] X. Liu and M. Fu, "Cuckoo search algorithm based on frog leaping local search and chaos theory," *Applied Mathematics & Computation*, vol.266, pp.1083–1092, Sept. 2015.
- [13] D.H. Wolpert and W.G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol.1, no.1, pp.67–82, April 1997.
- [14] Z. Zhao, J. Wang, J. Zhao, and Z. Su, "Using a Grey model optimized by Differential Evolution algorithm to forecast the per capita annual net income of rural households in China," *Omega*, vol.40, no.5, pp.525–532, Oct. 2012.
- [15] M. Ali, P. Siarry, and M. Pant, "An efficient Differential Evolution based algorithm for solving multi-objective optimization problems," *European Journal of Operational Research*, vol.217 no.2, pp.404–416, March 2012.
- [16] A.W. Mohamed and P.N. Suganthan, "Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation," *Soft Computing*, vol.22, no.10, pp.3215–3235, May 2018.
- [17] X.S. Yang and S. Deb, "Cuckoo Search via Lévy Flight," *Proc. World Congress on Nature & Biologically Inspired Computing*, pp.210–214, IEEE, India, 2009.
- [18] L.d.S. Coelho, T.C. Bora, and V.C. Mariani, "Differential evolution based on truncated Lévy-type flights a dispatch problems[J]," *Int. J. Electrical Power & Energy Systems*, vol.57, pp.178–188, May 2014.
- [19] W. Chu, X. Gao, and S. Sorooshian, "Handling boundary constraints for particle swarm optimization in high-dimensional search space," *Information Sciences*, vol.181, no.20, pp.4569–4581, Oct. 2011.
- [20] N. Padhye, K. Deb, and P. Mittal, "Boundary Handling Approaches in Particle Swarm Optimization," *Proc. Seventh Int. Conf. Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Springer India, pp.287–298, 2013.
- [21] S. Helwig and R. Wanka, "Particle Swarm Optimization in High-Dimensional Bounded Search Spaces," *Proc. 2007 IEEE Swarm Intelligence Symposium*, pp.198–205, 2007.
- [22] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol.10, no.3, pp.281–295, June 2006.
- [23] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar, "Evaluation of Comprehensive Learning Particle Swarm Optimizer[C]," *Neural Information Processing, 11th International Conference, ICONIP 2004, Lecture Notes in Computer Science*, pp.230–235, Calcutta, India, 2004.
- [24] W.H. Lim and N.A.M. Isa, "An adaptive two-layer particle swarm optimization with elitist learning strategy," *Information Sciences*, vol.273, no.3, pp.49–72, July 2014.
- [25] J.J. Liang, B.Y. Qu, and P.N. Suganthan, A.G. Hernández-Díaz "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," *Int. J. Computer Assisted Radiology Surgery*, 2013.
- [26] S. Sun and J. Li, "A two-swarm cooperative particle swarms optimization," *Swarm & Evolutionary Computation*, vol.15, pp.1–18, April 2014.



Li Huang received her B.S degree from Anhui University of Technology, PR China, and her MA.Sc. degree in computer science and technology from Guangxi Normal University. She is currently a lecturer at School of Management Science and Engineering, Anhui University of Technology, PR China. Her research interests include optimization method and machine learning.



Xiao Zheng received the B.S degree from Anhui University, PR China, in 1997, and the Ph.D. degree in computer science and technology from Southeast University in 2014. He is currently a professor at the School of Computer Science and Technology, Anhui University of Technology. His research interests include service computing, mobile cloud computing and machine learning.



Shuai Ding received his Ph.D. in MIS from Hefei University of Technology in 2011. He has been a visiting scholar at the University of Pittsburgh. He is an associate professor of information systems at the School of Management, Hefei University of Technology, China. His research interests include social networks, information systems, artificial intelligence, cloud computing, and business intelligence.



Zhi Liu received the B.E. from University of Science and Technology of China and the Ph.D. degree in informatics from National Institute of Informatics. He is currently an assistant professor at Shizuoka University. He was a Junior Researcher (Assistant Professor) at Waseda University and a JSPS research fellow in National Institute of Informatics. His research interest includes video network transmission, vehicular networks and mobile edge computing. He was the recipient of the IEEE

StreamComm2011 best student paper award, 2015 IEICE Young Researcher Award and ICOIN2018 best paper award. He has been a Guest Editor of journals including Wireless Communications and Mobile Computing, Sensors and IEICE Transactions on Information and Systems. He has been serving as the chair for number of international conference and workshops. He is a member of IEEE and IEICE.



Jun Huang received the M.S. degree in computer science from Anhui University of Technology, Ma'anshan, China, and Ph.D. degree in computer science from University of the Chinese Academy of Sciences, Beijing, China. His research interests include machine learning and data mining.