International Journal of Innovative Computing, Information and Control
Volume **12**, Number **6**, December **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 1809–1819

CUCKOO SEARCH ALGORITHM BASED ON MOBILE CLOUD MODEL

Shoulin Yin[1], Jie Liu[1,2,*], Ye Zhang[2] and Lin Teng[1]

[1]Software College
Shenyang Normal University
No. 253, Huanghe Bei Street, Huanggu District, Shenyang 110034, P. R. China
{ 352720214; 1532554069 }@qq.com; *Corresponding author: nan127@sohu.com

[2]Department of Information Engineering
Harbin Institute of Technology
No. 92, West Dazhi Street, Nangang District, Harbin 150001, P. R. China
zhye@hit.edu.cn

Received April 2016; revised August 2016

Abstract. *Cuckoo search is a new intelligence metaheuristic search algorithm which is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Levy flight behavior of some birds and fruit flies. However, there are some disadvantages such as inaccuracy results, and easily falling into local solution. In order to improve the performance of solving continuous function problem using cuckoo search algorithm and offset the drawbacks of cuckoo search, we propose a new model based on mobile cloud model. Mobile cloud model has a better transition performance between quantitative and qualitative aspects. The principle of the new model is to realize evolutionary learning process of cuckoo by using mobile cloud model. Finally, we utilize 10 test functions to make experiments and verify the high efficiency of our method. The experiment results show that the new scheme has a fast convergence speed and has a strong advantage on the function optimization. Its test function is very close to optimal solution. Also it has true engineering practical value.*

Keywords: Cuckoo search, Mobile cloud model, Transition performance, Evolutionary learning process

1. **Introduction.** The cuckoo search (CS) algorithm is a new heuristic swarm intelligence algorithm proposed by Yang and Deb [1]. CS algorithm is derived from the behavior simulation of cuckoo parasitic brood behavior and birds or drosophila melanogaster. CS algorithm has been used in engineering practice because of its relatively simple structure, less control parameters, the optimized search path and strong ability to jump out of local extremum value. The performance of CS algorithm is very close to differential evolution, so CS has become a hot topic in the field of intelligent algorithms. However, CS algorithm has some disadvantages, such as slow search speed in late of algorithm, low convergence precision and search energy shortage. Therefore, lots of researchers have proposed many improved CS algorithms to further improve the performance of the CS algorithm.

Marichelvam et al. [2] represented a developed cuckoo search metaheuristic algorithm in their paper to minimize the makespan for the hybrid flow shop (HFS) scheduling problems. A constructive heuristic called NEH heuristic was incorporated with the initial solutions to obtain the optimal or near optimal solutions rapidly in the improved cuckoo search (ICS) algorithm. The proposed algorithm was validated with the data from a leading furniture manufacturing company. Bourguignon [3] showed an improved cuckoo search (ICS) algorithm for solving problems of constrained engineering optimization. The dynamic change of parameters of probability and step size were introduced, which were constant in the standard algorithm. It gave a better solution than the standard CS algorithm and other optimization algorithms. Enireddy and Kumar [4] proposed improved image classification algorithm for content based image retrieval system using Haar wavelet and cuckoo search algorithm was proposed to optimize the learning rate of the neural network. Zhang and Chen [5] presented an improved cuckoo search algorithm with adaptive method. The self-adaptive machine was used to control the scaling factor and found probability so as to improve population diversity and avoid premature. Also there are other improved CS algorithms [6-10]. In conclusion, all these new methods are improved from parameters and fusion of algorithm, which makes up the deficiency of CS algorithm to a great extent.

Original cuckoo search algorithm has disadvantages such as easily falling into local solution, long convergence time and big result error. Most extant cuckoo search algorithms are improved in one step of its operation processes, for example, introducing inertia weight, and adaptive dynamic factor. However, there are few references studying cuckoo search algorithm with other methods under cloud model, especially combining mobile cloud model. Mobile cloud model has a better transition performance between quantitative and qualitative aspects. We use mobile cloud model to realize the evolutionary learning processes of cuckoo. Therefore, we firstly propose the new cuckoo search algorithm based on mobile cloud model (MCS). The new method guides the search capabilities of cuckoo search algorithm by introducing mobile cloud model, which has enhanced the search ability and optimized the performance of the CS algorithm. The structure of this paper is as follows. Section 2 introduces the principle of CS algorithm and mobile cloud respectively. Section 3 illustrates the CS algorithm based on mobile cloud model detailed. There are experiments in Section 4 to show the performance of MCS algorithm. There is a conclusion in the last section.

2. Overview of Cuckoo Search Algorithm and Mobile Cloud Model.

2.1. **Basic cuckoo search algorithm.** The most special habit of cuckoo is parasitic brood. At the time of reproduction, the cuckoo cannot hatch for themselves, rather than lay their eggs in other birds' nest, let them hatch. Hatched chicks instinctively can damage other eggs in the nest, and be able to emit a more loudly scream than chicks of the host. In order to reduce the probability that they are found, some cuckoos mimic their eggs as the selected birds' eggs. Many hosts judge the health degree of next generations through chirp size, and healthy offspring can get more food. So there is a higher survival rate. Under some certain conditions, when the host finds some strange eggs in the nest, host will abandon the nest and choose another place as a nest. Competing with host, the cuckoo eggs and chicks scream will have a development toward the direction of simulating host to against the increased ability of distinguishing next generation of host. The cuckoo search algorithm is derived from the reproduction behavior of cuckoo and Levy flight mode. Levy flight is a kind of random flight way combining long time and small scale local search with occasionally a larger scale exploring. In biological swarm intelligent optimization algorithms, this way can make a bigger search filed, more diversity of population. Also it is easy to jump out of local optimal solution. In the cuckoo search algorithm, in order to make a virtualization for laying eggs cuckoo, it needs to make three idealized assumptions as follows.

1) Every cuckoo lays only one egg, and then randomly finds a nest of other birds to hatch it.

2) In a group of randomly selected nest, the best bird's nest will be retained.
3) The number of available nest n is constant, and the probability of finding other eggs by the host is pa.

Based on the above assumptions, the mind of CS algorithm is that an egg in bird nest is a candidate solution, a cuckoo egg is a new solution, CS uses a new solution and a better solution in the next generation to replace the worst candidate solution in nest. Finally, it finds the best solution in the nest.

Assuming that D-dimension vector $X_i = [x_1\ x_2\ \dots\ x_D]$ denotes every egg or cuckoo. We use Levy method to produce offspring,

$$x_i^{t+1} = x_i^t + a \otimes levy(\lambda) \tag{1}$$

where x_i^{t+1} and x_i^t are the i-th nest position in the $t+1$ iteration and t iteration respectively. $\otimes$ is point to point multiplication. a is step length control. $levy(\lambda)$ denotes jumping path of random search, $levy \sim u = t^{-\lambda}$. When position updating, random number $r = [0, 1]$ compares with pa. If $r > pa$, it makes a random transformation for x_i^t. Otherwise, it keeps unchanged.

2.2. **The mobile cloud model.** The mobile cloud model is an uncertain transformation model describing qualitative knowledge, qualitative concepts and quantitative values, which can reflect fuzziness and randomness of people and things in the world. It can provide some methods for qualitative quantitative analysis. Mobile cloud model has been used in the fuzzy evaluation, intelligent control [11] and other fields.

Definition 2.1. *Mobile cloud and cloud mass. Assuming that U is a quantitative domain expressed by numerical value. C is a qualitative concept in U. If quantitative value $x \in U$ is a random implement in C. The certainty degree $\mu(x) \in [0, 1]$ of x for C is a random number with stable tendency, $\mu : U \to [0, 1]$. For $x \in U$, x distribution of $x \to \mu(x)$ in U can be called mobile cloud. Each x is cloud mass. If corresponding domain of concept is n-dimension space, it can be expanded to n-dimension cloud.*

Mobile cloud has three numeric characters to present a cloud cluster as: $C(Ex, En, He)$. It mainly reflects the uncertainty of concept. Here Ex is expected value, which denotes qualitative sample points in the most high energy domain space. En is entropy, which represents measurable granularity of a qualitative concept. The bigger entropy value is, the larger span of cloud cluster is. What is more, En also can show uncertainty of the qualitative concept. In the domain space, the size of value range accepted by qualitative concept can be called fuzziness. He is hyper entropy (namely entropy of entropy). It denotes the uncertainty of He to describe the thickness of cloud and reflects the sample randomness of qualitative concept value, reveals the association between fuzziness and randomness.

Definition 2.2. *One-dimensional normal cloud operators $A(C(Ex, En, He))$ is a mapping transforming overall features of the qualitative concept into quantitative representation $\pi : C \to \pi$. Its processes are as follows.*

a) Generate a normal random number En' based on En and He (now En is expected value, and He is standard deviation).
b) Generate a normal random number x based on Ex, En' (now Ex is expected value, and the absolute value of En' is standard deviation), x is a cloud mass in domain space.
c) Calculate $y = e^{\frac{-(x-Ex)^2}{2(En')^2}}$. Let y be a certainty degree in qualitative concept A.
d) Repeat a)$\sim$c), until produce N cloud mass.

We use the normal cloud operator that can realize the transformation from conceptual space to numerical space and qualitative concept can be transformed into cloud set expressed by numerical value.

3. **The CS Algorithm Based on Mobile Cloud Model.** In this section, MCS algorithm is divided into three parts: designing a cuckoo mobile cloud model; studying the communication in group with MCS and detailed processes of MCS. Each process is a new step for MCS different from CS algorithm.

3.1. **Building mobile cloud of cuckoo.** In this subsection, we introduce CS algorithm based on mobile cloud. When simulating cuckoo spawning in other bird nest, the individual expects to lay its eggs toward the direction of the optimal solution. However, the whole group are trying to approximate optimal solution in the search space. In the group, the position of each individual moves toward the direction of the optimal location. Unified dimension of each individual has the similarity in essence; however, in that every implementation of individual has the characteristics of randomness. So approximation process characteristics of cuckoo spawning can be described by mobile cloud model reflecting randomness and fuzziness of cuckoo approximating optimal solution concept. The normal mobile cloud of cuckoo approximation characteristics can be explained that it transforms qualitative concept into quantitative numeric using mobile cloud model. Each dimension of the nest optimal position at the beginning of initial population is regarded as a population cuckoo mobile cloud $PCM_j(Ex, En, He)$. Its expected value is Ex. For each individual, we select some cloud masses randomly in the generated cloud cluster. Certainty degree of cloud mass approximation expect is multiplied by Ex and we select its average value as current dimension of next nest position. The mobile cloud model formula of the CS algorithm is:

$$Ex = X_{gbest,j} \tag{2}$$

$$En = AVG\left(\sum r_i\right) \tag{3}$$

$$He = AVG\left(\sum Re_i\right) \tag{4}$$

$$x_{ij}^{t+1} = AVG\left(\sum Ex \times T(PCM_j(Ex, En, He))\right) \tag{5}$$

where i denotes the i-th nest. j is the j-th dimension. r is the group information rate of cuckoo. n is the number of group. Re is the responsive of cuckoo. $T(\cdot)$ is randomly selected function. $AVG(\cdot)$ denotes average function.

3.2. **Group communication.** Under ideal conditions, cuckoo can communicate with each other in one group in CS algorithm. Through experience transformation between individual and group, all the cuckoos can obtain the position information of other nest and find the nest rapidly. Cuckoo will emit yell, and the cuckoo within a certain range can receive the cuckoo's message. The received responsivity of algorithm among information transforming changes in the range of $[j_{\min}, j_{\max}]$ as :

$$j_{1i}^t = \left((j_{1,\min} - j_{1,\max})\frac{t}{nt} + j_{1,\max}\right)\varphi_1 \tag{6}$$

$$j_{2i}^t = \left((j_{2,\min} - j_{2,\max})\frac{t}{nt} + j_{2,\max}\right)\varphi_2 \tag{7}$$

where $\varphi_1, \varphi_2 \in [0, 1]$ are random variable of uniform distribution. nt is a constant parameter. $j_{1,\max} = j_{2,\max} = j_{\max}$. $j_{1,\min} = j_{2,\min} = j_{\min}$.

When cuckoo shares group nest location information, we utilize the difference operator which is similar to the difference algorithm to make the information transmission, and it is summarized as shown in (8):

$$x_i^{t+1} = x_{gbest}^t + j_{1i}^t \left(x_{r1}^t - x_{r2}^t\right) + j_{2i}^t \left(x_{r3}^t - x_{r4}^t\right) \tag{8}$$

where x_{gbest}^t denotes the best nest position after updating cuckoo mobile cloud. x_{ri}^t is the randomly selected individual of the i-th nest after cuckoo mobile cloud updating.

3.3. **The CS algorithm based on mobile cloud model.** The detailed processes of MCS algorithm are as follows.

Step 1: Input. Population size n, maximum iteration T, search space dimension D, discovering probability Pa, constant parameter nt, responsivity from received information transformation $j_{\min} = 0$, $j_{\max} = 1$.
Step 2: Initialization. In the search space, randomly generate n initial positions of nest x_i $(i = 1, 2, \ldots, n)$ and relevant parameters.
Step 3: Initial evolution. Calculate the initial fitness value of bird nest. The best nest position in the whole nest is as the initial value of x_{gbest}^*.
Step 4: Cuckoo mobile cloud update. Utilize mobile cloud model to generate cuckoo mobile cloud model and update the cuckoo position.
Step 5: Group information transformation. Cuckoo group uses difference operator similar to difference algorithm to transform information.
Step 6: Cuckoo position update. If $rand > Pa$, we randomly change the position of nest according to Formula (1) and get a new nest, then the new nest will make a comparison with the nest position of Step 4 and keep the best nest position.
Step 7: Group evolution. Calculate the fitness value from Step 3 to Step 5 and record the individual experience and select current optimal nest position x_{gbest}^*.
Step 8: Judge the iteration. $T = T + 1$, if it satisfies stop condition, then it breaks out. Otherwise, it returns to Step 3 and continues to search optimal position.
Step 9: Finish. Output x_{gbest}^* and optimal value $f_{\min}$.

This new MCS algorithm is inspired by cuckoo behavior. Based on the CS method and mobile cloud model, we redefine the model framework for mobile cloud model and cuckoo spawning features and build a new theoretical rules, a new optimization mechanism. We propose a novel MCS (mobile cuckoo search algorithm) method. MCS uses the information provided from optimal solution to do that it takes received information rate of cuckoo and responsivity of cuckoo as entropy and hyper entropy of MCS to control the generation of the cuckoo cloud mass. Group received information rate decreasing and responsivity of cuckoo increasing shows cuckoo approaching target. Using mobile cloud model denotes a qualitative concept. Cuckoo individuals around the current optimal position are gathered into the best position of MCS. Through the cuckoo mobile cloud, it is able to make the algorithm have a better stability and approximate the optimal value gradually. In the MCS algorithm, group communication operator guides the cuckoo group moving towards the optimal solution. The changing of information transmission responsivity controls the change of cuckoo finding nest. Exchange of information between individuals in the group promotes them to move toward the goal direction eventually. From Formula (8), cuckoo communicates each other on the basis of the best position. Its updated solution has a fast convergence speed; however, it easily falls into local optimal solution. MCS algorithm uses Levy random search method, which can jump out of local solution. Therefore, MCS algorithm is a better choice in solving optimal problems.

4. **Experiments and Analysis.** In order to verify the performance of MCS algorithm, we make a comparison with cuckoo algorithm (CS) and two typical improved CS algorithms: (CS based on Gamma distribution) GCS in reference [11]; NCS in reference [12]. We select 10 typical Benchmark function optimization problems as test function of this algorithm performance. The detailed functions are as Table 1.

Table 1. Test functions

Function	Expression	Range of x_i	Function type
Axis parallel ellipsoid	$f_1(X) = \sum_{i=1}^{n} ix_i^2$	$[-5.12, 5.12]$	Uni-modal function
Schwefel	$f_2(X) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	$[-10, 10]$	Uni-modal function
Rastrigin	$f_3(X) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	Jump function
Ackley	$f_4(X) = -20e^{-\frac{1}{5}\sqrt{1/n\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$	$[-30, 30]$	Multi-modal function
Sphere	$f_5(X) = \sum_{i=1}^{n} x_i^2$	$[-5.12, 5.12]$	Uni-modal function
Griewank	$f_6(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	Multi-modal function
Rotated hyper-ellipsoid	$f_7(X) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-10, 10]$	Uni-modal function
Zakharow	$f_8(X) = \sum_{i=1}^{n} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n} ix_i\right)^4$	$[-5, 5]$	Uni-modal function
Exponential	$f_9(X) = -e^{-0.5\sum_{i=1}^{n} x_i^2}$	$[-1, 1]$	Uni-modal function
Step	$f_{10}(X) = \sum_{i=1}^{n} (\lfloor x_i + 0.5\rfloor)^2$	$[-100, 100]$	Step function

We set the same experiment parameters: nest size $n = 100$, $D = 30$ (number of dimension), iteration $T = 200$. The upper and lower of Pa is 0.95 and 0.15 respectively. Scaling factor $0.4 \leq r \leq 0.9$. In MCS, $j_{\min} = 0$, $j_{\max} = 1$, $nt = 5000$. When the algorithm searches the optimal solution or it reaches the maximum number of iterations, algorithm will stop. Each test function independently runs 20 times and gets optimal value, average value, worst value and standard deviation. We use the five values to make a comparison in MATLAB platform. And we get the results of the four algorithms as Table 2.

From Table 2 we can see that the optimal value with CS is the biggest compared with GCS, NCS and MCS. However, for function f_3, optimal value of GCS is 1.231e^2 which is the smallest compared with that of CS (1.396e^2), NCS (1.375e^2) and MCS (15.897 is the biggest value). Similar to optimal value, worst value, average value and standard deviation of MCS are the biggest. In that f_3 is jump function. However, MCS algorithm can find global minimum value in this test except f_3. Especially, MCS almost finds the optimal extreme value 0 in function f_5. In other functions, it gets the global optimal value with MCS algorithm, which is better than CS, GCS and NCS algorithm. It has the ability of jumping out of local optimal value.

Table 2. Test results with different algorithms when $D = 30$

Function	Algorithm	Optimal value	Worst value	Average value	Standard deviation
f_1	CS	60.123	1.0905	81.139	14.612
	GCS	49.531	95.864	74.124	14.191
	NCS	24.381	62.322	44.978	8.9983
	MCS	0.0142	0.0064	0.0042	0.0021
f_2	CS	40.123	66.038	53.302	6.7123
	GCS	30.325	45.206	36.187	4.0101
	NCS	30.762	61.205	48.021	8.6521
	MCS	0.0641	0.1213	0.0798	0.0211
f_3	CS	$1.396e^2$	$1.9182e^2$	$1.6821e^2$	12.284
	GCS	$1.231e^2$	$1.8195e^2$	$1.5482e^2$	15.121
	NCS	$1.375e^2$	$1.9732e^2$	$1.7354e^2$	17.652
	MCS	15.897	95.628	57.312	20.898
f_4	CS	13.412	17.612	15.728	1.0227
	GCS	12.309	15.221	14.637	0.9543
	NCS	14.112	16.861	15.221	0.7496
	MCS	0.0350	2.5193	1.4987	0.7512
f_5	CS	3.6147	8.4287	6.2158	1.2856
	GCS	3.3291	7.1552	5.4811	1.1642
	NCS	3.0101	5.3615	4.0452	0.7876
	MCS	$4.9187e^{-5}$	$1.3197e^{-4}$	$8.7805e^{-5}$	$2.1661e^{-5}$
f_6	CS	15.835	32.682	23.473	4.7382
	GCS	12.534	28.705	20.472	4.3365
	NCS	24.618	54.866	39.318	7.8132
	MCS	0.0328	0.1348	0.0509	0.02153
f_7	CS	$1.6375e^3$	$3.3204e^3$	$2.2434e^3$	$4.6243e^2$
	GCS	$1.0428e^3$	$2.6452e^3$	$1.8214e^3$	$4.5276e^2$
	NCS	$5.3596e^2$	$9.6654e^2$	$7.8246e^2$	$1.3877e^2$
	MCS	0.1187	0.6256	0.3121	0.1165
f_8	CS	$4.3521e^3$	$2.6185e^4$	$1.2231e^4$	$5.5389e^3$
	GCS	$3.2195e^3$	$7.5027e^3$	$5.2067e^3$	$1.3325e^3$
	NCS	$1.2015e^3$	$1.1546e^4$	$6.3185e^3$	$2.7326e^3$
	MCS	0.0142	0.0745	0.0318	0.0209
f_9	CS	−0.9213	−0.8574	−0.8877	0.0196
	GCS	−0.9225	−0.8634	−0.8934	0.0164
	NCS	−0.9319	−0.8395	−0.8733	0.0315
	MCS	−0.9999	−0.9999	−0.9999	$2.5123e^{-6}$
f_{10}	CS	1328.02	3121.62	$2.3384e^3$	$4.5564e^2$
	GCS	1339.37	3002.49	$2.1865e^3$	$4.3512e^2$
	NCS	2416.84	4192.52	$3.0312e^3$	$5.1643e^2$
	MCS	0	1	0.16	0.3672

In order to further observe the better performance of MCS than CS, GCS and NCS, this section gives fitness change curve when solving the ten functions with MCS, CS, GCS and NCS algorithm as Figures 1-10. Figure 1 shows that MCS has a fast convergence speed within 15 iterations. Other methods have the similar convergence time for function f_1. In Figure 2, the curves of four algorithms are very similar to each other, which shows that they all have a better performance for function f_2. The curves of MCS in Figure 3 and Figure 4 are similar. Obviously, MCS is superior to other three methods. In Figure

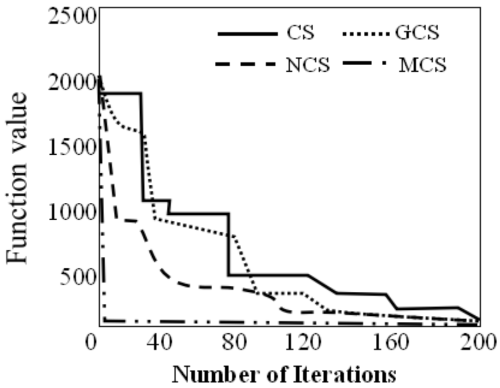

FIGURE 1. Comparison of f_1

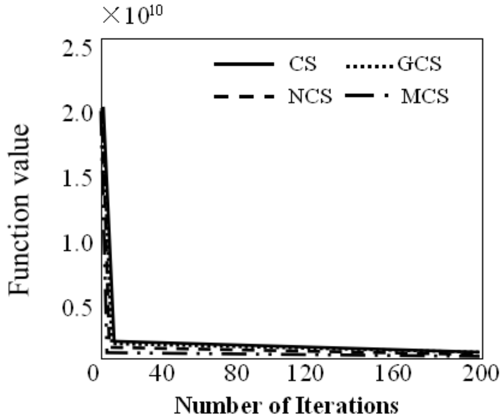

FIGURE 2. Comparison of f_2

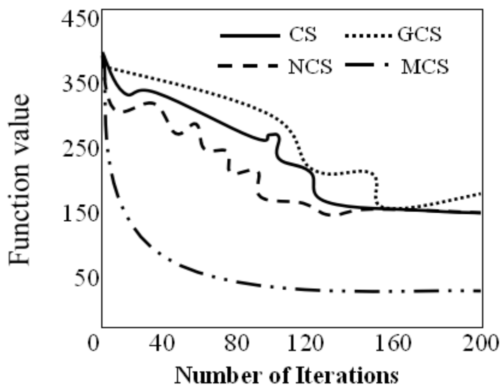

FIGURE 3. Comparison of f_3

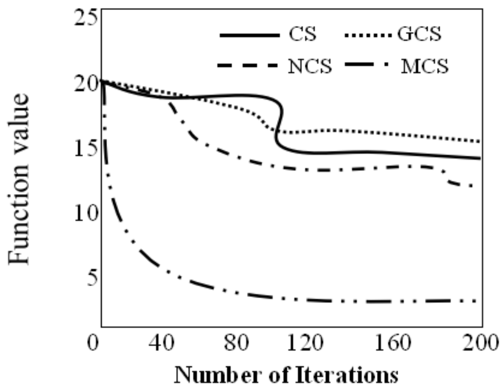

FIGURE 4. Comparison of f_4

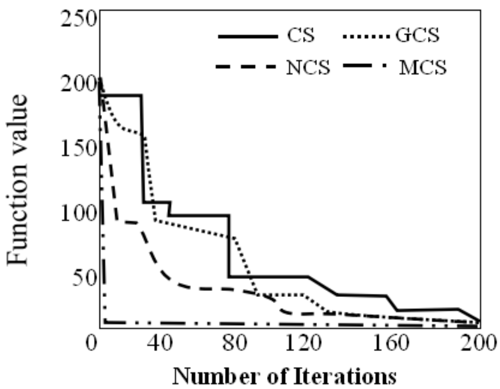

FIGURE 5. Comparison of f_5

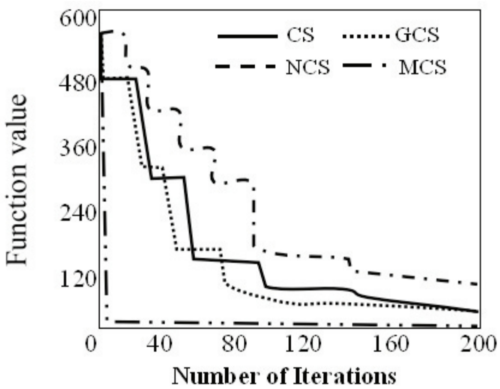

FIGURE 6. Comparison of f_6

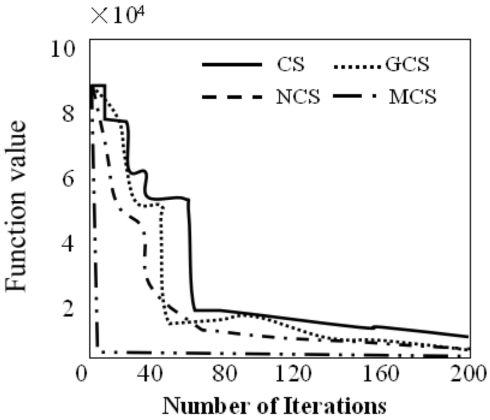

FIGURE 7. Comparison of f_7

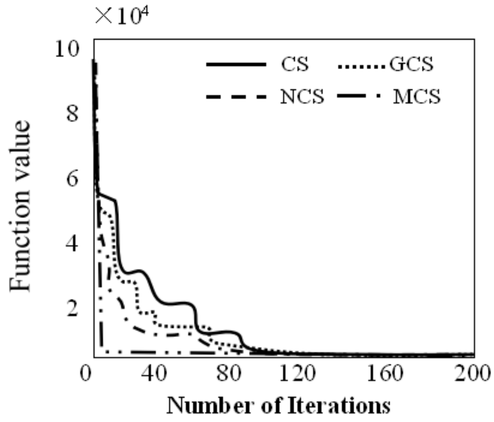

FIGURE 8. Comparison of f_8

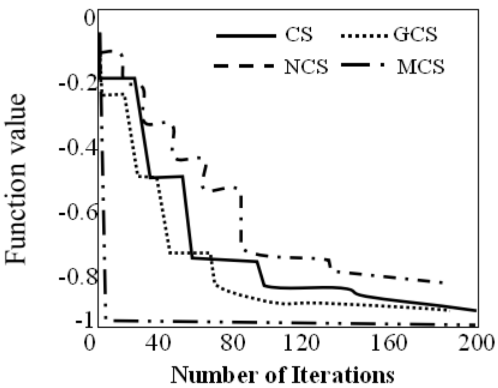

FIGURE 9. Comparison of f_9

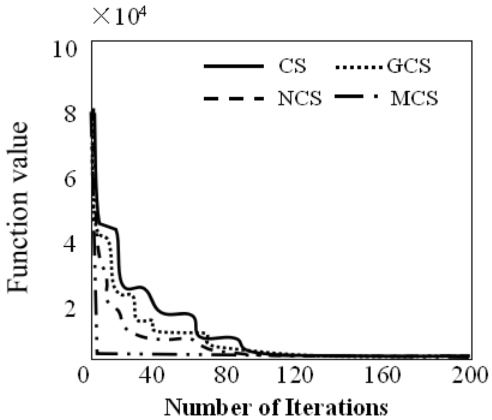

FIGURE 10. Comparison of f_{10}

5, function value of CS algorithm decreases from 200 to approximately 110 before running 40 times and that of GCS reaches about 99. Although convergence time of NCS is shorter than CS and GCS, MCS still has a fast convergence time. In Figure 6, what is different from Figure 5 is that NCS algorithm is the worst method for function f_6. However, they all experience a decreasing trend.

Figure 7 and Figure 9 are similar to Figure 5 and Figure 6, which shows the same curve changing trend. Figures 8 and 10 have the same iteration start and the end. However, the changing of MCS is superior to the other three algorithms.

From the above figures, we can know that our method MCS algorithm has faster convergence speed and better global search ability than other CS algorithms. Functions with MCS algorithm reach a steady low value in a short time. Also we select four (f_2, f_4, f_6, f_8) of the ten functions to test the performance of MCS with the high dimension. $D = 128$, $T = 500$. The results are as Table 3.

In Table 3, optimal value of f_2, f_4, f_6, f_8 are 0.4635, 61.856, 0.0123, -0.9997 respectively. These data show that MCS can obtain the best value than CS, GCS and NCS. Especially, standard deviation of MCS is 0.0926 (f_2), 0.0012 (f_6), 1.7315e^{-4} (f_8). That is less than other three algorithms. Only in f_4, the standard deviation of MCS is slightly more than CS, GCS and NCS. The results with MCS are superior to CS, GCS and NCS from Table 3 in the high dimension test. We also obtain the fitness change curve about the four functions as Figures 11-13. It obtains the similar results to above figures. In Figure

TABLE 3. Test results with different algorithms when $D = 128$

Function	Algorithm	Optimal value	Worst value	Average value	Standard deviation
f_2	CS	8.5123e^2	1.3207e^3	1.1423e^3	1.5028e^2
	GCS	1.0724e^3	2.1893e^3	1.6376e^3	2.9354e^2
	NCS	1.4471e^3	2.4545e^3	1.9923e^3	3.6157e^2
	MCS	0.4635	0.8167	0.6357	0.0926
f_4	CS	8.5513e^2	1.0528e^3	9.4351e^2	43.352
	GCS	1.0459e^3	1.1545e^3	1.0985e^3	30.526
	NCS	7.4358e^2	1.0284e^3	9.1978e^2	72.168
	MCS	61.856	3.2509e^2	1.9573e^2	75.429
f_6	CS	12.529	16.375	14.545	1.2913
	GCS	15.321	19.238	23.099	1.9382
	NCS	23.905	41.205	31.767	4.3397
	MCS	0.0123	0.0181	0.0158	0.0012
f_8	CS	-0.7915	-0.7421	-0.7754	0.0166
	GCS	-0.1287	-0.0841	-0.1075	0.0128
	NCS	-0.6314	-0.5121	-0.5549	0.0338
	MCS	-0.9997	-0.9987	-0.9985	1.7315e^{-4}

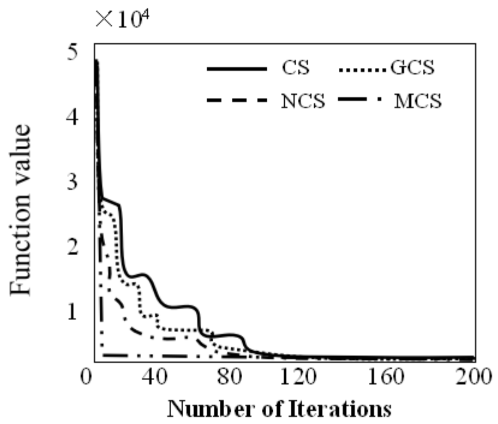

FIGURE 11. Comparison of f_2

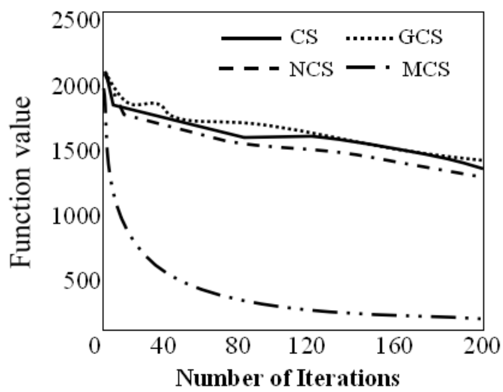

FIGURE 12. Comparison of f_4

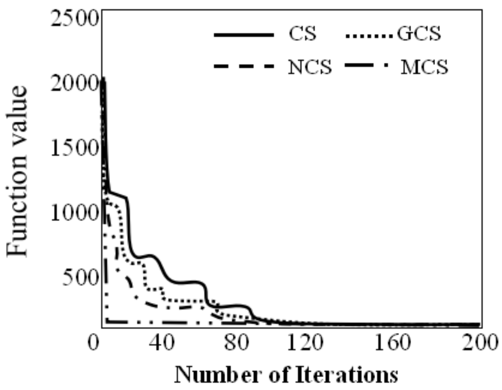

FIGURE 13. Comparison of f_6

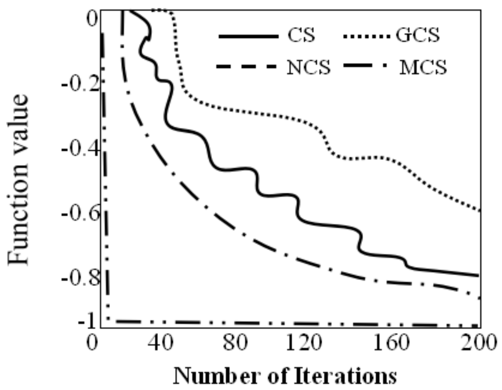

FIGURE 14. Comparison of f_8

14, GCS method is very poor. When iteration number is 200, it only reaches nearly -0.6. Followed by CS, the minimum value can be stable at about -0.8. Then value of NCS is -0.9 less than CS and greatly more than MCS (nearly -1). MCS algorithm still has the strong ability of global search and faster convergence speed in high dimension test, which offsets the disadvantage of CS algorithm and avoids premature convergence phenomenon to a certain extent. Therefore, our method has the best performance compared with other methods and is the best choice for cuckoo search.

5. **Conclusion.** In this paper, we propose cuckoo search algorithm based on mobile cloud model to improve the disadvantage of cuckoo search method. We improve the updating method of nest position combining mobile cloud model and use Levy way resemble differential evolution to realize the information transforming among group, which insures the cooperation of group, improves the cuckoo search ability and search efficiency and optimizes the performance of the algorithm. To demonstrate its performance, we use 10 high dimension complex problems to make experiments. From the results, we know that MCS algorithm has the advantage of jumping out of local optimal solution and high search efficiency. It will be applied into many practical engineering works in the future and we will also study more advanced cuckoo search methods to improve the current optimization problems.

REFERENCES

[1] X. Yang and S. Deb, Cuckoo search via Levy flights, *World Congress on Nature & Biologically Inspired Computing*, pp.210-214, 2010.

[2] M. K. Marichelvam, T. Prabaharan and X. S. Yang, Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan, *Applied Soft Computing*, vol.19, no.1, pp.93-101, 2014.

[3] G. Bourguignon, Improved cuckoo search (ICS) algorithm for constrained optimization problems, *Latin American Journal of Solids and Structures*, vol.11, no.8, pp.1349-1362, 2014.

[4] V. Enireddy and R. K. Kumar, Improved cuckoo search with particle swarm optimization for classification of compressed images, *Sadhana*, vol.4, no.8, pp.2271-2285, 2015.

[5] Z. Zhang and Y. Chen, An improved cuckoo search algorithm with adaptive method, *The 7th International Joint Conference on Computational Sciences and Optimization*, pp.204-207, 2014.

[6] S. Haring and R. Hochreiter, Efficient and robust calibration of the Heston option pricing model for American options using an improved cuckoo search algorithm, *American Journal of Roentgenology Radium Therapy & Nuclear Medicine*, vol.89, no.1, pp.797-803, 2015.

[7] J. Wang, B. Zhou and S. Zhou, An improved cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation, *Computational Intelligence & Neuroscience*, vol.2016, no.2, pp.1-8, 2016.

[8] P. Mohapatra, S. Chakravarty and P. K. Dash, An improved cuckoo search based extreme learning machine for medical data classification, *Swarm & Evolutionary Computation*, no.24, pp.25-49, 2015.

[9] D. Laha and D. K. Behera, An improved cuckoo search algorithm for parallel machine scheduling, *The International Conference on Swarm, Evolutionary and Memetic Computing*, pp.788-800, 2015.

[10] Q. Jin, L. Qi, B. Jiang et al., Novel improved cuckoo search for PID controller design, *Transactions of the Institute of Measurement & Control*, vol.37, no.6, 2014.

[11] Q. Zhang, L. T. Yang and Z. Chen, Privacy preserving deep computation model on cloud for big data feature learning, *IEEE Trans. Computers*, vol.65, no.5, pp.1351-1362, 2016.

[12] S. Roy, A. Mallick, S. S. Chowdhury et al., A novel approach on cuckoo search algorithm using Gamma distribution, *The 2nd International Conference on Electronics and Communication Systems*, 2015.

[13] H. Wang, W. Wang, H. Sun et al., A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems, *Soft Computing*, pp.1-11, 2016.