

## Research Article

# Cuckoo Search Algorithm Based on Repeat-Cycle Asymptotic Self-Learning and Self-Evolving Disturbance for Function Optimization

Jie-sheng Wang,<sup>1,2</sup> Shu-xia Li,<sup>1</sup> and Jiang-di Song<sup>1</sup>

<sup>1</sup>School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114044, China

<sup>2</sup>National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning, Anshan 114044, China

Correspondence should be addressed to Jie-sheng Wang; wang\_jiesheng@126.com

Received 29 October 2014; Accepted 19 January 2015

Academic Editor: Rahib H. Abiyev

Copyright © 2015 Jie-sheng Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to improve convergence velocity and optimization accuracy of the cuckoo search (CS) algorithm for solving the function optimization problems, a new improved cuckoo search algorithm based on the repeat-cycle asymptotic self-learning and self-evolving disturbance (RC-SSCS) is proposed. A disturbance operation is added into the algorithm by constructing a disturbance factor to make a more careful and thorough search near the bird's nests location. In order to select a reasonable repeat-cycled disturbance number, a further study on the choice of disturbance times is made. Finally, six typical test functions are adopted to carry out simulation experiments, meanwhile, compare algorithms of this paper with two typical swarm intelligence algorithms particle swarm optimization (PSO) algorithm and artificial bee colony (ABC) algorithm. The results show that the improved cuckoo search algorithm has better convergence velocity and optimization accuracy.

## 1. Introduction

Cuckoo search (CS) algorithm, a new biological heuristic algorithm, is put forward by Yang and Deb in 2009. It simulates the cuckoo's seeking nest and spawning behavior and introduces Levy flight mechanism into it, which is able to quickly and efficiently find the optimal solution [1, 2]. Studies have proved that CS algorithm is better than other swarm intelligence algorithms in convergence rate and optimization accuracy, such as genetic algorithm (GA), particle swarm optimization (PSO) algorithm, and artificial bee colony (ABC) algorithm [3]. Because this algorithm has the characteristics of fewer parameters and being simple and easy to implement, now it has been successfully applied in a variety of engineering optimization problems. So the CS algorithm has a very high potential research value [4, 5].

CS algorithm is a new type of bionic algorithm. Many scholars carry out many researches in CS algorithm and put forward the corresponding improvement strategies. In literature [6], it gains insight into search mechanisms of

CS algorithm and analyzes why it is efficient; meanwhile, it discusses the essence of algorithms and its link to self-organizing systems [6]. In literature [7], in order to increase CS efficiency, it exploits several parameters of the CS algorithm involving the Levy distribution factor beta ( $\beta$ ) and the probability factor ( $P$ ) and by seeking optimum values of these parameters efficiency of CS algorithm are improved [7]. In literature [8], it studied the algorithmic structure and behavior of CS and Levy distribution in detail, and then by comparing with widely used optimization algorithms (i.e., DE and GA) statistical results verified that CS has a more superior problem-solving ability [8]. For the purpose of enhancing the search ability of the cuckoo search (CS) algorithm, an improved robust approach, called harmony search (HS), is put forward, in which method a mutation operator is added to the process of the cuckoos updating to speed up convergence [9]. In literature [10], in order to solve combinatorial problems, it extends and improves CS by reconstructing its population and introducing a new category of cuckoos [10]. A cuckoo optimal algorithm based

on the exchange operator and the chaotic disturbance is proposed, which introduces the exchange operator theory of the particle swarm optimization algorithm to improve convergence rate and optimization accuracy [11]. A cooperative coevolutionary cuckoo search algorithm is put forward by applying the framework of cooperative coevolutionary, which divides the solution vectors of population into several subvectors and constructs the corresponding subswarms [12]. A novel cuckoo search optimization algorithm based on Gauss distribution is proposed by adding a Gauss distribution to CS algorithm to improve convergence rate [13]. A new self-adaptive cuckoo search algorithm is proposed by using a self-adaptive parameter control strategy to adjust the step size of CS to enhance its search ability [14]. Because the CS algorithm is highly random search according to Levy flight mechanism and shows a strong leaping, it is easy to jump from one region to another region, which makes the search that around each bird's nest location not careful and thorough and cannot make full use of information nearby the bird's nest locations. So the CS algorithm has characteristics of weak local searching ability, slow convergence rate, and low optimization accuracy.

In order to make up the defect of the algorithm on this aspect, a kind of improved cuckoo search algorithm based on the repeat-cycle asymptotic self-learning and self-evolving disturbance (RC-SSCS) is proposed. In order to obtain better disturbance effect, the learning and updating strategy of the worst frog in the shuffled frog leaping algorithm (SFLA) and part of differential evolution (DE) thought are introduced into the constructing of disturbance factor, thus, which makes every disturbance with the effect of nest's self-learning and self-evolving. Finally, the six typical test functions are chosen for simulation experiment and the simulation results proved that the improved cuckoo search algorithm has better convergence rate and optimization accuracy. The paper is organized as follows. In Section 2, the cuckoo search algorithm is introduced. The improved cuckoo search algorithm based on the repeat-cycle asymptotic self-learning and self-evolving disturbance is presented in Section 3. In Section 4, the simulation experiments and results analysis are introduced in details. Finally, the conclusion illustrates the last part.

## 2. Cuckoo Search Algorithm

Cuckoo search algorithm (CS) is developed by Xin-she Yang by observing the magical nature phenomenon and then giving an artificial processing, which is a new type of heuristic search algorithm [15, 16]. This algorithm is mainly based on two aspects: the cuckoo's parasitic reproduction mechanism and Levy flights search principle. In nature, cuckoos use a random manner or a quasirandom manner to seek bird's nest location [10]. Most of cuckoos lay their eggs in other bird nests and let the host raise their cubs instead of them. If a host found that the eggs are not its owns, it will either throw these alien eggs away from the nest or abandon its nest and build a new nest in other places. However, there are some cuckoos choosing nest that the color and shape of the host's egg are similar with their own to win the host's love, which can reduce

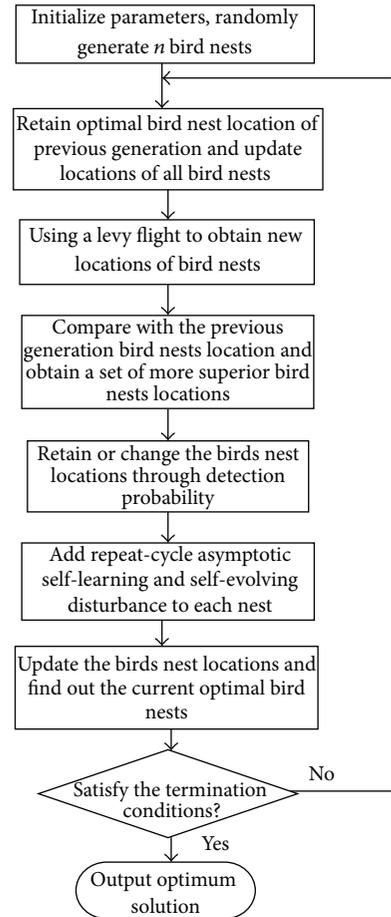


FIGURE 1: Flow chart of RC-SSCS algorithm.

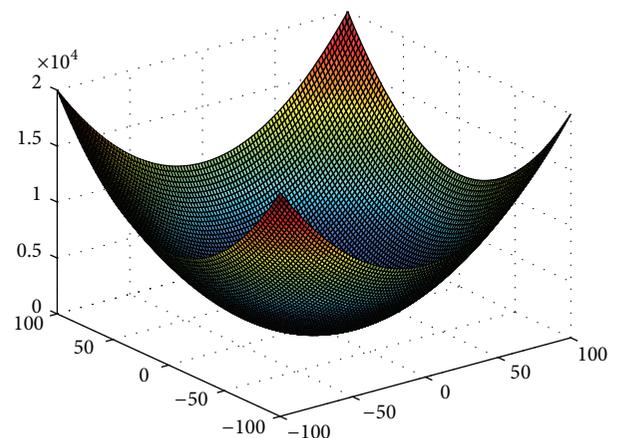


FIGURE 2: 3D surface figure of Sphere function.

the possibility of their eggs being abandoned and increase the reproduction rate of the cuckoos.

In general, each cuckoo can only lay one egg and each egg on behalf of one solution (cuckoo). The purpose is to make the new and potentially better solutions replace the not-so-good solutions (cuckoos). In order to study the cuckoo search algorithm better, the simplest method is adopted, that is to

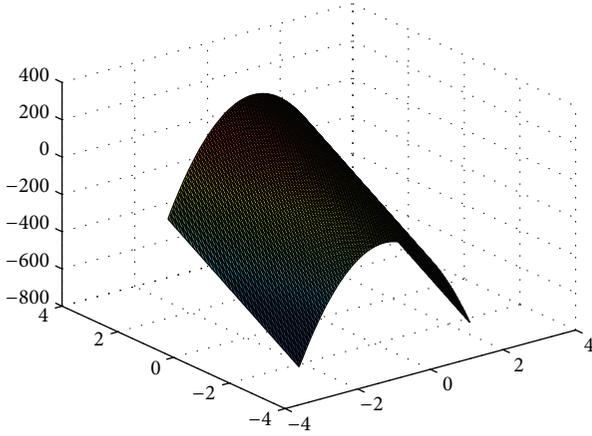


FIGURE 3: 3D surface figure of Rosenbrock function.

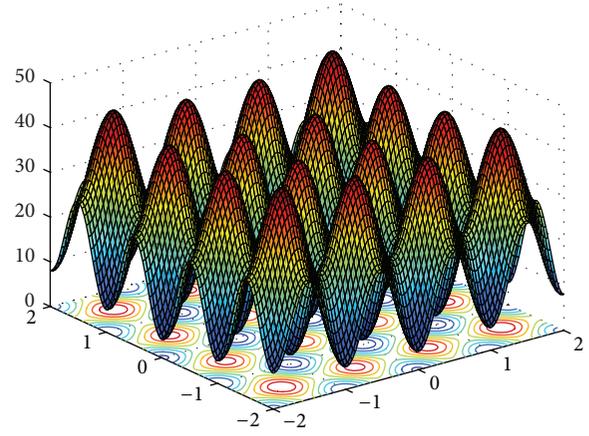


FIGURE 5: 3D surface figure of Rastrigrin function.

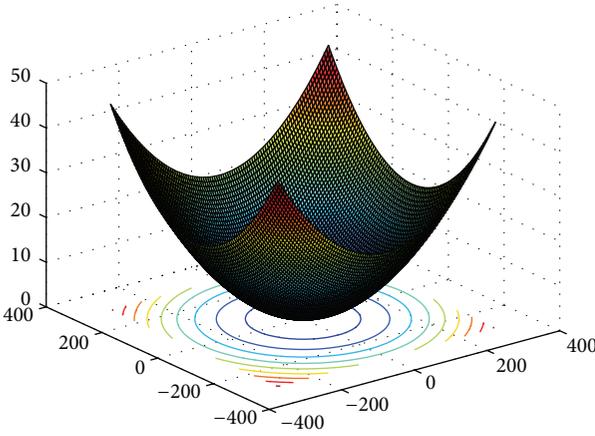


FIGURE 4: 3D surface figure of Griewank function.

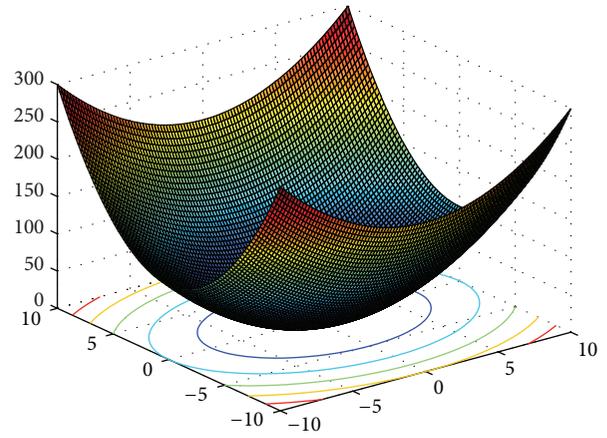


FIGURE 6: 3D surface figure of Sumsquares function.

say, only one egg is in each nest. In this case, an egg, a bird's nest, or a cuckoo is no different, which is to say, each nest corresponding to a cuckoo's egg. For simplicity in describing the cuckoo search algorithm, Yang and Deb use the following three idealized rules to construct the cuckoo algorithm [17].

- (1) Each cuckoo only lays one egg at a time and randomly chooses bird's nest to hatch the egg.
- (2) The best nest will carry over to the next generation.
- (3) The number of available host nests is fixed, and the probability of a host discovering an alien egg is  $P_a = [0, 1]$ . In this case, the host bird may either throw the alien egg away or abandon its nest so as to build a new nest in a new location.

Under the above conditions, the specific steps of CS algorithm are described as follows.

(1) *Initialization Setting.* Randomly generate  $N$  bird's nest location  $X_0 = (x_1^0, x_2^0, \dots, x_N^0)$ , and then take the  $N$  bird's nest positions into test functions for choosing experiments. Through testing, the best bird's nest location is chosen and carried over to the next generation.

(2) *Searching of Bird's Nest Location.* Equation (1) is used to realize the location update and search the bird's nest location of the next generation in order to gain a new set of bird's nest locations, which are taken into the test function again for testing experiments. After comparing with the last generation nest locations, the best bird's nest location is chosen and entered into the next step. Consider

$$x_i^{(t+1)} = x_i^t + \alpha \oplus \text{Levy}(\lambda), \quad i = 1, 2, \dots, n. \quad (1)$$

(3) *Selection of Bird's Nest Locations.* The probability of a host discovering an alien egg  $P_a = 0.25$  is compared with the random number that obeys uniform distribution  $r \in [0, 1]$ . If  $r$ , the value of  $x_i^{t+1}$  is randomly changed. Otherwise it does not need to be changed.

Then the changed bird's nest locations are calculated with the test functions, which are compared with the optimal position of previous generation and best bird's nest locations  $X_t = (x_1^t, x_2^t, \dots, x_N^t)$  is recorded. Finally, the optimal nest position  $pb_t^*$  is chosen.

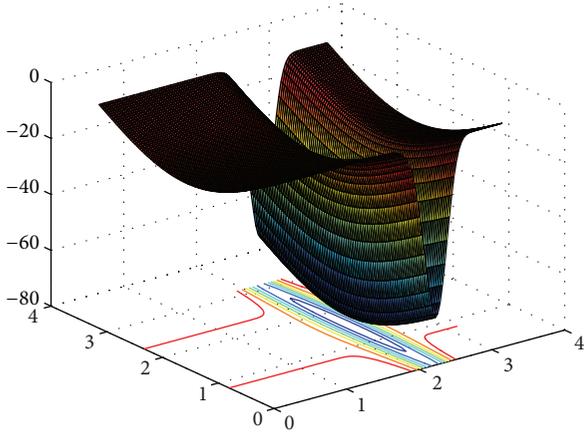


FIGURE 7: 3D surface figure of Michalewicz function.

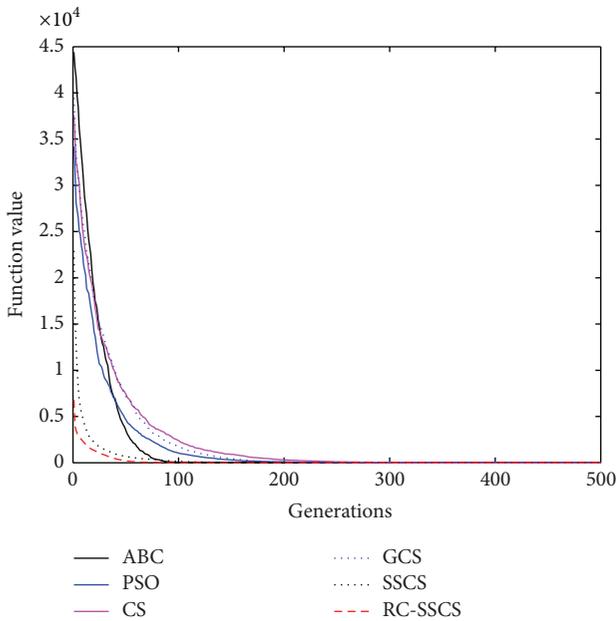


FIGURE 8: Convergence curves of Function  $f_1$ .

(4) *Accuracy or Iterations Judgment.* Calculate and judge whether  $f(pb_i^*)$  achieves the object accuracy or the terminating conditions. If it meets the requirements,  $pb_i^*$  is the global optimal solution  $gb$ ; if it is not be met,  $pb_i^*$  is kept to the next generation and return to step (2) and start the next loop iteration and update again.

According to these four steps described above of cuckoo search algorithm, the cuckoo algorithm not only uses the Levy flight (global search) search method, but also introduces the elite reserved strategy (local search), which makes the algorithm have both global search ability and local search ability. The purpose of step (3) in this algorithm is to increase the diversity of solution so that the algorithm is prevented from being caught into local optimum and achieving the global optimum.

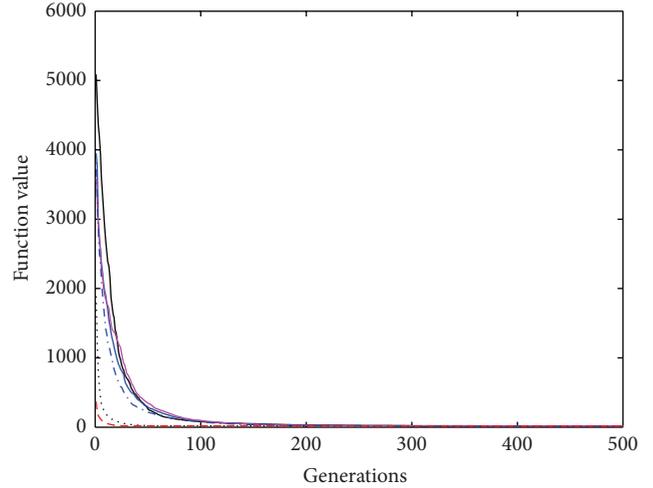


FIGURE 9: Convergence curves of Function  $f_2$ .

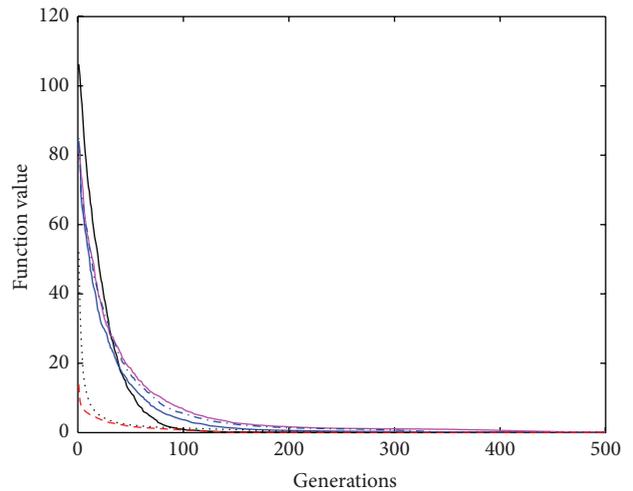
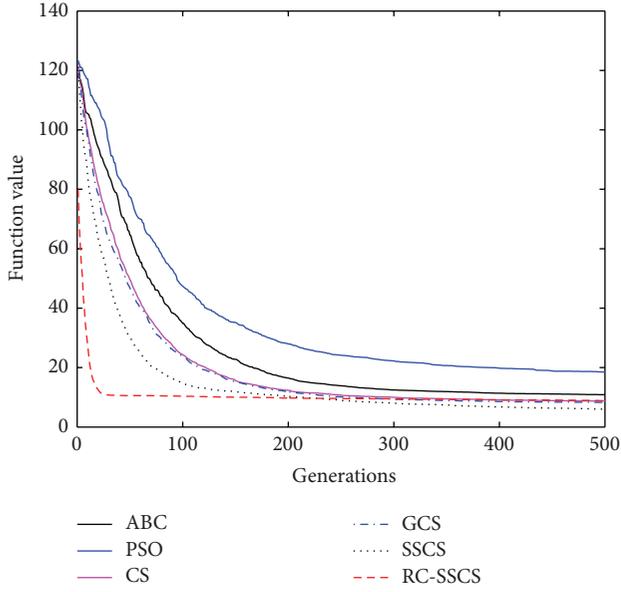
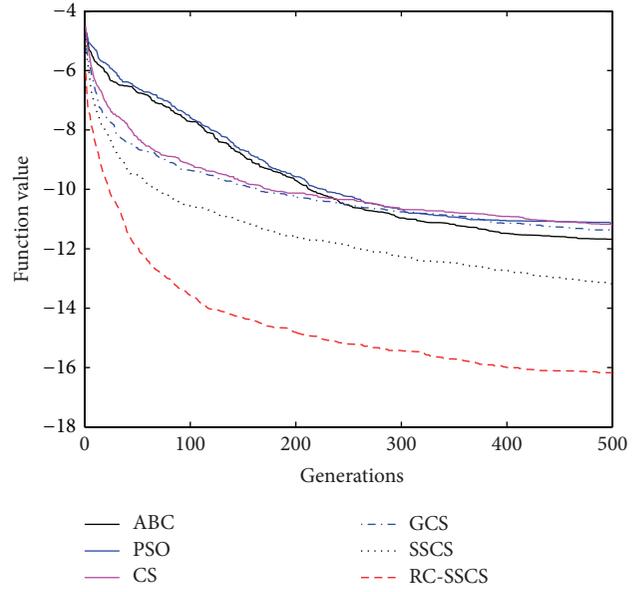
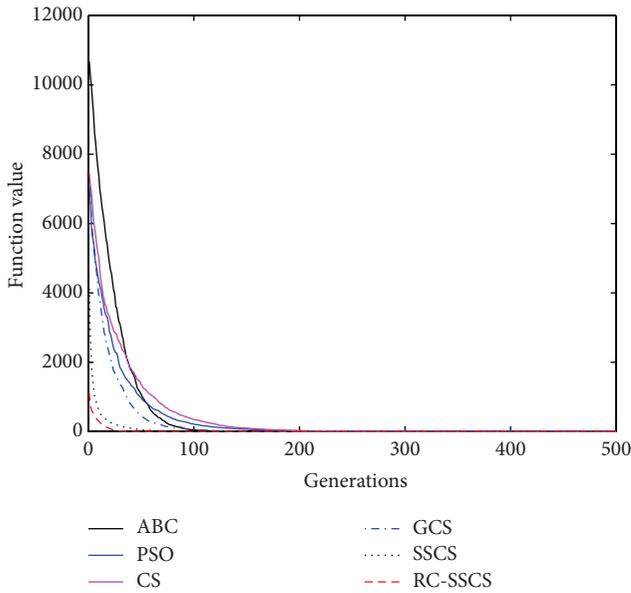


FIGURE 10: Convergence curves of Function  $f_3$ .

The search path of CS algorithm is different with other swarm algorithms. The CS algorithm uses Levy flight, which has character of strong randomness. Broadly speaking, Levy flight is a random walk, whose step size obeys Levy distribution, and the direction of travel is subject to uniform distribution. The step size vector of CS algorithm is determined by Mantegna rule of Levy distribution characteristics. In Mantegna rules, the step size  $s$  is designed as follows:

$$s = \frac{u}{|v|^{1/\beta}}, \tag{2}$$


 FIGURE 11: Convergence curves of Function  $f_4$ .

 FIGURE 13: Convergence curves of Function  $f_6$ .

 FIGURE 12: Convergence curves of Function  $f_5$ .

where  $u$  and  $v$  obey the normal distribution, that is to say

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2), \quad (3)$$

where

$$\sigma_v = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2] \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad (4)$$

$$\sigma_u = 1.$$

But here the chosen method of the direction obeys uniform distribution. The searching pattern of CS algorithm

is Levy flight, for instance, the  $i$ th cuckoo of the  $t$ th generation generates the next generation solution  $x_i^{t+1}$ :

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Levy}(\lambda), \quad (5)$$

where  $\oplus$  is a point to point multiplication and the step size of  $\text{Levy}(\lambda)$  obeys the Levy distribution, which can be expressed as follows:

$$\text{Levy} \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3). \quad (6)$$

Here, the Mantegna rules are used to calculate the step size. In (5),  $\alpha$  is the step controlled quantity mainly used to control the direction and step size. In that its distribution is a power function, Levy distribution has infinite variance and its increment obeys heavy-tailed distribution [18, 19]. Levy flight is seemingly Brownian motion under the status of a long-distance flight, or it may be described that Levy flight consists of frequent short-jump and occasional long-jump. The long-jump can help the CS algorithm jump out of local optimum. Consider

$$\alpha = O\left(\frac{L}{10}\right), \quad (7)$$

where  $L$  is the search space size of optimization problems.

Thus, the generation of some new solutions is gradually through the Levy flight and rand walk around the optimal solution to obtain the optimal solution, which can speed up the local searching. On the contrary, a part of new solutions is far away from the current optimal solution, because they are randomly generated by deviating from remote locations. The main purpose of these solutions is to ensure that the system does not fall into the local optimal solution.

A large number of simulation experiments prove that when the bird's nest groups values  $n = 15 \sim 40$  and the detection probability  $P_a = 0.25$ , it can solve many optimization

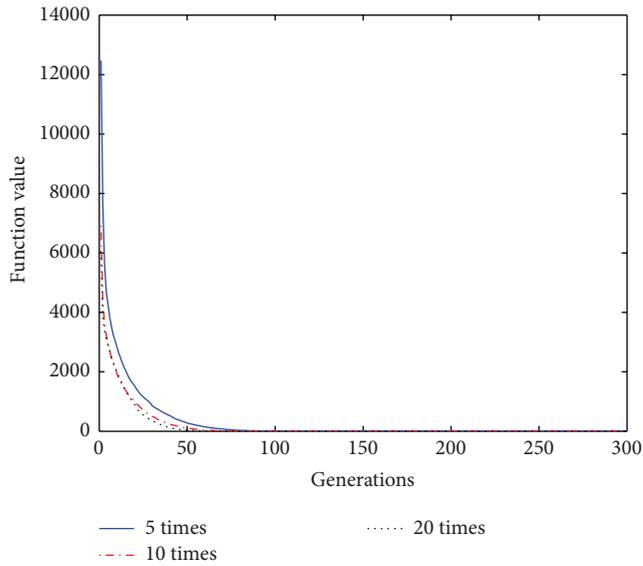


FIGURE 14: Convergence curves of Function  $f_1$  under different disturbance number.

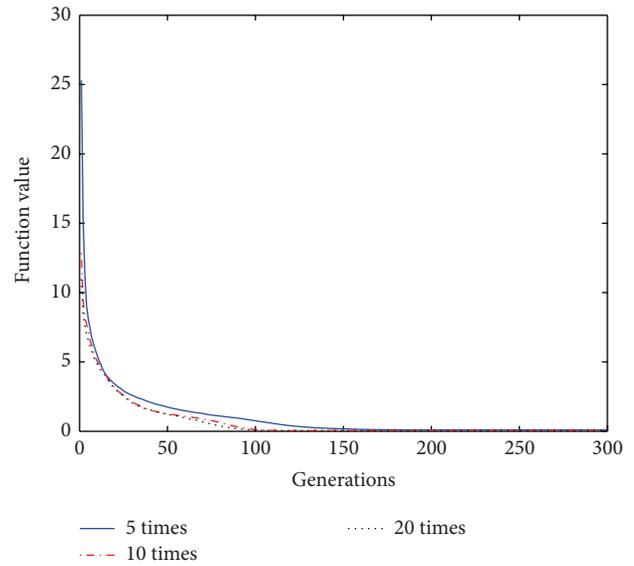


FIGURE 16: Convergence curves of Function  $f_3$  under different disturbance number.

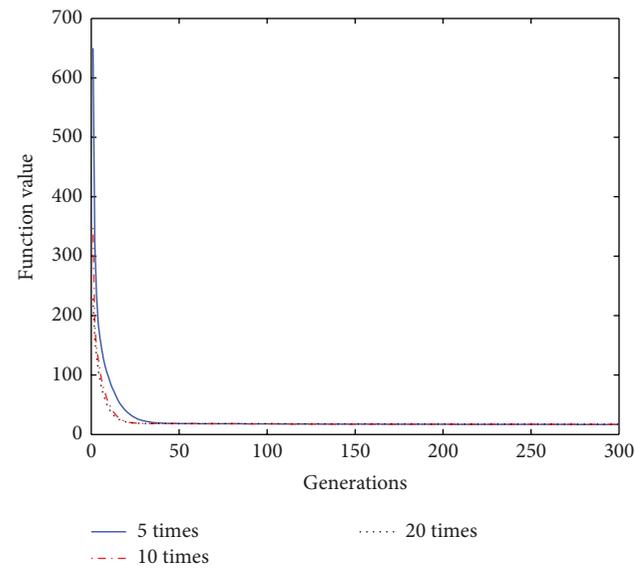


FIGURE 15: Convergence curves of Function  $f_2$  under different disturbance number.

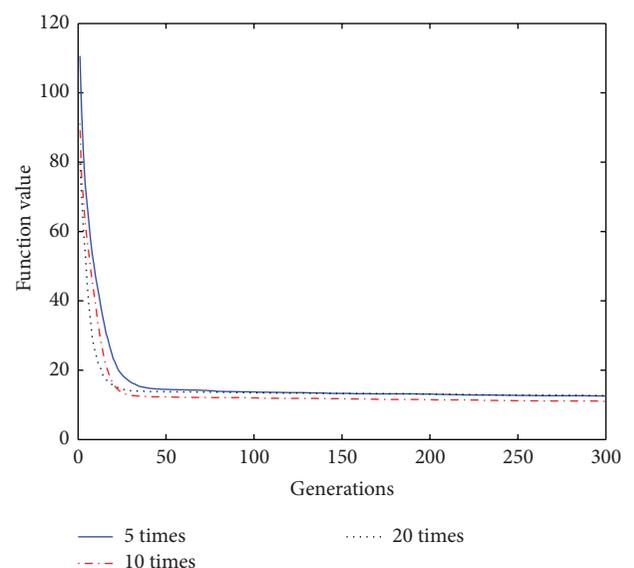


FIGURE 17: Convergence curves of Function  $f_4$  under different disturbance number.

problems [16]. Once the population size  $n$  is fixed, the discovery probability  $P_a$  is an important parameter to balance the global search and the local search and control the elite selection. Therefore, the CS algorithm has characteristic of less parameters, excellent searching path and strong global optimization ability, and so forth [20].

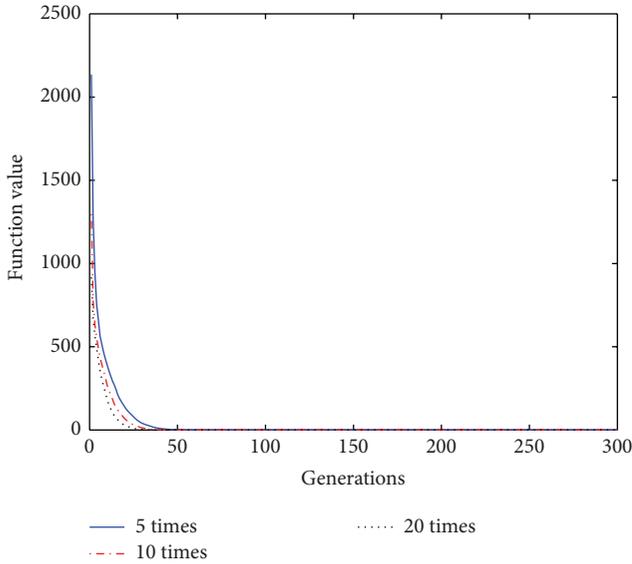
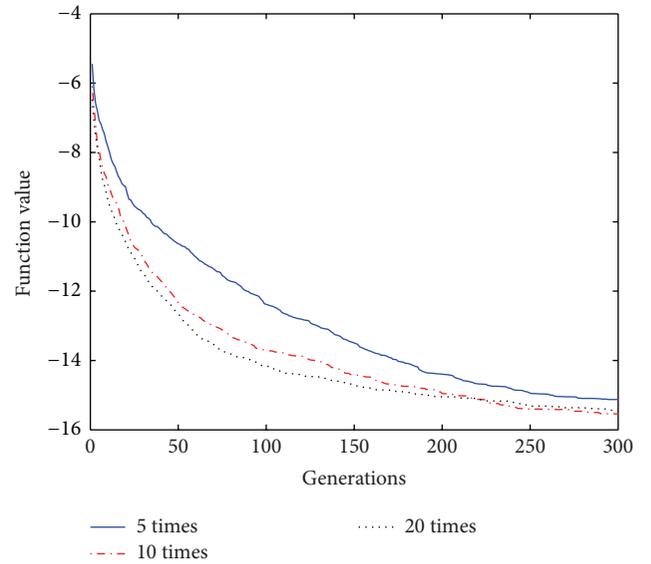
### 3. The Improved Cuckoo Search Algorithm

For each time, the length and direction of cuckoos' searching path are highly randomly changed based on Levy flight mechanism, so they are easy to jump from one region to

another, which is beneficial to the global search in the early stage of optimization and make the CS algorithm have strong global search ability [16, 21]. It is just because the CS algorithm shows a strong jumping in the search process that makes the local search around each bird's nest location no careful and no thorough. Therefore, the local optimization information near bird's nest location has not been fully utilized, which leads to that the local search ability is not strong, the optimization accuracy of the later period is not high, and the convergence speed is slow. In order to improve the convergence velocity and the optimization accuracy of the CS algorithm, a self-learning and self-evolving disturbance operation is added to

TABLE 1: Typical test functions.

Function name	Expression	Scope
Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	$[-100, 100]$
Rosenbrock	$f_2(x) = \sum_{i=1}^{d-1} (100(x_{i+1} - x_i^2) + (x_i - 1)^2)$	$[-2.08, 2.08]$
Griewank	$f_3(x) = \frac{1}{4000} \left( \sum_{i=1}^d (x_i^2) \right) - \left( \prod_{i=1}^d \cos \left( \frac{x_i}{\sqrt{i}} \right) \right) + 1$	$[-300, 300]$
Rastrigrin	$f_4(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-1.25, 1.25]$
Sumsquares	$f_5(x) = \sum_{i=1}^d i x_i^2$	$[-10, 10]$
Michalewicz	$f_6(x) = -\sum_{i=1}^d \sin(x_i) \left[ \sin \left( \frac{i x_i^2}{\pi} \right) \right]^{2m}, (m = 10)$	$[0, \pi]$

FIGURE 18: Convergence curves of Function  $f_5$  under different disturbance number.FIGURE 19: Convergence curves of Function  $f_6$  under different disturbance number.

the algorithm, and a further study for the improvement of disturbance is also discussed.

**3.1. Cuckoo Search Algorithm Based on Self-Learning and Self-Evolving Disturbance.** In order to make the algorithm carry on more careful and thorough searches near bird's nest locations, after each iteration of CS algorithm a set of obtained preponderant bird's nest locations  $X_i(t)$  instead of letting  $X_i(t)$  directly into the next iteration, a disturbance operation is applied to it for making a further search on the neighborhood of  $X_i(t)$ .

Due to the general disturbance, such as Gauss perturbation and random perturbation, all having the great randomness and blindness, in order to obtain a better disturbance effect, the learning and updating strategy of the worst frog in the shuffled frog leaping algorithm (SFLA) and a part of differential evolution (DE) thought are introduced into the

constructing of the disturbance factor. The introduction of learning and updating strategy of the worst frog can increase bird's self-learning ability of each bird's nest learning from the optimal nest [22]. That is to say it can increase the speed of other solutions approaching to the best solution and improve the algorithm's convergence rate. The introduction of the differential evolution thought can increase the diversity of bird's nest location, which makes every bird's nest have the evolution ability. The good learning and evolving ability are bound to cause high search ability.

Based on the above analysis, the disturbance factor is constituted of two parts: one part is the learning factor from SFLA learning and updating strategy of the worst frog; the other part is the evolution factor based on a part of differential evolution (DE) thoughts. Thus, the whole disturbance factor makes every disturbance with the effect of bird's self-learning

TABLE 2: Comparison of numerical testing results.

Function	Algorithm	Best	Worst	Mean
$f_1$	ABC	$3.1519e - 004$	0.0704	$9.0266e - 004$
	PSO	0.0048	7.0889	0.6284
	CS	0.0382	0.7001	0.2170
	GCS	$5.8627e - 004$	0.0047	0.0016
	SSCS	$4.8773e - 007$	$9.1317e - 005$	$1.6997e - 005$
	RC-SSCS	$3.6450e - 012$	$3.0061e - 005$	$1.7847e - 008$
$f_2$	ABC	11.3082	18.1271	13.7651
	PSO	10.4223	34.5360	17.4946
	CS	11.1806	17.3151	15.2039
	GCS	8.5219	16.8620	14.2017
	SSCS	6.2653	14.5681	13.3126
	RC-SSCS	3.3432	14.2079	10.8277
$f_3$	ABC	$4.6297e - 005$	0.6980	0.1859
	PSO	$5.8949e - 005$	1.8793	0.3322
	CS	0.0189	0.3452	0.2010
	GCS	$2.7105e - 004$	0.0077	0.1057
	SSCS	$2.2377e - 009$	0.3909	0.0814
	RC-SSCS	$1.0616e - 011$	0.1954	0.0454
$f_4$	ABC	5.2919	24.7673	10.3917
	PSO	6.0188	29.7667	18.5742
	CS	5.0454	11.4347	9.4766
	GCS	4.5346	9.4692	8.7840
	SSCS	1.2841	9.3480	6.0351
	RC-SSCS	0.6950	10.3400	5.8859
$f_5$	ABC	$2.5878e - 005$	$8.4891e - 004$	$3.3576e - 004$
	PSO	$5.6717e - 004$	4.2837	0.2335
	CS	$8.7353e - 004$	0.0068	0.0030
	GCS	$1.5714e - 004$	$9.2999e - 004$	$4.2927e - 004$
	SSCS	$9.3420e - 007$	$9.5495e - 004$	$9.8643e - 005$
	RC-SSCS	$2.4677e - 009$	$8.4555e - 004$	$5.6853e - 006$
$f_6$	ABC	-14.7812	-8.1498	-11.0840
	PSO	-15.1576	-7.8576	-10.8402
	CS	-12.7182	-10.1137	-11.3479
	GCS	-13.9297	-10.4565	-11.6951
	SSCS	-15.1779	-11.4004	-13.6241
	RC-SSCS	-18.0889	-13.1681	-16.7740

and self-evolving ability. Disturbance factor is structured as follows:

$$\varepsilon = c_1 u_1 (x_i - x_{\text{best}}) + c_2 u_2 (x_{r_1} - x_{r_2}), \quad (8)$$

where  $x_i$  is the  $i$ th disturbed bird's nest location,  $x_{\text{best}}$  is the current best location,  $r_1$  and  $r_2$  are random number from  $(1, n)$ ,  $r_1 \neq r_2$ ,  $n$  is the number of bird's nest population,  $x_{r_1}$  and  $x_{r_2}$  are the bird's nest locations corresponded to a random number  $r_1$  and  $r_2$ ,  $u_1$ , and  $u_2$  obey the Gaussian distribution,  $c_1$  is the learning scale, and  $c_2$  is evolution scale.

For better controlling of the disturbance range, a controlled quantity of disturbance scope  $\gamma$  is introduced to

control the search scope size around a bird's nest. After being disturbed, the bird's nest location is express as

$$x'_i = x_i + \gamma \otimes \varepsilon, \quad (9)$$

where  $x'_i$  is the bird's nest location after being disturbed,  $\gamma$  is the controlled quantity of the disturbance range, and  $\otimes$  is the point to point multiplication.

According to (8) and (9), after searching and selection operation obtain a bird's nest location  $x_i$ , do not let this nest go into the next generation directly. Instead, add to it a disturbance that take  $x_i$  as the foundation and  $\varepsilon$  disturbance factor, within the distribute scope that is controlled by  $\gamma$ . Finally after disturbance get a new bird's nest location  $x'_i$ , and let  $x'_i$  go into the next generation.

TABLE 3: Comparison of numerical testing results under different cycled disturbance number.

Function	Cycle number	Best	Worst	Mean
$f_1$	5	$1.0950e - 006$	$4.5927e - 004$	$8.4772e - 005$
	10	$8.0914e - 008$	$3.9593e - 005$	$3.3471e - 006$
	20	$8.1817e - 007$	$1.1113e - 004$	$2.4506e - 005$
$f_2$	5	15.5346	19.4028	18.2092
	10	14.5140	19.4748	16.6741
	20	14.9356	19.4993	16.6707
$f_3$	5	$2.0991e - 007$	1.2703	0.1498
	10	$1.0955e - 008$	0.3909	0.0847
	20	$3.1836e - 008$	1.3680	0.0847
$f_4$	5	5.9698	16.9143	12.4617
	10	1.9899	14.9247	11.2695
	20	1.9900	17.9210	11.7947
$f_5$	5	$1.5066e - 007$	$1.3715e - 004$	$5.1063e - 005$
	10	$5.5967e - 008$	0.0052	$3.6228e - 005$
	20	$3.2597e - 007$	0.0022	$2.9710e - 004$
$f_6$	5	-17.3839	-11.9059	-15.1020
	10	-18.2436	-12.5260	-15.5739
	20	-18.0307	-13.0349	-15.5256

In different stages, the bird nest's learning ability, the evolving ability, and the disturbance scope can be controlled by adjusting the values of  $c_1$ ,  $c_2$ , and  $\gamma$ . For example, when  $\gamma$  is a given value, if  $c_1 = 0$  and  $c_2$  is a certain constant, then within this scope of search, the bird's nest only has evolution ability, without learning ability; by the same token, if  $c_2 = 0$  and  $c_1$  is a certain constant, within the search scope, the algorithm only has learning ability and no evolving ability.

The disturbance range controller  $\gamma$  not only controls the search scope size, but also affects a bird nest's learning ability and evolution ability. When  $\gamma$  is given a big value, the coefficient before learning factor  $x_i - x_{\text{best}}$  and evolving factor  $x_{r_1} - x_{r_2}$  can be driven larger, so the learning and evolution ability become stronger. Under this condition, the search scope of the algorithm is larger, the search ability is stronger but the search fineness is lower. By the same token, when  $\gamma$  is given a small value, the learning and evolution ability are relatively weaker, the search scope is smaller, the search ability becomes weaker, but the search fineness is much higher.

**3.2. Cuckoo Search Algorithm Based on Repeat-Cycle Asymptotic Self-Learning and Self-Evolving Disturbance.** The ideal disturbance effect should be that in the early stage of disturbances, it has higher searching ability and in the later stage of disturbances has higher search accuracy. In this way, it can obtain a better bird's nest location fast and then carry on a more careful and thorough search around the better nest location. Based on the above analysis, a repeat-cycle asymptotic disturbance method is proposed. Adopting a dynamic adjustment  $\gamma$  makes the search scope gradually change from big to small after disturbance. In other words, based on the results of last disturbance, narrowing the disturbance scope, go on the next disturbance. The repeat-cycle asymptotic disturbance search is carried out in turn.

In order to obtain a better bird's nest position at a faster speed, at the beginning of disturbance  $\gamma$  is given a larger value. With the disturbance continuing, the bird's nests gradually get better and the adjustment of bird's nest locations is more and more subtle. Particularly in the condition that the response of fitness value is sensitive to parameter changing, it needs to use a very small amount of control  $\gamma$  to make the position have a fine-tuning near optimal value. The  $\gamma$  is adjusted according to the following:

$$\gamma = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \times \frac{N - n}{N}, \quad (10)$$

where  $\gamma_{\min}$  is the minimum value of control amount of disturbance scope,  $\gamma_{\max}$  is the maximum value,  $N$  is the total number of repeat-cycle disturbances, and  $n$  is the  $n$ th disturbance.

In the circulation disturbance, the controlled quantity of the disturbance range  $\gamma$  is controlled by disturbance number  $n$ . Make  $\gamma$  changes between  $\gamma_{\max}$  and  $\gamma_{\min}$ . When  $n = 1$ ,  $\gamma$  is maximum  $\gamma = \gamma_{\max}$ . With the disturbance number  $n$  increasing gradually  $\gamma$  decrease little by little. Finally when  $n = N$ ,  $\gamma$  reaches minimum value  $\gamma = \gamma_{\min}$ .

In the early stage of the repeat-cycle disturbances, the bird's nest self-learning and self-evolving function plays a leading role, which can make it surely find a better bird's nest faster. With disturbance number increasing, the search scope gradually turns smaller and the fineness degree of search is gradually enhanced. In the later stage of the repeat-cycle disturbances, the fineness search plays a leading role.

**3.3. Algorithm Procedure.** The algorithm procedure of the improved cuckoo search algorithm (RC-SSCS) is shown in Figure 1. The specific steps of RC-SSCS algorithm are described as follows.

*Step 1* (initialization). Randomly generate  $N$  bird's nest location  $X^0 = (x_1^0, x_2^0, \dots, x_N^0)$ , select the best bird's nest location, and carry it over to the next generation.

*Step 2* (searching operation). Use the location update (1) to search for the next generation bird's nest position, obtain a new set of bird's nests, and test them. Then compare the testing results with the bird's nest position of previous generation and obtain a set of better positions.

*Step 3* (selection operation). Generate the random number  $r \in [0, 1]$ , which obeys the uniform distribution. Contrast it with the detection probability  $P_a = 0.25$ , if  $r > P_a$ ,  $x_i^{(t+1)}$  is changed randomly, otherwise  $x_i^{(t+1)}$  is unchanged. Test the changed nest positions, compare them with locations of the last step, and choose the better nest locations.

*Step 4* (repeat-cycle disturbance operation). Add a self-learning and self-evolving disturbance to each bird's nest location, test new bird's nest locations that have been disturbed, and then compare the test results with locations of the last disturbance results and choose the better bird's nest locations. After many times disturbance, obtain a set of the best bird's nest locations, and then choose a best location  $pb$  from the set.

*Step 5* (judgment operation). Calculate the fitness value of  $f(pb)$  and judge whether it achieves the termination condition. If it is satisfied,  $pb$  is the optimal solution, otherwise return to Step 2 and start the next iteration.

#### 4. Simulation Results and Analysis

In order to verify the performances of the improved CS algorithm, six typical continuous test functions are chosen for carrying out the simulation research, meanwhile, which is compare simulation results with ABC, PSO, CS and GCS. These six test functions are shown in Table 1. Their 3D surface figures are shown in Figures 2–7.

Sphere function is a simple unimodal function. Rosenbrock function is an inseparable single mode function, and its global extreme value is in steep valleys. For the most search algorithms, it is difficult to acquire the right search direction within the canyon. Griewank function is a multimodal function with multiple local optimal points, and due to the correlation between variables, it will be very hard to obtain the global optimal solution. Rastrigrin function is a typical inseparable multimodal function, and in its searching domain, there are a large number of local minimum values, which leads to the fact that it is difficult to obtain the global optimum. Michalewicz function has  $d!$  local extreme values.

The experimental parameters are set as follows. For particle swarm optimization (PSO) algorithm the particle number is  $n = 30$ ; learning factors  $c_1 = 2$ ,  $c_2 = 2$ ; the inertia weight  $w = 0.9$ . For artificial bee colony (ABC) algorithm total number of colonies is  $n = 20$ , the number of following bees and leading bees is the same  $n/2 = 10$ . For cuckoo search (CS) algorithm and its 3 improved CS algorithm the

total bird's nest population  $n = 25$ , the detection probability  $P_a = 0.25$ , the step length controlled parameter  $\alpha = 0.01$ . The cuckoo search algorithm based on Gauss disturbance (GCS) and the cuckoo search algorithm based on self-learning and self-evolving disturbance (SSCS) adopt the same disturbance scope control quantity  $\gamma = 0.75$ . For the cuckoo search algorithm based on repeat-cycle asymptotic self-learning and self-evolving disturbance (RC-SSCS), the scope of  $\gamma$  is  $[0.25, 1.5]$ , the scale of learning and evolution is set  $c_1 = c_2 = 0.75$ . The number of the circulation disturbance  $N = 10$ . For all algorithms, the dimension of these six test functions is all set  $D = 20$ . The number of iterations  $iter = 500$ . For each algorithm, its program run 30 times independently.

Evaluate the performances of algorithms through statistics of the best value, average value and worst value among 30 times running, and the convergence curves of each function. The convergence curves of six functions  $f_1$ – $f_6$  are shown in Figures 8–13. The numerical test results of each algorithm are shown in Table 2.

After 500 iterations and 30 times independently running, it can be seen from the convergence curves and numerical statistics results of six functions that the convergence rate and the optimization accuracy of RC-SSCS algorithm is the best. And the convergence rate and optimization accuracy of the two algorithms SSCS and RC-SSCS proposed in this paper are obviously better than the original CS algorithm, GCS algorithm, ABC algorithm, and PSO algorithm.

Seen from six convergence curves, the convergence rate of the six functions is all obviously improved. The convergence curves of function  $f_2$  show that the optimization accuracy achieved by RC-SSCS algorithm after 20 times iteration is equal to that achieved by GCS algorithm and CS algorithm after 200 iterations. The function  $f_3$  convergence curve shows that the optimization accuracy by RC-SSCS algorithm after 100 iterations reached is equal to the accuracy by original CS algorithm after 400 iterations reached. From the convergence curve of function  $f_4$ , It can be seen that the optimization accuracy achieved by RC-SSCS algorithm after 25 iterations is equal to that by SSCS algorithm after 175 iterations achieved and by CS algorithm after 300 iterations achieved. The convergence rate of function  $f_5$  and  $f_6$  is also changed obviously. The results show that the improved algorithm makes the convergence rate be greatly improved.

Seen from the numerical results of Table 2, the improved algorithm makes the optimization accuracy of six typical functions be improved. The SSCS algorithm and RC-SSCS algorithm relative to CS algorithm, respectively, make the best value of single-mode function  $f_1$  increased by 5 and 10 orders of magnitude, and the average value, respectively, increased by 4 and 7 orders of magnitude. For multimode function  $f_3$  with multiple local optimal points, compared with the original CS algorithm, SSCS, and RC-SSCS algorithm, respectively, make its best value and average value improved by 5 and 9 orders of magnitude. Optimization accuracy of other functions has been improved. It shows that the improved algorithm can make the optimization accuracy be improved.

Based on the above analysis of the improved algorithm, it is known that, within a certain range of disturbance, the

number of the repeat-cycle disturbance affects the balance between the search capability and the optimization accuracy in disturbance process. An appropriate disturbance time can make a good balance between search ability and search accuracy and play a best optimization effect. In process of disturbance search, if the algorithm only has strong search ability but no high search precision, it will not get an ideal search effect. Similarly, if the precision is very high, but search ability is weak, it will be also no ideal search effect. In order to select a reasonable disturbance time, in this paper we carry out the further studies on the relationship between the repeat-cycle disturbance times and the convergence rate and optimization precision. The same parameter settings are chosen as described above, except the number of iterations set  $\text{iter} = 300$ . The cycle disturbances to each function are carried out 5 times, 10 times, and 20 times, respectively. The results of the six functions under different disturbance are shown in Figures 14–19 and Table 3.

It can be seen from the convergence curves in Figures 14–19 that with the increase of cycle's times, the convergence rate of six functions will be gradually improved. But by looking carefully at each function convergence curve, it can be discovered that the changing of convergence rate when number of loops increases from 5 to 10 is bigger than that when loops number increases from 10 to 20. That is to say, the changing of convergence produced by increasing 5 times loops in front of repeat-cycle disturbance is bigger than that by increasing 10 times in later of it. It also can be seen from the numerical results in Table 3 that the optimization accuracy is the highest when cycle disturbance times is 10. The optimization accuracy of six functions is all improved when the disturbance times are increased from 5 to 10. However, when the disturbance times are increased from 10 to 20, the optimization accuracy all decreases instead of increasing.

In conclusion, the more disturbance times may not obtain the better results. Within the same disturbance scope, when the disturbance times reach a certain number, if the disturbance number increases again, the convergence speed of the algorithm does not have an obvious improvement. However, the optimization accuracy will be reduced. Integrally considering the convergence rate, the optimization accuracy and the optimizing time, it is better to choose the cycled disturbance times about 10.

## 5. Conclusion

In order to improve the convergence rate and optimization accuracy of the cuckoo search (CS) algorithm for solving function optimization problems, a kind of cuckoo search algorithm based on the repeat-cycle asymptotic self-learning and self-evolving disturbance (RC-SSCS) is proposed. Six typical test functions are chosen for simulation experiments. Simulation results show the effectiveness of the proposed improved cuckoo search algorithm in convergence rate and optimization accuracy.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

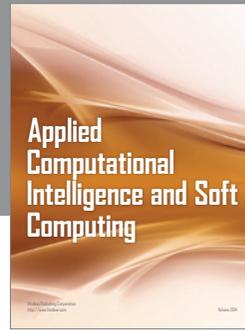
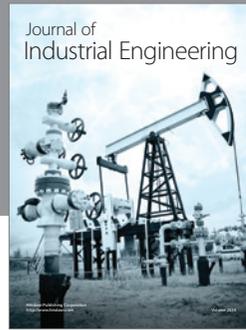
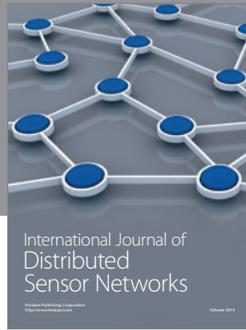
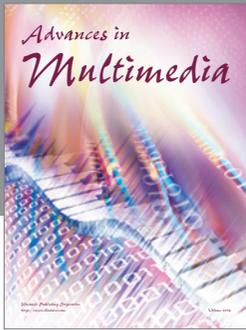
## Acknowledgments

This work is partially supported by The Program for China Postdoctoral Science Foundation (Grant no. 20110491510), The Program for Liaoning Excellent Talents in University (Grant no. LR2014008), and The Project by Liaoning Provincial Natural Science Foundation of China (Grant no. 2014020177).

## References

- [1] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, IEEE Publications, December 2009.
- [2] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.
- [3] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research*, vol. 40, no. 6, pp. 1616–1624, 2013.
- [4] X. S. Yang, "Cuckoo search and firefly algorithm: overview and analysis," in *Cuckoo Search and Firefly Algorithm*, vol. 516 of *Studies in Computational Intelligence*, pp. 1–26, Springer International Publishing, Cham, Switzerland, 2014.
- [5] E. Valian, S. Mohanna, and S. Tavakoli, "Improved cuckoo search algorithm for feedforward neural network training," *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 3, pp. 36–43, 2011.
- [6] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, 2014.
- [7] A. Mallick, S. Roy, S. S. Chaudhuri, and S. Roy, "Study of parametric optimization of the Cuckoo Search algorithm," in *Proceedings of the International Conference on Control, Instrumentation, Energy and Communication (CIEC '14)*, pp. 767–772, Calcutta, India, January–February 2014.
- [8] P. Civicioglu and E. Besdok, "Comparative analysis of the cuckoo search algorithm," in *Cuckoo Search and Firefly Algorithm*, vol. 516 of *Studies in Computational Intelligence*, pp. 85–113, Springer International Publishing, 2014.
- [9] G.-G. Wang, A. H. Gandomi, X. Zhao, and H. C. E. Chu, "Hybridizing harmony search algorithm with cuckoo search for global numerical optimization," *Soft Computing*, 2014.
- [10] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1659–1669, 2014.
- [11] C.-W. Qu, Y.-M. Fu, and X.-L. Hang, "Cuckoo optimal algorithm based on exchange operator and chaotic disturbance," *Journal of Chinese Computer Systems*, vol. 2, no. 27, pp. 384–387, 2014.
- [12] X.-X. Hu and Y.-L. Yin, "Cooperative co-evolutionary cuckoo search algorithm for continuous function optimization problems," *Pattern Recognition and Artificial Intelligence*, vol. 26, no. 11, pp. 1041–1049, 2013.

- [13] H. Zheng and Y. Zhou, "A novel cuckoo search optimization algorithm based on Gauss distribution," *Journal of Computational Information Systems*, vol. 8, no. 10, pp. 4193–4200, 2012.
- [14] W.-Y. Qian, H.-C. Hou, and S.-Y. Jiang, "New self-adaptive cuckoo search algorithm," *Computer Science*, vol. 41, no. 7, pp. 279–282, 2014.
- [15] A. R. Yildiz, "Cuckoo search algorithm for the selection of optimal machining parameters in milling operations," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 1–4, pp. 55–61, 2013.
- [16] 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 Systems with Applications*, vol. 41, no. 7, pp. 3538–3560, 2014.
- [17] J. Ahmed and Z. Salam, "A Maximum Power Point Tracking (MPPT) for PV system using Cuckoo Search with partial shading capability," *Applied Energy*, vol. 119, pp. 118–130, 2014.
- [18] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.
- [19] A. Kaveh and T. Bakhshpoori, "Optimum design of steel frames using Cuckoo Search algorithm with Lévy flights," *The Structural Design of Tall and Special Buildings*, vol. 22, no. 13, pp. 1023–1036, 2013.
- [20] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 315–346, 2013.
- [21] R. R. Bulatović, S. R. Dordević, and V. S. Dordević, "Cuckoo Search algorithm: a metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage," *Mechanism and Machine Theory*, vol. 61, pp. 1–13, 2013.
- [22] J. Luo and M.-R. Chen, "Improved shuffled Frog Leaping algorithm and its multi-phase model for multi-depot vehicle routing problem," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2535–2545, 2014.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

