Chaotic Cuckoo Search Algorithm for High-dimensional Functions

Aijia Ouyang[a,b]
a. College of Computer, Hunan Science & Technology Economy Trade Vocation College, Hengyang, China
b. College of Information Science and Engineering, Hunan University, Changsha, China
Email: ouyangaijia@163.com

Guo Pan[a,b,*]
a. Logistics Information Dept., Hunan Vocational College of Modern Logistics, Changsha, China
b. College of Information Science and Engineering, Hunan University, Changsha, China
Email: 690849909@qq.com

Guangxue Yue[a,b] and Jiayi Du[b]
a. College of Mathematics, Physics and Information Engineering, Jiaxing University, Jiaxing, China
b. College of Information Science and Engineering, Hunan University, Changsha, China
Email: {guangxueyue, dujiayi}@163.com

***Abstract*—Given that the basic cuckoo search algorithm is vulnerable to local optimum and unsatisfactory calculation precision, chaotic operator (CO), employed as local algorism to optimize elitist individuals in the population, can effectively enhance the properties of cuckoo search (CS) algorithm. Tested by 15 benchmark high-dimensional functions, the experiment result indicates that chaotic cuckoo search (CCS) algorithm can increase the calculation precision and step up the convergent speed with improved robustness, which can be applied to other engineering optimization problems.**

***Index Terms*—chaos, cuckoo search, lévy flight, high-dimensional function, optimization**

I. INTRODUCTION

Swarm intelligence algorithm is an efficient method to solve the global optimization problems. It mainly includes particle swarm optimization (PSO) [1], artificial immune (AI) [2], genetic algorithm (GA) [3], differential evolution (DE) [4], invasive weed optimization (IWO)[5] and so on.

After having studied the reproductive behavior of cuckoos and flying properties of lévy, Yang from Cambridge University puts forward cuckoo search algorithm, the properties of which were tested by many functions [6]. It reveals that the algorithm exceeds particle swarm algorithm and genetic algorithm with more powerful global searching ability and solving ability of multi-objective problems, fewer selected parameter as well as better search path [7]. It has been widely used in scientific research and industry. Multi-objective cuckoo search algorithm is applied to solve engineering design optimization in Literature [8] and Jiles-Atherton vector hysteresis parameters estimation problem in Literature [9] respectively. Directed cuckoo search algorithm successfully deals with the sheet nesting problem in Literature [10]. Cuckoo search algorithm handles distributed generation allocation problem so as to reduce power consumption in Literature [11] and image segmentation problem in Literature [12] respectively.

Chaos is an essential characteristic of nonlinear system with a series of particular features such as randomness, ergodicity and regularity, the discovery of which has profound effects on scientific development [13]. As an effective mechanism to avoid being in local optimum, chaos has been introduced to evolutionary computation, providing a new research field and application method for it [14, 15]. Most researches involved, however, only replace the random sequence in mutation operator with the chaos sequence to indicate that chaotic mutation is an effective realization of mutation operator with real number code evolutionary algorithm . Although clear to understand, simple to implement and easy to adapt itself, these algorithms still have some problems as chaos is not made full use of, such as neglecting the regularity of chaos and little use of prior knowledge to improve the local search ability of the algorithm [16].

In this paper, chaos optimization method is applied to make cuckoo search algorithm free from local optimum. It is discovered that rather than in a complicated mess, chaos enjoys fine internal structure. As a singular attractor, chaos attracts systematic movement and confines it to a specific range. Not following a particular orbit, particles of the system wander the chaotic area with freedom, so its future status is unpredictable. Even a slight difference of the original condition will cause immense change. Therefore, two aspects are being studied at present. First, chaos is not expected, and systematic movement is controlled to make chaos

∗ Corresponding author, Guo Pan, E-mail: panguo@hnu.edu.cn

doi:10.4304/jcp.9.5.1282-1290

unlikely to appear. Second, chaos is regarded as something beneficial, which uses less energy to make greater benefits. Meanwhile, characterized with ergodicity, chaos movement is ergodic with its own regularity in a certain range. As is seen in Literature [17] and Literature [18], using chaos variables to obtain optimization search can undoubtedly get rid of local optimum. A hybrid optimization algorithm is proposed in this paper by combining chaos optimization algorithm with cuckoo search algorithm, which can enhance the optimization efficiency by avoiding the disadvantage of all the statuses are ergodic.

The paper consists of five sections, the second describes the basic cuckoo search algorithm and chaos operator, the third gives the process of the chaotic cuckoo search algorithm, the fourth deals with the experimental results and analysis, and the last is the conclusion.

II. THE BASIC ALGORITHM

A. Cuckoo Search Algorithm

1) Reproductive behavior of cuckoos

According to entomologists' observation, some cuckoos breed their offspring by brood parasitism, the way certain birds lay eggs in other birds' nest and have them hatched and raised by other birds. Cuckoos are often alone. They seek hosts with similar hatching and breeding period, feeding habit and egg shape and color, mainly passerine birds. When the host is away from the nest, the cuckoo will quickly lay an egg in it, one every time. Mistaking the egg with its own because of the obscure impression, the host is then in charge of hatching it. The cuckoo always removes one or all the eggs of the host to force it to relay eggs. The newly-born cuckoo tends to push out the host's own baby bird so as to enjoy exclusive breeding.

2) Lévy flight

Viswanathan and some other researchers proved that the albatross adopts the pattern of Lévy flight for foraging. By using satellite positioning system they discovered their flying intervals follow power-law distribution, explained their findings through the space distribution property of invariable food scale on the sea surface and published some symbolic papers [19-21]. After having researched the foraging path of bees [22] and drosophilas [23], Reynolds discovered the appearing occurrence of the straight line portion in their flight path corresponds with scale-free inverse square of lévy distribution in that they both take on lévy flight properties. When the target positions are distributed randomly and sparsely, lévy flight is the ideal search strategy to M independent explorers [24]. In addition, lévy flight is found in creatures such as spider monkeys, gray seals and reindeer as well in human being's behavior [25,26]. Lévy flight is a sort of random walk. The step size meets a heavy-tailed stable distribution. Short-distance exploration and occasional long-distance walk interphase in it. Lévy flight used in intelligent optimization algorithm can enlarge search area, increase population diversity and jump out of local optimal point easily [27].

3) Features of the algorithm

Cuckoo algorithm is evolved form cuckoo's reproductive behavior and lévy flight mechanism. Cuckoos have special reproductive behavior in the nature. They lay eggs in the nest of other birds on which they rely to hatch the eggs. If the host bird spots the secret, severe collision will take place. If the host finds that the egg is not its own, it will remove the egg or rebuild a nest to breed its own offspring. Therefore, the cuckoo inevitably lays eggs the instant the host lays its own in the nest [28]. Once the cuckoo egg is kept in the nest, the host will hatch both the eggs at the same time. In addition, the cuckoo egg always enjoys the priority to be hatched. The newly-born cuckoo baby has an intuition to push out other birds' eggs, and in this way, it enjoys the exclusive breeding [29, 30]. Moreover, the flight of many animals and insects follow the flight properties [23]. Inspired by the two phenomena, Xin-She Yang as well as some other researchers present cuckoo search algorithm and make the following three ideal assumption,

a) Each cuckoo lays only one egg in a certain nest at random.

b) The cuckoo egg in the host nest with high quality will be hatched and then reproduce cuckoos of the next generation.

c) n standing for the number of the host nests used by the cuckoo is determined. The probability that the cuckoo egg is spotted by the host is $pa \in [0,1]$.

Given the three presumptions, the nest-seeking path and position updating formula can be expressed as

$$x_{ij}^{m+1} = x_{ij}^{m} + \alpha \times \text{Levy}(\lambda) \qquad (1)$$

where x_{ij}^{m} and x_{ij}^{m+1} stand for the position of dimension j(j=1, 2, …, d) of the nest i (i=1, 2, …, n) in the generation of m and m+1, and lévy(λ) is the skipping path of the flight search at random. The distance and direction of the path is undetermined, and sometimes the skipping path lévy(λ) is considerably long. To make it applied in CS algorithm successfully, Literature [6] defines an adaptive amount α, which is a constant more than zero. The value of it varies, and generally, α=0.01.

Lévy distribution was put forward by French mathematician Lévy in the 1930s. It is believed that the relationship between the continuous skipping path of lévy flight and the time t follows lévy distribution. Many other scholars conduct research on it and try to explain randomly phenomenon in the nature with it, such as Brownian Movement and random walk. Yang obtained probability density function in the forms of power through simplified distribution function and Fourier transform,

$$\text{levy} \sim u = t^{-\lambda};\ 1<\lambda<3 \qquad (2)$$

Here, λ is the power coefficient. Formula 2 is a probability distribution with heavy tail. Although it describes the randomly wandering process of the cuckoo essentially, it fails to describe the distribution mathematically with simplicity and easy programming to realize the algorithm. Yang, therefore, adopted formula of lévy flight skipping path put forward by Mantegna in 1992 and realized the algorithm [31].

$$s = \frac{\mu}{|v|^{1/\beta}} \quad (3)$$

In Formula 3, s is the lévy flight skipping path lévy(λ) . The relation between Parameter β and λ in Formula 2 is λ=1+β. The value of β is 0<β<2. Make β=1.5[7] in CS algorithm. Parameter μ and ν are random numbers of normal distribution, following the normal distribution expressed in Formula 4. The value of corresponding Standard deviation of normal distribution σ_μ and σ_υ is reflected in formula 5.

$$\begin{cases} \mu \sim N(0,\sigma_\mu^2) \\ \nu \sim N(0,\sigma_\nu^2) \end{cases} \quad (4)$$

$$\begin{cases} \sigma_\mu = \{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]2^{(\beta-1)/2}\beta}\} \\ \sigma_\nu = 1 \end{cases} \quad (5)$$

Make *Step*=α×lévy(λ), and *Step* is the path in which the cuckoo starts form the old nest position x_{ij}^m and searches for new nest position x_{ij}^{m+1} in the solution space by Formula 1. As lévy(λ) depends on the two normal distribution random number μ and ν, which can be various, passive and negative, the path length and direction of the cuckoo following lévy flight search mechanism varies randomly, which is apt to jump form one region to another, making the global optimization ability of CS algorithm extremely powerful. CS algorithm learns from the reproductive behavior of the cuckoo, defines the probability that the cuckoo egg will be spotted by the host as *pa*=0.25[6]. The cuckoo egg with worse adaptive ability will be ruled out, while the one with better adaptation will be hatched, making the newly-born cuckoos consists of excellent individuals. Thus CS algorithm enjoys great contraction ability. In actual optimization problems, the position of the nest x_{ij} stands for effective value space of all the variables, and the fitness of the nest represents the corresponding objective function when the value of the variable varies. Details of CS algorithm go to Literature [6] and [7].

B. Chaos Operator

Chaos is a unique movement pattern of nonlinear system with particular features of sensitivity to the initial value, randomness and ergodicity. If the features are made full use of, the optimization solution of CS algorithm will be promoted.

Chaotic search generates by iteration chaos sequence through certain particular format. Extend the numerical range of the chaos variables to the value range of the optimization variables through the form of carrier wave.

The math procedure of chaotic search is as follows:

If some individual pauses, d-dimension is generated and it randomly initializes vectors $y_0=[y_{0,1}, y_{0,2}, \ldots, y_{0,D}]'$, $y_{0,d}\in[0,1]$. There lie slight differences in the values. As the iteration initial value, Vector y_0 starts the chaos sequence iteration according to Logistic equation.

$$y_{n+1,d} = \mu y_{n,d}(1-y_{n,d}) \quad (6)$$

Thus iteration sequence $y_{n,d}$ is obtained. In the formula, n=0, 1,…, $N_{\max}$; d=0, 1,…, D.

The formula can generate many fields around the local optimal solution by iteration. Through the form of carrier wave, and according to Eq.(7),

$$y_{n,d}' = x_{i,d} + R_{i,d}(2y_{n,d}-1) \quad (7)$$

Chaos iteration variable $y_{n,d}$ is transformed into optimization variable $y_{n,d}'$,namely, extending the value of chaos variable $y_{n,d}$ to a region where the current position of the individual $x_{i,d}$ is made to be the center and $R_{i,d}$ as to be the radius. $R_{i,d}$ is the chaotic search radius and $y_{n,d}'$ is determined by the initialized range of the function variable $x_{i,d}$. The value range is as follows,

$$y_{n,d}' \in [x_{i,d} - R_{i,d}, x_{i,d} + R_{i,d}] \quad (8)$$

Compute the adaptive value of the function $f(y_n')$, and update the historical optimal adaptive value $f*$ in the chaos iteration process and the historical optimal position x_i^* . If $f*$ is superior to F_i , position x_i^* and velocity v_i^* are used to replace the original position and velocity of the individual. Here,

$$v_i^* = \frac{x_i^* - x_i}{\|x_i^* - x_i\|} \quad (9)$$

III. Chaotic Cuckoo Search Algorithm

Although chaotic search can avoid being caught in local minimum because of its ergodicity, pure chaotic search can obtain good solution only through huge iteration step numbers and it is sensitive to initial solution in particular. Therefore, a two-stage chaotic CS algorithm is put forward by combining CS algorithm with the above chaotic search, in which CS algorithm is used to lead global search and CO leads local search according to the result of CS algorithm. In order to maintain population diversity and strengthen the dispersion of the search, the algorithm keeps some superior individuals, dynamically contracts search range in view of the best position of the population, and replaces the worse nest position with the one generated in the contract region randomly. The steps of chaotic cuckoo search algorithm can be described as follows,

Step 1. The objective function is $f(X)$, and $X=(x_1, \ldots, x_d)^T$. Initialize the population, generate randomly n initialized positions $X_i (i = 1, 2, \ldots, n)$, and set up parameters of the algorithm.

Step2. Compute the objective function of every bird nest, and record the current optimum solution.

Step 3. Maintain the optimal nest position of previous generation, and update other nest positions according to the position updating formula (1).

Step4. Compare the current bird nest position with that of the previous nest. If better, it is made to be the current optimal position.

Step 5. R stands for the possibility that the nest host will recognize the cuckoo egg. Compare it with probability P_a. If $R>P_a$, change the nest position randomly and obtain a set of new nest positions.

Step 6. Maintain 20% nest positions with the optimal properties in the population.

Step 7. Conduct CO search to the optimal nest position in the population, and update the new nest position.

Step 8. If the stopping criterion of the algorithm is met, output the optimal nest position, otherwise, continue the following steps.

Step 9. Contract search region according to formula (10) and (11)

$$x_{ij}^{\min} = \max\{x_{ij}^{\min}, x_{g,j} - r(x_{ij}^{\max} - x_{ij}^{\min})\} \quad (10)$$

$$x_{ij}^{\max} = \min\{x_{ij}^{\max}, x_{g,j} + r(x_{ij}^{\max} - x_{ij}^{\min})\} \quad (11)$$

Step 10. Generate the rest 80% nest positions of the population randomly in the contracted space, and evaluate it. If the stopping criterion is not met, return to Step 2.

Step 11. Output the global optimal position.

IV. Experiments and Analyses

A. Benchmark Functions

Fifteen widely used test functions are chosen from [32], and the proposed algorithm in this paper, genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO) and cuckoo search (CS) are executed for them. These functions are shown in the equations as follows.

Note that functions F1-F9 have many local minima so that they are challenging enough for performance evaluation. For example, F7 has *n*! local minima and both F8 and F9 have 2*n* local minima, where *n*=100.

$$F_1 = \sum_{i=1}^{d} -x_i \sin(\sqrt{|x_i|}) \qquad F_2 = \sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i) + 10) \qquad F_3 = -20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{d}\sqrt{\sum_{i=1}^{d}\cos(2\pi x_i)}) + 20 + \exp(1)$$

$$F_4 = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}}) + 1 \qquad F_5 = \frac{\pi}{d}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{d-1}(y_i - 1)^2 \bullet [1 + 10\sin^2(\pi y_{i+1})] + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10, 100, 4)$$

Where, $y_i = 1 + \frac{1}{4}(x_i + 1)$， $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$

$$F_6 = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{d-1}(x_i - 1)^2 \bullet [1 + \sin^2(3\pi x_{i+1})] + (x_d - 1)^2[1 + \sin^2(2\pi x_d)]\right\} + \sum_{i=1}^{d} u(x_i, 5, 100, 4) \qquad F_7 = -\sum_{i=1}^{d}\sin(x_i)\sin^{20}(\frac{i \times x_i^2}{\pi})$$

$$F_8 = \sum_{i=1}^{d}\left[\sum_{j=1}^{d}(\chi_{ij}\sin\omega_j + \psi_{ij}\cos\omega_j) - \sum_{j=1}^{d}(\chi_{ij}\sin x_j + \psi_{ij}\cos x_j)\right]^2 \qquad F_9 = \frac{1}{d}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i) \qquad F_{10} = \sum_{i=1}^{d-1}\left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$$

$$F_{11} = \sum_{i=1}^{d} x_i^2 \qquad F_{12} = \sum_{i=1}^{d} x_i^4 + \mathrm{rand}[0,1) \qquad F_{13} = \sum_{i=1}^{d}|x_i| + \prod_{i=1}^{d}|x_i| \qquad F_{14} = \sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2 \qquad F_{15} = \max\left\{|x_i|, i = 1, 2, \cdots, d\right\}$$

B. Experimental Results and Discussion

TABLE I
COMPARISON OF RESULTS BY FIVE METHODS FOR F1-F5 ON 30 DIMENSION FUNCTION

Function	Algorithm	Best	Mean	Worst	Std.	Convergence
F1	GA	-9.8969E+03	-9.6331E+03	-9.3970E+03	2.0421E+02	195
	DE	-1.2569E+04	-1.2569E+04	-1.2569E+04	**0.0000E+00**	84
	PSO	-3.4249E+03	-3.2998E+03	-3.0110E+03	1.7331E+02	123
	CS	**-9.9789E+03**	**-9.6541E+03**	**-8.8397E+03**	4.5917E+02	141
	CCS	-1.2560E+04	-1.2034E+04	-1.1785E+04	3.2312E+02	**78**
F2	GA	2.0011E+01	2.2439E+01	2.4610E+01	1.8786E+00	203
	DE	9.8814E+00	1.0520E+01	1.2718E+01	1.2151E+00	99
	PSO	4.3767E+01	5.1612E+01	5.3535E+01	4.2250E+00	141
	CS	1.3770E-01	1.1153E+00	2.3152E+00	8.9050E-01	170
	CCS	**2.6545E-05**	**5.1887E-05**	**8.2446E-05**	**2.2871E-05**	**91**
F3	GA	8.7461E+00	1.0836E+01	1.1443E+01	1.1550E+00	232
	DE	1.9000E-06	2.0200E-06	2.0700E-06	7.1300E-08	121
	PSO	4.4440E-01	4.9860E-01	5.2340E-01	3.2988E-02	173
	CS	6.9286E-05	2.3578E-04	4.5698E-03	2.0830E-03	189
	CCS	**2.1316E-14**	**5.8776E-14**	**8.8284E-14**	**2.5200E-14**	**108**
F4	GA	1.7719E+01	2.8074E+01	2.9334E+01	5.2037E+00	217
	DE	3.1600E-12	3.7300E-12	1.4800E-11	5.3600E-12	117
	PSO	1.1796E+01	1.3323E+01	1.4862E+01	1.2518E+00	158
	CS	2.5547E-06	9.8740E-05	1.2354E-05	4.3200E-05	169
	CCS	**1.0842E-15**	**3.5478E-15**	**7.8462E-15**	**2.3212E-15**	**95**
F5	GA	2.1000E+05	2.8900E+05	3.5100E+05	5.7702E+04	194
	DE	**5.5000E-13**	**1.1100E-12**	**3.3900E-12**	1.2300E-12	101
	PSO	1.0470E-01	2.0890E-01	4.1660E-01	1.2965E-01	129
	CS	1.3614E-05	8.2345E-04	2.5401E-03	1.0530E-03	134
	CCS	1.0604E-10	7.2014E-10	5.5561E-09	**2.1047E-09**	**78**

TABLE II
COMPARISON OF RESULTS BY FIVE METHODS FOR F6-F10 ON 30 DIMENSION FUNCTION

Function	Algorithm	Best	Mean	Worst	Std.	Convergence
F6	GA	2.9100E+06	3.8900E+06	8.4900E+06	2.4326E+06	185
	DE	2.3565E-04	4.5620E-04	7.8899E-04	2.2700E-04	103
	PSO	1.5000E-02	2.1500E-02	2.2500E-02	3.3250E-03	114
	CS	8.8321E+04	5.4668E+04	9.1122E+03	3.2458E+04	151
	CCS	**7.3200E-12**	**8.2900E-12**	**9.9100E-12**	**1.0014E-12**	**76**
F7	GA	-2.8577E+01	-2.8286E+01	-2.7861E+01	2.9404E-01	158
	DE	-2.7408E+01	-2.7128E+01	-2.7076E+01	1.4558E-01	81
	PSO	-1.8290E+01	-1.6178E+01	-1.3520E+01	1.9517E+00	95
	CS	-2.1698E+01	-2.1161E+01	-2.0667E+01	4.2108E-01	130
	CCS	**-3.8541E+01**	**-3.8454E+01**	**-3.8322E+01**	**9.0007E-02**	**55**
F8	GA	8.3358E+03	2.8004E+04	6.5488E+04	2.3707E+04	425
	DE	3.1663E+01	5.0399E+01	6.7792E+01	1.4753E+01	**257**
	PSO	6.3445E+03	2.6571E+04	6.2989E+04	2.3438E+04	329
	CS	2.1047E+03	7.6873E+03	9.0806E+03	3.0142E+03	399
	CCS	**3.2170E-05**	**4.9867E-05**	**6.1243E-05**	**1.1008E-05**	277
F9	GA	-7.6387E+01	-7.6334E+01	-7.4475E+01	8.8931E-01	201
	DE	-7.8332E+01	-7.8332E+01	-7.8332E+01	0.0000E+00	108
	PSO	-6.8735E+01	-6.8674E+01	-6.5024E+01	1.7352E+00	105
	CS	-6.3215E+01	-6.1024E+01	-5.8014E+01	2.1321E+00	147
	CCS	**-7.2565E+01**	**-7.1902E+01**	**-7.1285E+01**	**4.5621E-01**	**66**
F10	GA	3.8969E+03	4.1243E+03	5.1371E+03	5.3910E+02	210
	DE	2.6995E+01	3.6461E+01	4.4843E+01	7.2911E+00	105
	PSO	4.1134E+01	6.2174E+01	1.1393E+02	3.0588E+01	95
	CS	1.2134E+02	1.0288E+02	9.8232E+01	9.9774E+00	124
	CCS	**2.3360E-02**	**3.1365E-02**	**3.8378E-02**	**1.9887E-01**	**71**

TABLE III
COMPARISON OF RESULTS BY FIVE METHODS FOR F11-F15 ON 30 DIMENSION FUNCTION

Function	Algorithm	Best	Mean	Worst	Std.	Convergence
F11	GA	1.0979E+03	2.3505E+03	2.9588E+03	7.7475E+02	156
	DE	4.3189E-11	5.4503E-11	6.3732E-11	8.4000E-12	82
	PSO	6.0821E-02	8.5370E-02	1.9144E-01	5.6681E-02	91
	CS	2.3147E-04	2.8510E-03	5.6201E-02	2.5789E-02	123
	CCS	**9.3257E-12**	**6.3479E-12**	**4.4323E-12**	**1.7886E-12**	**54**
F12	GA	2.8304E-01	3.1541E-01	9.4890E-01	3.0655E-01	174
	DE	2.6124E-01	4.7580E-01	6.5037E-01	1.5914E-01	95
	PSO	4.8274E-02	4.3897E-01	5.6509E-01	2.2001E-01	94
	CS	4.2357E-01	6.5478E-01	8.1479E-01	1.6059E-01	130
	CCS	**2.0361E-02**	**5.2014E-02**	**8.5433E-02**	**2.1314E-02**	**70**
F13	GA	9.3551E+00	1.1939E+01	1.2270E+01	1.3032E+00	204
	DE	2.6667E-07	2.9055E-07	3.9517E-07	5.5800E-08	124
	PSO	1.1879E+00	1.3636E+00	1.5306E+00	1.3992E-01	112
	CS	1.0287E-07	6.3144E-08	8.7293E-08	1.6300E-08	150
	CCS	**3.2154E-15**	**6.1981E-15**	**8.7557E-15**	**1.8769E-15**	**96**
F14	GA	3.5832E+04	3.7522E+04	4.2208E+04	2.6971E+03	304
	DE	1.2238E+04	1.6988E+04	1.7988E+04	2.5083E+03	182
	PSO	8.5820E-01	1.3556E+00	1.7450E+00	3.6293E-01	241
	CS	4.6965E+00	5.4284E+00	9.9811E+00	2.3378E+00	289
	CCS	**5.2112E-12**	**8.9876E-12**	**4.5231E-11**	**1.2365E-11**	**111**
F15	GA	2.7267E+01	2.7349E+01	3.3072E+01	2.7174E+00	155
	DE	3.4494E+00	3.7802E+00	3.9989E+00	2.2588E-01	104
	PSO	2.2730E-01	2.5260E-01	2.6940E-01	1.7304E-02	140
	CS	2.1935E+00	3.6503E+00	7.3986E-01	1.1882E+00	125
	CCS	**1.2387E-11**	**4.2551E-11**	**7.5884E-11**	**2.3017E-11**	**84**

To facilitate the experiments, we used the g++ to program some cpp files for implementing the five algorithms on a personal computer with a 32-bit ubuntu 10.04 operating system, a 4GB of RAM, and a 3.10GHz-core(TM) i5-based processor.

The parameter setting of GA, PSO, DE, CS and CCS are as follows. The maximum generations and population size of the five algorithms are the same: the maximum generation G_{max}=1000, population size N_p=100. It can ensure the fairness of competition for each algorithm. For GA, the crossover rate c=0.95, the mutation factor m=0.1; for DE, the crossover rate CR=0.1, the mutation factor F=0.5; for PSO, constriction factor c_1=1.49, c_2=1.49, the maximum flying velocity of particles V_{min}=-0.5, the minimum flying velocity of particles V_{max}=0.5; the maximum inertia weight factor of population ω_{max}=0.9, the minimum inertia weight factor of population ω_{min}=0.4; for CS and CCS, discovery probability of alien egg p_a=0.004, route length β=1.34.

TABLE IV
COMPARISON OF RESULTS BY FIVE METHODS FOR F1-F5 ON 100 DIMENSION FUNCTION

Function	Algorithm	Best	Mean	Worst	Std.	Convergence
F1	GA	-3.1046E+04	-2.9297E+04	-2.8968E+04	9.1198E+02	401
	DE	-2.2140E+04	-2.2072E+04	-2.1287E+04	3.8708E+02	178
	PSO	-6.1125E+03	-5.4758E+03	-4.8518E+03	5.1469E+02	250
	CS	-2.1989E+04	-2.1465E+04	-2.1439E+04	2.5337E+02	291
	CCS	**-2.9596E+04**	**-2.9587E+04**	**-2.9458E+04**	**5.9668E+01**	**147**
F2	GA	1.5654E+02	1.6064E+02	1.6378E+02	2.9641E+00	410
	DE	4.6918E+02	4.8911E+02	5.0298E+02	1.3874E+01	201
	PSO	4.3103E+02	4.4606E+02	4.7216E+02	1.6991E+01	299
	CS	1.7136E+02	1.7231E+02	1.7447E+02	**1.1210E+00**	351
	CCS	**1.2357E+02**	**1.2826E+02**	**1.3420E+02**	4.3519E+00	**197**
F3	GA	1.1801E+01	1.2974E+01	1.3001E+01	5.5943E-01	480
	DE	5.4024E-01	5.7761E-01	5.7917E-01	1.7995E-02	265
	PSO	2.5611E+00	3.1076E+00	3.1999E+00	2.8191E-01	384
	CS	2.8904E+00	3.1235E+00	3.4471E+00	2.2827E-01	427
	CCS	**2.3665E-02**	**3.7889E-02**	**5.2017E-02**	**1.0505E-02**	**201**
F4	GA	7.8163E+01	9.9526E+01	1.5024E+02	3.0226E+01	456
	DE	9.2770E-01	9.3300E-01	9.6640E-01	1.7131E-02	240
	PSO	7.5466E+01	7.8627E+01	8.0636E+01	2.1281E+00	321
	CS	1.4695E-01	2.3126E-01	8.2708E-01	3.0271E-01	364
	CCS	**8.5654E-03**	**5.2551E-03**	**4.6886E-03**	**1.1125E-03**	**214**
F5	GA	6.6758E+05	7.0339E+05	9.0341E+05	1.0377E+05	405
	DE	1.6189E+00	1.9531E+00	1.9859E+00	1.6582E-01	225
	PSO	1.5607E+00	1.9689E+00	2.3562E+00	3.2480E-01	281
	CS	2.8715E-01	7.9651E-01	1.2521E+00	3.9414E-01	257
	CCS	**1.2556E-04**	**4.2017E-04**	**8.0221E-04**	**2.2243E-04**	**186**

TABLE V
COMPARISON OF RESULTS BY FIVE METHODS FOR F6-F10 ON 100 DIMENSION FUNCTION

Function	Algorithm	Best	Mean	Worst	Std.	Convergence
F6	GA	1.2882E+07	1.4777E+07	1.5765E+07	1.1962E+06	421
	DE	7.8299E+00	8.5995E+00	8.9695E+00	4.7468E-01	213
	PSO	6.3090E-01	6.4300E-01	7.2470E-01	4.1660E-02	249
	CS	3.7961E+01	8.9404E+01	1.1219E+02	3.1048E+01	321
	CCS	**2.3224E-02**	**4.2517E-02**	**6.1204E-02**	**1.1228E-02**	**178**
F7	GA	-8.4439E+01	-8.2837E+01	-8.1177E+01	1.3318E+00	388
	DE	-5.0084E+01	-4.9558E+01	-4.9109E+01	**3.3663E-01**	178
	PSO	-2.7925E+01	-2.7906E+01	-2.6478E+01	6.7769E-01	226
	CS	-4.1175E+01	-3.8993E+01	-3.7071E+01	1.6766E+00	279
	CCS	**-9.3654E+01**	**-9.1201E+01**	**-9.0879E+01**	1.2392E+00	**112**
F8	GA	9.8754E+05	1.5426E+06	1.9083E+06	3.7854E+05	801
	DE	3.4724E+05	5.7849E+05	6.2045E+05	1.2013E+05	**405**
	PSO	9.9542E+05	1.7632E+06	2.1474E+06	4.7891E+05	666
	CS	1.9769E+06	1.9970E+06	2.0367E+06	2.4847E+04	804
	CCS	**3.2145E-01**	**5.6887E-01**	**8.3214E-01**	**1.7426E-01**	421
F9	GA	-7.4533E+01	-7.4454E+01	-7.4237E+01	1.2514E-01	423
	DE	-7.0789E+01	-7.0170E+01	-6.9612E+01	4.8072E-01	199
	PSO	-6.7447E+01	-6.6435E+01	-6.3643E+01	1.6087E+00	200
	CS	-6.3032E+01	-6.2786E+01	-6.2715E+01	1.3583E-01	269
	CCS	**-7.7920E+01**	**-7.7915E+01**	**-7.7911E+01**	**2.0108E-03**	**125**
F10	GA	1.2483E+04	1.2555E+04	1.7918E+04	2.5453E+03	431
	DE	7.5285E+02	8.3170E+02	9.7427E+02	9.1634E+01	228
	PSO	8.8102E+02	1.3039E+03	1.3593E+03	2.1361E+02	205
	CS	5.5349E+02	6.2408E+02	9.0421E+02	1.5146E+02	258
	CCS	**4.2365E-01**	**4.5142E-01**	**4.8997E-01**	**2.1106E-02**	**157**

The experiment is divided into two parts, part 1 is for solving the 30 dimension functions. The experimental result is displayed from Table I to Table III. CCS is better than GA, DE, PSO and CS in terms of precision except for F5, for F5, the precision of DE is the highest. the standard deviation of CCS is smaller than the four methods except for F1, for F1, the standard deviation of DE is the smallest. It shows that the robustness of CCS is strong. The convergent speed is higher than the four methods except for F8, for F8, the convergence of DE is the highest. The second part is for solving the 100 dimension functions. The experimental result is displayed from Table IV to Table VI. CCS shows that its super performance in precision, robustness and convergence. CCS is the most precise algorithm among the five methods. CCS is smaller than GA, DE, PSO and CS in terms of standard deviation except for F2 and F7, for F2 and F7, CS and DE is the smallest respectively. The convergence of CCS is faster than the four methods except for F8 and F14, for F8 and F14, DE is the fastest in convergence.

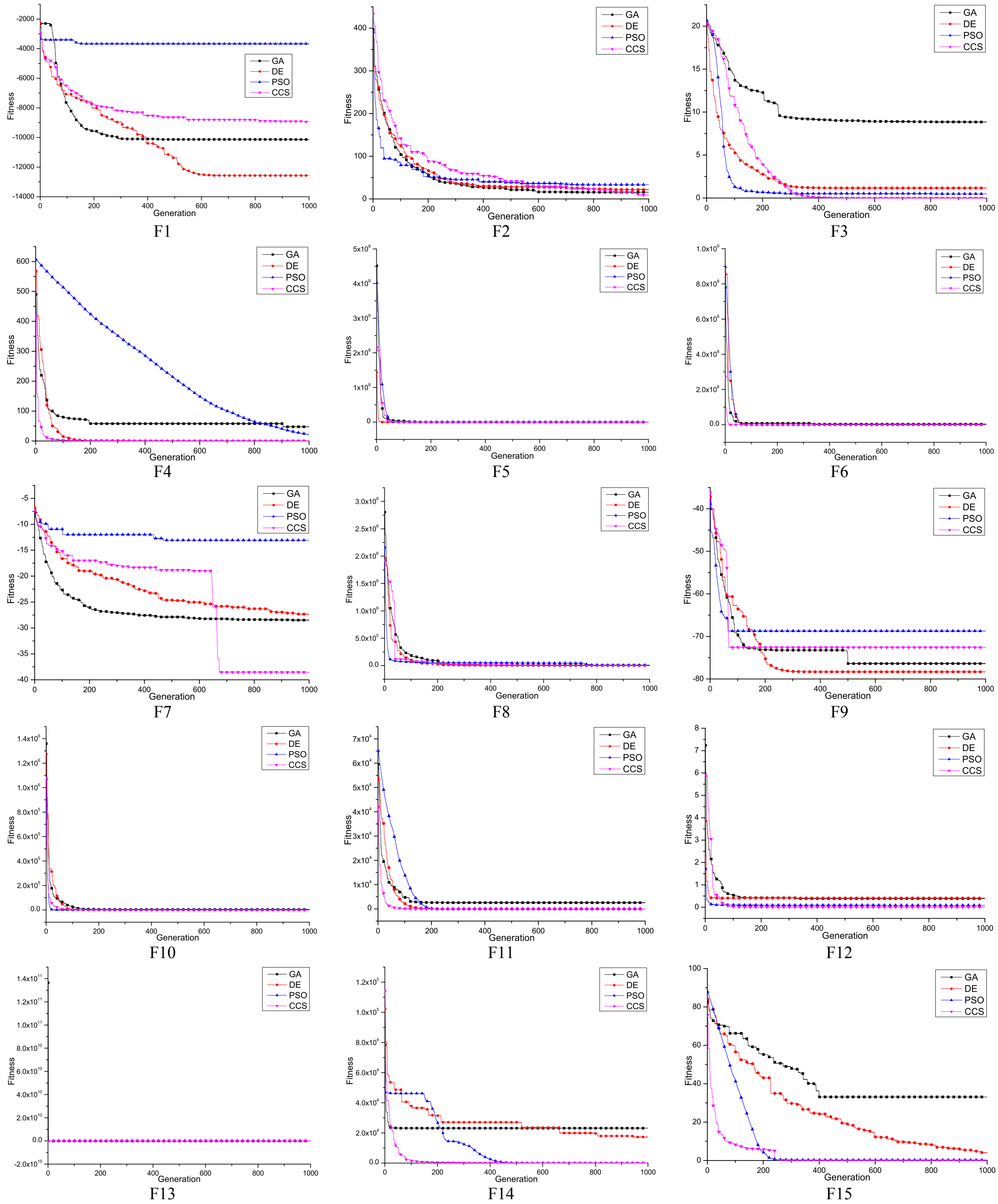

Figure 1. Convergence comparison of four algorihtms for benchmarks

The comparisons of convergent curves of four algorithms (GA, DE, PSO, CCS) for solving the benchmarks of high dimensional functions from F1 to F15 are listed in Figure 1. In subfigure F1, the convergence of DE is the fastest, but the fitness value deviates the theoretical optimum. The convergent curves of CCS for F2, F5, F9, F10 and F15 are rounded to the nearest the theoretical optimum. The convergent speeds of CCS for F3, F4, F6, F7, F11, F12, F13 and F14 are the fastest among the four algorithms. In subfigure F8, the convergent speed of DE is the fastest, the convergent speed of CCS is only faster than GA, the experimental results in the paper show that CCS has no advantage in convergence for F8. In subfigure F5, the fitness value of DE is the smallest. In Table I and Table V, the standard deviation of DE is the smallest and the robustness of DE is stronger than the other four methods for F1 and F7 respectively.

TABLE VI
COMPARISON OF RESULTS BY FIVE METHODS FOR F11-F15 ON 100 DIMENSION FUNCTION

Function	Algorithm	Best	Mean	Worst	Std.	Convergence
F11	GA	1.1314E+04	1.2385E+04	1.8144E+04	2.9993E+03	294
	DE	3.4943E+00	4.0955E+00	4.7194E+00	5.0017E-01	178
	PSO	3.5648E+00	3.6542E+00	3.6737E+00	4.7413E-02	201
	CS	3.2145E+03	5.6541E+03	9.8987E+03	2.7618E+03	261
	CCS	**1.8218E-01**	**3.0205E-01**	**3.6683E-01**	**3.2598E-02**	**101**
F12	GA	6.3992E-01	7.0922E-01	7.3981E-01	4.1788E-02	365
	DE	2.0997E-01	5.2664E-01	9.0501E-01	2.8412E-01	204
	PSO	6.9893E-01	9.9771E-01	1.5638E+00	3.5866E-01	210
	CS	6.1301E-01	7.4751E-01	5.1749E-01	9.4354E-02	248
	CCS	**1.2587E-01**	**1.3014E-01**	**1.4201E-01**	**4.5689E-03**	**142**
F13	GA	4.7663E+00	5.6044E+00	6.1119E+00	5.5484E-01	427
	DE	7.7861E-01	8.6395E-01	8.7092E-01	4.1969E-02	199
	PSO	1.3387E+01	1.5045E+01	1.7128E+01	1.5305E+00	247
	CS	2.5647E+01	3.8775E+01	5.6817E+01	1.2778E+01	270
	CCS	**4.5139E-02**	**6.1567E-02**	**8.9302E-02**	**1.4884E-02**	**173**
F14	GA	2.9544E+05	3.0252E+05	3.2549E+05	1.2827E+04	688
	DE	2.8302E+05	2.8791E+05	3.1719E+05	1.5088E+04	**294**
	PSO	8.6048E+02	8.7005E+02	1.4564E+03	2.7869E+02	541
	CS	3.7669E+03	3.8340E+03	4.4435E+03	3.0437E+02	600
	CCS	**4.2635E-06**	**4.3650E-06**	**4.4198E-06**	**3.0208E-08**	301
F15	GA	5.7815E+01	6.2661E+01	6.5075E+01	3.0188E+00	327
	DE	6.5758E+01	6.7462E+01	6.8746E+01	1.2239E+00	187
	PSO	3.2187E+00	3.5279E+00	3.8999E+00	2.7849E-01	301
	CS	1.2293E+01	1.3073E+01	1.4176E+01	7.7249E-01	214
	CCS	**2.3667E-04**	**2.7881E-04**	**2.8710E-04**	**9.2080E-06**	**184**

V. CONCLUSIONS

Cuckoo search algorithm is a novel intelligence algorithm, it has been successfully applied to scientific research and industrial technology. A chaotic cuckoo search algorithm is presented in the paper, which is employed to solve the high-dimensional functions. 15 Benchmarks of high-dimensional functions are used to verify the performance of CCS. The simulation results show that CCS have competitive advantages over GA, DE, PSO and CS in terms of computation precision, robustness, convergent speed.

Many real-world optimization problems involve a high dimension function of decision variables. For example, in shape optimization, a large number of shape design variables are often used to represent complex shapes, such as turbine blades, aircraft wings, and heat exchangers, etc. CCS can be applied to solve this kind of problems.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (No.61202109), A Project Supported by Scientific Research Fund of Hunan Provincial Education Department (Grant No. 08D092, No.13C333), A Project Supported by Scientific Research Fund of Hunan Provincial Statistical Bureau (2008C36), A Project Supported by the Science and Technology Research Foundation of Hunan Province (No.2008GK2019 and No. 2013GK3082). A Project supported by the Natural Science Foundation of Zhejiang Province, China (No.LY12F02019), A Project supported by the public technology applied research project of Zhejiang province (No.2011C23130) Project supported by the Science and Technology Foundation of Jiaxing City, China (No.2012AY1027).

REFERENCES

[1] X. Qiu, X. Qiu and F. Liao, "A Fractal Evolutionary Particle Swarm Optimizer," *Journal of Computers*, vol.8, no.5, pp.1303-1308, 2013.

[2] L. Ye, Y. Xing, and W. Xiang, "An Artificial Immune Classification Algorithm based on Particle Swarm Optimization," *Journal of Computers*, vol.8, no.3, pp.772-778, 2013.

[3] T. Chen and G. Zhou, "Modeling Production scheduling problem and its solution by genetic algorithm," *Journal of Computers*, vol.8, no.8, pp.2126-2133, 2013.

[4] F. Gu, and H. Liu, "An Adaptive Multiobjective Differential Evolution Algorithm," *Journal of Computers*, vol.8, no.2, pp.294-301, 2013.

[5] L. Xie, A. Ouyang, L. Liu, M. He, X. Peng and X. Zhou, "Chaotic Hybrid Invasive Weed Optimization for Machinery Optimizing," *Journal of Computers*, vol.8, no.8, pp.2093-2100, 2013.

[6] X.-S. Yang and S. Deb, "Cuckoo Search via Lévy flights," *2009 World Congress on Nature & Biologically Inspired Computing*, pp.210-214, 2009.

[7] 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.

[8] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research*, vol.40, no.6, pp.1616-1624, 2013.

[9] L.S. Coelho, F.A. Guerra, N.J. Batistela and J.V. Leite, "Multiobjective cuckoo search algorithm based on Duffing's oscillator applied to Jiles-Atherton vector hysteresis parameters estimation," *IEEE Transactions on Magnetics*, vol.49, no.5, pp.1745-1748, 2013.

[10] A. Elkeran, "A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering," *European Journal of Operational Research*, vol.231, no.3, pp.757-769, 2013.

[11] Z. Moravej and A. Akhlaghi, “A novel approach based on cuckoo search for DG allocation in distribution network,” *International Journal of Electrical Power & Energy Systems*, vol.44, no.1, pp.672-679, 2013.

[12] S. Agrawal, R. Panda, S. Bhuyan and B.K. Panigrahi, “Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm,” *Swarm and Evolutionary Computation*, vol.11, pp.16-30, 2013.

[13] R. Vali, S.M. Berber, S.-K. Nguang, “Analysis of Chaos-Based Code Tracking Using Chaotic Correlation Statistics,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol.59, no.4, pp.796-805, 2012.

[14] G.-C. Liao and T.-P. Tsao, “Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting,” *IEEE Transactions on Evolutionary Computation*, vol.10, no.3, pp.330-340, 2006.

[15] M. Shen, W.-N. Chen, J. Zhang, H.S.-H. Chung and O.Kaynak, “Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization,” *IEEE Transactions on Cybernetics*, vol.43, no.2, pp.790-802, 2013.

[16] R. Caponetto, L. Fortuna, S. Fazzino, M.G. Xibilia, “Chaotic sequences to improve the performance of evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol.7, no.3, pp.289-304, 2003.

[17] L.D.S. Coelho, T.C. Bora, L. Lebensztajn, “A Chaotic Approach of Differential Evolution Optimization Applied to Loudspeaker Design Problem,” *IEEE Transactions on Magnetics*, vol.48, no.2, pp.751-754, 2012.

[18] L.D.S. Coelho, V.C. Mariani, “Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect,” *IEEE Transactions on power systems*, vol.21, no.2, pp.989-996, 2006.

[19] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince and H. E. Stanley, “lévy flights search patterns of wandering albatrosses,” *Nature*, vol. 381, pp. 413-415, 1996.

[20] G.M. Viswanathan, V. Afanasyev, S.V. Buldyrev, S. Havlin, M.G.E da Luz, E.P Raposo, and H.E. Stanley, “Lévy flights in random searches,” *Physica A: Statistical Mechanics and its Applications*, Vol.282, no.1–2, pp.1-12, 2000.

[21] G.M. Viswanathan, V. Afanasyev, S. V Buldyrev, S. Havlin, M.G.E da Luz, E.P Raposo, and H.E. Stanley, “Lévy flights search patterns of biological organisms,” *Physica A: Statistical Mechanics and its Applications*, Vol.295, no.1-2, pp.85-88, 2001.

[22] A.M. Reynolds, “Cooperative random Lévy flight searches and the flight patterns of honeybees,” *Physics Letters A*, Vol.354, no.5-6, pp.384-388, 2006.

[23] A.M. Reynolds and M.A. Frye, “Free-flight odor tracking indrosophila is consistent with an optimal intermittent scale-free search,” *PLoS ONE*, vol.2, no.4, pp.1-9, 2007.

[24] A.M. Raynolds, A.D. Smith, R. Menzel, U. Greggers, D.R. Reynolds and J.R. Riley, “Displaced honey bees performoptimal scal-free search flights,” *Ecology*, vol.88, no.8, pp.1955-1961, 2007.

[25] A.L. Sellreier and M. Grove, “Ranging patterns of hamadryas baboons: Randomwalk analyses,” *Animal Behavior*,vol.80, no.l, pp.75-87, 2010.

[26] G. Ramos-Fernandez, J.L. Mateos, O. Miramontes, G. Cocho, H. Larralde and B. Ayala-Orozco, “Lévy walk patterns in the foraging movements of spider monkeys,” *Behavioral Ecology and Sociobioblogy*, vol.55, no.3, pp.1743-1750, 2004.

[27] M.F. Shlesinger, “Search research,” *Nature*, vol.443, pp.281-282, 2006.

[28] R.B. Payne, M.D. Sorenson and K. Klitz, “The cuckoo,” Oxford: Oxford University Press, 2005.

[29] C.T. Brown, L.S. Liebovitch and R. Glendon, “Lévy Flights in Dobe Ju/’hoansi Foraging Patterns,” *Human Ecology*, vol.35, no.1, pp.129-138, 2007.

[30] I. Pavlyukevich, “Lévy flights, non-local search and simulated annealing,” *Journal of Computational Physics*, Vol.226, no.2, pp.1830-1844, 2007.

[31] R.N. Mantegna, “Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes,” *Physical Review E*, vol.49, no.5, pp. 4677-4683, 1994.

[32] Y. Wang and C. Dang, “An Evolutionary Algorithm for Global Optimization Based on Level-Set Evolution and Latin Squares,” *IEEE Transactions on Evolutionary Computation*, vol.11, no.5, pp.579-595, 2007.

Aijia Ouyang received the M.E. degree in computer science and technology from Guangxi University for Nationalities, Nanning, China, in 2010. He is currently a Ph.D. candidate of Hunan University, Changsha, China. His research interests include parallel computing and artificial intelligence. He has published research articles in international conference and journals of parallel computing.

Guo Pan is an associate professor in the Logistics Information Department, Huna n Vocational college of Modern Logistic s, Changsha, Hunan Province, China. Sh e received the B.E. degree in the Depart ment of Computer from National Univer sity of Defense Technology, Changsha, China, in 2001, and the M.E. degree in th e Department of Computer Science and Technology from Hunan University, in 2 007, respectively. She is currently a Ph.D. candidate in the Colle ge in the College of Information Science and Engineering at Hu nan University, Changsha, China. Her research interests include parallel computing, big data, DNA computing and intelligence algorithms.

Guangxue Yue is a Professor in the Department of Computer Science and Technology at Jiaxing University, Jiaxing, China. He received the M.E. degree, the Ph.D. degree in the Department of Computer Science and Technology from Hunan University, Changsha, China, in 2006 and 2012, respectively. His research interests include computer network, distributed system, QoS of streaming media. He has published more than 70 international journals/conference papers on computer network, distributed system and information security.

Jiayi Du received his MSc and BSc in computer science from Hunan University, China, in 2004 and 2010. He is currently a PhD candidate in Hunan University, China. His research interest includes modeling and scheduling for parallel and distributed computing systems, embedded system computing, cloud computing, parallel system reliability, and parallel algorithms.