Journal of Software Engineering

ISSN 1819-4311

Academic
Journals Inc.

www.academicjournals.com

Journal of Software Engineering 10 (1): 16-28, 2016
ISSN 1819-4311 / DOI: 10.3923/jse.2016.16.28

A Cuckoo Optimization Algorithm Using Elite Opposition-Based Learning and Chaotic Disturbance

[1,2]Juan Li, [1]Tao Chen, [1,2]Ting Zhang and [2]Yuan Xiang Li
[1]College of Information Engineering, Wuhan Technology and Business University, Wuhan, 430065, China
[2]State Key Laboratory of Software Engineering, College of Computer Science, Wuhan University, Wuhan, Hubei, 430072, China

Corresponding Author: Juan Li, College of Information Engineering, Wuhan Technology and Business University, Wuhan, 430065, China

ABSTRACT

To solve the problem of slow convergence rate before reaching the global optimum in the conventional Cuckoo Search algorithm (CS). This study presents an enhanced CS algorithm called CH-EOBCCS, where elite members are introduced to generate their opposite solutions by elite opposition-based learning. Some excellent members will carry over to the next generations from the current solutions and the opposite solutions. This mechanism is helpful to enhance the ability of global optimum for CS algorithm and to expand the search area. At the same time, in order to expand diversity of the population, CH-EOBCCS algorithm introduces the chaotic disturbance to nest location in the iteration to improve the convergence speed. The experiments are conducted on 8 classic benchmark functions and the results show that the elite opposition-based learning and chaotic disturbance strategy has much better search performance than the generalized opposition-based learning strategy. The novel CH-EOBCCS algorithms improve the ability of CS to jump out of the local optima.

Key words: Cuckoo algorithm, chaotic disturbance, elite opposition-based learning

INTRODUCTION

Cuckoo Search optimization (CS) is a relatively new optimization technique which was developed by Yang and Deb (2009). The basic idea of this algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. The CS has obtained good performance on many optimization problems, Yang and Deb (2010) compared CS to PSO over 100 trials for each objective function, this number of objective function evaluations would not be feasible for application to practical engineering problems with costly objective functions. Chandrasekaran and Simon (2012) has successfully applied CS in multi-objective scheduling problem. A improved CS has been applied in reliability optimization problems by Valian *et al.* (2013). A new hybrid algorithm of cuckoo search and Particle Swarm Intelligence (PSO) was introduced by Wang and Cai (2009) to remedy the defect of PSO. Layeb (2011) introduced a new hybridization between quantum inspired and cuckoo search for the knapsack problem. The experimental results shows that this hybrid algorithm achieves better balance between exploration and exploitation. Wang *et al.* (2011) put CS algorithm based on Gaussian disturbance forward which adds vitality nest position change. Zheng and Zhou (2013) presents a self-adaptive step adjustment cuckoo search algorithm which speeds up the cuckoo search algorithm speed and improves the calculation accuracy. Qu *et al.* (2014) put forward the hybrid cuckoo algorithm based on the crossover strategy and chaos disturbance which effectively improved the precision of algorithm and rate of convergence. However, these methods may easily become trapped at local extreme. Many real world problems can be converted into optimization problems. As their complexity increases, traditional optimization algorithms cannot sufficiently satisfy the problem requirements and more effective algorithms are needed. Opposition-Based Learning (OBL) is proposed by Tizhoosh (2005) which is a new concept in computational intelligence and has been proven to be an effective concept to enhance various optimization approaches. A elite opposition-based learning for particle swarm optimization is proposed by Zhou *et al.* (2013). Tuo (2013) proposed a dynamic self-adaptive harmony search algorithm based on opposition-based computing and Gaussian distribution estimation. A multi-dimensional dynamic self-adaptive adjustment operator was employed to improvise a new harmony, harmony memory was selectively updated dynamic opposition harmony vector. In addition, some scholars have proposed improvements for the cuckoo (Lu and Gao, 2010; Wang *et al.*, 2013; Ahandani and Alavi-Rad, 2015).

These methods are only improved from some local aspects which are easy to fall into local extreme and slow convergence speed. In order to enhance the performance of CS on complex problems, this study presents a novel CS algorithm called CH-EOBCCS by using EOBL which is an elite opposition-based learning algorithm. The main idea behind EOBL is to transform solutions in the current search space to a new search space. By simultaneously considering the solutions in the current search space and the transformed search space, EOBL can provide a higher chance of finding solutions which are closer to the global optimum. At the same time, the algorithm brings the chaotic disturbance to accelerate the diversity of population. After parasitic nest owners found foreign egg in the T generation of CS algorithm, it obtain a set of optimum location of the nest .at this time, algorithm is not directly into the next iteration but continue to perform chaotic disturbance for better parasitic nest position. Experimental simulations on 8 functions show that CH-EOBCCS obtains better performance on the all of the test problems.

This study makes the extensive and innovative works in the following areas:

- In this study, exploration position space search area was expanded by using chaotic disturbance strategy which expand diversity of the population and which avoid falling into local extreme
- Chaotic disturbance process is bound to slow down the convergence process of algorithm in a certain extent. This problem can be overcome effectively by using the elite opposition-based learning strategy which accelerates the convergence rate. Because of search information space of elite individuals is better than ordinary individuals, algorithm can improve the search ability to accelerate the convergence speed

MATERIALS AND METHODS

Cuckoo Search (CS) algorithm: The CS algorithm is a new global search algorithm which is suitable for solving optimization problems and which is proposed by Yang and Deb (2009). This algorithm simulated process of cuckoo looking for nest for spawning, in this process, N nest location is randomly initialized in feasible solution space and the fitness of each nest location is calculated. The best nests with high quality of eggs (solutions) will carry over to the next generations. Some host birds can engage direct conflict with the intruding cuckoos. The algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of fruit flies and some of the birds. In order to Lévy flight distribution, it is found that animals and birds search for food in a random or quasi random manner and essentially follow a random walk because the next step is based on the current place and the transition probability to the next state. The CS is based on three idealized rules:

- Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest
- The best nests with high quality of eggs (solutions) will carry over to the next generations
- The number of available host nests is fixed and a host can discover an alien egg with probability pa ϵ [0 1]. In such case, the host bird either abandons the nest to build a completely new nest in a new location or throw the egg away

Yang X used D-dimensional vector Xi = (xi1, xi2,..., xid), 1≤i≤n indicates the position of the number i nest. Yang uses Lévy fly and randomly walks to produce offspring, execute formulas operations:

$$X_t^{t+1} = X_i^t + \alpha \oplus \text{levy}(\lambda) \quad (i=1,2,...,n) \tag{1}$$

where, X^t_i is represents location of number I nest in the 7th generation, levy (λ) indicate the random walk route and obey probability distribution levy~u = $t^{-\lambda}$ ($1<\lambda\leq 3$). The ⊕ represent dot product, α is the step-size information which is for controlling the range of random search, in order to obtain more useful step-size information, the Eq. 2 is used to calculate step-size information:

$$\alpha = \alpha_0 \alpha\left(x_i^t - x_{\text{best}}\right) \tag{2}$$

Opposition-based learning: Opposition-based strategy in the optimization algorithms uses the concept of Opposition-Based Learning (OBL) which is a new concept in computational intelligence and has been proven to be an effective concept to enhance various optimization approaches. When evaluating a solution x to a given problem, simultaneously computing its opposite solution will provide another chance for finding a candidate solution which is closer to the global optimum. It mainly is used to initial population in intelligent evolutionary algorithms. The OBL by comparing the fitness of a member to its opposite and retaining the fitter one in the population accelerates the EAs. The OBL was recently applied to different EAs. In order to explain easier opposition-based learning, we need to define clearly the concept of opposite numbers.

Definition 1: The opposite number: Let Ā ϵ [x,y] be a real number. The opposition number is defined by:

$$\bar{A} = x + y - \alpha \tag{3}$$

Similarly, the opposite point in D-dimensional space can be defined as follows.

Definition 2: Opposite point: Let A_i = ($\alpha_{i,1}$ $\alpha_{i,1}$,...,$\alpha_{i,D}$) is the point in D-dimensional space (i.e., candidate solution) and $\alpha_{i,j}$, ϵ [x_j,y_j] where the opposite point, $\bar{A} = \left(\overline{\alpha_{i,1}}, \overline{\alpha_{i,1}, ..., \alpha_{i,D}}\right)$ in D-dimensional space can be defined as follows:

$$\bar{A} = x_j + y_j + \alpha_{i,j} \tag{4}$$

Definition 3: Opposite optimization: Let $A_j = (\alpha_{i,1}\ \alpha_{i,1},...,\alpha_{i,D})$ is the point in D-dimensional space (i.e., candidate solution) which is an effective solution for optimization problem. For the function optimization problem, according to the definition of the opposite point, $\bar{A}=(\overline{\alpha_{i,1}},\overline{\alpha_{i,1}},...,\overline{\alpha_{i,D}})$ is the opposite point of A in D-dimensional space, if $f(\bar{A}) \le f(A)$, then f(A) can be replaced with $f(\bar{A})$.

Elite opposition-based learning: Because of the opposition-based learning method that is proposed by Tizhoosh (2005) can improve the efficiency of the algorithm and keep the population diversity, it is blind that all individuals form the opposite solution. The reason is that population size is enough large that search space of opposite solution is not conducive to search space of the current solution.

Therefore, in order to form a better search space, in this study, the optimal adaptive value for the CS algorithm is as elite member which obtained elite opposite solution by opposition-based learning of elite member.

Definition 4: Elite Opposition-Based Learning (EOBL): Let B_i $(b_{i,1}\ b_{i,2}\ ,...,b_{i,D})$ is the elite members in current populations, $\bar{B}l=(\overline{b_{1,1}},\overline{b_{1,2}},...\overline{b_{1,D}})$ then is the solution with B_i corresponding of elite opposition-based learning:

$$\overline{B_{1,j}} = \delta.(xm_j + yn_j) - b_{i,j} \tag{5}$$

where, $\delta \in (0,1)$ $B_{i,j} \in [mj,nj]$, $[Xm_j, yn_j]$ is Dynamic boundary in D-dimensional space. It is calculated with the Eq. 6:

$$xm_j = \min(B_{i,j}), yn_j = \max(B_{i,j}) \tag{6}$$

Chaotic disturbances CS algorithm (CH-CS): The algorithm brings the chaotic disturbance to accelerate the diversity of population. After parasitic nest owners found foreign egg in the T generation of CS algorithm, it obtain a set of optimum location of the nest. At this time, algorithm is not directly into the next iteration but continue to perform chaotic disturbance for better parasitic nest position.

Standard CS algorithm is easy to fall into local extreme. To avoid falling into local extreme, after parasitic nest owners found foreign egg in the T generation of CS algorithm, it obtain a set of optimum location of the nest x_i^t, i=1,2,... n. The x_i^t is not made direct access to the next iteration but continue to perform chaotic disturbance for better parasitic nest position. First, a set of chaotic variables that have same number of optimization nest were produced.

Chaos is introduced into optimization variables by means of the carrier, then direct to search by using chaotic variables. It is calculated with the Eq. 7:

$$nest_{best,d} = nest_{best,d} + R_d \cdot (2 \cdot x_d(i) - 1) \tag{7}$$

where, x_d (i) is chaotic sequence.

Chaotic sequences were generated as follows:

- Generated Randomly D dimensional vector x(1)=(x1(1),x2(1),...,xD(1))
- Using Kent chaotic mapping to optimize

$$x_n + 1\begin{cases} x_n/\alpha, 0<x_n \le \alpha \\ (1-x_n)/(1-\alpha), \alpha<x_n \le 1 \end{cases} \tag{8}$$

where, $\alpha = 0.4$. After the j iteration, the j chaotic sequences which are j were generated.

Binary sequence are obtained after quantifying the output of the Tent chaotic map. Set the output of chaotic map $x_n \epsilon [m,M]$, quantization function is:

$$S(x_n) = \begin{cases} 0 & x_n < (m+M)/2 \\ 1 & x_n \ge (m+M)/2 \end{cases} \tag{9}$$

Randomness of common chaotic mapping output sequence were studied, the results showed that Tent map sequence randomness is not ideal. Therefore, a new measure, Piecewise Tent Chaotic Map, is proposed for improving the randomness of Tent chaotic sequences. Piecewise Tent map; the performances under computer finite precision are analyzed. In order to improve the randomness of the output sequence, the following two piecewise Tent map were structured.

$$x_{n+1} = \begin{cases} \dfrac{2x_n}{\alpha} & 0 < x_n \le 0.5\alpha \\ \dfrac{1-2x_n}{1-\alpha} & 0.5\alpha < x_n \le 0.5 \\ \dfrac{\{1-2[x_n-(0.5-0.5\alpha)]\}}{1-\alpha} & 0.5 < x_n \le 1-0.5\alpha \\ \dfrac{2[x_n-(1-0.5\alpha)]}{\alpha} & 1-0.5\alpha < x_n \le 1 \end{cases} \tag{10}$$

The curve of Tent Chaotic Map and curve of two piecewise Tent map can be shown from Fig. 1a-b.

It is important to determine the chaos disturbance radius when there is chaotic disturbance.

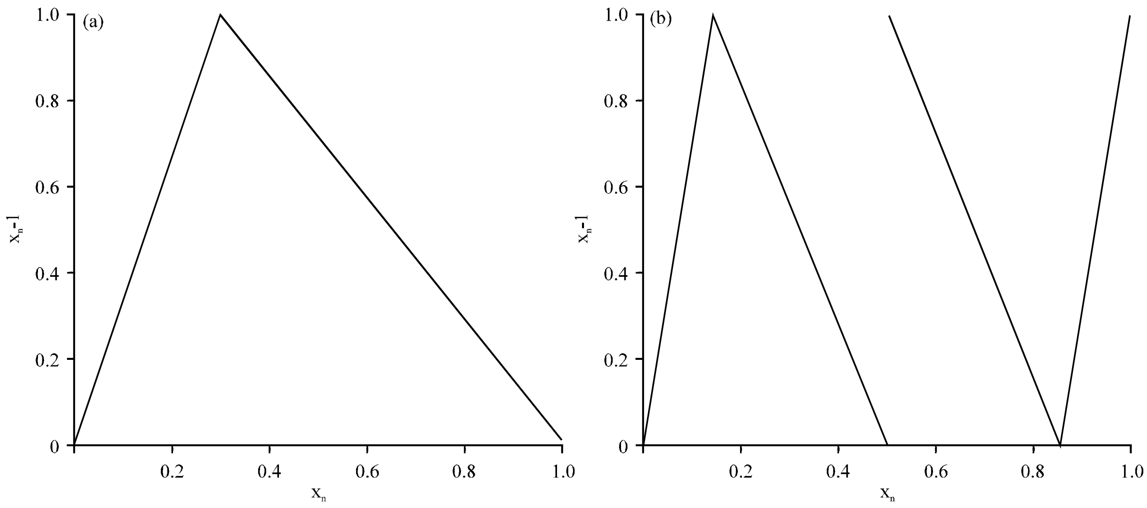

Fig. 1(a-b): (a) Tent chaotic map and (b) Two piecewise Tent map

If regional radius is too large, deviation of nest location is too much. Because of each dimension is different, each dimension take different disturbance radius. In this study, the method with Dimension by Dimension Improvement is used to determine the chaos disturbance radius. It is calculated with the Eq. 11:

$$R_d = \delta \cdot \left| \frac{1}{n} \cdot \sum_{t=1}^{n} nest_{t,d} - nest_{best,d} \right| \qquad (11)$$

where, δ is Factor de escala, $\sum_{t=1}^{n} nest_{t,d}$ is average values of current nest on D dimension. The $nest_{best,d}$ is variables values of current the best nest on D dimension.

The structure of CH-CS algorithm shown in Algorithm 1.

Algorithm 1: CH-CS algorithm

```
Begin
    Randomly initialize population of n host nests $x_i$
    While (stopping criterion is not meet do)
        Generate $x_i$ as new solution by Lévy flights
        Evaluate its fitness $f_i$;
        Choose a solution (say xj) randomly among N new and evaluate its fitness (Fj)
        if Fi>Fj then
            Replace j by new solution
        end if
        A fraction (1-pa) of nests are abandoned at random and new ones are built via random walk
        Keep the best solutions
        Rank the solutions and find the current best
        A set of optimal nest location P perform chaotic disturbance operations using the Eq. 7 which get a new set of nest position
        P1. The values of P1 are compare with the values of P2 and poor nest the nest were replaced by better nest location. The
        optimal nest location will were will carry over to the next generations
    End while
End
```

CH-CS algorithm applied elite opposition-based learning (CH-EOBCCS): Chaotic disturbance process is bound to slow down the convergence process of algorithm in a certain extent. This problem can be overcome effectively by using the elite opposition-based learning strategy which accelerates the convergence rate. Because of elite members have the better search information space than ordinary members and it can improve algorithm search performance, the elite opposite solution is more close to the global optimal solution than ordinary opposite solution. In order to prove the theory, generally it could be validated by $f(x) = \|x\text{-}1\|^2$ as the optimized function.

The first example: Assume that A1 is the one point of two-dimensional space, A1 = (x1,y1,), x1,y1ϵ[0,30], t = (15,17) is global optimal solution. Defining x1,x2 get A1 = (5,5), then get the opposite solution $\overline{A1} = (25,25)$ according to Eq. 3. You can calculate the result f(A1) = 244, $\overline{A1} = (25,25)$ according to function f (x).

The second example: Assume there is elite member A2 = (12,8), you can figure out the opposite solution. The $\overline{A2} = (18,22)$ then you can get f(A2) = 90, $f(\overline{A2}) = 34$ according to f (x).

Obviously, according to the above two examples, the conclusion is that opposite solution of elite members is better than opposite solution of ordinary members. Opposite solution of elite members are more optimized than ordinary members. The examples can be shown from Fig. 2.

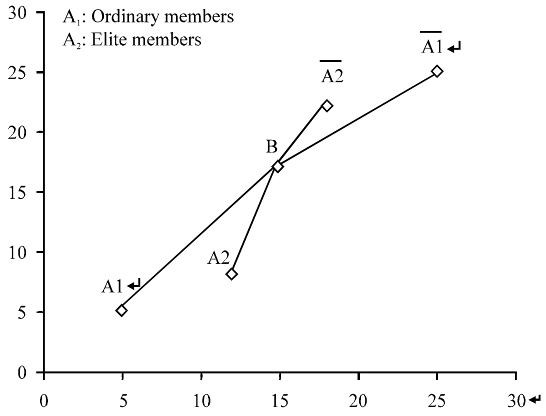

Fig. 2: Example of elite opposition-based learning

Elite opposition-based learning strategy is applied to the CH-CS algorithm and the framework based on the group of elite opposition-based learning is built in this study. It is necessary that how to choose N solution as the next populations from the current populations and opposite populations. In this study, group selection mechanism is adopted. Individuals of the current population and individuals of opposite populations sorted by fitness. The best N individuals are selected into the next generation. Let p(g) is current population, the corresponding opposite population, Eop(g) is produced by the elite opposite opposition-based strategy. The N best parasitic nest are chosen to constitute next member p(g+1) from P(g)∪Eop(g). Let N is population size, p_m is the probability to execute elite opposition-based learning strategy, X_e is the elite individual which is the best individual selected from current population, rand∈[0,1], rand is random numbers which are distributed uniformly.

The structure of CH-EOBCCS algorithm shown in Algorithm 2.

Algorithm 2: CH-EOBCCS algorithm

```
Randomly initial position of parasitic nest P(g)
While stopping criterion is not meet do
    if rand≤$p_m$ then
        select the best individual as elite individual $X_e$ from P(g)
        update [$xm_j$,$yn_j$] according to Eq. 6
        generate the opposite solution of $X_e$ from the current swarm (g) by Eq. 5
         if $\overline{B_{i,j}} < xm_j$ or $\overline{B_{i,j}} > yn_j$ then
          $\overline{B_{i,j}} = rand(xm_j, yn_j)$
        enf if
        Calculate the fitness value of opposite solution and add to opposite population Eop(g)
        select the N best fittest solutions from p(g)∪Eop(g) as the next swarm p(g+1) by population-based selection mechanism
    Else
        Execute the canonical CH-CS algorithm
    end if
End while
```

RESULTS AND DISCUSSION

Experimental setting: In this experiment, a comprehensive set of benchmark problems, including 8 different global optimization problems, was chosen for the following experimental studies. These function are Sphere, Griewank, Rastrigrin, Rosenbrock, Ackley, Quartic, Step and Schwefel 2.22. All the functions used in this study are minimization problems. A brief description of these benchmark problems is listed in Table 1.

The experiments are carried out on a P4 Dual-core platform with a 1.75 GHz processor and 1 GB memory, running under the Windows 7.0 operating system. The algorithms are developed using MATLAB Release 2010. Populations size is 50 and the maximum number of iterations is1000. the probability that Foreign eggs were found is Pn = 0.25.

Comparison of CH-EOBCCS, CS, PSO and CH-CS: In this section, in order to analyze the effectiveness of the CH-EOBCCS algorithm, for all the test functions, three different dimension sizes are tested: 10, 20 and 30. The maximum number of generations is set as 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30. In order to eliminate stochastic discrepancy, a total of 50 runs for each experimental setting were conducted. The results are presented in Table 2.

Table 2 list the testing results on the functions f1-f8. The algorithm in this study can find the global optimal solutions for f1, f2, f3, f4, f7 and f4 the success rate attains 100% in 50 runs. M-iteration denotes mean number of iterations before finding the global optimal value.

The testing results on the function of f1-f8 is given in Table 2. For the function f5, the proposed algorithm demonstrates a far better average convergence precision than the standard CS and improved CS algorithms. For function f6, the proposed algorithm also yields a compelling result in terms of average convergence precision and success rate. The M-iteration in the table denotes mean number of iterations before success (i.e., the iteration number to converge to a given threshold). The proposed CS has a strong ability to move out of the local optima and it can effectively prevent the premature convergence and significantly enhance the convergence rate and accuracy in the evolutionary process.

Table 1: Brief description of functions

Name	Function	Dom	Best
Sphere	$f1=\sum_{i=1}^{n} x_i^2$	[-1, 1]	0
Griewank	$f2=\frac{1}{400}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos(\frac{xi}{\sqrt{x}})+1$	[-10, 10]	0
Rastrigin	$f3=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)10]$	[-5.12, 5.12]	0
Rosenbrock	$f4=\sum_{i}^{n}(100x_i+1-x_i^2)+(1-x_i)^2$	[-10, 10]	0
Ackley	$f5=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})$ $-\exp(\frac{1}{n}\sum_{1}^{n}\cos(2\pi x_i))+20+e$	[-32, 32]	0
Quartic	$f6=\sum_{i=1}^{n} ix_i^4+\text{random}[0,1]$	[-1.28, 1.28]	0
Step	$f7=\sum_{i}^{n}(\lvert x_i 0.5\rvert)^2$	[-100, 100]	0
Schwefel 2.22	$f8=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10, 10]	0

Table 2: Testing results for the test function f1-f8

Function	Name	Pop. size	Dim	Gene	Pn	Mean	M-iteration	Success ratio (%)
f1	Sphere	20	10	1000	0.25	0.00E+00	44.82	100
f1	Sphere	20	20	1500	0.25	0.00E+00	127.49	100
f1	Sphere	20	30	2000	0.25	0.00E+00	177.21	100
f2	Griewank	20	10	1000	0.25	0.00E+00	60.23	100
f2	Griewank	20	20	1500	0.25	0.00E+00	117.88	100
f2	Griewank	20	30	2000	0.25	0.00E+00	178.24	100
f3	Rastrigin	20	10	1000	0.25	0.00E+00	25.68	100
f3	Rastrigin	20	20	1500	0.25	0.00E+00	57.45	100
f3	Rastrigin	20	30	2000	0.25	0.00E+00	75.77	100
f4	Rosenbrock	20	10	1000	0.25	0.00E+00	114.34	100
f4	Rosenbrock	20	20	1500	0.25	0.00E+00	207.19	100
f4	Rosenbrock	20	30	2000	0.25	0.00E+00	237.72	100
f5	Ackley	20	10	1000	0.25	0.892E-17	49.09	94
f5	Ackley	20	20	1500	0.25	7.661E-17	127.28	98
f5	Ackley	20	30	2000	0.25	9.535E-18	145.96	96
f6	Quartic	20	10	1000	0.25	2.853E-04	98.06	95
f6	Quartic	20	20	1500	0.25	6.125E-04	146.43	98
f6	Quartic	20	30	2000	0.25	5.811E-05	197.08	97
f7	Step	20	10	1000	0.25	0.00E+00	68.90	100
f7	Step	20	20	1500	0.25	0.00E+00	98.22	100
f7	Step	20	30	2000	0.25	0.00E+00	165.83	100
f8	Schwefel 2.22	20	10	1000	0.25	0.00E+00	280.88	100
f8	Schwefel 2.22	20	20	1500	0.25	0.00E+00	377.35	100
f8	Schwefel 2.22	20	30	2000	0.25	0.00E+00	478.21	100

We have solved the same problem by using the Particle Swarm Algorithm (PSO), Cuckoo Search (CS) algorithm, Cuckoo optimization algorithm based on Chaotic disturbance (CH-CS) and CH-EOBCCS algorithm addressed in the literature. The results were obtained by the CH-EOBCCS algorithm for the case study problem is compared with PSO, CH-CS, CS algorithms. We tested the algorithms with the similar parameters used in the respective literature studies and the same number of iterations. The results that four kinds of algorithms were conducted 50 times for the case study problem are presented in Table 3, where "Mean" indicates the mean function value, "Std" stands for the standard deviation. Graph optimization search of four kinds of algorithms for the eight functions are displayed in Fig. 3(a-h). The results are presented in Table 4. The best results among the four algorithms are shown in bold.

From the results in the Fig. 3(a-h), it can be concluded that the CH-EOBCCS algorithm performs better than the PSO, CH-CS, CS algorithms in all test cases. Especially, it can be seen that CH-EOBCCS outperforms PSO, CH-CS and CS on f1, f2, f3, f4, f5 and f8. The CH-EOBCCS shows excellent search abilities on f1, f2, f3, f4 and f8. Although, the evolution rate of CH-CS is close to CH-EOBCCS on f1, CH-CS is premature convergence and CH-EOBCCS convergence accuracy is much higher than CH-CS. The CH-EOBCCS found the global optimum at about 250 generations on function f2. However, other algorithms are fall into the local minima. The CH-EOBCCS found the global optimum at about 250 generations at about 180 generations on function f3; it is faster than other algorithms. On function f4, CH-EOBCCS can fast find the global optimum while CS, PSO and CH-CS fall into the local minima; on function f5, CH-EOBCCS can find the global optimum CH-EOBCCS before the 430 generations. Although, CH-EOBCCS is slower than CS and PSO on the convergence rate, finding the optimal solution is superior to other algorithms. In f7, although the four algorithm can converge to the global minimum, the CH-EOBCCS is faster than other algorithms significantly. From the results in Fig. 3(a-h), PSO,

Table 3: Result comparison of different algorithm

Function	Mean/Std	PSO	CS	CH-CS	CH-EOBCCS
f1	Mean	3.343E-40	1.152E-59	5.957E-100	**0.00E+00**
	Std	5.221E-40	3.970E-59	6.349E-101	**0.00E+00**
f2	Mean	2.254E-04	1.776E-04	3.306E-12	**0.00E+00**
	Std	4.112E-03	2.103E-03	7.513E-11	**0.00E+00**
f3	Mean	1.876E+01	7.700E+00	5.774E-09	**0.00E+00**
	Std	3.901E+00	1.993E+00	5.712E-08	**0.00E+00**
f4	Mean	1.177E+01	2.771E-05	2.211E-12	**0.00E+00**
	Std	0.875E+00	8.588E-06	0.965E-11	**0.00E+00**
f5	Mean	2.103E-10	2.334E-15	2.594E-17	9.535E-18
	Std	5.417E-11	5.680E-16	1.060E-16	1.773E-18
f6	Mean	7.902E-02	2.661E-03	9.241E-04	5.811E-05
	Std	1.981E-02	6.337E-03	4.568E-05	3.387E-05
f7	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f8	Mean	1.082E-01	1.259E-02	7.129E-06	**0.00E+00**
	Std	3.855E-02	1.102E-02	3.728E-05	**0.00E+00**

Table 4: Result comparison of different algorithm in literature

Function	Mean/generate	GCS	ASCS	ECCS	MACS	CH-EOBCCS
f1	Mean	2.679E-10	1.006E-006	5.224E-18	2.0307E -7	**0.00E+00**
	Generate	408.3	**180.9**	237.75	1200	641
f2	Mean	4.599E-11	1.2444	6.247E-14	3.0335E-8	**0.00E+00**
	Generate	230.5	140.8	302.75	500	132
f3	Mean	3.704E-07	17.9013	1.615E-12	1.3833E-3	**0.00E+00**
	Generate	450.8	176.25	309.25	1500	96
f4	Mean	3.704E-07	17.9013	1.615E-12	0.4895	**0.00E+00**
	Generate	450.8	176.25	309.25	1500	95
f5	Mean	5.324E-08	0.0019	5.482E-14	16.5568	**9.535E-18**
	Generate	428.15	313.25	**296.2**	1100	1566

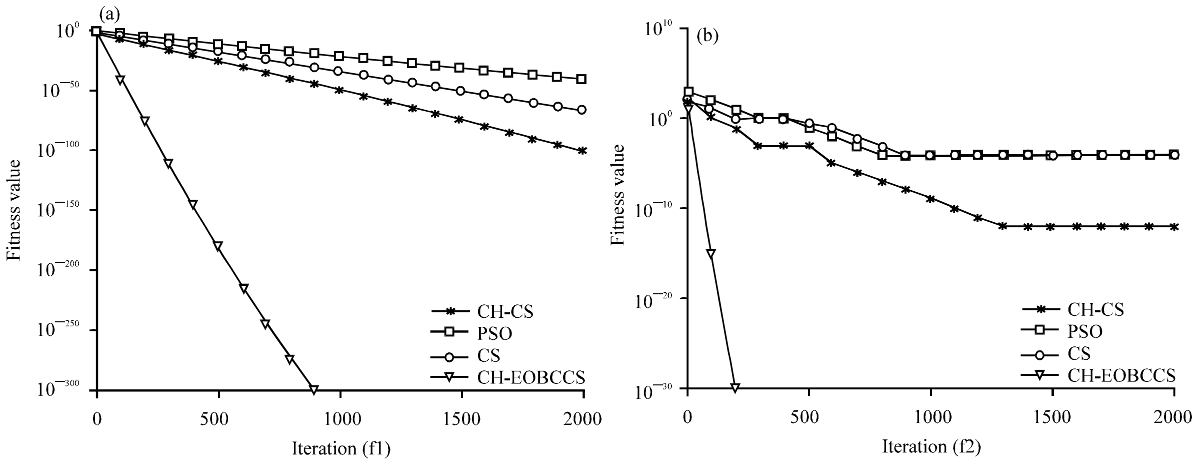

Fig. 3(a-h): Continue

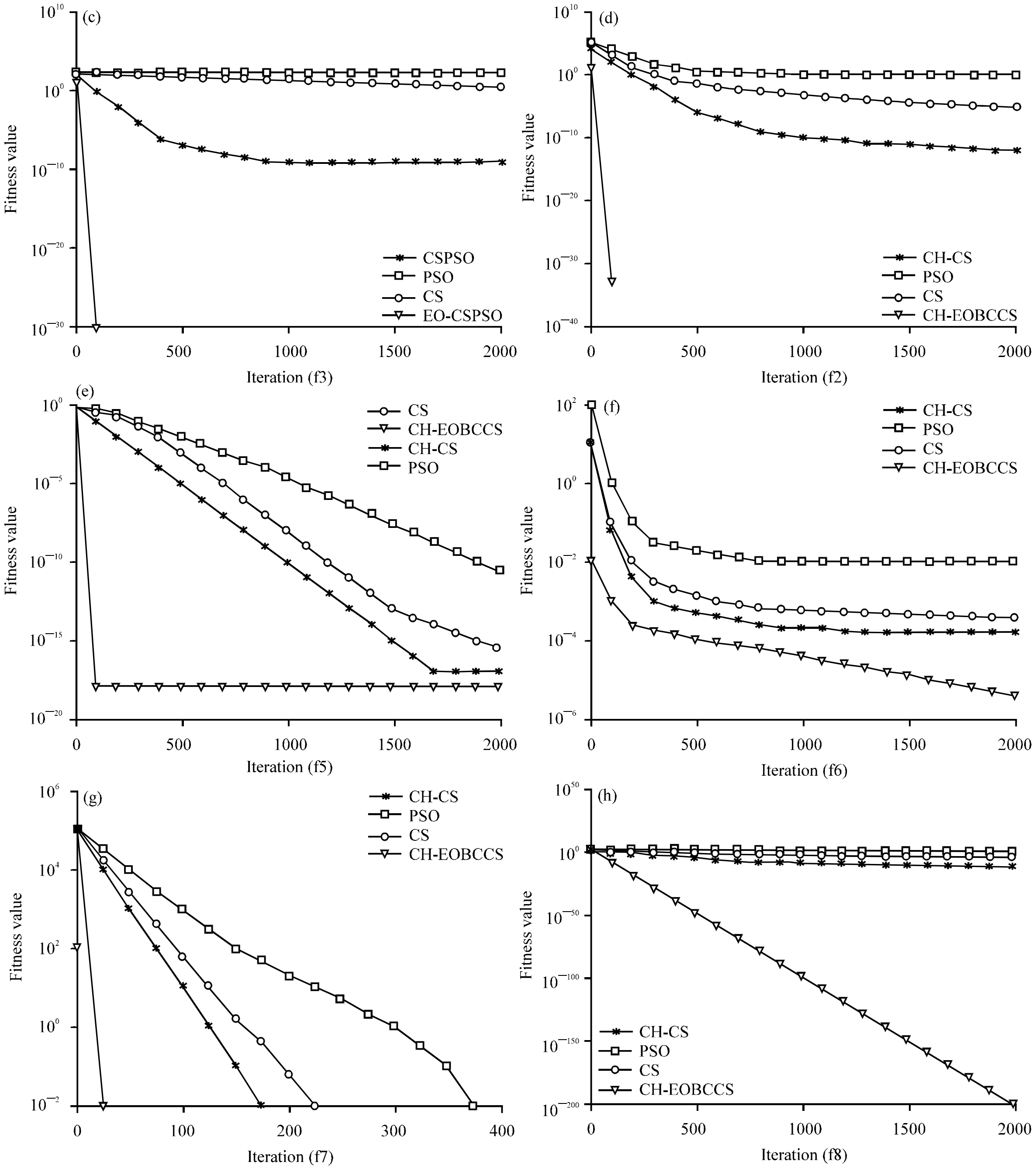

Fig. 3(a-h): Convergence trend of each algorithm on (a) Sphere/f1, (b) Griewank/f2, (c) Rastrigrin/f3, (d) Rosenbrock/f4, (e) Ackley/f5, (f) Quartic/f6, (g) Step /f7 and Schwefel2.22/f8

CH-CS and CS fall into the local minima on function f1-f6, f8, CH-EOBCCS shows excellent search abilities; either the convergence rate or convergence precision.

It can be seen that CH-EOBCCS outperforms PSO, CH-CS and CS on Mean and Std in Table 2. The CH-EOBCCS obtains theoretical extremes 0 on these functions and standard deviation is also 0. Especially, CH-EOBCCS shows excellent search abilities and search precision on f1, f2, f3, f4, f5, f6 and f8. Although CH-EOBCCS did not obtain optimal theoretical extremes 0 on f5 and f6, it obtain better solution than other algorithms. The result of CH-EOBCCS algorithm improves 60 the number of level than other algorithm on f1 on function f2. The result of CH-EOBCCS algorithm improves 9 number level than other algorithm. On function f3, although, CH-EOBCCS performs slightly better than CH-CS on Mean, CH-EOBCCS is obvious better than other two algorithms. The result of CH-EOBCCS algorithm improves 5 number level than other algorithm on function f4, the result of CH-EOBCCS algorithm improves 8 number level than other algorithm on function f5. The result of CH-EOBCCS algorithm improves 3 number level than other algorithm on function f6. The result of CH-EOBCCS algorithm improves 6 number level than other algorithm on function f8. In addition, through the comparison of Std, the stability of the CH-EOBCCS is higher than other algorithms.

Comparison of CH-EOBCCS and previous best-known improved CS algorithms: To further verify the effectiveness of the algorithm, we compare the performance of CH-EOBCCS with 4 previous best-known algorithms from the literature. They are: GCS algorithm (Wang *et al.*, 2011), ASCS algorithm (Zheng and Zhou, 2013), ECCS algorithm (Qu *et al.*, 2014) and MACS algorithm (Zhang *et al.*, 2012). The numerical results are shown in Table 4, where the best solutions of each problem are reported and compared with solutions reported previously in the literature. The results of each algorithm are as follows.

From Table 4, we observe that the mean solution obtained by CH-EOBCCS for f_1-f_4 is 0.00E+00 which is superior to those all by the other typical approaches in the literature.

For function f1, although, the ASCS algorithm demonstrates a far better average convergence precision than CH-EOBCCS algorithm, the ASCS prematurely into local optima. In the optimization process the convergence rate of CH-EOBCCS algorithm are 133.1, 95.7 and 94.9 on f2, f3 and f4. For the convergence rate, the CH-EOBCCS algorithm demonstrates a far better average convergence precision than GCS, ASCS, ECCS, CS-PSO and MACS and CH-EOBCCS obtains theoretical extremes 0. For function f5, although, the CH-EOBCCS algorithm demonstrates slower than CH-EOBCCS algorithm for average convergence precision. The mean solution improves 4 number level than ECCS algorithm. According to the analysis results of optimal solution and results of the convergence rate: It conclude that CH-EOBCCS algorithm has more advantages than these improved CS algorithm.

CONCLUSION

This study presents an enhanced CS algorithm called CH-EOBCCS by using elite opposition-based learning and chaotic disturbance. Exploration position space search area was expanded by using chaotic disturbance strategy which expand diversity of the population and which avoid falling into local extreme. Opposition-based learning is used to accelerate the convergence rate by simultaneously evaluating the current population and the opposite population. The algorithm generates current population with a certain probability and searches other members among field of opposite solution to form opposite populations. This method use fully the characteristics those elite members are better than ordinary members search space. The elite opposition-based learning sampling scheme could provide more chance of finding better solutions by transforming candidate solutions from current population into a new search space. With the help of a new elite selection mechanism, we can select better eggs. The CH-EOBCCS algorithm can improve the search ability to accelerate the convergence speed. From the analysis and experiments, we observe that the elite opposition-based learning enables the CH-EOBCCS to achieve better results on search abilities when CH-EOBCCS is compared with three other algorithms. Elite opposition-based learning is used to accelerate the convergence rate by simultaneously evaluating the current population and the opposite population. It performs badly on shifted and large scale problems. Possible future work is to investigate the effectiveness of elite opposition-based learning on many different kinds of problems.

ACKNOWLEDGMENT

Collaborative Innovation Center for Modern Logistics and Business of Hubei (Cultivation); (2011A201315), Wuhan Technology and Business University (A2014021) are acknowledged for their help.

REFERENCES

Ahandani, M.A. and H. Alavi-Rad, 2015. Opposition-based learning in shuffled frog leaping: An application for parameter identification. Inform. Sci., 291: 19-42.

Chandrasekaran, K. and S.P. Simon, 2012. Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm. Swarm Evol. Comput., 5: 1-16.

Layeb, A., 2011. A novel quantum inspired cuckoo search for knapsack problems. Int. J. Bio-Inspired Comput., 3: 297-305.

Lu, F. and L.Q. Gao, 2010. Adaptive differential evolution algorithm based on multiple subpopulation with parallel policy. J. Northeastern Univ. (Nat. Sci.), 31: 1538-1541.

Qu, C.W., Y.M. Fu and X.L. Huang, 2014. Cuckoo optimization algorithm with chaos disturbance based on the communication operator. J. Chin. Comput. Syst., 35: 134-140.

Tizhoosh, H.R., 2005. Opposition-based learning: A new scheme for machine intelligence. Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Volume 1, November 28-30, 2005, Vienna, Austria, pp: 695-701.

Tuo, S.H., 2013. Dynamic self-adaptive harmony search algorithm based on opposition-based computing and Gaussian distribution estimation. J. Chin. Comput. Syst., 34: 1158-1162.

Valian, E., S. Tavakoli, S. Mohanna and A. Haghi, 2013. Improved cuckoo search for reliability optimization problems. Comput. Ind. Eng., 64: 459-468.

Wang, Y. and Z. Cai, 2009. A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems. Front. Comput. Sci. Chin., 3: 38-52.

Wang, F., X.S. He and Y. Wang, 2011. The cuckoo search algorithm based on Gaussian disturbance. J. Xi'an Polytechnic Univ., 25: 566-569.

Wang, L.J., Y.L. Yin and Y.W. Zhong, 2013. Cuckoo search algorithm with dimension by dimension improvement. J. Software, 24: 2687-2698.

Yang, X.S. and S. Deb, 2009. Cuckoo search via levy flights. Proceedings of the World Congress on Nature and Biologically Inspired Computing, December 9-11, 2009, Coimbatore, India, pp: 210-214.

Yang, X.S. and S. Deb, 2010. Engineering optimisation by cuckoo search. Int. J. Math. Modell. Numer. Optim., 1: 330-343.

Zhang, Y., L. Wang and Q. Wu, 2012. Modified Adaptive Cuckoo Search (MACS) algorithm and formal description for Global optimisation. Int. J. Comput. Applic. Technol., 44: 73-79.

Zheng, H.Q. and Y.Q. Zhou, 2013. Self-adaptive step cuckoo search algorithm. Comput. Eng. Applic., 49: 68-71.

Zhou, X.Y., Z.J. Wu, H. Wang, K.S. Li and H.Y. Zhang, 2013. Elite opposition-based particle swarm optimization. Acta Electronica Sinica, 41: 1647-1652.