Cockroach Swarm Optimization

Chen ZhaoHui
Department of Mathematics and Physics
Chongqing University of Science and Technology
Chongqing, China
chen.zhh@163.com

Tang HaiYan
Department of Mathematics and Physics
Chongqing University of Science and Technology
Chongqing, China
thy13579@163.com

Abstract—**This paper presents a new bionic algorithm, entitled Cockroach Swarm Optimization (CSO), that is inspired by the social behavior of cockroaches. We construct some models by imitating the foraging behaviors of cockroach, and describe the steps of CSO. Simulation results illustrate that CSO has stronger convergence performance and highly-accuracy optimization results compares with Particle Swarm Optimization (PSO), Chaotic Particle Swarm Optimization (CPSO) and Artificial Fish-swarm Algorithm (AFSA).**

Keywords-Cockroach Swarm Optimization; Particle Swarm Optimization; Chaotic Particle Swarm Optimization; Artificial Fish-swarm Algorithm; Global Optimization

I INTRODUCTION

With the deep research of biology, many algorithms derived from natural biological processes were proposed , for example, the Genetic Algorithms (GA)[1] inspired by the process of biological evolution, Particle Swarm Optimization (PSO) algorithm[2][3], Ant Colony Optimization (ACO) algorithm[4], Artificial Fish Swarm Algorithm (AFSA)[5] and Bacterial Chemotaxis Algorithm (BCA)[6] inspired by animals foraging. These algorithms based on biological collective wisdom are known as bionic optimization algorithm, which have been widely used in many fields, such as numerical optimization, system control and robotics.

The Cockroach Swarm Optimization (CSO) presented in this paper, is an optimization algorithm inspired by the behaviors of cockroach swarm foraging. The cockroach belongs to Insecta Blattodea, likes warm, dark and moist places, has the habits, such as omnivorous, swarming, chasing, dispersing, ruthless. Its swarming and chasing habits reflect cockroaches can communicate with each other effectively. Cockroach can feel the change of surroundings because of its sensitive antennae. The dispersing habit refers to cockroaches can quickly disperse to the surroundings while the environment has a sudden change, this can reduce the harm from predators which may improve the survival rate [7]. The ruthless habit is defined as while food shortage there will occur the bigger eat the smaller, the stronger eat the weaker. It is these habits that can be used as a model for our function optimization algorithm.

CSO is constructed mainly through imitating the chase-swarming behavior of cockroach individuals, to search the global optimum. But if only carrying out this behavior, the CSO may fall into local optimum, while dispersing behavior may keep individuals diversity, at the same time, to imitate ruthless behavior can improve the results.

II COCKROACH SWARM OPTIMIZATION

CSO is inspired by the social behavior of cockroaches, and its model is constructed by imitating the behaviors which are chase-swarming, dispersing, ruthless behavior. Located in the D-dimensional search space R^D, there is a cockroach cluster contain N cockroach individuals, the ith individual represents a D-dimensional vector $X(i)=(x_{i1},x_{i2},\ldots,x_{iD})$ (i = 1,2, ..., N), the location each of the individual is a potential solution.

A. Cockroach Behavior Models

1) Chase-Swarming behavior: Each individual $X(i)$ will chase the local optimum individual $P(i)$ within its visual scope, of course, such chasing behavior also produced cluster simultaneously. There is an assumption, when an individual is the best one within its visual scope, it will chase the global optimum individual Pg. The model is:

$$X'(i)=\begin{cases} X(i)+step\cdot rand\cdot(P(i)-X(i)), X(i)\neq P(i) \\ X(i)+step\cdot rand\cdot(Pg-X(i)), X(i)=P(i) \end{cases} \quad (1)$$

where $step$ is a fixed value, $rand$ is a random number within [0,1],

$$P(i)=\mathrm{Opt}_j\{X(j)\,|\,|X(i)-X(j)|\leq visual, j=1,2,\ldots,N\} \quad (i=1,2,\ldots,N) \quad (2)$$

$$Pg=\mathrm{Opt}_i\{X(i), i=1,2,\ldots,N\} \quad (3)$$

2) Dispersing behavior: At intervals of certain time, each individual be dispersed randomly, so that it may keep the current individual diversity. The model is:

$$X'(i)=X(i)+rand(1,D), i=1,2,\cdots,N \quad (4)$$

where $rand(1,D)$ is a D-dimensional random vector that can be set within a certain range.

3) Ruthless behavior: At intervals of certain time, the current best individual replaces an individual that be selected randomly, and that is the stronger eat the weaker. The model is:

$$X(k)=Pg \quad (5)$$

where k is a random integer within [1, N].

B. Cockroach Swarm Optimization Algorithm

1) Parameters

$X(i)$ denotes the current location of the ith cockroach individual, $X'(i)$ denotes the new location, $visual$ denotes the visual distance of cockroach, $P(i)$ denotes the optimal individual within the visual scope of $X(i)$, Pg

denotes the global optimal individual, *step* is a fixed parameter, *N* is the population size, D is the space dimension.

2) Description of the CSO algorithm

According to the cockroach behavior models, we give the steps of CSO algorithm as follow:

Step 1: Parameters setting and population initializing. Set the value of parameters *step*, *N*, *D*; generate a cockroach population $X(i)=(x_{i1},x_{i2},\ldots,x_{iD})$ $(i=1,2,\ldots,N)$ within feasible region randomly.

Step 2: To search for $P(i)$ and Pg by the equation (2), (3).

Step 3: To carry out chase-swarming behavior by the equation (1). $X(i) \leftarrow X'(i)$, Update Pg.

Step 4: To carry out dispersing behavior by the equation (4). If the new position $X'(i)$ superior than the original location $X(i)$, then, $X(i) \leftarrow X'(i)$,otherwise, return to the original location $X(i)$. Update Pg.

Step 5: To carry out ruthless behavior by the equation (5).

Step 6: Termination checking: Whether meet the terminate condition? Yes, output; otherwise, return to step 2.

C. Performance and Complexity Analysis

CSO algorithm can be illustrated by 2-*D* model (see Fig.1). The chase-swarming behavior lead to the fact that $X(i)$ always approach to $P(i)$. Similarly, $P(i)$ approach to Pg. But chase-swarming may lead to all individuals gather around a position, fail to get global optimum and get an accurate result. While dispersing behavior may keep individuals diversity, which can improve the ability of avoiding fall into local optimum. Ruthless behavior can enhance the ability of local searching. The latter two behaviors not only can help CSO escape from local optimum, but also improve the accuracy. Its convergence will be illustrated in Section Ⅲ by the results of solving benchmark Problems. Its time complexity is a polynomial of *N*, that is $O(C\cdot N^2)$, where C represents a constant.

III EXPERIMENTS

This section presents several experiments that show the relative ability of the CSO, PSO, CPSO[8] and AFSA in finding global minima of Benchmark Problems as TABLE Ⅳ.

The experiments were performed on PC of which CPU2.80GHz, memory 1.0G with MATLAB7.0. Parameters were designed as follow: perception distance *visual*=2, the largest step size *step*=1.5 while solving Rastrigin's problem for CSO and AFSA. In solving other problems, population size *N*=50, perception distance *visual*=5, the largest step size *step*=2 for CSO; Population size *N*=50, learning factor $c_1=c_2$=1.4962, inertia weight *w*=0.7298 for PSO and CPSO; Population size *N*=50, perception distance *visual*=3.5, the largest step size *step*=1.5, crowed factor $\delta=0.618$, try number *trynumber*=6 for AFSA. Maximum iteration degree is 1000.

Test function F_1 has numerous local minima within the entire range (see Fig.2), which can easily lead to find many local minima and also visit the global one but does not stop here[6]. These four algorithms run 20 times to solving F_1 respectively, CSO find the optima by the ratio of 60%, the other three algorithms cannot find optima success, the mean results in TABLE Ⅰ, which illustrate that the accuracy of CSO is superior to the other three.

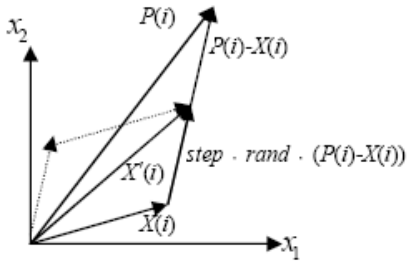

Figure 1. 2-D path of a cockroach consisting of the $X(i)$ chase $P(i)$ yielding the new position $X(i)$

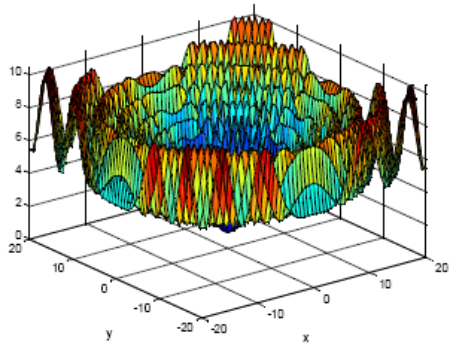

Figure 2. Figure 2. The surface figure of function

Test function F_2, that is, Easom Function [9], there are numerous local minima located on concentric circles (Fig.3).

These four algorithms run 20 times to find the minima of F_2 respectively, and solving Rosenbrock's problem (2-D) with precision of 10^{-6} for the stop criteria, the mean iteration degree in TABLE Ⅱ. These four algorithms iterative 100 times respectively, obtain the curve of optimal value shown in Fig.4. All these show CSO convergent faster than the other three algorithms.

TABLE Ⅲ outlines the results of the testing of CSO, PSO, CPSO, and AFSA, on Sphere, Rastrigin, Rosenbrock, Schwefel functions. All these functions were tested in 2, 10 and 30 dimensions respectively. The data in brackets denotes the radio of reached optima. The values in TABLE Ⅲ indicate CSO algorithm performed best for the latter two functions, performed best for each function in 30 dimensions. But CSO algorithm is difficult to escape from local minima for Rastrigin function in 2 dimensions. In essence, these show that CSO algorithm performed as good or better than PSO and CPSO for some functions, better than AFSA, especially for optimizing highly-dimension functions.

TABLE I. MEAN RESULTS FOR THE DIFFERENT ALGORITHMS

Test Function	PSO	CPSO	AFSA	CSO
F_1	8.1×10^{-24}	1.5×10^{-25}	0.020738	**0（60%）**

TABLE II. TERATIVE DEGREE OF SOLVING F_1 AND ROSENBROCK FOR THE DIFFERENT ALGORITHMS

Test Function	PSO	CPSO	AFSA	CSO
F_2	60	48.7	232.6	9.2
Rosenbrock	78.6	77.1	177.2	18

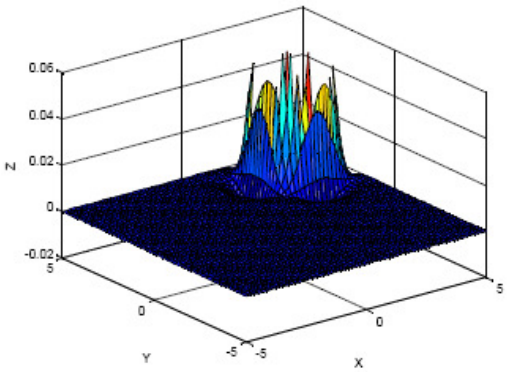

Figure 3. . The surface figure of function -F_2

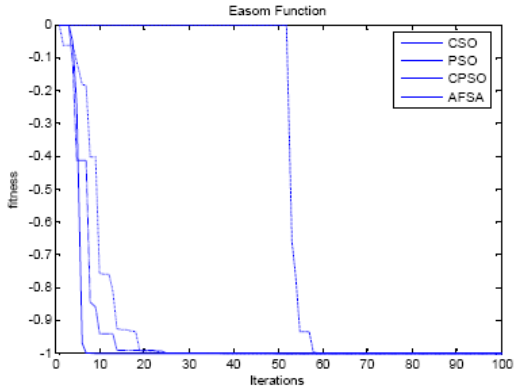

Figure 4. The curve of Easom function optimal value for the different algorithms

IV CONCLUSIONS

The results presented in Table Ⅱ-Ⅳ show that CSO not only possesses the ability of escaping from local optima, but also can reach highly-accuracy results. Thus it can be concluded that the social behavior of cockroaches is an effective model for constructing CSO algorithm. But analysis from the structure of CSO algorithm, the minima within the scope of each cockroach individual its own perception must be searched for each iteration. So this will lead to iteration takes a long time. In generally, by the preliminary study on the swarm intelligence optimization method CSO in this paper, the results show that it's necessary to further the study, and is worth developing and applying.

REFERENCES

[1] Holland J. Adaptation in Natural and Artificial Systems. Ann Arbor, MI：University of Michigan Press, 1975.

[2] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway：IEEE Service Center, 1995, 1942-1948.

[3] Shi Y, Eberhart R C. A modified particle swarm optimizer. Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, 1998, 69-73.

[4] M.Dorigo, "Learning and Natural Algorithms," Ph.D. dissertation (in Italian), Dipartimento di Elettromazione, Politecnico di Milano, Milano, Italy, 1922.

[5] [5] Xiaolei Li, Zhijiang Shao, Jixin Qian. An Optimizing Method Based on Autonomous Animals: Fish-swarm Algorithm. Systems Engineering Theory and Practice, 2002, 22(11): 32-38.

[6] Muller S D, J. Airaghi Marchetto S, Koumoutsakos P. Optimization Based on Bacterial Chemotaxis[J]. IEEE Transaction of Evolutionary Computation, 2002, 6(1):16-29.

[7] Dawei Li, Zhaohui Chen, Binyan Zhang. Particle Swarm Optimization Algorithm with Chaos and Its Performance Testing. The Second International Conference on Impulsive Dynamical Systems and Applications, 2005, 1330-1335.

[8] E. Easom, A survey of global optimization techniques, Master's thesis, Univ. Louisville, Louisville, KY,1990.

TABLE III. MEAN RESULTS FOR THE DIFFERENT ALGORITHMS

Test Function	Dimensions	PSO	CPSO	AFSA	CSO	Actual Value

Sphere	2	2.1×10^{-91}	3.0×10^{-99}	5.3×10^{-8}	**0**	0
	10	3.6×10^{-7}	**1.6×10^{-7}**	12.572955	4.2×10^{-6}	0
	30	1795.443	60.8606	1212.083	**9.5×10^{-5}**	0
Rastrigin	2	0（85%）	**0（100%）**	2.1×10^{-5}	0（25%）	0
	10	30.478828	**20.797431**	21.424781	21.890013	0
	30	275.020595	75.120430	154.749154	**58.023333**	0
Rosenbrock	2	0	0	2.0×10^{-7}	0	0
	10	5.208933	2.374827	346.15508	**0.813931**	0
	30	2.7×10^{6}	6361.67	7.5×10^{5}	**86.58846**	0
Schwefel	2	6.4×10^{-92}	1.6×10^{-99}	2.1×10^{-8}	**0**	0
	10	2.714103	4.4×10^{-5}	166.6706	**4.9×10^{-6}**	0
	30	4700.5620	576.7357	754.8838	**0.001124**	0

TABLE IV. BENCHMARK PROBLEMS

Test Function	Range	Minimum
$F_1(x_1,x_2)=(x_1^{\,2}+x_2^{\,2})^{0.25}\cdot(\sin^2(50(x_1^{\,2}+x_2^{\,2})^{0.1})+1.0)$	[-20,20]	$F_1(0,0)=0$
Easom: $F_2(x_1,x_2)=-\cos x_1\cos x_2\cdot e^{-(x_1-\pi)^2}e^{-(x_2-\pi)^2}$	[-100,100]	$F_2(\pi,\pi)=-1$
Sphere: $f(\vec{x})=\sum_{i=1}^{D}x_i^{\,2}$	[-100,100]	$f(\vec{0})=0$
Rastrigin: $f(\vec{x})=\sum_{i=1}^{D}[x_i^{\,2}-10\cos(2\pi x_i)+10]$	[-5.12，5.12]	$f(\vec{0})=0$
Rosenbrock: $f(\vec{x})=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^{\,2})^2+(1-x_i)^2]$	[-50,50]	$f(\vec{0})=0$
Schwefel: $f(\vec{x})=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	[-100,100]	$f(\vec{0})=0$