Available online at www.sciencedirect.com

SciVerse ScienceDirect

Energy Procedia 11 (2011) 4 – 9

Energy Procedia

ICEES 2011: 14–16 October 2011, Singapore

A modified cockroach swarm optimization

Zhaohui Chen*

Dept. of Mathematics & Physics, Chongqing University of Science and Technology, Chongqing, China

Abstract

In order to apply cockroach swarm optimization (CSO) algorithm to more function optimization and improve its accuracy, this paper improve CSO by modifying chase-swarming behavior of cockroach, and apply the modified CSO to optimize several Benchmark Problems. The results of simulation experiments illustrate that modified CSO has stronger convergence performance and highly-accuracy optimization results compared with Particle Swarm Optimization (PSO) and Chaotic Particle Swarm Optimization (CPSO).

Keywords: Cockroach swarm optimization; Particle swarm optimization; Chaotic particle swarm optimization

1. Introduction

There are many algorithms derived from natural biological processes, 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], Bacterial Chemotaxis Algorithm (BCA)[6], artificial bee colony (ABC) algorithm[7][8] and Roach Infestation Optimization[9] 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)[10] is an optimization algorithm inspired by the behaviors of cockroach swarm foraging. The cockroach belongs to Insecta Blattodea, likes warm, dark and moist places, have the habits, such as omnivorous, swarming, chasing, dispersing, ruthless behavior. 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 have a sudden

* *E-mail address:* chen.zhh@163.com

doi:10.1016/j.egypro.2011.10.201

change, this can reduce the harm from predators to improve the survival rate. 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 to construct an 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 trap in local optimum, while dispersing behavior may keep individuals diversity, at the same time, to imitate ruthless behavior can improve the accuracy.

2. Cockroach swarm optimization

CSO is inspired by the social behavior of cockroaches, and its model is constructed by imitating the behaviors that 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.

2.1. Cockroach behavior models

2.1.1. 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:

$$X_i = \begin{cases} X_i + step \cdot rand \cdot (P_i - X_i), X_i \neq P_i \\ X_i + step \cdot rand \cdot (Pg - X_i), X_i = P_i \end{cases}, i = 1,2,\cdots,N \tag{1}$$

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

$$P_i=\mathrm{Opt}_j\{X_j \mid |X_i-X_j|\leq visual, j=1,2,\ldots,N\} \quad (i=1,2,\ldots,N) \tag{2}$$

$$Pg=\mathrm{Opt}_i\{X_i, i=1,2,\ldots,N\} \tag{3}$$

2.1.2. Dispersing behavior

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

$$X_i = X_i + rand(1,D), i = 1,2,\cdots,N \tag{4}$$

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

2.1.3. 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:

$$X_k=Pg \tag{5}$$

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

2.2. The modified cockroach swarm optimization algorithm

2.2.1. Parameters

X_i denotes the current location of the ith cockroach individual, $visual$ 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.

2.2.2. Description of the CSO algorithm

According to the cockroach behavior models, we modify (1) as:

$$X_i = \begin{cases} w \cdot X_i + step \cdot rand \cdot (P_i - X_i), X_i \neq P_i \\ w \cdot X_i + step \cdot rand \cdot (Pg - X_i), X_i = P_i \end{cases}, i = 1,2,\cdots,N \quad (6)$$

where w is inertia weight. The steps of modified CSO algorithm as follows:

Step1: Parameters setting and population initializing. Set the value of parameters w, $step$, N, D; generate a cockroach population $X_i=(x_{i1},x_{i2},\ldots,x_{iD})$ (i=1,2,…,N) within feasible region randomly.

Step2: To search for P_i and Pg by the equation (2), (3).

Step3: To carry out chase-swarming behavior by the equation (6), Update Pg.

Step4: To carry out dispersing behavior by the equation (4). Update Pg.

Step5: To carry out ruthless behavior by the equation (5).

Step6: Termination checking: Whether meet the terminate condition? Yes, output; otherwise, return to step2.

The modified CSO algorithm is outlined as follow:

```
Algorithm: Cockroach Swarm Optimization
Input: Fitness Function F(X), X∈R^D
Parameters:
T_max=200, Maximum iterations
N=60, Number of cockroach agents
w=0.618, step=1.5, visual=1, cockroach parameters
Initialization:
set population X_i, i=1,2,…,N  randomly
set Pg=X_1
for i=2 to N do
  if F(X_i)<F(Pg) then Pg= X_i
for t=1 to T_max do
  for i=1 to N do
     for j=1 to N do
       if | X_i – X_j |<visual & F(X_j)<F(X_i) then
            P_i= X_j
       if P_i= = X_i then
            X_i = w×X_i +step×rand×(Pg- X_i)
       else X_i= w×X_i +step×rand×(P_i - X_i)
     if F(X_i)<F(Pg) then Pg= X_i
   for i=1 to N do
     X_i = X_i +rand(1,D)
     if F(X_i)<F(Pg) then Pg= X_i
   k=randint([1,N])
   X_k =Pg
```

3. Experiments

This section presents several experiments that show the relative ability of the modified CSO, PSO and CPSO[11] in finding global minima of Benchmark Problems as table1.

The experiments were performed on PC of which CPU2.80GHz, memory 1.0G with MATLAB7.0. Parameters were designed as follow: inertia weight w=1, perception distance $visual$=5, the largest step size $step$=1.5 while solving Rosenbrock's problem; w=1, $visual$=1, $step$=1.5 while solving Easom function; w=0.618, *visual*=0.1, *step*=1.5 while solving Rastrigin function for CSO. In solving other problems, w=0.618, *step*=1.5, *visual*=1. Learning factor c_1= c_2=1.4962, inertia weight w=0.7298 for PSO and CPSO. Population size N=60, maximum iteration degree is 200 for all algorithms, where the values of the inertia weight w are the suitable value obtained from many experiments.

Easom function[12], there are numerous local minima located around (π, π), many algorithms usually trap in a local optimum. CSO to solving the problem, the mean iteration is 10 to obtain the global minima. These three algorithms iterative 50 times respectively, obtain the curve of optimal value shown in Fig.1. Which illustrate that the evolution speed of CSO is superior to PSO and CPSO.

Schaffer function has numerous local minima located around (0, 0). These three algorithms run 20 times to solving it respectively, the mean run time, iterations and optimal rate of PSO, CPSO and CSO obtained the optima shown in table2. The same comparison is shown in table2 for Ackley's function.

Table3 shows the results of the testing of CSO, PSO and CPSO on Sphere, Rastrigin, Rosenbrock, Schwefel, Ackley, Schaffer, Griewangk functions. All these functions were tested in 2, 10 and 30 dimensions respectively. The data in brackets denotes the radio of reached optima. The results indicate modified CSO algorithm performed best for the latter two functions, performed best for each function in 30 dimensions. These show that CSO algorithm performed better than PSO and CPSO for some functions, especially for optimizing high-dimension functions.

4. Conclusions

The results presented in table 2-3 show that CSO not only possesses the ability of escaping from local optima, but also can achieve 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.

By several experiments, the results show that the parameters of CSO will serious influence the results for some problems. By experience *visual* usually value around 1, *visual* not only influence the speed but also influence the accuracy. *Step* usually value the number 1 to 1.5, w usually value the number 0.5 to 1.

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] Li Xiaolei， Shao Zhijiang，Qian Jixin．An Optimizing Method Based on Autonomous Animats: Fish-swarm Algorithm. Systems Engineering-theory & Practice，2002，22(11)：32-38

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

[7] Basturk, B., Karaboga, D.. An artificial bee colony (ABC) algorithm for numeric function optimization. Proceedings of the IEEE Swarm Intelligence Symposium 2006, Indianapolis, Indiana, USA, 12–14 May 2006

[8] Dervis Karaboga, Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 2007, 39:459–471

[9] Timothy C. Havens, Christopher J. Spain, Nathan G. Salmon, James M. Keller. Roach Infestation Optimization. IEEE Swarm Intelligence Symposium, St. Louis MO USA, 2008

[10] Zhaohui Chen, Haiyan Tang. Cockroach swarm optimization. The Second International Conference on Computer Engineering and Technology, 2010, 6:652-655

[11] 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

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

Table1 Benchmark Problems

Test Function	**Range**	**Minimum**
Sphere: $f(X)=\sum_{i=1}^{D} x_i^{\ 2}$	[-100,100]	$f(\mathbf{0})=0$
Rastrigin： $f(X)=\sum_{i=1}^{D}[x_i^{\ 2}-10\cos(2\pi x_i)+10]$	[-5.12，5.12]	$f(\mathbf{0})=0$
Rosenbrock: $f(X)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^{\ 2})^2+(1-x_i)^2]$	[-50,50]	$f(\mathbf{0})=0$
Ackley： $f(X)=-20e^{-0.2\sqrt{\sum_{i=1}^{D}\frac{x_i^2}{D}}}-e^{\sum_{i=1}^{D}\cos\frac{2\pi x_i}{D}}+20+e$	[-50,50]	$f(\mathbf{0})=0$
Schaffer： $f(x_1,x_2)=\dfrac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{[1+0.001(x_1^2+x_2^2)]^2-0.5}$	[-100,100]	$f(0,0)=-1$
Easom: $f(x_1,x_2)=-\cos x_1\cos x_2\cdot e^{-(x_1-\pi)^2}e^{-(x_2-\pi)^2}$	[-100,100]	$f(\pi,\pi)=-1$
Griewangk: $f(X)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	$f(\mathbf{0})=0$
Schwefel 1.2: $f(X)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	[-100,100]	$f(\mathbf{0})=0$

Table2 . Mean run time, iterations and optimal rate of PSO, CPSO, CSO for Schaffer and Ackley

Test Function	algorithm	run time	iterations	Optimal rate
Schaffer	PSO	0.69	184	20%
	CPSO	0.63	172	30%
	CSO	3.23	15.8	100%
Ackley	PSO	0.51	112.85	100%
	CPSO	0.53	105	100%
	CSO	2.12	9.55	100%

Table3 Mean results of PSO, CPSO, CSO for some benchmark problems

Test Function	Dimensions	PSO	CPSO	CSO	Actual Value
Sphere	2	8.07×10^{-21}	4.42×10^{-22}	$\mathbf{2.32\times10^{-302}}$	0
	10	2.07×10^{-5}	1.85×10^{-8}	$\mathbf{5.08\times10^{-301}}$	0
	30	1.39×10^{3}	1.08×10^{3}	$\mathbf{6.28\times10^{-304}}$	0
Rastrigin	2	**0**	**0**	**0**	0
	10	27.41677	21.32524	**0**	0
	30	257.5718	142.7576	**0**	0
Rosenbrock	2	5.48×10^{-6}	1.59×10^{-16}	**0**	0
	10	74.57964	**27.8381**	77.69203	0
	30	1.86×10^{6}	5.41×10^{5}	$\mathbf{6.77\times10^{3}}$	0
Ackley	2	7.79×10^{-11}	2.22×10^{-11}	$\mathbf{-8.88\times10^{-16}}$	0
	10	3.551093	3.5200000	$\mathbf{-8.88\times10^{-16}}$	0
	30	12.61584	12.849876	$\mathbf{-8.88\times10^{-16}}$	0
Schaffer	2	-1(20%)	-1（30%）	**-1 (100%)**	-1
Easom	2	-1	-1	**-1**	-1
Griewangk	2	0.003082	0.002219	**0**	0
	10	0.213207	0.232714	**0**	0
	30	14.26552	11.88633	**0**	0
Schwefel 1.2	2	1.45×10^{-20}	1.61×10^{-22}	$\mathbf{1.022\times10^{-297}}$	0
	10	35.14274	19.49408	$\mathbf{1.75\times10^{-280}}$	0
	30	1.14×10^{4}	9.57×10^{3}	$\mathbf{1.2\times10^{-298}}$	0

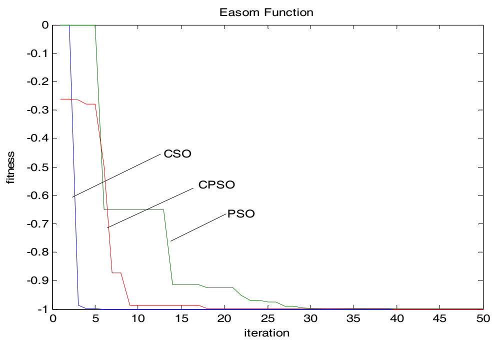

Figure 1. The curve of PSO, CPSO, CSO algorithms solving Easom function