PoS PROCEEDINGS OF SCIENCE

A Study on Improved Cockroach Swarm Optimization Algorithm

Le Cheng[1]

Department of Computer Science and Engineering, Huaian Vocational College of Information Technology，Huaian 223003, China
College of Computer and Information, Hohai University, Nanjing 210098, China
E-mail: cl211282@163.com

Yan-hong Song[a], **Meng-an Shi**[b]

Department of Computer Science and Engineering, Huaian Vocational College of Information Technology
Huaian 223003, China

Yu-jia Zhai

Department of Chemistry, Duke University
Durham, NC 27708, USA

Yue-tang Bian

School of Business, Nanjing Normal University
Nanjing, 210023, China

The original cockroach swarm optimization (CSO) algorithm has long been criticized for the problems of "early mature". This paper described a new cockroach-inspired algorithm, which is called CSO with global and local neighborhoods (CSOGL). In CSOGL, two kinds of neighborhood models are designedin order to increase the diversity of promising solution. Based on above two neighborhood models, two kinds of novel chase-swarming behaviors are proposed and applied to CSOGL. The comparison results show that the CSOGL algorithm outperform theoriginal CSO algorithms.

CENet2017
22-23 July 2017
Shanghai, China

[1] This work is supported by the National Natural Science Foundation of China (project 71301078 and 60971088), the Qing Lan Project, the Natural Science Foundation of Education Bureau of Jiangsu Province (Project 16KJB520049) and Innovation Foundation of Huaian College of Information Technology (Project hxyc2015001).

1.Introduction

In the past two decades, many nature- nspired algorithms have been proposed and applied to solve optimization problems, e.g., Genetic Algorithm (GA) [1], Differential Evolution (DE) [52, Particle Swarm Optimization (PSO) [2-3], ant Colony Optimization (ACO) [4], etc. One of the recent developments in biologically- inspired algorithms is a series of cockroach-inspired algorithms. Chen first proposed a cockroach swarm optimization (CSO) [5]. The CSO algorithm simulated the general behavior of cockroach, e.g., chase-swarming, dispersion and ruthless, etc. Furthermore, the modified CSO (MCSO) is presented by introducing the inertial weight in chase-swarming operation of CSO [6]. By studying the recent discoveries in the behavior of cockroaches, Havens et al. proposed a new cockroach-inspired algorithm, called roach infestation optimization (RIO) [7]. In essence, RIO has the different searching strategy with CSO. That is, RIO was not the improved version of CSO, but can be regarded as an improved version of PSO. Based on RIO, Havens proposed the hungry RIO (HRIO). It has proved that HRIO has better performance than that RIO and PSO[7].

We have proposed a series of cockroach-inspired algorithms for the robot path planning (RPP) and rod-like robot path planning (RLRPP) problems[8]. However, practical experience shows that the existing cockroach-inspired algorithms for numerical optimization problem generally bear the problem of of "early mature", which leads to the loss of diversity in the population. Therefore, the strategy of neighborhood can be introduced in the original CSO algorithm. In this paper, we propose a new variant of CSO, which is called CSO with global and local neighborhoods (CSOGL). The CSOGL algorithm extends the original CSO, and a novel neighborhood scheme is proposed and applied in CSOGL. The neighborhood scheme can generate a local optimal global optimum P_i, which is significantly different from the CSO. Furthermore, a new Chase-Swarming behavior is designed for CSOGL, which can implement the better tradeoff between the local and global search during the computing process.

The organization of this paper is as follows: Section 2 describes the original version of CSO; Section 3 gives the swarm scheme and the chase-swarming strategy of CSOGL;The experimental results are illustrated in section 4;Section 5 concludes this paper.

2.Cockroach Swarm Optimization

The original version of CSO simulates some basic biological behaviors of the cockroach, which include chase-swarming, dispersing, ruthless behavior. The CSO model is described as follows：

(1) Chase-swarming behavior:

$$X_{i,G+1}=\begin{cases} X_{i,G}+step\cdot r_1\cdot(P_{i,G}-X_{i,G}),\ X_{i,G}\neq P_{i,G} \\ X_{i,G}+step\cdot r_2\cdot(P_{g,G}-X_{i,G}),\ X_{i,G}=P_{i,G} \end{cases} \qquad (2.1)$$

Where, $X_{i,G}$ is the cockroach current position at the G-th generation, *step* is a constant value, and the random numbers r_1 and r_2 ∈ [0,1]. P_i is the best position of $X_{i,G}$, which can be computed by following equation:

$$P_i = Opt_j\{X_j \mid |X_i - X_j| \quad visual\} \qquad (2.2)$$

here, *visual* is the perception constant. $P_{g,G}$ is the global best position at the G-th iteration.

$$P_g = Opt\{X_i\} \qquad (2.3)$$

(2) Dispersion behavior

$$X_i = X_i + rand(1,D) \qquad (2.4)$$

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

(3) Ruthless behavior

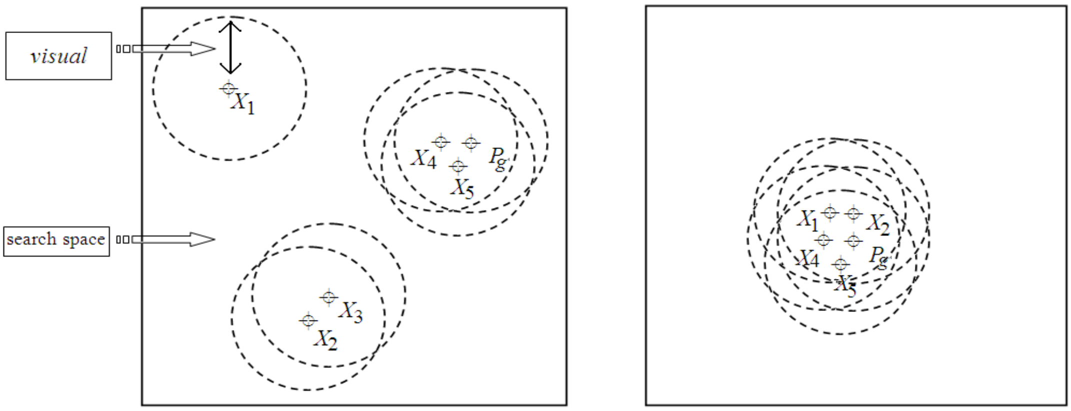
$$X_r = P_g \qquad (2.5)$$

where, r is a random integer within [1, N], P_g is the global best position.

3. Cso with Global and Local Neighborhoods

3.1 The Neighborhood Model of CSO

By many experimental researches, we found that CSO featured the problems of "early mature". Furthermore, we found that the swarm strategy of CSO is unreasonable.



(a) (b)

Figure 1: Swarm Strategy of CSO

In Equation (2), the swarm strategy $|X_i - X_j| \leq visual$ does not guarantee that each cockroach is in a sub-population or a sub-population with a certain size at the initial stage of CSO (See Fig.1(a)). However, after some iteration, all cockroaches are near around P_g and there exists only one sub-population that is the entire population (See Fig. 1 (b)).

In Fig. 1(a), the cockroach X_1 has no neighborhood in the range that is controlled by Equation 2 and X_1 hence does not belong to any sub-population. It Means that many cockroach individuals are in the sub-population that only includes itself, and other cockroach individuals may be in a sub-population with a big size. This problem can seriously deteriorate the diversity of promising solution.

3.2 Neighborhood Structures and Search Strategy of CSOGL

In CSOGL, two kinds of neighborhood structures are used, which are similar to the idea of literature [9]. One is called the local structure, where each $X_{i,G+1}$ is computed by the best position $P_{i,G}$. On the other hand, the second one is the entire population at current generation G.

K is defined as the scale of the neighborhood. Thus, the neighborhood structure can be illustrated as Fig. (2), which is a ring topology struct. In Fig. (2), N is the number of all cockroaches in CSOGL. We assume K=5, then cockroach X_i has five neighborhoods X_{i+0}, X_{i+1}, X_{i+2}, X_{i+3} and X_{i+4}. There exist some overlapping neighborhoods (See Fig.3).

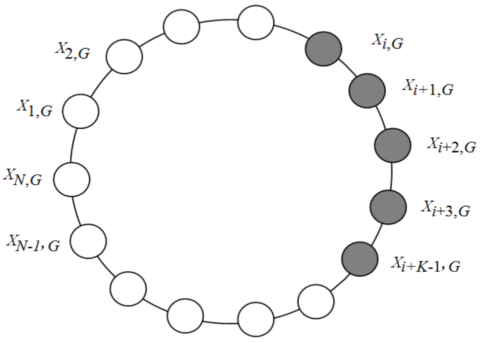


Figure 2: Neighborhood Model

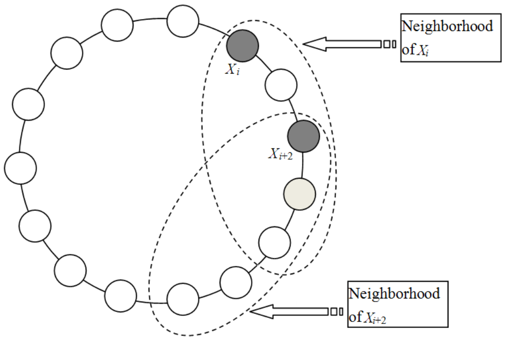


Figure 3: Overlapping Neighborhoods

According to above two neighborhood models, there are two kinds of chase-swarming behaviors.

The chase-swarming behavior for local neighborhood is as Equation (3.1):

$$F_{i,G} = X_{i,G} + (P_{i,G} - X_{i,G}) + (X_{r1} - X_{r2}) \tag{3.1}$$

Tere, $L_{i,G}$ denotes the new location found by the i-th cockroach. Similarly, the chase-swarming behavior for global neighborhood is as Equation (3.2):

$$F_{i,G} = X_{i,G} + (P_{g,G} - X_{i,G}) + (X_{r3} - X_{r4}) \tag{3.2}$$

In Equation (3.1) and Equation (3.2), the indices r_1,r_2,r_3,r_4∈[1, N].

Above two chase-swarming operations are chosen with a random method. Thus, the whole chase-swarming behavior of CSOGL can be described as Equation (3.3)

$$F_{i,G}=\begin{cases} X_{i,G}+(P_{i,G}-X_{i,G})+(X_{r1}-X_{r2}), \, rand(0,1)<0.5 \\ X_{i,G}+(P_{g,G}-X_{i,G})+(X_{r3}-X_{r4}), \, otherwise \end{cases} \tag{3.3}$$

where, *rand* (0, 1) is a uniformly distributed random number. In essence, the chase-swarming behaviors of local and global neighborhood correspond to the local and global searching on CSOGL.

Notice that $F_{i,G,}$ is the new location found by the *i*-th cockroach, which doesn't mean that $F_{i,G,}$ must be as position of *i*-th cockroach in *G*+1 generation. Any one of $F_{i,G,}$ and $X_{i,G,}$ is chosen to be $X_{i,G,+1}$ by the greedy selection scheme(See Equation (3.4)).

$$X_{i,G+1}=\begin{cases} F_{i,G}, \, if\ (f(F_{i,G})<f(X_{i,G})), \, minimization\ problems \\ X_{i,G}, \, otherwise \end{cases} \tag{3.4}$$

The complete pseudo-code of CSOGL is given in Algorithm 1.

Algorithm 1: *CSOGL*

```
INPUT: Fitness function: $f(X)$, $X \in$
Set parameters and generate an
Choose the $p_g$ from whole
Choose the $p_i$ for each cockraoch;
FOR $t$ = 1 to $G_{max}$
    FOR $i$= 1 to $N$
        chase-swarming operations (E.q(8));
        greedy selection scheme(See E.q(9))
        IF $f(X_{i,}) < f(P_i)$ THEN
            $P_i = X_i$;
        END IF
        IF $f(X_i) < f(P_g)$ THEN
            $P_g = X_i$;
        END IF
    END FOR
END FOR
Check termination condition
```

Table1: Benchmark Test Functions

Fn	**Description**	Search Range	Optimum Value
Rastrigin	$f(x)=10\cdot D+\sum_{i=1}^{D}((x_i^2)-10\cdot\cos(2\pi x_i))$	$-5.12 \le x_i \le 5.12$	$f_2(X^*)=0$
Rosenbrock	$f(x)=\sum_{i=1}^{D-1}(100\cdot(x_{i+1}-x_i^2)^2+(1-x_i)^2)$	$-30 \le x_i \le 30$	$f_3(X^*)=0$
Schwefel2.22	$f(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i}^{D}x_i$	$-10 \le x_i \le 10$	$f_6(X^*)=0$
Griewangk	$f(x)=1+\sum_{i=1}^{D}(\frac{x_i^2}{4000})-\prod_{i}^{D}(\cos(\frac{x_i}{\sqrt{i}}))$	$-600 \le x_i \le 600$	$f_7(X^*)=0$

We perform large number of simulation experiments, and compare the CSOGL with original CSO, MCSO, RIO and HRIO. All the benchmark functions are tested with 30 dimensions, and each test is run 20 times with maximum iteration 1000. Cockroach population size $N = 50$ is used in this paper for all the algorithms. Other control parameters of original CSO and CSO variants are set according to literature [5-7]. The test results are demonstrated in Table 2.

In Table 2, *Mean* denotes the mean function value, *STD* is the standard deviation of the function value during the 20 runs, and *Best* means the best function values. For most of test functions, MCSO demonstrates better performance than that of CSO, RIO and HRIO. Compared with MCSO, CSOGL can give smaller function values by using the same numbers of function evaluations. That is, the performance of CSOGL is significantly superior to the existing cockroach-inspired algorithm. The standard deviation of the function value shows that CSOGL is stable.

Table 2: Test results of CSO, MCSO, RIO, HRIO, and CSOGL

Function		CSO	MCSO	RIO	HRIO	CSOGL
Rastrigin	*Mean*	3.602E+03	9.199E-11	3.814E-05	3.215E-04	0.000E+00
	STD	5.573E+03	3.946E-10	3.444E-05	3.000E-04	0.000E+00
	Best	3.134E-04	0.000E+00	2.710E-07	2.145E-07	0.000E+00
Rosenbrock	*Mean*	9.507E+11	2.900E+01	2.528E+06	3.357E+06	3.400E+01
	STD	2.271E+12	0.000E+00	4.053E+06	7.115E+06	0.000E+00
	Best	4.407E+01	2.900E+01	1.677E+04	3.756E+04	2.700E+01
Schwefel2.22	*Mean*	2.901E+54	6.359E-06	2.313E+02	2.440E+02	4.030E-17
	STD	1.297E+55	1.194E-05	1.319E+02	1.234E+02	2.314E-15
	Best	3.685E+01	5.941E-08	6.740E+01	1.735E+01	4.251E-23
Griewangk	*Mean*	2.615E+01	3.315E-11	7.951E-01	7.775E-01	0.000E+00
	STD	3.663E+01	1.467E-10	3.758E-01	2.545E-01	0.000E+00
	Best	6.391E-05	0.000E+00	2.932E-01	3.203E-01	0.00E+000

5. Smmary and Conclusion

In this paper, we present the CSOGL algorithm for global numerical optimization with continuous variables. CSOGL is an improved version of CSO. However, CSOGL hasa novel swarm strategy and all-new chase-Swarming scheme, which are significantly different from existing cockroach -inspired algorithms. We have compared the performance of CSOGL with those of CSO, MCSO, RIO, and HRIO over a suite of 11 numerical optimization problems and concluded that CSOGL is more effective in obtaining better quality solutions. In most cases, CSOGL is more stable with the relatively smaller standard deviation.

References

[1] Shihchia H., Mingkai J., Chihsiang L., "*Optimization of the Carpool Service Problem via a Fuzzy-Controlled Genetic Algorithm*", IEEE Transactions on Fuzzy Systems, vol. 23, no. 5, pp. 1698-1712, 2015.

[2] Parashar E. A. K. , Patro B. , Rawat E. A., "*Is differential evolution with penalty function really better than CMODE ?*", Proc. 9th International Conference on Computer Science & Education, Vancouver, BC, pp. 20-23, 2014.

[3] Ibrahim I., Yusof Z. M. , Nawawi S.W., "*A Novel Multi-state Particle Swarm Optimization for Discrete Combinatorial Optimization Problems*", Proc. Fourth International Conference on Computational Intelligence, Modelling and Simulation, Kuantan, pp. 18-23,2012.

[4] Anantathanavit M., Munlin, M. A., "*Radius Particle Swarm Optimization*", Nakorn Pathom, pp. 126-130, 2013.

[5] Haibin D , Guanjun M , Delin L, "*Optimal Formation Reconfiguration Control of Multiple UCAVs Using Improved Particle Swarm Optimization*", Journal of Bionic Engineering, vol. 5, no. 4, pp. 341-347, 2008.

[6] Dorigo M., Maniezzo V., Colorni A., "*The Ant System: Optimization by A Colony of Cooperating Agents*", IEEE Transactions on Systems, Man, and Cybernetics, vol. 26, no. 1, pp. 29-41, 1996.

[7] Chiafeng J. , Chiwei H., Chiahung H. "*Rule-Based Cooperative Continuous Ant Colony Optimization to Improve the Accuracy of Fuzzy System Design*", IEEE Transactions on Fuzzy Systems, vol. 22, no. 4, pp. 723-735, 2013.

[8] Le C., Lixin H., Xiaoqin Z., Yuetang B., Hong Y., "*Adaptive Cockroach Colony Optimization for Rod-like Robot Vavigation*". Journal of Bionic Engineering, vol. 12, no. 2, pp. 324-337, 2015.

[9] Das S., Abraham A., Chakraborty U.K.; Konar A., "*Differential Evolution Using a Neighborhood-Based Mutation Operator*", IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, pp. 526-553, 2009.