Cockroach Swarm Optimization Using A Neighborhood-Based Strategy

Le Cheng[1,2], Yanhong Song[1], and Yuetang Bian[3]
[1]College of Computer and Communication Engineering, Huaian Vocational College of Information Technology, China
[2]College of Computer and Information, Hohai University, China
[3]School of Business, Nanjing Normal University, China

Abstract: *The original Cockroach Swarm Optimization (CSO) algorithm suffers from the problems of slow or premature convergence. 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 designed, in 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. Moreover, this paper also provides a formal convergence proof for the CSOGL algorithm. The comparison results show that the CSOGL algorithm outperform the existing cockroach-inspired algorithms.*

Keywords: *Cockroach swarm optimization, cockroach-inspired algorithm, CSO with global and local neighborhoods, premature convergence.*

Received January 24, 2016; accepted June 13, 2017

1. Introduction

In past two decades, many nature-inspired algorithms have been proposed and applied to solve optimization problems, e.g., Simulated Annealing (SA) [3, 17], Evolutionary Algorithms [4, 14], Genetic Algorithm (GA) [1, 8, 11, 15, 23], Differential Evolution (DE) [20, 22], Particle Swarm Optimization (PSO) [2, 10, 12, 13], Ant Colony Optimization (ACO) [16, 21, 24], etc., Among these algorithms, PSO and ACO belong to biologically- inspired algorithms. This kind of optimization algorithms generally simulates some collective behavior of natural systems, e.g., the ACO algorithm is inspired by the biological behaviors of ant and PSO simulates the collective behavior of bird.

One of recent developments in biologically-inspired algorithms is a series of cockroach-inspired algorithms. Chen first proposed a Cockroach Swarm Optimization (CSO) [25]. 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 [5]. By studying the recent discoveries in the behavior of cockroaches, Havens et al. proposed a new cockroach-inspired algorithm, called Roach Infestation Optimization (RIO) [9]. In essence, RIO has the different searching strategy with CSO. That is, RIO is 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). Literature [9] has proved that HRIO has better performance than that RIO and PSO.

In literature [6], we have proposed a series of cockroach-inspired algorithms for the Robot Path Planning (RPP) and Rod-Like Robot Path Planning (RLRPP) problems. However, practical experience shows that the existing cockroach-inspired algorithms for numerical optimization problem generally suffer from the problem of premature convergence. That is, the population converges to some local optima of a multimodal objective function, loses its diversity. We consider that the probability of stagnation depends on how many different potential solutions are available and also on their capability to enter into the population of the subsequent generations. 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. Section 4 provides a formal convergence proof for the CSOGL algorithm. The experimental results are illustrated in section 5. Section 6 concluded 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 follow:

1. Chase-swarming behavior:

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

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

$$P_i = Opt_j\{X_j \mid |X_i - X_j| \le visual\} \qquad (2)$$

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

$$P_g = Opt\{X_i\} \qquad (3)$$

2. Dispersion behavior

$$X_i = X_i + rand(1, D) \qquad (4)$$

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

3. Ruthless behavior

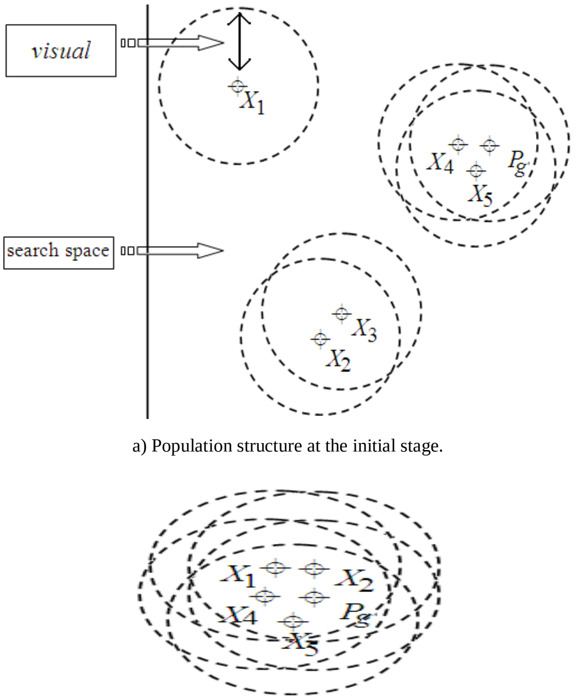
$$X_r = P_g \qquad (5)$$

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 suffers from the problems of slow or premature convergence. Furthermore, we found that the swarm strategy of CSO is unreasonable. In Equation (2), the swarm strategy $|X_i - X_j| \le visual$ 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 Figure 1-a). However, after some iteration, all cockroaches are near around P_g and there exist only one sub-population that is the entire population (See Figure 1-b).



a) Population structure at the initial stage.

b) Population structure after some iteration.

Figure 1. The swarm strategy of CSO.

In Figure 1-a, the cockroach X_1 has no neighborhood in the range that controlled by Equation (2) and X_1 hence does not belong to any sub-population. It Means that many cockroach individuals is 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 Model and Search Strategy of CSOGL

In CSOGL, two kinds of neighborhood models are used, which are similar to the idea of literature [7]. The first one is called the local neighborhood model, where each $X_{i,G+1}$ is computed by the best position $P_{i,G}$ found so far in a small neighborhood of it. On the other hand, the second one takes into account the globally best position $P_{g,G}$ of the entire population at current generation G.

We define a number of neighborhood K. The positions are organized on a ring topology with respect to their indices. Thus, the neighborhood structure can be illustrated as Figure 2. In Figure 2, N is the number of all cockroaches in CSOGL. We assume K=5, then cockroach Xi has five neighborhoods Xi+0, Xi+1, Xi+2, Xi+3 and Xi+4. There exist some overlapping neighborhoods (See Figure 3).

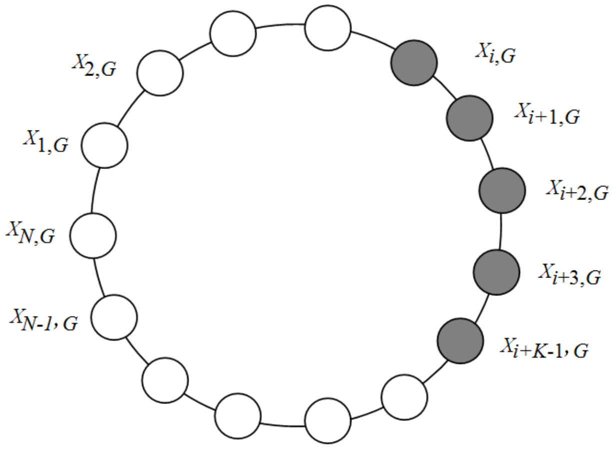


Figure 2. The neighborhood model of CSOGL.

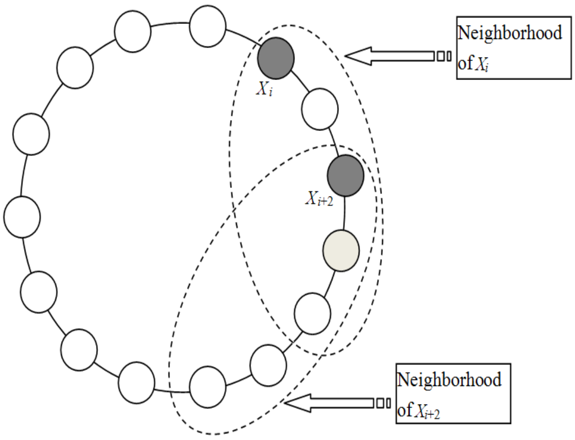


Figure 3. The overlapping neighborhoods of CSOGL.

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

The chase-swarming behavior for local neighborhood is as Equation (6):

$$F_{i,G} = X_{i,G} + (P_{i,G} - X_{i,G}) + (X_{r1} - X_{r2}) \tag{6}$$

here, $L_{i,G}$ denotes the new location found by the i-th cockroach. Similarly, the chase-swarming behavior for global neighborhood is as Equation (7):

$$F_{i,G} = X_{i,G} + (P_{g,G} - X_{i,G}) + (X_{r3} - X_{r4}) \tag{7}$$

In Equations (6) and (7), the indices r_1, r_2, r_3, r_4 are mutually exclusive integers randomly generated within the range [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 (8)

$$F_{i,G} = \begin{cases} X_{i,G} + (P_{i,G} - X_{i,G}) + (X_{r1} - X_{r2}), rand(0,1) < 0.5 \\ X_{i,G} + (P_{g,G} - X_{i,G}) + (X_{r3} - X_{r4}), \text{otherwise} \end{cases} \tag{8}$$

where, *rand* (0, 1) is a uniformly distributed random number lying between 0 and 1. 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 don'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 (9)).

$$X_{i,G+1} = \begin{cases} F_{i,G}, \text{ if } (f(L_{i,G}) < f(X_{i,G})), \text{ minimization problems} \\ X_{i,G}, \text{ otherwise} \end{cases} \tag{9}$$

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

Algorithm 1: CSOGL

```
INPUT: Fitness function: $f(X), X \in RD$
Set parameters and generate an initial population of cockroach
Choose the pg from whole population;
Choose the pi for each cockraoch;
FOR t = 1 to Gmax
    FOR i = 1 to N
        chase-swarming operations (E.q(8));
        greedy selection scheme(See E.q(9))
        IF f(Xi,) < f(Pi) THEN
            Pi = Xi;
        END IF
        IF f(Xi) < f(Pg) THEN
            Pg = Xi;
        END IF
    END FOR
END FOR
Check termination condition
```

4. Convergence Analysis on CSOGL

In order to get a better understanding of the optimal strategy of an algorithm, it is necessary to conduct a theoretical analysis. In CSOGL, the greedy selection scheme can guarantee that the off springs are more optimal than the parents. Therefore, the stochastic functional theory can is used to analysis on CSOGL. For the D-dimensional problem, each individual in CSOGL is a D-dimensional vector $X_{i=} x_{i1},x_{i2},...,x_{iD})$ (i=I,...,N) in essence. The minimization problem can be described as

$$\min\{f(X) \mid \forall X \in S, 0 < f(X) < +\infty\} \tag{10}$$

Where, S is the solution space and $S = \prod_{j=1}^{D}[L_j, U_j]$. Each X_i subjects to $L< X_i <U$. Let $U = \max\{\overline{U}_j | \overline{U}_j = \lceil U_j \rceil, j = 1,2\ldots D\}$ and $L =$ $\max\{\overline{L}_j | \overline{L}_j = \lfloor L_j \rfloor, j = 1,2\ldots D\}$, then

$$|S| \le \prod_{j=1}^{D}[(U_j - L_j)10^k + 1] \le [10^k(U - L) + 1]^D \tag{11}$$

For each iteration, the Equations (8) and (9) are all executed. Equation (8) is composed of Equations (6) and (7). Note that Equation (6) and Equation (7) are similar in components. For conveniences, Equations (6) and (9) are only discussed. According to stochastic functional theory, once iteration of CSOGL is regarded as the stochastic mapping that composed of Equations (6) and (9). The mapping on Equation (6) can is defined as Ψ_1.

$$\begin{aligned}&\mu\{\omega \mid \psi_1(\omega, X) = < X, F >\}\\ &= \mu\{F = X_i + (P_i - X_i) + (X_{r1} - X_{r2})\}\end{aligned} \qquad (12)$$

Where, (Ω, A, μ) is the space of probability measures and Ω is a abstract set. Ω is the basic events and A denotes σ- algebra. μ is the probability measures of A. On the other hand, the mapping on Equation (9) can is defined as Ψ_2:

$$\begin{aligned}&\psi_2(< X, F >) = Z \wedge Z \in \{X, F\} \wedge f(F)\\ &= \min\{f(X), f(F)\}\end{aligned} \qquad (13)$$

Once iteration of CSOGL can be regard as the mapping $\psi = (\psi_2 \circ \psi_1)$. For each iteration, P_g recorded the best cockroach individual. By the mapping Ψ, the fitness of P_g from 1 to $G_{\max}$ are non-monotonic decreasing. That is, $\{f(P_{g,t})\}_{1 \le t \le G\max}$ is a non-monotonic decreasing sequence. P_g is introduced in mapping Ψ. Thus, the mapping Ψ can be defined as $P_{g,t+1} = \psi(\omega, P_{g,t}) = \psi_2(\psi_1(\omega, P_{g,t}))$

- *Lemma 1: Suppose λ: $S \times S \rightarrow R$ is the distance defined on the space S and* $\lambda(X_i, X_j) = | f(X_i) - f(X_j) |, \forall X_i, X_j \in S$ *, then (S, λ) is the Perfect Metric Space [18].*
- *Theorem 1: Mapping Ψ: $\Omega \times S \rightarrow S$ is a randomly contractive operator.*
- *PROOF: According to Equations (6) and* (9), the off springs are more optimal than parents. For the mapping Ψ, there exist a non-negative variable $0 \le K(\omega) < 1$, a. s.

$$\begin{aligned}&\lambda(\psi(\omega, P_{g,t-1}), \psi(\omega, P_{g,t}))\\ &= \lambda(P_{g,t}, P_{g,t+1})\\ &= | f(P_{g,t}) - f(P_{g,t+1}) | \le K(\omega) \, | f(P_{g,t-1}) - f(P_{g,t}) |\\ &= K(\omega)\lambda(P_{g,t-1}, P_{g,t})\end{aligned}$$

here

$$\begin{aligned}&\Omega_0 = \{\omega \mid \lambda(\psi(\omega, P_{g,t-1}), \psi(\omega, P_{g,t}))\\ &\le K(\omega)\lambda(P_{g,t-1}, P_{g,t})\}\\ &\mu(\Omega_0) = 1\end{aligned}$$

The result of the above proof shows that mapping Ψ is the randomly contractive operator.

- *Lemma* 2: Ψ: $\Omega \times S \rightarrow S$ is a randomly operator. For each ω ($\omega \in \Omega$) , If $\Psi(\omega)$ is the contractive operator, then $\Psi(\omega)$ has the unique fixed point $g(\omega)$, that is, $\Psi(\omega, g(\omega)) = g(\omega)$ [18, 19].
- *Theorem* 1: If Ψ is the randomly contractive operator of CSOGL, then Ψ has the unique fixed point, that is, CSOGL has the property of gradual-approach convergence.

5. Simulation Studies

In order to evaluate the performance of CSOGL, the paper selected eleven classical benchmark test functions. These functions include various types of complex problems, e.g., the single-mode and multi-mode, regular and irregular, separation and non-separated, etc., all the test functions are shown in Table 1.

Table 1. Benchmark test functions.

Fn	Description	Search range	Optimum value
Sphere	$f_1(X) = \sum_{i=1}^{D} x_i^2$	$-100 \le x_i \le 100$	$f_1(X^*) = 0$
Rastrigin	$f_2(X) = 10 \cdot D + \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i))$	$-5.12 \le x_i \le 5.12$	$f_2(X^*) = 0$
Rosenbrock	$f_3(X) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$-30 \le x_i \le 30$	$f_3(X^*) = 0$
Ackley	$f_4(X) = -20\exp\left(0.2\sqrt{1/D \sum_{i=1}^{D} x_i^2}\right) - \exp\left(1/D \sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	$-100 \le x_i \le 100$	$f_4(X^*) = 0$
Schwefel1.2	$f_5(X) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$-100 \le x_i \le 100$	$f_5(X^*) = 0$
Schwefel2.22	$f_6(X) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} x_i$	$-10 \le x_i \le 10$	$f_6(X^*) = 0$
Griewangk	$f_7(X) = 1 + \sum_{i=1}^{D} (\frac{x_i^2}{4000}) - \prod_{i=1}^{D} (\cos(\frac{x_i}{\sqrt{i}}))$	$-600 \le x_i \le 600$	$f_7(X^*) = 0$
Sumsquares	$f_8(X) = \sum_{i=1}^{D} i \cdot x_i^2$	$-10 \le x_i \le 10$	$f_8(X^*) = 0$
Sinusoidal	$f_9(X) = -\left[A\prod_{i=1}^{D} \sin(x_i - z) + \prod_{i=1}^{D} \sin(B(x_i - z))\right]; A = 2.5, B = 5, z = 30$	$0 \le x_i \le 180$	$f_9(X^*) = -3.5$
Zakharov	$f_{10}(X) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$	$-5 \le x_i \le 10$	$f_{10}(X^*) = 0$
Step	$f_{11}(X) = \sum_{i=1}^{D-1} (\lfloor x_i + 0.5 \rfloor)^2$	$-100 \le x_i \le 100$	$f_{11}(X^*) = 0$

We perform large number of simulation experiments, and compare the CSOGL with original CSO, MCSO, RIO and HRIO. All the benchmark functions are test 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 literature [5, 9, 25]. 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 standarddeviation 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**
Sphere	*Mean*	1.812E+02	1.520E-12	2.217E-05	1.668E-04	1.427E-30
	STD	8.105E+02	6.722E-12	2.453E-05	2.402E-04	1.332E-30
	Best	4.920E-07	2.998E-24	5.763E-09	5.564E-08	4.643E-44
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
Ackley	*Mean*	1.922E+01	5.159E-06	2.000E+01	2.001E+01	2.146E-14
	STD	5.826E+00	1.915E-05	3.046E-03	1.567E-02	6.822E-16
	Best	2.013E+01	6.462E-09	2.000E+01	2.000E+01	3.294E-16
Schwefel1.2	*Mean*	3.499E-04	4.475E-13	2.450E-05	2.271E-04	5.309E-27
	STD	3.373E-04	1.975E-12	2.796E-05	2.364E-04	2.417E-27
	Best	4.155E-08	5.631E-23	1.136E-08	5.823E-07	4.506E-50
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	0.000E+00
Sum Squares	*Mean*	9.050E+05	4.245E-11	1.982E+03	4.677E+03	2.124E-27
	STD	1.025E+06	1.293E-10	2.837E+03	6.710E+03	5.441E-24
	Best	1.873E+02	1.500E-16	1.646E+01	2.052E+02	2.464E-46
Sinusoidal	*Mean*	-2.449E+00	-3.103E+00	-4.259E-01	-3.790E-01	-2.034E+00
	STD	1.020E+00	5.047E-05	2.663E-01	1.979E-01	2.115E-14
	Best	-3.309E+00	-3.103E+00	-1.192E+00	-8.311E-01	-4.142E+00
Zakharov	*Mean*	6.366E+18	2.388E-09	1.017E+04	1.022E+04	3.065E-24
	STD	2.273E+19	8.853E-09	3.864E+03	5.101E+03	0.446E-24
	Best	1.358E+09	2.095E-15	2.663E+03	2.315E+03	5.239E-57
Step	*Mean*	2.000E+04	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	STD	8.482E+04	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00

6. Summary and Conclusions

In this paper, we present the CSOGL algorithm for global numerical optimization with continuous variables. CSOGL is an improved version of CSO. However, CSOGL has a novel swarm strategy and all-new chase-swarming scheme, which are significantly different from existing cockroach - inspired algorithms. This paper provides a formal convergence proof for the CSOGL algorithm. 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, CCO is more stable with the relatively smaller standard deviation.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 71301078), the Natural Science Foundation of Education Bureau of Jiangsu Province (Grant No. 16KJB520049) and Innovation Foundation of Huaian College of Information Technology (Grant No. hxyc2015001).

References

[1] Ali A. and Hassanien A., “Minimizing Molecular Potential Energy Function Using Genetic Nelder-Mead Algorithm,” *in Proceedings of the 8th International Conference on Computer Engineering and Systems*, Cairo, pp. 177-183, 2013.

[2] Anantathanavit M. and Munlin M., “Radius Particle Swarm Optimization,” *in Proceedings of the International Computer Science and Engineering Conference*, Nakorn Pathom, pp. 126-130, 2013.

[3] Biswas S. and Acharyya S., “Gene Expression Profiling by Estimating Parameters of Gene Regulatory Network Using Simulated Annealing: A Comparative Study,” *in Proceedings of the IEEE International Advance Computing Conference*, Gurgaon, pp. 56-61, 2014.

[4] Chakradeo S., Hendre A., and Deshpande S., “Generalized Theory for Hybridization of Evolutionary Algorithms,” *in Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research*, Coimbatore, pp. 1-5, 2014.

[5] Chen Z., “A Modified Cockroach Swarm Optimization,” *Advances in Engineering Software*, vol. 32, no. 1, pp. 49-60, 2011.

[6] Cheng L., Han L., Zheng X., Bian Y., and Yan H., “Adaptive Cockroach Colony Optimization for Rod-like Robot Vavigation,” *Journal of Bionic Engineering*, vol. 12, no. 2, pp. 324-337, 2015.

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

[8] Ding Z., Liu J., Sun Y., Jiang C., and Zhou M., "A Transaction and QoS-Aware Service Selection Approach Based on Genetic Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1035-1046, 2015.

[9] Havens T., Spain C., Salmon N., and Keller J., "Roach Infestation Optimization," *in Proceedings of the IEEE Swarm Intelligence Symposium*, St. Louis, pp. 1-7, 2008.

[10] Huang H., "A Taguchi-Based Heterogeneous Parallel Metaheuristic ACO-PSO and its FPGA Realization to Optimal Polar-Space Locomotion Control of Four-Wheeled Redundant Mobile Robots," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 915-922, 2015.

[11] Huang S., Jiau M., and Lin C., "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.

[12] Ibrahim I., Yusof Z., Nawawi S., Rahim M., Khalil K., Ahmad H., and Ibrahim Z., "A Novel Multi-State Particle Swarm Optimization for Discrete Combinatorial Optimization Problems," *in Proceedings of the 4th International Conference on Computational Intelligence, Modelling and Simulation*, Kuantan, pp. 18-23, 2012.

[13] Jamian J., Mustafa M., Mokhlis H., and Abdullah M., "Comparative Study on Distributed Generator Sizing Using Three Types of Particle Swarm Optimization," *in Proceedings of the 3th International Conference on Intelligent Systems, Modelling and Simulation*, Kota Kinabalu, pp. 131-136, 2012.

[14] Jiang S., Zhang J., Ong Y., Zhang A., and Tan P., "A Simple and Fast Hypervolume Indicator-Based Multi Objective Evolutionary Algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2202- 2213, 2014.

[15] Jin Z. and Fan H., "A New Immune Genetic Algorithm for 0-1 Knapsack Problem," *in Proceedings of the 6th International Symposium on Computational Intelligence and Design*, Hangzhou, pp. 31-33, 2013.

[16] Juang C., Hung C., and Hsu C., "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.

[17] Kerrache S. and Benhidour H., "Topology-Aware Simulated Annealing," *in Proceedings of the 2nd International Conference on Artificial Intelligence*, Madrid, pp. 19-24, 2014.

[18] Li M., Kou J., and Lin D., *the Basic Theory and Application of Genetic Algorithm*, Science Press, 2003.

[19] Lu T., "*Random Functional Analysis and Applications*, Qingdao Ocean University Press, 1990.

[20] Parashar E., Patro B., and Rawat E., "Is Differential Evolution with Penalty Function Really Better than CMODE?," *in Proceedings of the 9th International Conference on Computer Science and Education*, Vancouver, pp. 20-23, 2014.

[21] Sales D., Dias M., and Osorio F., "Grid Ant Colony Optimization Applied to a Multi-robotic Garbage Collection System," *in Proceedings of the Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol*, Sao Carlos, pp. 187-192, 2014.

[22] Skanderova L., Fabian T., and Zelinka I., "Differential Evolution Enhanced by the Closeness Centrality: Initial Study," *in Proceedings of the International Conference on Intelligent Networking and Collaborative Systems*, Taipei, pp. 346-353, 2015.

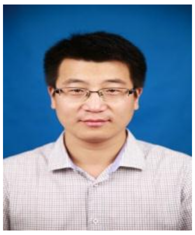
[23] Todmal S. and Patil S., "Enhancing the Optimal Robust Watermarking Algorithm to High Payload," *The International Arab Journal of Information Technology*, vol. 13, no. 1, pp. 108-117, 2016.

[24] Xiao G., Guo B., Jing Z., and Chaopeng Yu., "An Improved Ant Colony Optimization Approach for Multi-Aircraft Horizontal Escape Maneuvers," *in Proceedings of the IEEE/AIAA 31st Digital Avionics Systems Conference*, Williamsburg, pp. 1 -8, 2012.

[25] Zhaohui C. and Haiyan T., "Cockroach Swarm Optimization," *in Proceedings of the 2nd International Conference on Computer Engineering and Technology*, Chengdu, pp. 652-655, 2010.

Le Cheng received his Ph.D. degree in College of Computer and Information in 2015 from Hohai University, Nanjing, P. R. China. He is currently a Associate Professor with the Department of Computer Science and Engineering, Huaian Vocational College of Information Technology, Huaian, China. His research interests include evolutionary algorithms, motion planning, planning algorithms and machine learning.

Yanhong Song received the B.S. degree in College of Computer and Information in 2009 from Hohai University, Nanjing, P. R. China. She is currently a Assistant Professor with the Department of Computer Science and Engineering, Huaian Vocational College of Information Technology, Huaian, China. Her research interests include evolutionary algorithms and machine learning.



Yuetang Bian received his Ph.D. degree from Southeast University, Nanjing, China. He is currently a Associate Professor with the Department of Management Science, School of Business, Nanjing Normal University, Nanjing, China. His current research interests include network theory and methods, Optimization Theory and Social behavior in Networks. He has hosted and participated in a number of project sponsored by the Natural Science Foundation of China. Dr. Bian has published more than ten papers in international journals