ICIC Express Letters
Volume **12**, Number **2**, February **2018**

ICIC International ©2018 ISSN 1881-803X
pp. 125–134

ARTIFICIAL BEE COLONY WITH CELLULAR AUTOMATA

Na Tian[1], Jing Sun[1], Meng Wang[1] and Likun Qiu[2]

[1]Research Center of Educational Informatization
Jiangnan University
No. 1800, Lihu Avenue, Wuxi 214122, P. R. China
tianna@jiangnan.edu.cn

[2]Shandong Key Lab of Language Resource Development and Application
Ludong University
No. 186, Hongqi Middle Road, Zhifu District, Yantai 264025, P. R. China

Received August 2017; accepted November 2017

Abstract. *Artificial bee colony (ABC) has shown competitive performance for handling complex optimization problems. However, it inevitably suffers from slow convergence and loses balance between exploitation and exploration. In this paper, a cellular structured neighborhood and a redefined probability calculation method are proposed for ABC (ABCCA). Individuals interact with specific neighbors in cellular automata (CA) model while maintaining the population diversity. A set of benchmark functions is used to test the algorithms and the results demonstrate that the proposed strategies help ABC improve in terms of convergence rate and global search ability while compared with other variants of ABC.*

Keywords: Artificial bee colony, Cellular structure, Probability calculation, Convergence

1. **Introduction.** Evolutionary optimization methods (EAs) [1,2], as a significant branch of derivative-free methods, have been proven to be efficient tools for solving complex optimization problems. The famous algorithms include genetic algorithm (GA) [3] evolution strategy (ES) [4,5], evolutionary programming (EP) [6], differential evolution (DE) [7,8], ant colony optimization (ACO) [9], particle swarm optimization (PSO) [10], artificial immune algorithm (AIA) [11], etc.

Artificial bee colony (ABC) was proposed by Karaboga [12] in 2005, inspired from the foraging behavior of honeybees. In ABC, there are three kinds of bees to perform different division of work. Employed bees take the responsibility for searching food sources in a given multidimensional continuous search space and propagating food information to onlooker bees. After receiving the information, onlooker bees make an exploitative search around the neighborhood of food. The scouts are designed to help jump out of local minima. Since it is easy to implement with fewer control parameters and simple structure, ABC has validated comparable performance to other EAs [13-15].

However, since the search equation performs well in exploration but poor in exploitation, ABC inevitably faces slow convergence. To address this, researchers have developed plenty of approaches from various aspects. In terms of modified search equations, *global best solution* inspired from PSO and DE is highlighted to improve the exploitation ability, such as GABC [16] and MABC [17]. Under the guidance of *global best solution*, individuals could be pulled towards the potential regions; thus, the convergence rate can be improved. Besides, in [18], Gaussian distribution was used for parameter turning to get better stability and exploitative behavior. In the case of hybrid ABC, some evolutionary operators involved in other EAs have been incorporated in ABC. For example, chaotic-based search widely used for initialization was proposed to enhance global convergence and keep population diversity [17,19,20]. Finally, new selection strategy for neighborhood was designed for updating equation to enhance the local search ability [20,21]. To avoid random search direction, previous successful experience of foraging was memorized for bees to provide favorable search guidance in [22-24].

However, to our best knowledge, little attention has been paid on the neighborhood structure of ABC in the past literature. Therefore, we try to use a cellular topology motivated by cellular automata (CA) [25] to decentralize the population, in which the population can be arranged in a 2-D lattice structure and individuals interact with each other in a particular neighborhood. This proposed algorithm is termed as ABCCA.

The rest of paper is structured as follows. Section 2 describes the ABC algorithm in detail. Section 3 introduces the motivation of this paper. Experimental studies are reported and discussed in Section 4. Finally, the conclusion is drawn in Section 5.

2. **Overview of ABC.**

In an ABC system, there are three groups of bees: employed, onlooker and scouts. The number of employed and onlooker bees account for half of the colony. The position of a bee represents a candidate solution of the optimization problem, and its corresponding fitness denotes the amount of nectar of a food source. It should be noted that one food source is assigned to only one employed bee. Assuming that the initial population, consisting of SN D-dimensional vectors $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$, is randomly generated by Equation (1),

$$x_{i,j} = XMIN_j + rand(0,1)\,(XMAX_j - XMIN_j) \tag{1}$$

where $i \in \{1, 2, \ldots, SN\}$, $j \in \{1, 2, \ldots, D\}$, and $XMIN_j$ and $XMAX_j$ are the lower and upper bounds of the jth dimension. The employed bees randomly choose a dimension to generate a new candidate solution V_i by Equation (2),

$$v_{i,j} = x_{i,j} + \phi_{i,j}\,(x_{i,j} - x_{k,j}) \tag{2}$$

where $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, SN\} \cap k \neq i$ are randomly selected indexes; $\phi_{i,j}$ is a uniformly random number in $[-1, 1]$. Then, a greedy selection based on the amount of nectar is adopted to select the better one between the candidate and the old solutions. After that, employed bees share food information with onlooker bees through dancing.

Onlookers select potential food sources to exploit according to the probabilities calculated by fitness. The probability p_i and fitness fit_i of solution X_i is calculated as follows:

$$p_i = fit_i \Bigg/ \sum_{i=1}^{SN} fit_i \tag{3}$$

$$fit_i = \begin{cases} 1/(1+f_i) & f_i > 0 \\ abs(f_i) & f_i < 0 \end{cases} \tag{4}$$

where f_i denotes the objective function value. In the scout phase, a solution would be abandoned if it had not been improved after consecutive $Limit$ iterations and the associated employed bee would become a scout to produce a new random solution by Equation (2).

3. **ABC with Cellular Automata Model.**

3.1. **Motivation.**

Two contradictory aspects that intensively influence the performance of EAs are exploration and exploitation. Exploration is considered as the ability to search the unknown region to find potential solutions, while exploitation refers to the ability to exploit better solutions in the neighborhood using information of previous good individuals. Onlookers selected by probabilities in Equation (3) take responsibility for exploiting better individuals. Nevertheless, the fitness values can be approximately equal to 1 when the corresponding objective function values are positive but too small, e.g., 1e-30. This has a directly negative effect on the calculation of probabilities, on which onlookers depend to select potential individuals to exploit. Furthermore, scouts are supposed to increase diversity when the population is getting stuck in a local optimum.

In order to validate the above discussion about the performance of onlooker bees and scouts, an experiment is conducted to record the mean number of scouts, the mean percentage of independent individuals regarded as onlookers and the mean convergence curves along with the search process. The experimental results on function Ackley with $D = 30$ in $[-32, 32]$ observed at three different values of *Limit*, e.g., $0.2 * SN * D$, $0.6 * SN * D$ and $1.0 * SN * D$ [16] are presented in Figures 1-3, from which inspiring conclusions can be drawn as follows:

1) Scouts appear transitorily after the stagnation of population, and the time occurring becomes later with *Limit* increasing;
2) The proportion of onlooker bees vibrates when scouts appear; otherwise, it keeps more than 60%;
3) The convergence curves keep unchangeable no matter what values *Limit* are.

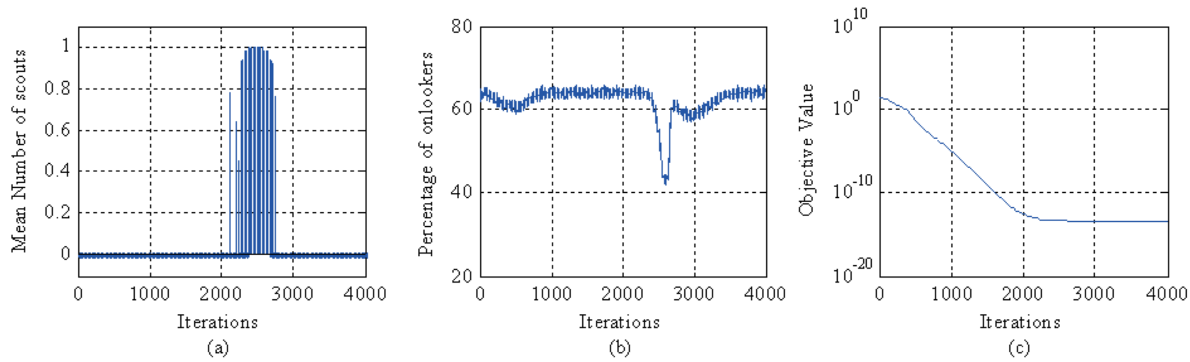

FIGURE 1. Results on Ackley with $Limit = 0.2*SN*D$: (a) mean number of scouts; (b) the proportion of selected onlookers; (c) convergence curve over iterations

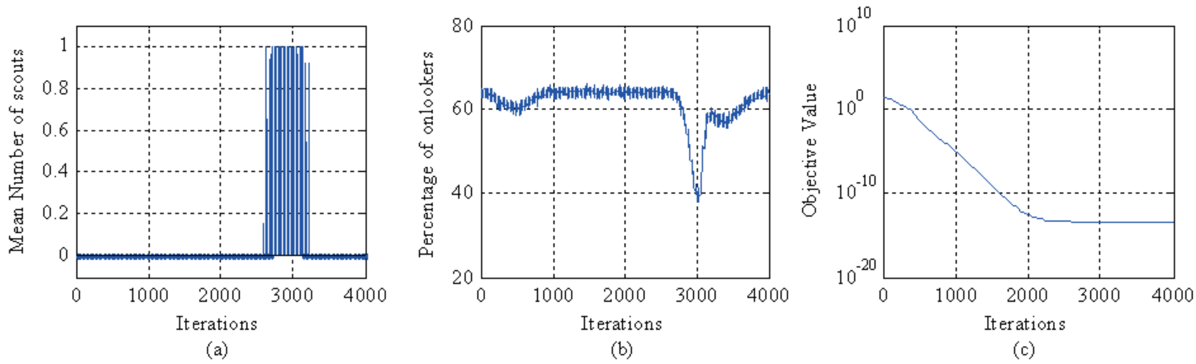

FIGURE 2. Results on Ackley with $Limit = 0.6*SN*D$: (a) mean number of scouts; (b) the proportion of selected onlookers; (c) convergence curve

3.2. **Hybridization of CA model and ABC.** The concept of CA model was first proposed by Von Neumann and Ulam, and the primary classification of CA was outlined by Wolfram [25]. As a discrete dynamical system, CA can stimulate micro-behavior by micro-dynamics using the interaction of individuals (cells) connected in particular neighborhood structures. Therefore, CA model is adopted to decentralize the population of ABC.

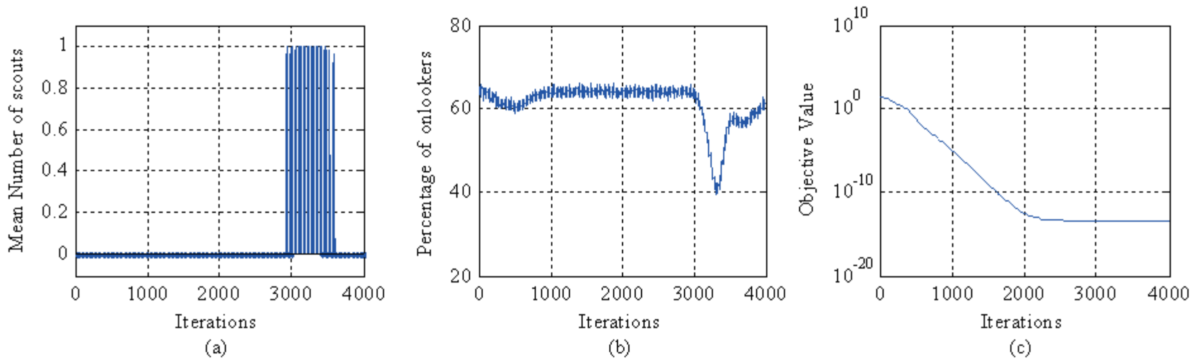

FIGURE 3. Results on Ackley with $Limit = 1.0 * SN * D$: (a) mean number of scouts; (b) proportion of selected onlookers; (c) convergence curve

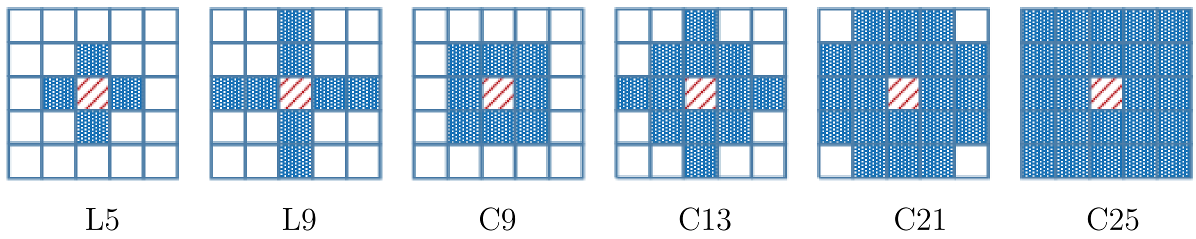

FIGURE 4. Six famous neighborhood in CA

TABLE 1. Radius and ratio of six neighborhoods in Figure 4

Shape / Neighborhood	5 × 20			10 × 10		
	rad_{shape}	$rad_{neighborhood}$	$ratio$	rad_{shape}	$rad_{neighborhood}$	$ratio$
L5	5.2783	0.8944	**0.1694**	4.062	0.8944	0.2202
L9	5.2783	1.4907	**0.2824**	4.062	1.4907	0.3670
C9	5.2783	1.1547	**0.2188**	4.062	1.1547	0.2843
C13	5.2783	1.4676	**0.2780**	4.062	2.0000	0.3613
C21	5.2783	1.7995	**0.3409**	4.062	1.4676	0.4430
C25	5.2783	2.0000	**0.3789**	4.062	1.7995	0.4924

3.2.1. *Cellular automata (CA).* There are four basic components in CA: cell space, neighborhood, cell state, and transition rule. Cell space presents the connecting structure of cells, and a checkboard-like lattice structure in two-dimension is employed in this paper. Neighborhood is defined as the cells surrounding a given cell, and six commonly used neighborhoods are shown in Figure 4, where L5 and L9 are linear neighborhood and C9, C13, C21, and C25 are compact neighborhood.

For a cell, the distribution of its neighbors is an important characteristic of a neighboring structure. For instance, the number of neighbors in L9 and C9 is the same, but the distribution is totally different. Alba and Troya [26] defined the 'radius' for both 2-D grid and the neighborhood by the dispersion of n^* points in a circle centered in $(\bar{x}, \bar{y})$ based on Equation (5), and then the grid-neighborhood relationship can be quantified by the relative ratio between their radius according to Equation (6). For comparison, the radius and ratio for different neighborhoods (i.e., $100 = 5 * 20$ and $100 = 10 * 10$) with 100 individuals are presented in Table 1. It is obviously noted that thinner grid shape gets smaller ratio because of the larger dispersion.

$$\begin{gathered} rad = \sqrt{\left(\sum_{i=1}^{n^*} (x_i - \bar{x})^2 + \sum_{i=1}^{n^*} (y_i - \bar{y})^2\right) \Big/ n^*}, \\ \bar{x} = \sum_{i=1}^{n^*} x_i \Big/ n^*, \ \bar{y} = \sum_{i=1}^{n^*} y_i \Big/ n^* \end{gathered} \tag{5}$$

$$ratio = rad_{neighborhood} / rad_{shape} \tag{6}$$

3.2.2. *ABC with cellular structured topology.* From the above description, it can be seen that CA is easy to implement without any strict mathematical reasoning. To enhance the cooperation of population, the concept of CA is applied in ABC to exploring the neighboring structure and diffusing mechanism. Some elaborated definitions for integrating ABC and CA are given as follows.

a) Cell space

An individual is thought of a cell, and the number of cells in CA is equal to the number of individuals in ABC. Then, each individual is randomly allocated to a unique cell of the lattice structure without duplication in the initial phase, and most importantly the position of an individual in the lattice structure is fixed during the search process.

b) Neighborhood

Taking C9 as an example, an explicit representation of the neighboring structure is depicted in Figure 5 with one hundred individuals. Individuals locally interact with their neighbors, and such limited information transmission has advantages of improving local search ability. Meanwhile, since the population has been decentralized, the information delivered by individuals could be slowly diffused to others through overlapping neighbors.

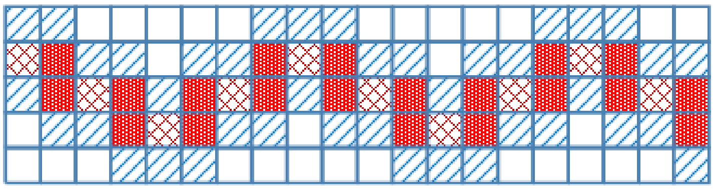

FIGURE 5. Neighboring structure in the Moore neighborhood C9

c) Cell state

According to the definition of ABC, the cell state can be briefly defined as employed, onlooker or scout. Naturally, switch of cell states is an important issue for the search process. It can be noted that the state transition between employed and onlooker is dominated by the parameter p in Equation (3), and the state transition between employed and scouts is determined by the parameter *Limit*.

d) Transition rule

The search equation updating the positions of individuals is regarded as transition rule. In order to improve the local exploitation in the neighborhood, a novel search equation is proposed, which gets inspiration from Gaussian distribution. The generalized formulation is defined as follows,

$$v_{i,j} = N\left(s_{i,j}, \alpha * \left|s_{i,j} - x_{k,j}\right|\right) \tag{7}$$

$$\alpha = 0.8 * \left(1 - \sqrt{\frac{FEs}{\max FEs}}\right) + 0.2 \tag{8}$$

where $j \in \{1, 2, \ldots, D\}$ is a randomly selected dimension; FEs denotes the index of current function evaluation; $\max FEs$ means the maximal number of function evaluations; $k \in \{1, 2, \ldots, SN\} \cap k \neq i$ and X_k is a randomly chosen solution from the whole population instead of the neighborhood; S_i is regarded as the local attractor of individual X_i, and $s_{i,j}$ is the jth element of vector S_i; α is a scaling factor to control the magnitude of variance operator, which iteratively decreases from 1.0 to 0.2 by Equation (8).

It should be emphasized that S_i is an abstract term with different definitions for employed and onlooker bees as shown in Equations (9) and (10), where $x_{lbest,j}$ and $x_{gbest,j}$ are the jth elements of the local best position X_{lbest} and the global best position X_{gbest}, respectively. λ, which controls the degree $x_{i,j}$ depending on $x_{lbest,j}$, decreases along with the iterations; $N(\cdot)$ is used as a noise for λ obeying a normal distribution whose mean and standard deviation both are 0.5.

$$s_{i,j} = \lambda * x_{lbest,j} + (1-\lambda) * x_{k,j} \tag{9}$$

$$s_{i,j} = \lambda * x_{lbest,j} + (1-\lambda) * x_{gbest,j} \tag{10}$$

$$\lambda = N(0.5, 0.5) * \left(1 - \left(\frac{FEs}{\max FEs}\right)^2\right) \tag{11}$$

4. Experiments and Analysis.

4.1. Test suites.

In order to comprehensively investigate the effectiveness of ABCCA, experiments are conducted on 15 benchmark functions with different properties (unimodal, multimodal, separated, shifted, rotated, and noisy) from [27].

4.2. Experimental settings.

In this paper, the population size of all the compared algorithms is set to be 100 ($SN = 50$). The dimension D is set as 30, and the parameter *Limit* of ABCCA is set as 100. Each algorithm runs 50 times independently for each function.

Three groups of experiments are designed for analysis and comparison. In order to find out the effect of the proposed strategies, the first group of experiments is for comparison among ABC and five new variants of ABC, including ABC with C25 cellular structured population (cABC), ABC with adaptive parameter Pr (pABC), ABC with Gaussian-based search equation (gABC), ABC with both Pr and C25 neighborhood structure (pcABC) as well as the proposed ABCCA. The next two groups of experiments are carried out to investigate the influence of different grid shapes and neighborhoods appointed on ABC. The maximal number of function evaluation on F1-F15 is set as 150000 [27].

4.3. Experimental results on benchmark functions.

4.3.1. *Performance comparison among different variants of ABC.*

For algorithms cABC, pcABC, and ABCCA, which are embedded with the cellular structured population, the 2-D grid shape is set as 6×8, and the neighborhood structure is C25. The results are listed in Table 2 in terms of mean best values (Mean) and standard deviations (Std) of the best solutions obtained on the fifteen functions. The superior algorithm for each function is highlighted in boldface.

From Table 2, we can see that ABCCA achieves the best performance on 11 functions with the exception of F6, F8, F9 and F10 where gABC performs best on the first three functions and pABC does well on the last one.

4.3.2. *Performance comparison of ABCCA with four grid shapes.*

In this group of experiments, the performance of ABCCA with different grid shapes and fixed number of neighbors is investigated. With a constant or approximated size of population 50, four 2-D grid shapes used here are: (a) $4 \times 12 \approx 48$; (b) $5 \times 10 = 50$; (c) $6 \times 8 \approx 48$; (d) $7 \times 7 \approx 49$, and the corresponding ABCCA with these grid shapes are denoted as ABCCA-G1, ABCCA-G2, ABCCA-G3, ABCCA-G4. For clarity, the ratios for different grid shapes are listed in Table 3. C25 is employed for this group of experiments.

The results are given in Table 4 in terms of mean best values and standard deviations out of 50 runs. From Table 4 it can be seen that all the four algorithms can reach the global optima on functions F7, F11, F12, and F13, and ABCCA-G3 performs the best on most unimodal functions, involving F1, F2, F3 and F5. Moreover, it seems that the relationship between the ratios and the results is not evident.

Table 2. Mean and standard deviation of six variants of ABC on 15 functions

Algorithms		ABC	pABC	cABC	pcABC	gABC	ABCCA
F1	Mean	8.81e-16	6.51e-16	2.06e-17	2.99e-21	1.16e-41	**3.88e-75**
	Std	1.59e-16	9.63e-17	1.79e-17	5.08e-21	6.91e-41	**5.69e-75**
F2	Mean	9.89e-09	2.41e-15	2.74e-09	5.28e-17	2.59e-37	**1.26e-71**
	Std	9.01e-09	1.77e-15	3.48e-09	6.91e-17	1.79e-36	**1.83e-71**
F3	Mean	5.71e-16	5.84e-16	3.90e-19	2.04e-22	9.38e-43	**5.18e-76**
	Std	8.79e-17	8.61e-17	3.04e-19	2.62e-22	3.56e-42	**8.82e-76**
F4	Mean	3.41e-08	1.77e-06	2.37e+00	1.13e-07	−3.02e-12	**−3.64e-12**
	Std	1.21e-07	1.04e-05	1.67e+01	7.41e-07	8.70e-13	**0.00e+00**
F5	Mean	2.13e-10	5.54e-11	1.20e-10	8.01e-12	2.21e-22	**1.27e-39**
	Std	6.69e-11	1.73e-11	3.51e-11	5.82e-12	8.00e-22	**9.49e-40**
F6	Mean	1.28e+01	1.07e+01	1.17e+01	1.07e+01	**1.25e+00**	5.46e+00
	Std	2.87e+00	2.33e+00	2.98e+00	2.22e+00	**2.21e-01**	9.00e-01
F7	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	**0**	**0**	**0**	**0**
F8	Mean	1.10e-01	1.08e-01	1.12e-01	1.04e-01	**1.32e-02**	1.60e-02
	Std	2.40e-02	2.11e-02	2.07e-02	2.21e-02	**3.53e-03**	4.35e-03
F9	Mean	3.66e-01	3.19e-01	3.59e-01	3.22e-01	**1.97e-01**	2.28e-01
	Std	**2.16e-02**	3.26e-02	2.62e-02	3.13e-02	3.81e-02	2.63e-02
F10	Mean	5.80e-02	**4.28e-02**	7.05e-02	4.39e-02	6.65e-01	1.65e-01
	Std	4.71e-02	**3.10e-02**	7.15e-02	3.41e-02	1.24e+00	2.52e-01
F11	Mean	1.44e-14	1.42e-14	1.57e-14	4.05e-15	**0**	**0**
	Std	1.43e-14	1.56e-14	1.64e-14	6.28e-15	**0**	**0**
F12	Mean	2.99e-13	2.48e-13	3.82e-13	3.72e-14	**0**	**0**
	Std	4.46e-13	3.23e-13	6.95e-13	4.93e-14	**0**	**0**
F13	Mean	6.16e-14	3.52e-15	2.17e-14	3.95e-15	**0**	**0**
	Std	1.35e-13	5.22e-15	4.95e-14	1.23e-14	**0**	**0**
F14	Mean	1.23e-16	9.23e-14	2.13e-21	1.33e-21	5.80e-61	**1.83e-133**
	Std	1.23e-16	1.96e-13	3.97e-21	3.23e-21	2.98e-60	**9.53e-133**
F15	Mean	1.51e-09	1.56e-09	9.85e-10	2.72e-10	1.56e-14	**1.06e-14**
	Std	5.59e-10	5.26e-10	3.82e-10	1.93e-10	3.11e-15	**2.74e-15**

Table 3. Radiuses and ratios of four gird shapes

2-D Grid / Neighborhood	4 × 12		5 × 10		6 × 8		7 × 7	
	rad_{shape}	*ratio*	rad_{shape}	*ratio*	rad_{shape}	*ratio*	rad_{shape}	*ratio*
L5	3.6429	0.2455	3.2879	0.2720	3.1160	0.2870	2.8284	0.3162
L9	3.6429	0.4092	3.2879	0.4534	3.1160	0.4784	2.8284	0.5270
C9	3.6429	0.3170	3.2879	0.3512	3.1160	0.3706	2.8284	0.4082
C13	3.6429	0.4029	3.2879	0.4464	3.1160	0.4710	2.8284	0.5189
C21	3.6429	0.4940	3.2879	0.5473	3.1160	0.5775	2.8284	0.6362
C25	3.6429	0.5490	3.2879	0.6083	3.1160	0.6418	2.8284	0.7071

4.3.3. *Performance comparison of ABCCA with six neighborhood structures.* In this subsection, six neighborhoods in Figure 4 are used in ABCCA to find out how these neighborhoods impact the performance of ABCCA, and the algorithms are represented as ABCCA-L5, ABCCA-L9, ABCCA-C9, ABCCA-C13, ABCCA-C21, and ABCCA-C25 respectively. The fixed 2-D grid shape for all algorithms is $6 \times 8 \approx 48$ according to the results obtained in the former subsection. The results of mean best values and standard deviations over 50 runs are listed in Table 5, from which, it can be noted that all the

Table 4. Performance comparisons among the ABCCA with four grid shapes

F	ABCCA-G1	ABCCA-G2	ABCCA-G3	ABCCA-G4
F1	1.12e-77	3.88e-75	**7.72e-78**	2.53e-76
	1.47e-77	5.69e-75	**1.23e-77**	4.13e-76
F2	4.59e-74	1.26e-71	**1.46e-74**	3.81e-73
	6.52e-74	1.83e-71	**1.95e-74**	6.05e-73
F3	**1.75e-78**	5.18e-76	2.73e-78	1.96e-77
	2.76e-78	8.82e-76	4.18e-78	2.57e-77
F4	**−3.64e-12**	**−3.64e-12**	**−3.64e-12**	**−3.64e-12**
	0	**0**	**0**	**0**
F5	1.02e-40	1.27e-39	**5.13e-41**	1.65e-40
	8.21e-41	9.49e-40	**4.20e-41**	1.36e-40
F6	5.42e+00	5.46e+00	**5.24e+00**	5.55e+00
	9.80e-01	**9.00e-01**	9.69e-01	9.14e-01
F7	**0**	**0**	**0**	**0**
	0	**0**	**0**	**0**
F8	1.48e-02	1.60e-02	**1.46e-02**	1.52e-02
	3.77e-03	4.35e-03	4.30e-03	4.11e-03
F9	2.30e-01	**2.28e-01**	2.34e-01	2.30e-01
	3.37e-02	**2.63e-02**	2.84e-02	3.07e-02
F10	4.07e-01	**1.65e-01**	2.78e-01	2.36e-01
	1.12e+00	**2.52e-01**	5.72e-01	4.08e-01
F11	**0**	**0**	**0**	**0**
	0	**0**	**0**	**0**
F12	**0**	**0**	**0**	**0**
	0	**0**	**0**	**0**
F13	**0**	**0**	**0**	**0**
	0	**0**	**0**	**0**
F14	4.99e-138	1.83e-133	**1.59e-139**	6.93e-137
	3.29e-137	9.53e-133	**5.38e-139**	3.28e-136
F15	1.09e-14	1.06e-14	1.08e-14	**1.05e-14**
	2.73e-15	2.74e-15	2.87e-15	2.96e-15

algorithms are able to reach the global optima on four functions, including F7, F11, F12 and F13. It can be also seen that both ABCCA-C25 and ABCCAC21 show the best performance and the difference between them is not obvious. ABCCA-L5 performs the worst, while the performance of the other three algorithms is between ABCCA-C25 and ABCCA-L5. The performance of these six algorithms shows a promising trend, which is related to the ratios of the neighborhood as presented in Table 1. The neighborhood C25 has the largest ratio, while L5 holds the lowest ratio. It can be concluded that an inherent relationship exists between the ratios of neighborhood and the performance of relevant algorithms.

5. **Conclusions.** In this paper, ABC has been extended to ABCCA after elaborately analyzing the effect of the onlooker bees and scouts with well-designed experiments. The cellular structured neighborhood is introduced to the ABCCA to make individuals only interact with their neighbors while preserving the population diversity. Besides, a more intelligent and robust probability calculation method based on rank ordering is developed to determine the qualified solutions regarded as onlooker bees. The experimental results conducted on 15 benchmark functions indicate that ABCCA has superior capabilities in terms of accuracy, robustness and efficiency.

Table 5. Performance comparison among ABCCA with six neighborhoods

Algorithms		ABCCA-L5	ABCCA-L9	ABCCA-C9	ABCCA-C13	ABCCA-C21	ABCCA-C25
F1	Mean	1.92e-72	2.41e-76	9.98e-76	2.86e-77	1.34e-77	**7.72e-78**
	SD	2.70e-72	4.04e-76	1.26e-75	4.26e-77	2.09e-77	**1.23e-77**
F2	Mean	1.21e-69	4.90e-73	2.01e-72	8.80e-74	**1.42e-74**	1.46e-74
	SD	1.11e-69	6.48e-73	2.36e-72	1.62e-73	2.09e-74	**1.95e-74**
F3	Mean	2.01e-73	2.45e-77	1.11e-76	3.62e-78	**9.35e-79**	2.73e-78
	SD	2.48e-73	3.28e-77	1.37e-76	5.67e-78	**1.78e-78**	4.18e-78
F4	Mean	**−3.64e-12**	**−3.64e-12**	**−3.64e-12**	**−3.64e-12**	**−3.64e-12**	**−3.64e-12**
	SD	**0**	**0**	**0**	**0**	**0**	**0**
F5	Mean	6.46e-38	4.04e-40	1.06e-39	1.17e-40	5.58e-41	**5.13e-41**
	SD	4.65e-38	2.38e-40	8.63e-40	6.88e-41	**3.79e-41**	4.20e-41
F6	Mean	7.18e+00	6.17e+00	6.39e+00	5.95e+00	5.28e+00	**5.24e+00**
	SD	9.88e-01	8.99e-01	1.21e+00	9.52e-01	**8.78e-01**	9.69e-01
F7	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	SD	**0**	**0**	**0**	**0**	**0**	**0**
F8	Mean	1.99e-02	1.61e-02	1.61e-02	1.57e-02	**1.44e-02**	1.46e-02
	SD	4.73e-03	4.15e-03	3.92e-03	3.94e-03	**3.62e-03**	4.30e-03
F9	Mean	2.51e-01	2.39e-01	2.43e-01	2.37e-01	**2.29e-01**	2.34e-01
	SD	3.11e-02	2.73e-02	**2.43e-02**	3.00e-02	3.14e-02	2.84e-02
F10	Mean	9.87e-02	1.28e-01	8.28e-02	**5.68e-02**	2.80e-01	2.78e-01
	SD	1.49e-01	2.25e-01	1.15e-01	**8.58e-02**	4.74e-01	5.72e-01
F11	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	SD	**0**	**0**	**0**	**0**	**0**	**0**
F12	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	SD	**0**	**0**	**0**	**0**	**0**	**0**
F13	Mean	**0**	**0**	**0**	**0**	**0**	**0**
	SD	**0**	**0**	**0**	**0**	**0**	**0**
F14	Mean	4.81e-112	4.53e-126	9.30e-129	3.95e-134	8.51e-138	**1.59e-139**
	SD	2.52e-111	3.15e-125	5.68e-128	1.68e-133	3.13e-137	**5.38e-139**
F15	Mean	1.26e-14	1.20e-14	1.15e-14	1.13e-14	**1.02e-14**	1.08e-14
	SD	**1.60e-15**	2.12e-15	2.06e-15	2.79e-15	2.93e-15	2.87e-15

Our future work will focus on applying ABCCA on feature selection in text sentiment analysis problems.

Acknowledgment. This work is supported by the Central University Basic Research Special Funding (Project ID: 2017JDZD07), the Open Foundation of Shandong Key Lab of Language Resource Development and Application, and Special Funding of Jiangnan University Basic Research (Project ID: 2014ZX07).

REFERENCES

[1] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.
[2] R. C. Eberhart, Y. Shi and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann, 2001.
[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
[4] I. Rechenberg, *Cybernetic Solution Path of an Experimental Problem*, Royal Aircraft Establishment Library Translation, 1965.
[5] H. P. Schwefel, *Kybernetische Evolution als Strategie der Exprimentellen Forschung in der Strömungstechnik*, 1965.
[6] L. J. Fogel, A. J. Owens and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley & Son, New York, 1966.
[7] R. Storn and K. Price, *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, Technical Report, International Computer Science Institute, Berkley, 1995.

[8] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, pp.341-359, 1997.

[9] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.

[10] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.

[11] L. N. D. Castro and F. J. V. Zuben, *Artificial Immune Systems: Part I – Basic Theory and Applications*, Technical Report Rt Dca 01/99, Feec/Unicamp, Brazil, 1999.

[12] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report-TR06, Erciyes University, 2005.

[13] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, vol.39, no.3, pp.459-471, 2007.

[14] D. Karaboga and B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, vol.8, no.1, pp.687-697, 2008.

[15] D. Karaboga and B. Basturk, A comparative study of artificial bee colony algorithm, *Applied Mathematics & Computation*, vol.214, no.1, pp.108-132, 2009.

[16] G. P. Zhu and S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics & Computation*, vol.217, no.7, pp.3166-3173, 2010.

[17] W. F. Gao and S. Y. Liu, Improved artificial bee colony algorithm for global optimization, *Information Processing Letters*, vol.111, no.17, pp.871-882, 2011.

[18] L. S. Coelho and P. Alotto, Gaussian artificial bee colony algorithm approach applied to Loney's solenoid benchmark problem, *IEEE Transactions Magnetics*, vol.47, no.5, pp.1326-1329, 2011.

[19] F. Kuang, Z. Jin, W. Xu et al., A novel chaotic artificial bee colony algorithm based on tent map, *Proc. of IEEE Congress on Evolutionary Computation (CEC)*, pp.235-241, 2014.

[20] Y. J. Shi, C. M. Pun, H. D. Hu and H. Gao, An improved artificial bee colony and its application, *Knowlede-Based Systems*, vol.107, pp.14-31, 2016.

[21] D. Karaboga and B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on the optimization problems, *Applied Soft Computing*, vol.23, no.5, pp.227-238, 2014.

[22] B. Akay and D. Karaboga, A modified artificial bee colony algorithm for real parameter optimization, *Information Sciences*, vol.192, no.6, pp.120-142, 2012.

[23] M. S. Kiran and O. Findik, A direct artificial bee colony algorithm, *Applied Soft Computing*, vol.26, pp.454-462, 2014.

[24] X. N. Li and G. F. Yang, Artificial bee colony algorithm with memory, *Applied Soft Computing*, vol.41, pp.362-372, 2016.

[25] S. Wolfram, A new kind of science, *Applied Mechanics Reviews*, vol.12, no.56, pp.449-451, 2006.

[26] E. Alba and J. Troya, Cellular evolutionary algorithms: Evaluating the influence of ratio, *Lecture Notes in Computer Science*, Springer, Berlin, Heidel-berg, pp.29-38, 2000.

[27] W. F. Gao, F. T. S. Chan, L. L. Huang and S. Y. Liu, Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood, *Information Sciences*, vol.316, pp.180-200, 2015.