Journal of Engineering Research Volume 23, Number 1, March 2018

Knowledge-Based Artificial Bee Colony Algorithm for Optimization Problems

[1]B. H. Adebiyi, [2]M. B. Mu'azu, [3]A. M. S. Tekanyi, [4]A. T. Salawudeen, [5]R. F. Adebiyi
[1, 2 and 4] Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria
[3] and [5] Department of Communication Engineering, Ahmadu Bello University, Zaria, Nigeria
Corresponding e-mail: busayoadebiyi@yahoo.com

Abstract
This paper presents a cultural algorithm-based artificial bee colony algorithm to modify the artificial bee colony (ABC). The normative and situational knowledge inherent in the cultural algorithm is utilized to guide the step size as well as the direction of evolution of ABC at different arrangements. This was done in order to combat the disparity between exploration and exploitation associated with the standard ABC, which results in poor convergence and optimization inefficiency. Four variants of Cultural Artificial Bee Colony Algorithm (CABCA) are accomplished in MATLAB/Simulink program using different configurations of cultural knowledge. A total of 20 standards applied mathematical optimization benchmark functions (Ackley, Michalewicz, Quartic, Sphere etc) are employed to evaluate the performance, and it was found that all the four variants of CABCA outperformed the standard ABC. The superiority of CABCA variants over ABC justifies the essence of knowledge introduction in the belief space for self-adaptation.

Keywords *- Artificial Bee Colony; Cultural Algorithm; Cultural Artificial Bee Colony Algorithm; Exploration and Exploitation.*

1.0 INTRODUCTION

The artificial bee colony (ABC) is one of the swarm intelligence algorithms used to solve optimization problems which is inspired by the foraging behaviour of the honey bees. Since its advent (Karaboga, 2005) ABC and its variants have attracted increasing interest and has been applied to solve many real-world optimization problems (Karaboga and Basturk, 2008; Lin and Su, 2012; El-Telbany, 2013). The ABC has the advantages of strong robustness, fast convergence and fewer setting parameters. Also, the performance of ABC has already been compared to other Evolution algorithms (EAs) such as Generic algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) (Karaboga and Basturk, 2007, 2008; Karaboga and Akay, 2009). The results show that ABC is better than or at least comparable to the other compared methods (Karaboga and Akay, 2009). However, like other swarm algorithms, ABC also faces convergence problems. The reasons are as follows. It is well known that both the exploration and the exploitation are necessary for a population-based optimization algorithm. In EAs, the exploration refers to the ability to investigate the various unknown regions in the solution space to discover the global optimum. The exploitation refers to the ability to apply the knowledge of the previous good solutions to find better solutions. Actually, the two aspects contradict to each other. For the sake of the good performance on the optimization problems, the main challenge is how to strike a good balance between the exploration and the exploitation in the search process Gao *et. al.* (2016). In order to address these shortcomings, an adaptive parameter is introduced to address the imbalance. This paper presents a knowledge based ABC called the CABCA using both situational and normative cultural knowledge. This research is an extension of Adebiyi *et. al.* (2017) as they did not explore various configurations of the situational and normative knowledge.

To improve the performance of the standard ABC, some ABC variants have been developed. Here are some of the few; In the work of Lee and Cai, (2011), a diversity strategy (DABC) was introduced to preserve sufficient amount of diversity among the candidate solutions by switching between exploration and exploitation. Also in the work of Banharnsakun *et. al.* (2011), a best-so-far method was proposed. The onlooker bees compared the information from all employed bees to select the best-so-far candidate food source. This will bias the solution handled by onlooker bees towards the optimal solution. Yan and Li proposed a chaotic local search method which was applied to solve the accuracy problem of global optimal value (Yan and Li, 2011). Gao *et al.* (2016) improved the exploitation ability of ABC by proposing a new algorithm, i.e. DGABC, which combined DE with best-guided ABC (GABC) by an evaluation strategy with an attempt to utilize more prior information of previous search experience to speed up the convergence. This was done by increasing the numbers of the scout to enhance better initialization, which in turns improved the convergence of ABC algorithm. Also, in the work of Liang *et. al.* (2017), an adaptive differential operator was embedded into the employed bee phase of ABC to enhance global convergence capability of the algorithm. A stair-step probability calculation method was designed for onlooker bees to differentiate the good solutions and bad solutions, such that more computational effort can be put into the local search of promising areas. A novel ABC variant namely ABCADE is formed by combing the basic framework of ABC and the two proposed algorithmic components. The experimental results show that ABCADE obtains superior or comparable performance to other representative state-of-the-art ABC and DE algorithm. Adebiyi *et. al.* (2017) proposed an improved ABC algorithm using knowledge inherent in Cultural Algorithm (CA) to develop two variants of ABC. The simulation results showed that the two variants have better performance than the standard ABC.

Our preliminary review of literature indicates that successful efforts have been made by researchers to address some of the problems associated with the ABC. However, the experience of individuals and the knowledge derived from the parent as a result of evolution has not been efficiently utilized. In this work, this individual experience will be employed through cultural evolution process to propose new variations of ABC called the CABCA. This is expected to improve the original ABC algorithm but not necessarily complicate the algorithm.

Artificial Bee Colony Algorithm

The standard ABC Algorithm consists of three kinds of bees; employed bees, onlooker bees and scout bees. Employed bees are responsible for exploiting the nectar sources explored and take the information obtained about the quality of the food source locations which they are exploiting to the waiting bees (onlooker bees). Onlooker bees wait in the hive and decide on a food source to be exploited based on the information received from the employed bees. Scouts bees randomly search the environment in order to find a new food source depending on an internal motivation or based on possible external traces. In the ABC algorithm, the position of a food source represents a possible solution to the optimization problem, and the nectar amount of a food source corresponds to the profitability (fitness) of the associated solution (Karaboga, 2005). The ABC as an iterative algorithm starts with the employed bee searching for food source within the neighborhood of the food source. Let suppose an employed bee is currently positioned at a food source position. During this stage, each employed bee searches in the neighborhood of its current position to produce new trial food source v_i using:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{kj} - x_{ij}\right) \tag{1}$$

Where, $j \in \{1,2,...,D\}$ and $k \in \{1,2,...SN\}$ are randomly picked indices, D is the dimension of the problem, SN is the number of food positions and ϕ_{ij} is a uniform random value [-1, 1] (Karaboga and Basturk, 2008).

Thus, the new solution v_i is produced from x_i by perturbing its randomly picked location j^{th} parameter and using the information of x_i and another randomly picked solution x_k. If v_i has better fitness than the old food position x_i, then x_i is replaced by v_i. Otherwise, the previous position x_i is retained.

$$fitness(x_i) = \begin{cases} \dfrac{1}{(1+f(x_i))} if & f(x_i) \geq 0 \\ 1+|f(x_i)| & otherwise \end{cases} \quad (2)$$

For the problem of function optimization, where f is the function to be minimized, ABC computes the fitness of a candidate solution x_i using (2) (Bansal and Mittal, 2017):

After all the employed bees complete their search, they share their information related to the nectar amounts and the positions of their sources with the onlooker bees in the dance area. The dance area is where the exchange of information about the quality of food sources occurs. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source location with a probability related to its nectar amount.

$$P_i = \frac{fitness(x_i)}{\sum_{n=0}^{SN} ftness(x_n)} \quad (3)$$

The probability P_i that the employed bee with food source x_i would be picked by an onlooker bee is computed using (3), making the probability P_i to be proportional to $fitness(x_i)$.

Like the employed bees, each onlooker bee also employs (2) to produce trial food source v_i in the vicinity of its current food source position. If v_i has better fitness then x_i is replaced by v_i. Otherwise, x_i is discarded.

A scout bee is created only when a particular food source cannot be improved through a predetermined number of trials limit.

$$x_{ij} = \min{}_j + rand(0,1)(\max{}_j - \min{}_j) \quad (4)$$

The employed bee now becomes a scout bee and its food source is positioned randomly across the search space using (4) where j = 1, 2… D and $\lfloor \max{}_j, \min{}_j \rfloor$ is the search space along the dimension (Alam *et al.,* 2015).

Below is the pseudo-code for the standard ABC algorithm.

1) *Begin*
2) *Initialize the solution population,* $i = 1,....,SN$
3) *Evaluate population*
4) *cycle = 1*
5) *Repeat*

6) *Generate new solutions v_{ij} for the employed bees using (1) and evaluate them. Keep the best solution between current and candidate*
7) *Select the visited solution for onlooker bees by their fitness*
8) *Generate new solutions v_{ij} for the Onlooker bees using (2) and evaluate them*
9) *Keep the best solution between current and candidate*
10) *Determine if there exist an abandoned food source and replace it using a scout bee*
11) *Save in memory the best solution so far*
12) *cycle = cycle + 1*
13) *Until cycle = M C N*

In a robust search process, exploration and exploitation processes must be carried out together in equal proportion. The onlooker and employed bees carried out the exploitation process in the search space, while the scouts are responsible for the exploration process. In the case of real honey bees, the recruitment rate represents a measure of how quickly the bee colony finds and exploits a newly discovered food source (Karaboga and Basturk, 2008). This recruiting could also represent the measurement of the speed of which the feasible solutions of the difficult optimization problems can be discovered.

Cultural Algorithm

Cultural Algorithms (CAs), is an evolutionary model which is inspired by the model of the cultural evolution process. The CAs has been developed in order to model the evolution of the cultural component of an evolutionary computational system over time as it accumulates experience. CAs can provide an explicit mechanism for global knowledge and a useful framework within which to model self-adaptation in an evolutionary or swarm intelligence system (Chung, 1997; Reynolds and Chung, 1997; Reynolds and Peng, 2005). CAs model has two levels of evolution (population level and belief space level). The population space consists of a set of possible solutions to the problem and can be modelled by using any population-based optimization method. The belief space is the place where the information about the knowledge on the solution of the problem is developed and stored. The belief space has the goal to guide individuals of the population in search of better regions. The five basic categories of cultural knowledge have been identified: Normative, Situational, Domain, History, and Topographical Knowledge (Reynolds and Peng, 2004). In this paper, the population component of the CA will be the ABC and the global knowledge that has been learned by the population will be expressed using situational and normative knowledge.

Situational Knowledge

Situational Knowledge represents the best individuals found at a certain time of evolution and it contains a number of individuals considered as a set of exemplars to the rest of the population. The Situational Knowledge is updated when the best individual of the population is found, that is at iteration t, $S = \prec S^t | S^t = \{s_1^t, s_2^t, ..., s_n^t\} \succ$. This situational knowledge equation can be represented as (Reynolds and Chung, 1997):

$$S_j^{t+1} = \begin{cases} X_{best,j} t+1 & if \quad f\left(X_{best,j}^{t+1}\right) \prec f\left(s^t\right) \\ s_j^t & otherwise \end{cases} \tag{5}$$

where X_{gbest}^{t+1} denotes the best artificial bee individuals in the colony at generation $t+1$

Normative Knowledge

Normative Knowledge describes how the individual should act in terms of ranges of acceptable behaviour. In other words, normative knowledge defines a standard or ideal way that can be used to judge which behaviour is desirable or undesirable. Normative Knowledge provides standards for interpreting and determining individual behaviours through guidelines within which individual adjustments can be made. The normative component is a set of interval information for each n parameter. Each of the intervals in the belief space is represented as a triple Reynolds and Chung, (1997).

$$N = \langle I, L, U \rangle \tag{6}$$

where I, L and U , are $n-$ dimensional vectors, and I_j denotes the closed interval for the variable j that is a continuous set of real numbers x represented as a number pair:

$$I_j = [l_j, u_j] = \{x | l_j \le x \le u_j\} \tag{7}$$

n is the number of the variables, l_j and u_j are the lower and upper bounds for the j^{th} variable, respectively, L_j and U_j are the values of the fitness function associated with the bound l_j and u_j are usually initialized with positive infinity. Usually, the normative knowledge leads individuals "to jump into the good range" if they are not there yet usually initialized with positive infinity (Reynolds and Peng, 2004).

The Normative Knowledge is updated as follows Chung, (1997):

$$l_j^{t+1} = \begin{cases} x_{i,j}^t & if\ x_{i,j}^t \le l_j^t \quad or \quad f(x_i^t) \prec L_j^t \\ l_j^t & otherwise \end{cases} \tag{8}$$

$$u_j^{t+1} = \begin{cases} x_{k,j} & if\ x_{k,j} \ge u_j^t \quad or \quad f(x_k) \prec U_j^t \\ u_j^t & otherwise \end{cases} \tag{9}$$

$$L_j^{t+1} = \begin{cases} f(X_i) & if\ \ x_{i,j}^t \le l_j^t \ \ or\ f(X_i) \prec L_j^t \\ L_j^t & otherwise \end{cases} \tag{10}$$

$$U_j^{t+1} = \begin{cases} f(X_k) & if\ \ x_{k,j} \ge u_j^t \ \ or\ f(X_k) \prec U_j^t \\ U_j^t & otherwise \end{cases} \tag{11}$$

where the i^{th} individual affects the lower bound for the variable j, and the k^{th} individual affects the upper bound for the variable j. It should be noted that t denotes the current generation of individuals.

2.0 METHODOLOGY

In this research, the Cultural Algorithm is used to propose an improvement on Artificial Bee Colony (ABC) Algorithm. The population level of the Cultural Algorithm component will be the ABC algorithm. The global knowledge that has been learned by the population is expressed in terms of both normative and situational knowledge.

The experience of the selected individuals from the population (bee) contributes to the cultural knowledge by means of acceptance function. An acceptance function is used to control which members of the population are permitted to impact the belief space. They determine which individuals in the current generation of the population will contribute with their knowledge to belief space. The experiences of the selected individuals are used to update the knowledge of the current belief space. In turns, the knowledge in the beliefs will be used to guide and influence the evolution of the population. In this research, the acceptance function for component N selects the best range of the 20% performing individuals in each generation is calculated and successive generations will be randomly generated in this promising range. This will increase the convergence and will preserve wasting time in discovering the good regions. The number of the individuals accepted for the update of the belief space is obtained according to the following function (Chung, 1997; Reynolds and Chung, 1997; Salawudeen, 2015).

$$\alpha(N,t) = N.\beta + \lfloor N.\beta/t \rfloor \quad (12)$$

where N is the size of the population, t is the current number of generation and β is a parameter given by the user t (in this work, 0.2 is adopted). That is, top 20% of the population.

CABCA Knowledge Adjustment and Variants

Each of the knowledge in the belief space has its own rules and approach of operations. This rule which is also referred to as update function as described above in situational and normative knowledge. Consequently, four variants of CABCA (CABCA(Ns), CABCA(Sd), CABCA(Ns+Sd) and CABCA(Ns+Nd)) were developed using the different influence function.

Below is the pseudo-code for the “cultured” ABC algorithm.

1) Begin
2) Initialize the solution population,
3) Evaluate population
4) cycle = 1
5) Repeat
6) Generate new solutions for the employed bees using (1) and evaluate them. Keep the best solution between current and candidate
7) Update the belief space with the given problem domain and candidate solutions.
8) Apply acceptance function (top 20%)
9) Select the visited solution for onlooker bees by their fitness
10) Generate new solutions for the Onlooker bees using (2) and evaluate them
11) Keep the best solution between current and candidate
12) Determine if there exist an abandoned food source and replace it using a scout bee
13) Save in memory the best solution so far
14) cycle = cycle + 1
15) Until cycle = M C N

The belief space knowledge can influence the evolutional operation, in two ways (Chung, 1997):

1) Determining the step size of the evolution
2) Determine the direction of evolution

These two operations are applied to the solution search equation for the ABC as given in equation (1) as adaptive parameters in order to balance exploration and exploitation which is often necessary for good results and sufficient convergence speed of the algorithm, especially for complex, high dimensional, multimodal problems with many local optimal points. The variants were developed using the following configurations:

CABCA(Ns): CABCA using normative knowledge

This is the simplest form of cultural influence. This variant uses only the normative knowledge to determine the step size of evolution. For all components $i = 1,...,t$ and $j = 1,...,n.$ This can be expressed mathematically as follows:

$$x_{ij}^{t+1} = x_{ij}^{t} + \left|size(I_j)\right| \cdot \phi_{ij}(x_{ij} - x_{kj}) \tag{13}$$

where $size(I_j) = u_i - l_i$ is the size of the belief space interval for the parameter $i,$ which is decided by the normative knowledge for i^{th} variable

CABCA(Sd): CABCA using situational knowledge

In this variant only situational knowledge of the current exemplar or best solution found so far, is used to decide the direction of evolution. The variation is influenced as follows:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^{t} + \left|\phi_{ij}(x_{ij} - x_{kj})\right| & if \quad x_{ij}^{t} \prec s_j^t \\ x_{ij}^{t} - \left|\phi_{ij}(x_{ij} - x_{kj})\right| & if \quad x_{ij}^{t} \succ s_j^t \\ x_{ij}^{t} + \phi_{ij}(x_{ij} - x_{kj}) & otherwise \end{cases} \tag{14}$$

CABCA(Ns+Sd): CABCA using normative and situational knowledge

This variant uses both normative knowledge to determine the step size, and situational knowledge to determine the direction as shown in the following, for all components $i = 1,..,t$ and $j = 1,...,n$

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^{t} + \left|\delta \times \phi_{ij}(x_{ij} - x_{kj})\right| & if \quad x_{ij}^{t} \prec s_j^t \\ x_{ij}^{t} - \left|\delta \times \phi_{ij}(x_{ij} - x_{kj})\right| & if \quad x_{ij}^{t} \succ s_j^t \\ x_{ij}^{t} + \delta \times \phi_{ij}(x_{ij} - x_{kj}) & otherwise \end{cases} \tag{15}$$

where $\delta = size(I_j)$ is the size of the belief space interval which is $u_i - l_i$

If an individual's parameter value is less than that of the current best, then the absolute value of the calculated step size, $size(I_j)$, is added to the current parameter value. If an individual's parameter value is greater than that of the current best, then the absolute value of the step size, $size(I_j)$, is subtracted from the current parameter value. If an individual's parameter value is equal to that of the current best, then $size(I_j) \cdot \phi_{ij}(x_{ij} - x_{kj})$ is just added to the current parameter value. In this case, the direction of evolution will be random.

CABCA(Ns+Nd): CABCA using normative for step size and direction of evolution

This variant uses only normative knowledge to influence the evolution of both step size and direction. The basic idea behind this variant is to perturb small in a random direction when the parameter value of a parent is in the acceptable range; otherwise, perturb according to the current belief range found toward the left or right boundary of the current range in the belief space. For all components $i = 1,...,t$ and $j = 1,...,n$.

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^{t} + \left|\omega \times \phi_{ij}\left(x_{ij} - x_{kj}\right)\right| & if \quad x_{ij}^{t} \prec l_j \\ x_{ij}^{t} - \left|\omega \times \phi_{ij}\left(x_{ij} - x_{kj}\right)\right| & if \quad x_{ij}^{t} \succ l_j \\ x_{ij}^{t} + \left|\omega\right| \times \phi_{ij}\left(x_{ij} - x_{kj}\right) & otherwise \end{cases} \tag{16}$$

where $|\omega| = size(I_k) = |u_k - l_k|$ which represents the size of the current upper limit and lower limit in the belief space for parameter j respectively.

Also if an individual's parameter value is less than that of the current best, then the absolute value of the calculated step size, $size(I_k)$, is added to the current parameter value. If an individual's parameter value is greater than that of the current best, then the absolute value of the step size, $size(I_k)$, is subtracted from the current parameter value. If an individual's parameter value is equal to that of the current best, then $size(I_k) \cdot \phi_{ij}(x_{ij} - x_{kj})$ is just added to the current parameter value. In this case, the direction of evolution will be random. The Flowchart of cultural artificial bee colony algorithm is shown in Figure 1.

Optimization Benchmark Test Functions

In order to validate any new optimization algorithm, one has to validate it against standard test functions so as to compare its performance with well-established or existing algorithms. There are many test functions, so there is no standard list or set of test functions one has to follow. However, various test functions do exist in literature, so new algorithms should be tested using at least a subset of functions with diverse properties so as to make sure whether or not the tested algorithm can solve a certain type of optimization efficiently. In this research, a collection of twenty (20) unconstrained optimization test functions which are used to evaluate the performance of the proposed algorithm. The purpose of this collection is to give the proposed algorithms a large number of general test functions to be used in testing the optimization algorithm and comparison studies. For each function, has its algebraic expression and the standard initial point, as well as optimal values, are given. The Flowchart of cultural artificial bee colony algorithm is shown in Figure 1.

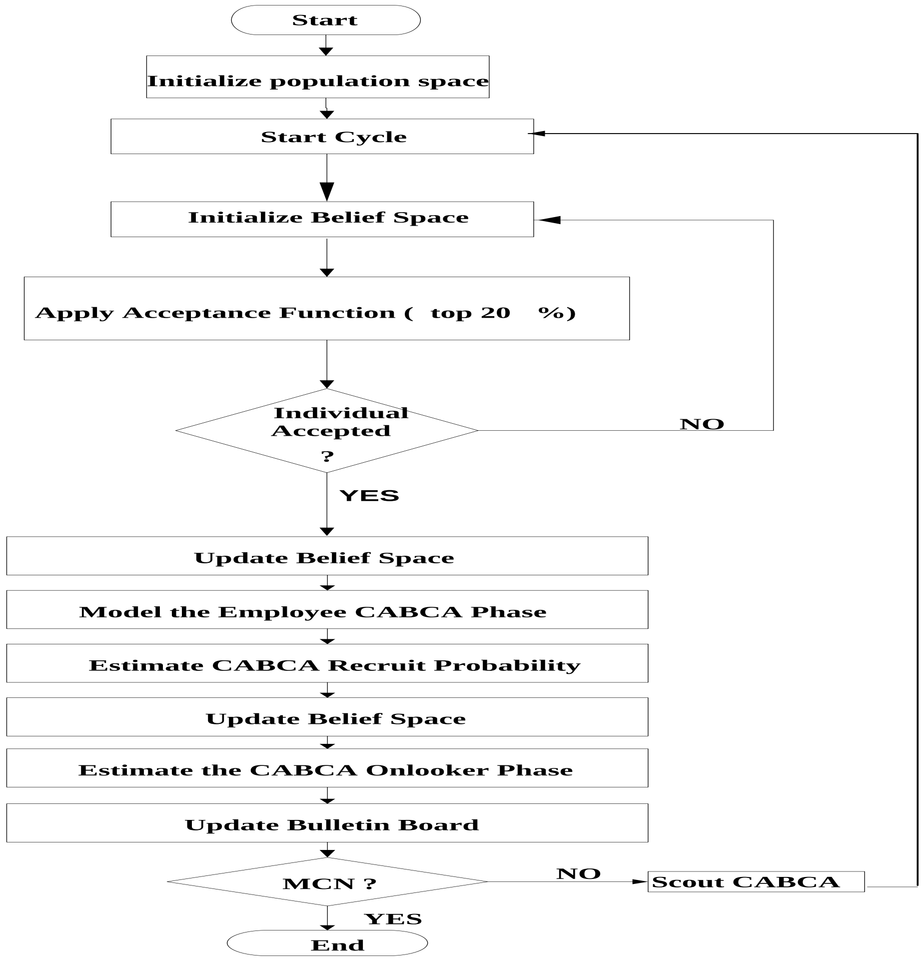

Figure 1: Flowchart of Cultural Artificial Bee Colony Algorithm (CABCA)

3.0 RESULTS AND DISCUSSION

As discussed above, the development of four variants of CABCA are simulated in MATLAB Control/ Optimization tool box. The performance of CABCA variants and standard ABC is evaluated using a collection of twenty (20) applied mathematical optimization test functions. Henceforth, the comparison between CABCA variants and the standard ABC algorithm with reference to the global solution is estimated. The parameters setting of this work is indicated in Table 1. For each system and for each test case, the average of twenty (20) tests performed using MATLAB R2015a was recorded. The simulations were performed on HP-Pavilion g7-2270us on an Intel(R) Core i3 with a 2.40GHz processor and 6.00GB RAM with 64-bit Windows 8 Pro Operating System (OS).

Table 1: Parameter Setting for ABC

S/N	Parameters	Values
1	Population size	50
2	Dimension	30
3	Maximum Cycle Number	5000
4	Limit	50

Comparison of ABC with CABCA

In this section, ABC is compared with all the variants of the proposed CABCA variants. The t-test is introduced to clearly indicate the statistical significance of the algorithms. This is clearly presented in Table 2.

- **ABC vs CABCA(Ns):** Out of the 20 functions, CABCA(Ns) performs better than ABC on 13 functions; while ABC performs better only on one (1) function. On the remaining six (6) functions, their results are similar (i.e., the performance difference is not statistically significant in t-tests with at least 99% degree of confidence). Thus the overall performance of CABCA(Ns) is better than standard ABC.
- **ABC vs CABCA(Sd):** It can be observed from Table 2 that on all of the 20 test functions, CABCA(Sd) produced the best result in 11 and ABC has the best performance in 2 functions. On the remaining 7 functions, their results are similar.
- **ABC vs CABCA(Ns+Sd):** From Table 2, it could be seen that the variant of CABCA that employ both normative and situational knowledge to guide the evolution produced the best results in eleven (11) functions while ABC produced best results in two (2) functions and both CABCA(Ns+Sd) and ABC produced similar results in seven (7) functions.
- **ABC vs CABCA(Ns+Nd):** In this variant, it could be observed that CABCA(Ns+Nd) outpaced the ABC in ten (10) and two (2)cases respectively while they provide similar result in eight (8) cases.

In total, the variants of CABCA (i.e. CABCA(Ns), CABCA(Sd), CABCA(Ns+Sd) and CABCA(Ns+Sd)) outperformed the standard ABC in all the optimization test functions. The superiority of CABCA justifies the essence of knowledge introduction in the belief space for self-adaptation. This has substantially improved the performance of the standard ABC in the search for the global solution.

The following points summarize the observations on the results:

1) Out of the 20 functions CABCA(Ns) performs best in 4 (20%) functions, CABCA(Ns+Nd) also in 4 (20%) functions, while CABCA(Sd) and CABCA(Ns+Sd) performed best in 3 (15%) and 2 (10%) functions respectively. On the remaining 7 (35%) functions, their results are similar (i.e., the performance difference is not statistically significant in t-tests with at least 99% degree of confidence). Thus the overall performance of CABCA is better than ABC.
2) Variants that employed normative knowledge prevailed in most of the test cases. Hence, normative knowledge seems to be the prevailing knowledge source for the optimization functions.
3) The unimodal and separable functions are relatively easier to optimize. On these functions, the performance of ABC and CABCA variants are mostly similar.
4) The best-performed variant is selected based on the success ratio, which is the number of successful runs that found the solution.

Table 2: Performance of the ABC algorithm and proposed CABCA algorithms on the benchmark functions.

Test Function	G_{min}	ABC	CABCA(N_s)	CABCA(S_d)	CABCA (Ns+Sd)	CABCA (Ns+Nd)	Best Performance (t-Test with λ =0.99)
Ackley	0.000E+00	2.1157E-13	***4.1466e-15***	1.9851E-13	5.7676E-15	4.3750E-15	CABCA(Ns)
Axis parallel	0.0000E+00	9.5846E-16	***1.0472E-16***	8.7434E-16	5.4133E-16	8.5830E-16	CABCA(Ns)
CM	-3.0000E+00	-3.0000E+00	-3.0000E+00	-3.0000E+00	**-3.0000E+00**	-3.0000E+00	Similar
DeJongf4	0.0000E+00	5.4743E-16	***4.3421E-16***	5.1616E-16	4.5198E-16	4.7587E-16	CABCA(Ns)
ExpFun	1.0000E+00	1.0000E+00	0.0000E+00	1.0000E+00	0.0000E+00	0.0000E+00	Similar
Griewangk	0.0000E+00	0.0000E+00	1.0000E+00	0.0000E+00	1.0000E+00	1.0000E+00	Similar
Hyperelliptic	0.0000E+00	1.0000E-15	***8.8185E-43***	9.1751E-16	1.51583-42	1.2385E-42	CABCA(Ns)
Michalewicz	-9.6602E+00	-26.5754E+00	-2.3832E-13	***-24.3579E+00***	-6.0315E-15	-2.4091E-13	CABCA(Sd)
Neumaier3	-4.9300E+03	-2.8331E+03	-2.9000E-04	***-2.9523E+03***	-2.9000E-04	-2.9000E-04	CABCA(Sd)
PM1	0.0000E+00	1.2349E-15	3.7832E-27	1.2310E-15	4.0409E-30	***1.3401E-45***	CABCA(Ns+Nd)
PM2	0.0000E+00	1.55319E-11	2.5959E-27	4.4354E-12	2.5959E-27	2.5959E-27	Similar
Quartic	0.0000E+00	10.9770E+00	8.4591E-01	8.3048E+00	2.4158E-02	***4.2308E-21***	CABCA(Ns+Nd)
Rastrigin	0.0000E+00	2.3628E-12	1.1806E-13	2.1652E-12	3.5527E-15	***7.1059E-15***	CABCA(Ns+Nd)
Rosenbrock	0.0000E+00	1.1236E+00	5.9490E-02	1.1179E+00	1.4246E-02	***2.0832E-13***	CABCA(Ns+Nd)
Sal	0.0000E+00	9.9873E-02	9.9873E-02	9.9873E-02	9.9873E-02	9.9873E-02	Similar
Schaffer	0.0000E+00	5.7452E-15	7.4541E-19	6.5000E-15	***7.2048E-19***	7.6013E-19	CABCA(Ns+Sd)
Schwefel	0.0000E+00	3.5536E+03	3.5542E+03	3.5535E+03	3.5537E+03	3.5533E+03	Similar
Sphere	0.0000E+00	1.0875E-15	9.8378E-16	***9.1795E-16***	9.2855E-16	1.0886E-15	CABCA(Sd)
Step	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	Similar
Zakharov	0.0000E+00	6.8857E-13	1.1469E-15	1.3933E-15	***1.6837E-19***	4.3144E-15	CABCA(Ns+Sd)

Results are averaged over 20 independent runs. Better performance by CABCA variants is marked with boldface and italic font. In case the best CABCA performance to ABC performance difference is not significant by t-Test with at least 99% level of confidence (i.e., λ = 0.99), it is marked as “Similar”

4.0 CONCLUSION

In this work, a Cultural Artificial Bee Colony Algorithm (CABCA) for global optimization has been proposed. In the algorithm, the normative and situational knowledge inherent in the cultural algorithm was utilized to guide the step size as well as the direction of evolution of ABC at the employed bee stage. This was done in order to combat the disparity between exploration and exploitation associated with the basic ABC, which often lead to poor convergence and optimization inefficiency. Consequently, four new variants CABCA(Ns), CABCA(Sd), CABCA(Ns+Sd) and CABCA(Ns+Nd) were developed in MATLAB R2015a. The exploration and the exploitation ability of the algorithm have been balanced, and the search performance of the approach is improved. Various benchmark functions including unimodal separable, unimodal non-separable, multi-modal separable and multi-modal non-separable test functions have been applied to test the effectiveness of the presented method. The simulation results show that all the variants of CABCA outperformed the ABC and CABCA(Ns+Nd) has the overall best performance. In the future work, we will focus on the application of the proposed algorithms to various engineering problems.

REFERENCES

Adebiyi, B. H., Tekanyi, A. M. S., and Salawudeen, T. A. (2017). An Improved Artificial Bee Colony using Cultural Algorithm for Optimization Problem. *International Journal of Computer Applications, 160*(8).

Alam, M. S., Islam, M. M., and Murase, K. (2015). Artificial Bee Colony Algorithm with Adaptive Explorations and Exploitations: A Novel Approach for Continuous Optimization. *International Journal of Applied Information System, 8*(2), 32 - 43.

Banharnsakun, A., Achalakul, T., and Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm. *Applied soft computing, 11*(2), 2888-2901.

Chung, C.-J. (1997). Knowledge-based approaches to self-adaptation in cultural algorithms.

Chung, C. (1997). *Fuzzy Approaches to Acquiring Experimental Knowledge in Cultural Algorithms.* Paper presented at the Proceedings of the 9th International Conference on Tools with Artificial Intelligence.

El-Telbany, M. E. (2013). Tuning PID controller for DC motor: An artificial bees optimization approach. *International Journal of Computer Applications, 77*(15).

Gao, W.-f., Huang, L.-l., Wang, J., Liu, S.-y., and Qin, C.-d. (2016). Enhanced artificial bee colony algorithm through differential evolution. *Applied soft computing, 48*, 137-150.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization: Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Karaboga, D., and Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*(1), 108-132.

Karaboga, D., and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization, 39*(3), 459-471.

Karaboga, D., and Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing, 8*(1), 687-697.

Lee, W.-P., and Cai, W.-T. (2011). *A novel artificial bee colony algorithm with diversity strategy.* Paper presented at the Natural Computation (ICNC), 2011 Seventh International Conference on.

Liang, Z., Hu, K., Zhu, Q., and Zhu, Z. (2017). An enhanced artificial bee colony algorithm with adaptive differential operators. *Applied soft computing, 58*, 480-494.

Lin, C.-J., and Su, S.-C. (2012). Using an efficient artificial bee colony algorithm for protein structure prediction on lattice models. *International journal of innovative computing, information and control, 8*, 2049-2064.

Priti Bansal, S. S., and Nitish Mittal. (2017). A Hybrid Artificial Bee Colony and Harmony Search Algorithm to Generate Covering Arrays for Pair-wise Testing. *International Journal of Intelligent Systems andApplications(IJISA), 9*(8), 59-70. doi: 10.5815.

Reynolds, R. G., and Chung, C. (1997). *Fuzzy approaches to acquiring experimental knowledge in cultural algorithms.* Paper presented at the Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on.

Reynolds, R. G., and Peng, B. (2004). *Cultural algorithms: modeling of how cultures learn to solve problems.* Paper presented at the Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on.

Reynolds, R. G., and Peng, B. (2005). Cultural algorithms: computational modeling of how cultures learn to solve problems: an engineering example. *Cybernetics and Systems: An International Journal, 36*(8), 753-771.

Salawudeen, A. T. (2015). *Development of an Improved Cultural Artificial Fish Swarm Algorithm with Crossover.* (M.Sc Dissertation). Department of Electrical and Computer Engineering, Ahmadu Bello University Zaria, Nigeria. p.154.

Yan, G., and Li, C. (2011). An effective refinement artificial bee colony optimization algorithm based on chaotic search and application for pid control tuning. *Journal of Computational Information Systems, 7*(9), 3309-3316.