Quantum Artificial Bee Colony Algorithm for Numerical Function Optimization

Nizar Hadi Abbas, Ph.D
Electrical Engineering
Department
College of Engineering
University of Baghdad, Iraq.

Haitham Saadoon Aftan
Electrical Engineering
Department
College of Engineering
University of Baghdad, Iraq.

ABSTRACT
The Artificial Bee Colony (ABC) algorithm is a swarm intelligence based algorithm, which simulate the foraging behavior of honey bee colonies. It has been widely applied to solve the real-world problem. However, ABC has good exploration but poor exploitation abilities, and its convergence speed is also an issue in some cases. In order to overcome these issues, this paper presents a new metaheuristic algorithm called Quantum Artificial Bee Colony (QABC) algorithm for global optimization problems inspired by quantum physics concepts. Simulations are conducted on a suite of unimodal/multimodal continuous benchmark functions. The results demonstrate the good performance of the QABC algorithm in solving complex numerical optimization problems when compared with other popular algorithms.

General Terms
Bio-inspired algorithm, Numerical optimization, meta-heuristics.

Keywords
Artificial bee colony algorithm, Swarm intelligence, Quantum physics, Benchmark functions.

1. INTRODUCTION
In the last two decades, there has been an increasing interest in the field of bio-inspired computation specially swarm based algorithms. Some of these algorithms gained much popularity because of their versatility, flexibility and powerful adaptive search technique and efficient in solving practical nonlinear problems. There are two noteworthy features of swarm based algorithms are decentralized control and self-organization that lead to a distinguished behavior [1]. This behavior is a property that appears through interactions among system component, and it is impossible to be achieved by any of the components of the system acting alone. Several possible approaches of bio-inspired algorithms have been reported in literature, and some popular approaches are Particle Swarm Optimization (PSO) [2], Ant Colony Optimization (ACO) [3] and Artificial Bee Colony (ABC) [4] strategies. All these techniques have shown their robustness and if properly developed and tuned, have also shown relatively fast convergence.

Despite the popularity and success of bio-inspired computing, there remain many challenging issues such as gaps between theory and practice, classifications, parameter tuning, lack of truly large-scale real-world applications and the selection of the appropriate algorithm for specific problem [1].

This paper focuses on the ABC algorithm which is a population-based meta-heuristic approach originally proposed by D. Karaboga [4]. It simulates foraging behavior of honey bee swarm. ABC has been used to solve many optimization problems. Several studies have been made to compare of ABC with the performance of other algorithms such as Genetic Algorithm (GA), ACO and Differential Evolution (DE) [5]. Several unimodal/multimodal benchmarks functions have been used to evaluate the performance of ABC. These studies have been revealed that ABC algorithm can achieve better results and it is more efficient than other algorithms. However, the main issue in ABC algorithm is that it is good at exploration but poor at exploitation. The convergence is also an issue in some cases. The exploration is defined as the ability to independently seeking for the global optimum, while the exploitation is defined as the ability to apply the existing knowledge to look for better solutions [1].

In order to find the optimal tradeoff between exploration and exploitation, several modified version of ABC have been proposed in the literature, such as gbest guided ABC (GABC) [6] and I-ABC [7]. Inspired by PSO, GABC takes advantages of the information collected from the global best (gbest) solution to improve the exploitation capability of the standard ABC algorithm. L. Guoqiang et al. [7], proposed an improved ABC algorithm called I-ABC. In I-ABC, the inertial weight and acceleration coefficients were introduced to modify the search process. In addition to the further balancing of search processes, the modification forms of the employed bees and the onlooker ones are different in the second acceleration coefficient.

In this paper, a new modified version of the ABC algorithm called Quantum Artificial Bee Colony (QABC) Algorithm is proposed. QABC exploits the quantum physics concepts to construct a new update equation which balances between exploration and exploitation capabilities to achieve better results. This algorithm is applied to several benchmark functions to compare its performance with other algorithms.

The remainder of this paper is organized as follows: section 2 describes the standard ABC algorithm; section 3 describes the behavior of the bee in the proposed algorithm; Section 4 describes the proposed QABC algorithm, and the simulation results are shown in section 5. Finally, section 6 concludes the paper.

2. Standard ABC Algorithm
The artificial bee colony algorithm is a robust, straightforward, population based and a stochastic optimization algorithm. ABC algorithm was proposed by D.

Karaboga [4]. In the ABC algorithm, the colony of artificial bees is classified into three groups: employed bees, onlookers and scouts. The employed bees constitute half of the population, and the onlookers constitute the remaining half. The position of a food source corresponds to a feasible solution in search space of the optimization problem, and the quality of each food source is represented by its nectar amount. The number of the food sources equals to the number of employed bees. When a food source has been abandoned by bees, the abandoned employed bee would become a scout [4].

In the beginning, the ABC algorithm produces a randomly distributed initial population of SN food source positions, where SN is the size of food sources. Every solution $x_i = (i = 1,2, \ldots, SN)$ is a D-dimensional vector, where D denotes the number of decision variables. In the subsequent steps, the population of solutions is subject to repeated cycles of the search phases of employed bees, onlookers and scouts. An employed bee could produce a modification on the solution in its memory depending on local information and test its fitness value of the new source. If the fitness value of the new one is better than that of the previous one, the employed bee would memorize the new position and forget the previous one. Otherwise, it keeps the position of the previous one in its memory. When all employed bees complete the search process, they will share the information about nectar amounts and positions of food sources with onlookers. An onlooker evaluates the nectar information which is owned by all employed bees, and then chooses a food source with a probability which is related to the nectar amount. As in the case of the employed bee, the onlooker can produce a modification on the position in its memory and check the nectar amount of the candidate source. If the nectar amount is more than that of the previous one, the bee would memorize the new position and forgets the previous one.

An onlooker chooses a food source completely depending on the probability value associated with the food source $\mathrm{p_i}$, which is calculated by the following formula [8]:

$$\mathrm{p_i} = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \qquad (1)$$

Where fit_i denotes the fitness value of the *ith* solution which is proportional to the nectar amount of the food source in the *ith* position. For the sake of producing a new food position from the previous one, the ABC could adopt the following modification form [8]:

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \qquad (2)$$

Where $k \in \{1,2, \ldots, SN\}$ and $j \in \{1,2, \ldots, D\}$ are randomly generated index, and k are different from j. $\emptyset_{ij}$ is random variable in the interval [-1,1]. It can control to produce a new food source around x_{ij} and represent the comparison of two food positions visually by a bee. As can be seen from Eq. (2), as the difference between the parameter x_{ij} and x_{kj}decreases, the perturbation on the position x_{ij} is also decreased. Thus, when the search approaches to the optimum solution in the search space, the step size is adaptively reduced. If a parameter value produced by the operation exceeds its predetermined limit, it is set to its limit value.

The food source which is abandoned by the bees would be replaced with a new food source found by scouts. In ABC algorithm, the foraging behavior is simulated by randomly producing a position and replacing the abandoned one with a new one. If a position cannot be improved further through a predetermined number of cycles, the food source should be abandoned. The predetermined number of cycles is an important control parameter in ABC algorithm, which is called "limit" for abandonment. Suppose that the abandoned source is x_{ij} and $j \in \{1,2, \ldots, D\}$, then the scout finds a new food source to be replaced with x_{ij}. This operation can be defined as [8]:

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \qquad (3)$$

After each new source position v_{ij} produced, it can be evaluated by the artificial bee, and its fitness is compared with that of its previous one. If the new food source equals or is better than the old one, it would be replaced with the previous one in its memory. Otherwise, the old one is retained in its memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the new one. The main steps of the ABC algorithm are outlined in Algorithm 1[8].

Algorithm 1: Artificial Bee Colony Algorithm

Initialize food sources;

Repeat while Termination criteria is not meet

Step 1: Employed bee phase for computing new food sources.

Step 2: Onlooker bees phase for updating the location of food sources based on their amount of nectar.

Step 3: Scout bee phase for searching about new food sources in place of rejected food sources.

Step 4: Memorize the best food source identified so far.

End of while

Output: The best solution obtained so far.

3. Bees in Quantum Delta Potential Field

In the concepts of Newtonian mechanics, the particle is described by its velocity vector *v* and position vector *x*. Where *v* and *x* determine the trajectory of the particle. The particle moves along a predictable trajectory in Newtonian mechanics, but this is not the case in quantum mechanics. In quantum physics, the term trajectory is meaningless, because *x* and *v* of a particle cannot be determined simultaneously according to Heisenberg's uncertainty principle. Hence, if individual particles in an ABC system have quantum behavior, the ABC algorithm is bound to work in a different manner. In this paper, motivated by the work done in the philosophy of PSO with delta potential of QPSO proposed in [9] and [10], a new modified ABC approach with bees having quantum behavior is proposed.

In a quantum model of ABC, each bee represents a particle which has a state depicted by wave function $\Psi(x,t)$, instead of position and velocity. The dynamic behavior of the bee is different from that of the bee in standard ABC algorithm; in that the accurate values of *x* and *v* cannot be calculated simultaneously. The probability density function of bee position x_i is $|\Psi(x,t)|^2$. The update equation can be found as follows:

In this algorithm, the delta potential is used and its equation is as follows:

$$V(x) = -\lambda\delta(x) \quad (4)$$

Where λ is a positive value and $\delta(x)$ is Dirac delta function.

The wave function of a particle in delta potential is represented by the following formula [11]:

$$\Psi(x) = \sqrt{\frac{m\lambda}{\hbar^2}}\,(e^{-\frac{m\lambda|x|}{\hbar^2}}) \quad (5)$$

Where ħ is the reduced Plank constant, m is the mass of the particle and E is the energy of the particle.

Motivated by the Eberhart's PSO convergence analysis, Let the center of the potential be around position p_j defined by [12]:

$$p_j = \frac{r_1 p_{ij} + r_2 p_{gj}}{r_1 + r_2},\ j = 1,2,\ldots\ldots D\ \ ,i = 1,2,\ldots SN/2 \quad (6)$$

Where p_{ij} is the *ith* bee best position in the *jth* dimension of the hyperspace and p_{gj} is value of *jth* dimension of the bees' global best position. r_1 *and* r_2 are random variables in the range (0,1].

Then Eq. (4) becomes,

$$V(z) = -\lambda\delta(x-p) = -\lambda\delta(z) \quad (7)$$

Where $z=x-p$

Replace x with z in Eq. (5) then:

$$\Psi(x) = \sqrt{\frac{m\lambda}{\hbar^2}}\,(e^{-\frac{m\lambda|x|}{\hbar^2}}) \quad (8)$$

Hence, the probability of finding particle Q(z) in any position can be obtained:

$$\text{Since } Q(z) = |\Psi(z)|^2, \quad (9)$$

$$\therefore Q(z) = \frac{m\lambda}{\hbar^2}\exp\left(-2\frac{m\lambda|z|}{\hbar^2}\right) \quad (10)$$

To obtain the position value from this distribution, Smirnov transform is used

The cumulative distribution function F(z) is:

$$F(z) = \int_{-\infty}^{z}\frac{m\lambda}{\hbar^2}\exp\left(-2\frac{m\lambda|z|}{\hbar^2}\right) \quad (11)$$

$$F(z) = e^{\frac{-2\hbar^2|z|}{m\lambda}} \quad (12)$$

Let u = rand (0, 1), then Substitute Eq. (7) into Eq. (12) yields:

$$v = P \pm \frac{\hbar^2}{2m\lambda}\ln\left(\frac{1}{u}\right) \quad (13)$$

let $c = \frac{\hbar^2}{2m\lambda}$ then Eq. (13) becomes:

$$v_i = P_i \pm c_i \ln\left(\frac{1}{u}\right) \quad (14)$$

Let MB equals to the mean of best SN/2 positions:

$$MB = \frac{2}{n}\sum_{i=1}^{SN/2} p_i \quad (15)$$

$$\text{Let } \vec{\boldsymbol{\mu}} = [\mu_1, \mu_2, \ldots, \mu_{(\frac{SN}{2})}] \quad (16)$$

Where $\vec{\boldsymbol{\mu}}$ is vector of tuning factor in decreasing order.

Where: $\mu_1 > \mu_2 > \mu_3,,, > \ \mu_{(N)}$

To change the variance of the distribution so that the algorithm can achieve the optimization task effectively let:

$$c_i = \mu_i * (|MB - x_i|) \quad (17)$$

$\vec{\boldsymbol{\mu}}$ is multiplied by $(|\overrightarrow{MB} - \vec{X}|)$ in element wise manner

Substituting c_i in Eq. (14) yields:

$$v_i = \begin{cases} p_i + \mu_i * (|MB - x_i|)\ln\left(\frac{1}{u}\right) & for\ k \geq 0.5 \\ p_i - \mu_i * (|MB - x_i|)\ln\left(\frac{1}{u}\right) & for\ k < 0.5 \end{cases} \quad (18)$$

where k is uniform random variable between 0 and 1.

4. Quantum Artificial Bee Colony (QABC)

It can be seen from Eq. (18) that the high best fitness motivates exploitation since the variance of the distribution becomes smaller while the variance increases as the term $|MB - x_i|$ increses and this motivates the exploration. Further, a new parameter which controls how many dimension to be changed for each bee in each iteration, called modification rate (MR). Eq. (18) can be used in employed phase and in onlooker phase or in the onlooker phase only. The procedure for implementing the QABC is given by the following steps,

Step 1: The user must choose the key parameters that control the QABC, namely population size of particles, boundary constraints of optimization variables, modification rate (*MR)*, tuning vector ($\vec{\boldsymbol{\mu}}$) and the stop criterion (tmax).

Step 2: Initialize a population (array) of food sources with random positions in the n-dimensional problem space using a uniform probability distribution function.

Step 3: Evaluate the fitness value of each food source.

Step 4: Update the positions of the employed bees using Eq. (2) then evaluate their fitness again.

Step 5: Using greedy selection to choose the best position between the current and the updated position.

Step 6: Distribute the onlookers on the food sources according to their fitness using wheel selection rule.

Step 7: Generate a new position for each onlooker using Eq. (18) then evaluate the fitness.

Step 8: Apply the greedy selection to choose the best position between the current food source position and the updated position found by the onlookers.

Step 9: In the scout phase, replace the abandoned food source by randomly generated position.

Step 10: Repeat 4-9 until the stopping condition is satisfied.

5. Simulation Results and Discussions

5.1 Benchmark Functions

A function is multimodal if it has two or more local optima. A function of variables is separable if it can be rewritten as a sum of functions of just one variable [8]. Nonseparable functions are difficult and the problem is even more difficult if the function is also multimodal. The algorithm must be able to avoid the regions around local minima in order to cover all and converge to the global minima. The most complex case appears when the local optima are randomly distributed in the search space. The dimensionality of the search space is another important factor in the complexity of the problem. In general, the complexity increases with the dimensionality.

Some of the optimization benchmark functions were used to validate and compare the performance of the QABC algorithm. The used functions have diverse properties; therefore, they are useful to test the proposed QABC algorithm without biasing. The functions are as follows:

5.1.1 Griewank function

It is a multimodal, continuous, differentiable and non-separable function. Its global minimum value is 0 and located at $\vec{X} = [0,0,0,..0]$ .The range of the decision variables is [-600,600]. Griewank function is highly unimodal when the dimensionality is less than 30 [12].

$$f_1(\vec{X}) = \frac{1}{4000}\left(\sum_{i=1}^{D} x_i^2\right) - \left(\prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1 \quad (19)$$

5.1.2 Rastrigin function

It is a multimodal, continuous, differentiable and separable function. The global minimum of the function is 0 and located at $\vec{X} = [0,0,0,..0]$.the initialization range of the decision variables is [-15, 15] [12].

$$f_2(\vec{X}) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10) \quad (20)$$

The cosine term generates many local minima. The minima are uniformly distributed.

5.1.3 Rosenbrock function

It is a unimodal, continuous, differentiable and non-separable function [12].

$$f_3(\vec{X}) = \sum_{i=1}^{D-1}(100(x_{i+1} - x_i{}^2)^2 + (x_i - 1)^2) \quad (21)$$

Subject to $-15 \leq x_i \leq 15$.

The global minimum of Rosenbrock function is located inside a narrow and long valley that has parabolic shape. Because it is hard to obtain the global minimum value, the variables are highly dependent, and the gradients usually do not direct towards global minimum, this function is used often to test the performance of the optimization algorithms.

5.1.4 Ackley function

It is a multimodal, continuous, differentiable and non-separable function whose global optimal value is 0 and located at $\vec{X} = [0,0,0,..0]$. The range of the decision variables is [−32.768, 32.768] [12].

$$f_4(\vec{X}) = 20 + e - 20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)} e^{\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)} \quad (22)$$

The exponential terms produces many local minima. Algorithms that depend on the gradient on their work can easily trapped in local minima. Hence, only the algorithm with good exploration and exploitation capability can find the global minimum value.

5.1.5 Schwefel function

It is a multimodal, continuous, differentiable and separable function whose global optimal value is 0 and located $\vec{X} = [420.9867, 420.9867, .., 420.9867]$. The range of the decision variables is [−500, 500] [12].

$$f_5(\vec{X}) = D * 418.9829 + \sum_{i=1}^{D} -x_i \sin(\sqrt{|x_i|}) \quad (23)$$

The function has many local maxima and local minima. Because the location of the global minimum value near the boundary and the existence of the second best minimum value far from the global minimum, this problem is difficult for algorithm with poor exploration mechanism.

5.1.6 Colville Function

This function is multimodal continuous and differentiable whose global optimal value is 0 and located t $\vec{X} = [1,1 \ldots, 1]$. The range of the decision variables is $[-10, 10]$ [13].

$$\begin{aligned} f_6(\vec{X}) = {} & 100(x_1 - x_2^2)^2 + (1 - x_1)^2 + \\ & 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + \\ & 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + \\ & 19.8(x_2 - 1)(x_4 - 1) \end{aligned} \quad (24)$$

5.1.7 Zakharov Function

It is multimodal, non-separable continuous and differentiable function. The global minimum of the function is 0 and located at $\vec{X} = [0,0,0,..0]$.the initialization range of the decision variables is [-5, 10] [13].

$$f_7(\vec{X}) = \sum_{i=1}^{D} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{D} i x_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{D} i x_i\right)^4 \quad (25)$$

5.1.8 Powell Singular Function

It is unimodal, continuous, differentiable and non-separable function whose global optimal value is 0 and located t $\vec{X} = [3, -1, 0, 1 \ldots 3, -1, 0, 1]$. The range of the decision variables is [−4, 5] [13].

$$f_8(\vec{X}) = \sum_{i=1}^{D-2}(x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4 \quad (26)$$

The mean function and the standard deviation values of the minimum solutions found by the QABC algorithm in different dimensions have been computed. The mean and the standard deviations of the function values obtained by the QABC are compared to the results obtained by other algorithms in [7] and [12] as shown in Table 1.

5.2 Simulation Parameters

The following simulation parameters are adopted for QABC:

- The size of colony (SN) = 6.
- Number of Employed bees = Number of Onlooker bees SN/2.
- The maximum number of cycles for foraging = 2000.
- Runtime = 100
- Limit = 70
- MR = 0.1
- $\vec{\mu}$ = varies linearly with the iteration from 1.2 to 0.8

Simulation setting for ABC, PSO, GABC are as in [7] and [12].

5.3 Simulation Results

The results of QABC with the simulation parameters discussed in section 5.2 are shown in Table 1. Table 1 shows the mean values and the standard deviation achieved by each algorithm for different functions and dimensions. Table 2 shows the significant of QABC results compared to the others. In Table 2, '+' indicates that the QABC is better than all the others, '=' indicates that QABC equals to at least one of the other algorithm and '-' indicates that QABC is worse than at least one of the other algorithm. QABC outperforms the others in 11 out of 18 cases as shown in Table 2. The results obtained demonstrate the superiority of QABC over ABC, PSO and GABC.

6. Conclusion

In this paper, a new modified version of ABC algorithm is introduced. The newly suggested strategy used in onlooker bee phase. Furthermore, the proposed algorithm modifies the update equation of the ABC in order to balance between the exploration and exploitation of the search mechanism. The bee in the QABC is considered to behave like a quantum particle in a delta potential well. Moreover, the modified strategy is applied to solve 8 well-known benchmark functions. The obtained result shows that the QABC is superior to the ABC, PSO and GABC. The proposed algorithm can be applied for optimization problems with different properties. Additionally, it can escape the local minima and converge to the global minima efficiently.

Table 1: Comparison of the Results obtained by QABC algorithm and other algorithms.

Fun.	D	ABC		PSO		GABC		QABC	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
F1	10	3.7196e-3	8.8334e-8	0.079393	0.033451	6.0279e-4	2.3313e-3	8.8334e-8	1.8151e-7
	20	3.8168e-3	8.8334e-8	0.30565	0.025419	6.9655e-4	2.2609-3	0	0
	30	1.1971e-2	1.9763e-2	0.011151	0.014209	1.0470e-3	2.7482e-3	0	0
F2	10	0	0	2.6559	1.3896	0	0	0	0
	20	1.4052e-11	4.0524e-11	12.059	3.3216	3.3165e-2	1.8165e-1	0	0
	30	0.45305	0.51495	32.476	6.9521	2.1733	1.0728	0	0
F3	10	1.1114	1.7952	4.3713	2.3811	1.6769	2.9037	0.0114638	0.0503656
	20	4.5509	4.8776	77.382	94.901	7.47961	19.092	0.00887047	0.0115089
	30	48.0307	46.65676	402.54	633.65	25.7164	31.75811	0.0041108	0.00390255
F4	10	2.8343e-9	2.5816e-9	9.8499E-13	9.6202E-13	2.7533e-14	3.35832e-15	0	0
	20	2.7591e-5	2.1321e-5	1.1778E-6	1.584E-6	7.7828e-10	2.9817e-10	0	0
	30	4.7018e-2	3.3957e-2	1.4917E-6	1.861E-6	1.11137e-4	3.8873e-5	0	0
F5	10	1.9519	1.3044	161.87	144.16	1.27E-09	4E-12	1.27E-09	4E-12
	20	7.285	2.9971	543.07	360.22	0.000255	0	0.0002546	0
	30	13.5346	4.9534	990.77	581.14	0.000382	1E-12	0.00038182	1E-12
F6	4	0.014938	0.007364	0	0	0.0929674	0.066277	0.001254	0.00125632
F7	10	0.013355	0.004532	0	0	0.0002476	0.000183	0	0
F8	24	9.703771	1.547983	0.00011004	0.000160	0.0031344	0.000503	0.000376	0.002264

Table 2: Summary of the results.

Fun.	F1			F2			F3			F4			F5			F6	F7	F8
D	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	4	10	24
QABC Performance	+	+	+	=	+	+	+	+	+	+	+	+	=	=	=	-	=	-

7. REFERENCES

[1] X. S. Yang, Z. H. Cui, R. B. Xiao, A. H. Gandomi, and M. Karamanoglu, "Swarm Intelligence and Bio-Inspired Computation," Elsevier, Waltham, Mass, USA, 2013.

[2] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," *Artificial Life*, vol. 5, no. 2, pp.137-172, 1999.

[3] R. C. Eberhart, and J. Kennedy, "A New Optimizer using Particle Swarm Theory," *In Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, vol. 1, pp. 39-43. 1995.

[4] D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimization," *Technical Report*, Erciyes University, Engineering Faculty, Computer Engineering Department, pp. 1-10, 2005.

[5] K.V. Price, R. M. Storn, and J. A. Lampinen, "Differential Evolution: A Practical Approach to Global optimization," Springer-Verlag, Berlin, Germany, 2005.

[6] Z. Guopu, and S. Kwong, "Gbest-guided Artificial Bee Colony Algorithm for Numerical Function Optimization," *Applied Mathematics and Computation*, vol. 217, pp. 3166-3173, 2010.

[7] L. Guoqiang, P. Niu, and X. Xiao, "Development and Investigation of Efficient Artificial Bee Colony Algorithm for Numerical Function Optimization," *Applied Soft Computing*, vol. 12, no. 1, pp. 320-332, 2012.

[8] 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.

[9] J. Sun, B. Feng, and W. Xu, "Particle Swarm Optimization with Particles having Quantum Behavior," *Congress on Evolutionary Computation,* Portland OR, USA, vol. 1, pp. 325-331, 2004.

[10] J. Sun, W. Xu, B. Feng, "Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level," *IEEE International Conference on Systems, Man and Cybernetics*, Hawaii, USA, vol. 4, 2005.

[11] D. Griffiths, and E. G. Harris, "Introduction to Quantum Mechanics," Prentice Hall, New Jersey, USA, vol.2, 1995.

[12] D. Karaboga, and B. Akay, "A Comparative Study of Artificial Bee Colony Algorithm," *Applied Mathematics and Computation*, vol. 214, no.1, pp. 108-132, 2009.

[13] M. Jamil, and X. S. Yang, "A Literature Survey of Benchmark Functions for Gobal Optimization Problems," *International Journal of Mathematical Modeling and Numerical Optimization*, vol.4, no. 2, pp. 150-194, 2013.