International Journal of Electronics and Communication Engineering.
ISSN 0974-2166 Volume 7, Number 1 (2014), pp. 63–74

http://www.irphouse.com

Accelerated Artificial Bee Colony Algorithm for Parameter Estimation of Frequency-modulated Sound Waves

Bharti Gothania
Jaipur Engineering College, Kukas, Jaipur
E-mail: gothania_bharti@yahoo.com

Garima Mathur and R.P. Yadav
EC Deptt., M.N.I.T, Jaipur
E-mail: garimamathurjec@gmail.com,rp_yadav@yahoo.com

Abstract

Artificial Bee Colony (ABC) optimization algorithm is a swarm intelligence based nature inspired algorithm which has been proved a competitive algorithm with some popular nature-inspired algorithms. However, it is found that the ABC algorithm prefers exploration at the cost of the exploitation. Therefore, in this paper a self adaptive fitness based position update strategy is presented. The proposed strategy is self-adaptive in nature and therefore no manual parameter setting is required. Further, the proposed strategy is used to solve a complex real world engineering optimization problems, namely, parameter estimation for frequency-modulated sound waves (FMSW).

AMS subject classification:
Keywords: Swarm intelligence, Self adaptive mutation, Engineering optimization problems, Artificial Bee Colony

1. Introduction

Artificial bee colony (ABC) optimization algorithm introduced by D.Karaboga [8] is a recent swarm intelligence based algorithm. This algorithm is inspired by the behavior of honey bees when seeking a quality food source. Like any other population based optimization algorithm, ABC consists of a population of possible solutions. The possible solutions are food sources of honey bees. The fitness is determined in terms of the quality (nectar amount) of the food source. ABC is relatively a simple, fast and population based stochastic search technique in the field of nature inspired algorithms.

There are two fundamental processes which drive the swarm to update in ABC: the variation process, which enables exploring different areas of the search space, and the selection process, which ensures the exploitation of the previous experience. However, it has been shown that the ABC may occasionally stop proceeding toward the global optimum even though the population has not converged to a local optimum [6]. It can be observed that the solution search equation of ABC algorithm is good at exploration but poor at exploitation [13]. Therefore, to maintain the proper balance between exploration and exploitation behavior of ABC, it is highly required to develop a strategy in which better solutions exploit the search space in close proximity while less fit solutions explore the search space. Therefore, in this paper, we proposed a self adaptive step size strategy to update a solution. In the proposed strategy, a solution takes small step sizes in position updating process if its fitness is high i.e. it searches the solution in its vicinity whereas a solution takes large step sizes if its fitness is low, hence explore the search space. The proposed strategy is used for finding the global optima of a unimodal and/or multimodal functions by adaptively modifying the step sizes in updating process of the candidate solution in the search space within which the optima is known to exist. In the proposed strategy, ABC algorithm's parameter '*limit*' is modified self adaptively based on the fitness of the solution. Now, there is separate '*limit*' for every solution according to the fitness. For high fitness solutions, value of '*limit*' is high while for less fit solutions, it is low. Hence, a better solution have more time to update itself in comparison to the less fit solutions. Further, to improve the diversity of the algorithm, number of scout bees are increased. The proposed strategy is compared with ABC.

Rest of the paper is organized as follows: Basic ABC is explained in section 2. In section 3, fitness based position update in ABC is proposed and explained. In Section 4, performance of the proposed strategy is analyzed. A complex real world engineering optimization problems, namely, parameter estimation for frequency-modulated sound waves (FMSW) is solved in section 5. Finally, in section 6, paper is concluded.

2. Artificial Bee Colony(ABC) algorithm

In ABC, honey bees are classified into three groups namely employed bees, onlooker bees and scout bees. The number of employed bees are equal to the onlooker bees. The employed bees are the bees which searches the food source and gather the information about the quality of the food source. Onlooker bees stay in the hive and search the food sources on the basis of the information gathered by the employed bees. The scout bee, searches new food sources randomly in places of the abandoned foods sources. Similar to the other population-based algorithms, ABC solution search process is an iterative process. After, initialization of the ABC parameters and swarm, it requires the repetitive iterations of the three phases namely employed bee phase, onlooker bee phase and scout bee phase. Each of the phase is described as follows:

2.1. Initialization of the swarm

The parameters for the ABC are the number of food sources, the number of trials after which a food source is considered to be abandoned and the termination criteria. In the basic ABC, the number of food sources are equal to the employed bees or onlooker bees. Initially, a uniformly distributed initial swarm of SN food sources where each food source $x_i (i = 1, 2, \ldots, SN)$ is a D-dimensional vector, generated. Here D is the number of variables in the optimization problem and x_i represent the i^{th} food source in the swarm. Each food source is generated as follows:

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}) \tag{1}$$

here x_{minj} and x_{maxj} are bounds of x_i in j^{th} direction and $rand[0, 1]$ is a uniformly distributed random number in the range [0, 1].

2.2. Employed bee phase

In the employed bee phase, modification in the current solution (food source) is done by employed bees according to the information of individual experience and the new solution fitness value. If the fitness value of the new solution is greater than that of the old solution, then the bee updates her position to the new solution and old one is dicarded. The position update equation for i^{th} candidate in this phase is

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

here $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indices. k must be different from i. ϕ_{ij} is a random number between $[-1, 1]$.

2.3. Onlooker bees phase

In this phase, the new fitness information (nectar) of the new solutions (food sources) and their position information is shared by all the employed bees with the onlooker bees in the hive. Onlooker bees analyze the available information and selects a solution with a probability $prob_i$ related to its fitness, which can be calculated using following expression (there may be some other but must be a function of fitness):

$$prob_i(G) = \frac{0.9 \times fitness_i}{maxfit} + 0.1, \tag{3}$$

here $fitness_i$ is the fitness value of the i^{th} solution and $maxfit$ is the maximum fitness of the solutions. As in the case of employed bee, it produces a modification on the position in its memory and checks the fitness of the new solution. If the fitness is higher than the previous one, the bee memorizes the new position and forgets the old one.

2.4. Scout bees phase

A food source is considered to be abandoned, if its position is not getting updated during a predetermined number of cycles. In this phase, the bee whose food source has been abandoned becomes scout bee and the abandoned food source is replaced by a randomly chosen food source within the search space. In ABC, predetermined number of cycles is a crucial control parameter which is called $limit$ for abandonment.

Assume that the abandoned source is x_i. The scout bee replaces this food source by a randomly chosen food source which is generated as follows:

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}), \text{ for } j \in \{1, 2, \ldots, D\} \tag{4}$$

where x_{minj} and x_{maxj} are bounds of x_i in j^{th} direction.

2.5. Main steps of the ABC algorithm

Based on the above explanation, it is clear that the ABC search process contains three important control parameters: The number of food sources SN (equal to number of onlooker or employed bees), the value of $limit$ and the maximum number of iterations. The pseudo-code of the ABC is shown in Algorithm 1 [6].

Algorithm 1 Artificial Bee Colony Algorithm:

```
Initialize the parameters;
while Termination criteria is not satisfied do
   Step 1: Employed bee phase for generating new food sources;
   Step 2: Onlooker bees phase for updating the food sources depending on their nectar
   amounts;
   Step 3: Scout bee phase for discovering the new food sources in place of abandoned food
   sources;
   Step 4: Memorize the best food source found so far;
end while
Output the best solution found so far.
```

3. Accelerating Artificial Bee Colony (AABC)

Exploration and exploitation are the two important characteristics of the population-based optimization algorithms such as GA [5], PSO [9], DE [11], BFO [10] and so on. In these optimization algorithms, the exploration represents the ability to discover the global optimum by investigating the various unknown regions in the solution search space. While, the exploitation represents the ability to find better solutions by implementing the knowledge of the previous good solutions. In behavior, the exploration and exploitation contradict with each other, however both abilities should be well balanced to achieve better optimization performance. Dervis Karaboga and Bahriye Akay [6] tested different variants of ABC for global optimization and found that the ABC shows poor performance and remains inefficient in exploring the search space. In ABC, any potential solution updates itself using the information provided by a randomly selected potential solution within the current swarm. In this process, a step size which is a linear combination of a random number $\phi_{ij} \in [-1, 1]$, current solution and a randomly selected solution are used. Now the quality of the updated solution highly depends upon this step size. If the step size is too large, which may occur if the difference of current solution and randomly selected solution is large with high absolute value of ϕ_{ij}, then updated solution can surpass the true solution and if this step size is too small then the convergence rate of ABC may significantly decrease. A proper balance of this step size can balance the exploration and exploitation capability of the ABC simultaneously. But, since this step size consists of random component so the balance can not be done manually. Therefore, to balance the exploration and exploitation, we modified the solution update strategy according to the fitness of the solution. In the basic ABC, the food sources are updated, as shown in equation (2). In order to improve the exploitation, take advantage of the information of the global best solution to guide the search of candidate solutions, the solution search equation described by equation (2) is modified as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + (2.0 - prob_i) \times (x_{bestj} - x_{ij}),$$

To enhance the exploitation capability of ABC, fitness based self adaptive mutation mechanism is introduced in the basic ABC and shown in Algorithm 2. In the proposed strategy, the perturbation in the solution is based on the fitness of the solution. It is clear from Algorithm 2 that the number of update in the dimensions of the i^{th} solution is depend on $prob_i$ and which is a function of fitness (refer equation 3). The strategy is based on the concept that the perturbation will be high for low fit solutions as for that the value of $prob_i$ will be low while the perturbation in high fit solutions will be low due to high value of $prob_i$. It is assumed that the global optima should be near about to the better fit solutions and if perturbation of better solutions will be high then there may be chance of skipping true solutions due to large step size. Therefore, the step sizes which are proportionally related to the perturbations in the solutions are less for good solutions and are high for worst solutions which are responsible for the exploration. Therefore in the proposed strategy, the better solutions exploit the search space while low fit solutions explore the search area.

Algorithm 2 Solution update in Employed bee phase:

Input: solution x_i, $prob_i$ and $j \in (1, D)$;
for $j \in \{1 \text{ to } D\}$ **do**
  if $U(0, 1) > \frac{prob_i}{2}$ **then**
    $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + (2.0 - prob_i) \times (x_{bestj} - x_{ij}),$
  else
    $v_{ij} = x_{ij};$
  end if
end for

Here, $prob_i$ is a function of fitness and calculated as shown in equation (3). In Algorithm 2, it is clear that for a solution if value of $prob_i$ is high and that is the case of high fitness solution then for that solution the step size will be small. Therefore, it is obvious that there is more chance for the high fitness solution to move in its neighborhood compare to the low fitness solution and hence, a better solution could exploit the search area in its vicinity. In other words, we can say that solutions exploit or explore the search area based on probability which is function of fitness.

4. Experimental results and discussion

4.1. Test problems under consideration

In order to analyze the performance of $AABC$, 16 unbiased optimization problems (solutions does not exists on axis, diagonal or origan) (f_1 to f_{16}) are selected (listed in Table 1).

Table 1: Test problems

Test Problem	Objective function	Search Range	Optimum Value	D	Acceptable Error
Beale function	$f_1(x)=[1.5-x_1(1-x_2)]^2+[2.25-x_1(1-x_2^2)]^2+[2.625-x_1(1-x_2^3)]^2$	[-4.5,4.5]	$f(3,0.5)=0$	2	$1.0E-05$
Colville function	$f_2(x)=100[x_2-x_1^2]^2+(1-x_1)^2+90(x_4-x_3^2)^2+(1-x_3)^2+10.1[(x_2-1)^2+(x_4-1)^2]+19.8(x_2-1)(x_4-1)$	[-10,10]	$f(\vec{1})=0$	4	$1.0E-05$
Kowalik function	$f_3(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	[-5,5]	$f(0.1928, 0.1908, 0.1231, 0.1357)=3.07E-04$	4	$1.0E-05$
Shifted Rosenbrock	$f_4(x)=\sum_{i=1}^{D-1}(100(z_i^2-z_{i+1})^2+(z_i-1)^2)+f_{bias}, z=x-o+1, x=[x_1,x_2,\ldots,x_D], o=[o_1,o_2,\ldots,o_D]$	[-100, 100]	$f(o)=f_{bias}=390$	10	$1.0E-01$
Shifted Sphere	$f_5(x)=\sum_{i=1}^{D}z_i^2+f_{bias}, z=x-o, x=[x_1,x_2,\ldots,x_D], o=[o_1,o_2,\ldots,o_D]$	[-100,100]	$f(o)=f_{bias}=-450$	10	$1.0E-05$
Shifted Rastrigin	$f_6(x)=\sum_{i=1}^{D}(z_i^2-10\cos(2\pi z_i)+10)+f_{bias}$ z=(x-o), x=$(x_1,x_2,\ldots,x_D), o=(o_1,o_2,\ldots,o_D)$	[-5,5]	$f(o)=f_{bias}=-330$	10	$1.0E-02$
Shifted Schwefel	$f_7(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}z_j)^2+f_{bias}, z=x-o, x=[x_1,x_2,\ldots,x_D], o=[o_1,o_2,\ldots,o_D]$	[-100,100]	$f(o)=f_{bias}=-450$	10	$1.0E-05$
Shifted Griewank	$f_8(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}})+1+f_{bias}, z=(x-o), x=[x_1,x_2,\ldots,x_D], o=[o_1,o_2,\ldots,o_D]$	[-600,600]	$f(o)=f_{bias}=-180$	10	$1.0E-05$
Shifted Ackley	$f_9(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}z_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i))+20+e+f_{bias}, z=(x-o), x=(x_1,x_2,\ldots,x_D), o=(o_1,o_2,\ldots,o_D)$	[-32,32]	$f(o)=f_{bias}=-140$	10	$1.0E-05$
Goldstein-Price	$f_{10}(x)=(1+(x_1+x_2+1)^2\cdot(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2))\cdot(30+(2x_1-3x_2)^2\cdot(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$	[-2,2]	$f(0,-1)=3$	2	$1.0E-14$
Easom's function	$f_{11}(x)=-\cos x_1\cos x_2 e^{((-(x_1-\Pi)^2-(x_2-\Pi)^2))}$	[-10,10]	$f(\pi,\pi)=-1$	2	$1.0E-13$
Dekkers and Aarts	$f_{12}(x)=10^5x_1^2+x_2^2-(x_1^2+x_2^2)^2+10^{-5}(x_1^2+x_2^2)^4$	[-20,20]	$f(0,15)=f(0,-15)=-24777$	2	$5.0E-01$
McCormick	$f_{13}(x)=\sin(x_1+x_2)+(x_1-x_2)^2-\frac{3}{2}x_1+\frac{5}{2}x_2+1$	$-1.5\leq x_1\leq 4, -3\leq x_2\leq 3$	$f(-0.547, -1.547)=-1.9133$	30	$1.0E-04$
Meyer and Roth Problem	$f_{14}(x)=\sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i}-y_i)^2$	[-10, 10]	$f(3.13,15.16,0.78)=0.4E-04$	3	$1.0E-03$

to be cont'd on next page

Table 1: Test problems (Cont.)

Test Problem	Objective function	Search Range	Optimum Value	D	Acceptable Error
Shubert	$f_{15}(x) = -\sum_{i=1}^{5} i \cos((i+1)x_1+1) \sum_{i=1}^{5} i \cos((i+1)x_2+1)$	[-10, 10]	$f(7.0835, 4.8580) = -186.7309$	2	$1.0E-05$
Sinusoidal	$f_{16}(x) = -[A \prod_{i=1}^{n} \sin(x_i - z) + \prod_{i=1}^{n} sin(B(x_i - z))]$, $A = 2.5, B = 5, z = 30$	[0, 180]	$f(\bar{90} + z) = -(A+1)$	10	1.00E-02

4.2. Experimental setting

To prove the efficiency of $AABC$, it is compared with ABC. To test $AABC$ and ABC over considered problems, following experimental setting is adopted:

- Colony size $NP = 50$ [3, 4],
- $\phi_{ij} = rand[-1, 1]$,
- Number of food sources $SN = NP/2$,
- $limit = D \times SN$ [1, 8],
- The stopping criteria is either maximum number of function evaluations (which is set to be 200000) is reached or the acceptable error (mentioned in Table 1) has been achieved,
- The number of simulations/run =100,
- Parameter settings for the algorithms ABC is similar to its original research paper.

4.3. Results Comparison

Numerical results with experimental setting of subsection 4.2 are given in Table 2. In Table 2, standard deviation (SD), mean error (ME), average function evaluations (AFE), and success rate (SR) are reported. Table 2 shows that most of the time $AABC$ outperforms in terms of reliability, efficiency and accuracy as compare to the basic ABC.

Table 2: Comparison of the results of test problems

Test Function	**Algorithm**	**SD**	**ME**	**AFE**	**SR**
f_1	ABC	1.66E-06	8.64E-06	16520.09	100
	AABC	3.05E-06	5.03E-06	9314.71	100
f_2	ABC	1.03E-01	1.67E-01	199254.48	1
	AABC	1.71E-02	1.95E-02	151300.35	46
f_3	ABC	7.33E-05	1.76E-04	180578.91	18
	AABC	2.15E-05	8.68E-05	90834.53	97
f_4	ABC	1.05E+00	6.36E-01	176098.02	23
	AABC	1.60E-02	8.45E-02	99219.48	99
f_5	ABC	2.42E-06	7.16E-06	9013.5	100
	AABC	2.08E-06	6.83E-06	5585.5	100
f_6	ABC	1.21E+01	8.91E+01	200011.71	0
	AABC	9.24E+00	8.56E+01	200006.8	0
f_7	ABC	3.54E+03	1.11E+04	200029.02	0
	AABC	3.00E+03	1.08E+04	200016.04	0
f_8	ABC	2.21E-03	6.95E-04	61650.9	90
	AABC	7.35E-04	7.88E-05	38328.96	99
f_9	ABC	1.80E-06	7.90E-06	16767	100
	AABC	1.37E-06	8.31E-06	9366	100
f_{10}	ABC	5.16E-06	1.04E-06	109879.46	62
	AABC	4.37E-15	4.87E-15	3956.05	100
f_{11}	ABC	4.44E-05	1.60E-05	181447.91	17
	AABC	2.79E-14	4.02E-14	46909.7	100
f_{12}	ABC	5.33E-03	4.91E-01	1460.56	100
	AABC	5.40E-03	4.90E-01	792	100

to be cont’d on next page

Table 2: Comparison of the results of test problems (Cont.)

Test Function	Algorithm	SD	ME	AFE	SR
f_{13}	ABC	6.67E-06	8.92E-05	1166.5	100
	AABC	6.45E-06	8.79E-05	622	100
f_{14}	ABC	2.89E-06	1.94E-03	24476.88	100
	AABC	2.74E-06	1.95E-03	5127.73	100
f_{15}	ABC	5.34E-06	4.86E-06	4752.21	100
	AABC	5.72E-06	5.07E-06	2550.57	100
f_{16}	ABC	1.83E-03	7.77E-03	54159.26	99
	AABC	2.09E-03	7.87E-03	49230.85	100

$AABC$ and ABC are compared through SR, ME and AFE in Table 2. First SR is compared for all these algorithms and if it is not possible to distinguish the algorithms based on SR then comparison is made on the basis of AFE. ME is used for comparison if it is not possible on the basis of SR and AFE both. Outcome of this comparison is summarized in Table 3. In Table 3, '+' indicates that the $AABC$ is better than the considered algorithms and '-' indicates that the algorithm is not better or the difference is very small. The last row of Table 3, establishes the superiority of $AABC$ over ABC.

Table 3: Summary of Table 2 outcome

Function	AABC Vs ABC
f_1	+
f_2	+
f_3	+
f_4	+
f_5	+
f_6	=
f_7	=
f_8	+
f_9	+
f_{10}	+
f_{11}	+
f_{12}	+
f_{13}	+
f_{14}	+
f_{15}	+
f_{16}	+
Total number of + sign	14

For the purpose of comparison in terms of consolidated performance, boxplot analyses have been carried out for all the considered algorithms. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [12]. The boxplots for ABC and $AABC$ are shown in Figure 1. It is clear from this figure that $AABC$ is better than the basic ABC as interquartile range and median are comparatively low.

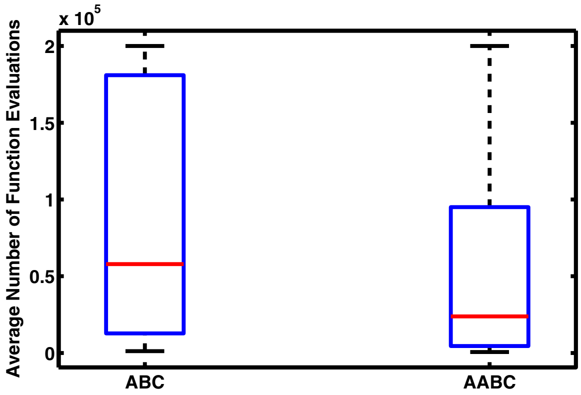

Figure 1: Boxplots graphs for average number of function evaluations

5. Applications of *AABC* to frequency-modulated (FM) sound waves

To see the robustness of the proposed strategy, a complex real world engineering optimization problems, namely, parameter estimation for frequency-modulated sound waves (FMSW) [2] is solved.

Frequency-Modulated (FM) sound wave synthesis has an important role in several modern music systems. The parameter optimization of an FM synthesizer is a six dimensional optimization problem where the vector to be optimized is $\vec{X} = \{a_1, w_1, a_2, w_2, a_3, w_3\}$ of the sound wave given in equation (5). The problem is to generate a sound (1) similar to target (2). This problem is a highly complex multimodal one having strong epistasis, with minimum value $f(\vec{X_{sol}}) = 0$. The expressions for the estimated sound and the target sound waves are given as:

$$y(t) = a_1 sin(w_1 t\theta + a_2 sin(w_2 t\theta + a_3 sin(w_3 t\theta))) \quad (5)$$

$$y_0(t) = (1.0)sin((5.0)t\theta - (1.5)sin((4.8)t\theta + (2.0)sin((4.9)t\theta))) \quad (6)$$

respectively where $\theta = 2\pi/100$ and the parameters are defined in the range [-6.4, 6.35]. The fitness function is the summation of square errors between the estimated wave (1) and the target wave (2) as follows:

$$E_2(\vec{X}) = \sum_{i=0}^{100} (y(t) - y_0(t))^2$$

Acceptable error for this problem is $1.0E-05$, i.e. an algorithm is considered successful if it finds the error less than acceptable error in a given number of generations.

Table 4 shows the experimental results. It is clear from Table 4 that the inclusion of fitness based updating strategy in the basic ABC performs better than the basic ABC algorithm.

Table 4: Comparison of the results of FMSW problem

Algorithm	Objective Value
AABC	0.366958
ABC	2.77254

6. Conclusion

In this paper, to improve the exploitation in ABC, a fitness based position update strategy is presented and incorporated with ABC. The so obtained modified ABC is named as Accelerating ABC (AABC). It is shown that, in the proposed strategy, better solutions exploits the search space in their neighborhood while less fit solutions explore the search area based on the fitness. The proposed algorithm is compared to ABC and with the help of experiments over test problems, it is shown that the $AABC$ outperforms to the considered algorithms in terms of reliability, efficiency and accuracy. Further, the proposed algorithm is used to solve a real world complex engineering optimization problem, namely parameter estimation of frequency modulated sound wave and performed better than the basic ABC algorithm.

References

[1] B. Akay and D. Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, doi:10.1016/j.ins.2010.07.015, 2010.

[2] S. Das and PN Suganthan. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. *Jadavpur University, Kolkata, India, and Nangyang Technological University, Singapore, Tech. Rep*, 2010.

[3] K. Diwold, A. Aderhold, A. Scheidler, and M. Middendorf. Performance evaluation of artificial bee colony optimization and new selection schemes. *Memetic Computing*, pages 1–14, 2011.

[4] M. El-Abd. Performance assessment of foraging algorithms vs. evolutionary algorithms. *Information Sciences*, 182(1):243–263, 2011.

[5] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.

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

[7] D. Karaboga and B. Basturk. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *Foundations of Fuzzy Logic and Soft Computing*, pages 789–798, 2007.

[8] D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.

[9] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[10] K.M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *Control Systems Magazine, IEEE*, 22(3):52–67, 2002.

[11] R. Storn and K. Price. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[12] D.F. Williamson, R.A. Parker, and J.S. Kendrick. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 110(11):916, 1989.

[13] G. Zhu and S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7):3166–3173, 2010.