Research Journal of Applied Sciences 10 (6): 241-250, 2015
ISSN: 1815-932X

Enhanced Artificial Bee Colony Algorithm for Constrained Optimization Problems

Soudeh Babaeizadeh and Rohanin Ahmad
Department of Mathematical Sciences, Faculty of Science,
Universiti Teknologi Malaysia (UTM), 81310 Johor, Malaysia

Abstract: Artificial Bee Colony (ABC) algorithm is a relatively new Swarm Intelligence algorithm that has attracted great deal of attention from researchers in recent years with the advantage of less control parameters and strong global optimization ability. However, there is still an insufficiency in ABC regarding its solution search equation which is good at exploration but poor at exploitation. This drawback can be even more significant when constraints are also involved. To address this issue, an Enhanced ABC algorithm (EABC) is proposed for constrained optimization problems where two new solution search equations are introduced for employed bee and onlooker bee phases, respectively. In addition, both Chaotic Search Method and opposition-based learning mechanism are employed to be used in population initialization in order to enhance the global convergence when producing initial population. This algorithm is tested on several benchmark functions where the numerical results demonstrate that the EABC is competitive with state of the art constrained ABC algorithm.

Key words: Artificial bee colony, constrained optimization, swarm intelligence, search equation, strong global

INTRODUCTION

Global optimization deals with optimization problems that might have more than one local minimum. Therefore, finding global minimum out of a set of local minima solutions in a certain feasible region can be challenging. While these problems can even be more challenging when constraints are also involved. In general, algorithms for solving constrained optimization problems can be classified into two main categories: derivative-based methods and derivative-free methods. There have always been many real world problems with non-differentiable constraints and disjoint feasible domains. These difficulties can make it very challenging for derivative-based methods to find even a feasible solution, let alone an optimal solution.

Furthermore, if derivative-based methods can obtain solutions they are usually only locally optimal. Derivative-free methods in contrast utilize a population of individuals in a search domain. Moreover, they only use the evaluations of the objective function to direct the search. Therefore, they do not usually pose limitations related to derivative-based methods and they do not easily fall into local optima.

Population based algorithms as significant branch of derivative-free methods capture much attention in recent years in solving constrained optimization problems. The most prominent population based algorithms suggested in the literatures are Genetic Algorithm (GA) (Holland, 1975), Particle Swarm Optimization (PSO) (Kennedy, 2011), Ant Colony Optimization (ACO) (Dorigo and Blum, 2005), Differential Evaluation (DE) (Stanarevic *et al*., 2011) and Artificial Bee Colony Algorithm (ABC) (Karaboga, 2005) and so on. ABC is a relatively new population-based algorithm developed by Karaboga (2005) based on simulating the foraging behaviour of honey bee swarm. Numerical performance demonstrated that ABC algorithm iscompetitive to that of other population-based algorithms with an advantage of employing fewer control parameters and the need for fewer function evaluations to arrive at an optimal solution (Karaboga and Basturk, 2007; Karaboga and Akay, 2009, 2011). Due to its simplicity and ease of implementation, ABC has captured much attention and has been employed to solve many numerical as well as practical optimization problems since its inception (Gao *et al*., 2014; Aydin *et al*., 2014; Li *et al*., 2012; Xiang and An, 2013).

Among optimization problems, the ones tackled in this study are Constrained Optimization Problems (COPs) for Non-Linear Programming (NLP) which can be formulated as in the following problem:

$$\begin{aligned} &\min f(x) \\ &\text{s.t}\quad g_j(x) \le 0,\ j = 1, 2,..., m \\ &\qquad h_j(x) = 0,\ j = m+1,..., l \end{aligned} \quad (1)$$

Corresponding Author: Rohanin Ahmad, Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia (UTM), 81310 Johor, Malaysia

where, $x = [x_1, x_2,..., x_n] \in R^n$ is an n-dimensional decision vector and each x_1 is bounded by lower and upper bounds as $[x_{min}, x_{max}]$. The objective function f(x) is defined on S and is an n-dimensional search space in R^n.

In general, most of the optimization algorithms have been initially introduced to address unconstrained optimization problems. Therefore, constraint handling techniques are employed to direct the search towards the feasible regions of the search space. Constraint handling methods were categorized into four groups by Koziel and Michalewicz (1999):

- Methods based on penalty functions which penalize constraints in order to solve a constrained problem as an unconstrained one
- Methods based on reservation of feasible solutions by transforming infeasible solutions to feasible ones with some operators
- Methods that separate feasible and infeasible solutions
- Other hybrid methods

In this study an enhanced constrained ABC algorithm is proposed by employing two new search equations for employed bee and onlooker bee phases. Moreover, chaotic search mechanism and opposition-based learning method are applied to initialize population with the aim of preventing algorithm from getting stuck at local minima.

ARTIFICIAL BEE COLONY

ABC algorithm is a recently proposed population-based algorithm introduced by Karaboga (2005) for real parameter optimization. This algorithm emulates the foraging behaviour of honey bee colonies. This algorithm classifies the artificial bees into three groups, employed bees, onlooker bees and scout bees. Half of the colony includes employed bee and the other half of the colony consist of onlooker bees. In ABC, the position of food source denotes a possible solution to the optimization problem and the nectar amount of food source represents fitness value of the associated solution. The number of employed bees or the onlooker bees is equal to the number of Solutions (SN) in the population. At initialization step, ABC generates a randomly distributed initial population of SN solutions using following equation:

$$x_{i,j} = x_{min,j} + rand(0, 1)(x_{max,j} - x_{min,j}) \tag{2}$$

where each solution x_i, i = 1, 2, ..., SN is d-dimensional. vector for j = 1, 2, ..., d. In addition, $x_{min,j}$ and $x_{max,j}$ are the lower and upper bounds for the dimension j, respectively. These food sources are randomly assigned to SN number of employed bees and their fitness are evaluated. After initialization, the population of the solutions is subjected to repeat the search processes for employed bee, the onlooker bees and the scout bee phases. The process continues until algorithm reaches the Maximum Cycle Number (MCN). In employed bee phase each employed bees produces a modification on the solution X_i using Eq. 3:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{3}$$

where, $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., d\}$ are randomly chosen indexes and k has to be different from i. $\phi_{i,j}$ is a random number in the range [-1, 1]. After V_i is obtained its fitness values is evaluated and a greedy selection mechanism is applied comparing X_i and V_i. If the fitness value of the new solution V_i is less than the current solution X_i then, the solution V_i is replaced with the X_i, otherwise the current solution remains. After the employed bee phase, the solution information is transferred to the onlooker bee phase.

In onlookers bee phase a solution is chosen depend on the probability value p_i associated with that solution calculated using the following equation:

$$p_i = \frac{f_i}{\sum_{j=1}^{SN} f_i} \tag{4}$$

where, f_i is the fitness value of solution i. Once the onlooker has selected solution X_i a modification is done on the solution using Eq. 3. Then, fitness values of generated solutions are evaluated and similar to employed bees phase, greedy selection mechanism is employed. If new solution has better fitness value than current solution, the new solution remains in population and the old solution is removed.

In the scout bee phase, if solution X_i cannot be improved further through a predetermined number of cycles (limit), then that solution is abandoned and replaced with a new solution generated randomly in using Eq. 2. According to the abovementioned description, ABC main procedure can be summarized in Algorithm 1.

Algorithm 1 (original artificial bee colony algorithm):

```
Initialize the population of solution
Evaluate the initial population
cycle = 1
Repeat
  Employed bee phase
  Apply greedy selection process
  Calculate the probability values for
i = 1, 2, ..., SN
  Onlooker bee phase
  Scout bee phase
  Memorize the best solution achieved so far
  cycle = cycle+1
until cycle = maximum cycle number
```

CONSTRAINED ARTIFICIAL BEE COLONY

ABC algorithm has been originally proposed to deal with unconstrained optimization problems (Karaboga, 2005). This algorithm is then adapted to tackle constrained optimization problems. The presence of various constraints and interferences between constraints makes COPs more difficult to tackle than unconstrained optimization problems. In this study, we present the available constrained ABC algorithms in the literature.

ABC algorithm for the first time was adapted by Karaboga and Basturk (2007) to solve constrained optimization problems. In this algorithm to cope with constraints, Deb's mechanism (Deb, 2000) is employed to be used instead of the greedy selection process due to its simplicity, computational cost and fine tuning requirement over other constraint handling methods. Because initialization with feasible solutions is very time consuming and in some situation impossible to generate a feasible solution randomly, the constrained ABC algorithm does not consider the initial population to be feasible. As an alternative Deb's rules are employed to direct the solutions to feasible region of search space. In addition, scout bee phase of the algorithm provides a diversity mechanism that allows new and probably infeasible individuals to be in the population. In this algorithm, artificial scouts are produced at a Scout Predetermined Period (SPP) of cycles for generating new solution randomly. The numerical performance of proposed ABC algorithm is evaluated and compared with the constrained PSO and DE algorithms and results show that ABC algorithm can be effectively applied for solving constrained optimization problems.

Mezura-Montes, Dami'an-Araoz and Cetina-Dom'ingez (Mezura-Montes *et al*., 2010) presented Smart Flight ABC (SF-ABC) algorithm to improve the performance of constrained ABC. In this algorithm to direct search towards the best-so-far solution, smart flight operator is applied in scout bee phase instead of uniform random search in ABC (Karaboga and Basturk, 2007). Based on this method, if the best solution is infeasible, the trial solution has the chance to be located near the boundaries of the feasible region of search space. However, if the best solution is infeasible, the smart flight will generate a solution in promising region of search space. In addition to aforementioned improvement on ABC, the combination of two dynamic tolerances are also applied in SF-ABC as constrained handling mechanism to transform the original CNOP into unconstrained optimization. The numerical results demonstrate the competitive performance of SF-ABC with original ABC.

Another modification on ABC algorithm was introduced by Karaboga and Akay (2011). What makes this algorithm different from the original ABC (Karaboga and Basturk, 2007) is related with the probability selection mechanism and parameter setting process. In this algorithm, a new probability selection mechanism is presented to enhance diversity by allowing infeasible solutions in the population where infeasible solutions are introduced inversely proportional to their constraint violations and feasible solution defined based on their fitness values. In addition, in this algorithm appropriate value for each parameter is obtained. To recognize this algorithm through this study, the abbreviation MABC is used to refer to this algorithm.

Modified constrained ABC by applying multiple onlooker bees (MO-ABC) was developed by Subotic (2011) to improve constrained ABC (Karaboga and Basturk, 2007). The numerical performance demonstrates comparative results with original ABC.

A modified ABC (M-ABC) introduced four modifications related with the selection mechanism, the equality and boundary constraints and scout bee operators to improve the behaviour of ABC in constrained search space. The numerical results show that M-ABC provides comparable results with respect to algorithms under comparison (Mezura-Montes and Cetina-Dominguez, 2012).

A genetically inspired ABC algorithm (GI-ABC) was presented for COP. In this algorithm uniform crossover and mutation operators from GA are applied to scout bee phase to improve the performance of ABC algorithm (Bacanin and Tuba, 2012).

Stanarevic *et al*. (2011) introduced a modified ABC algorithm in a form of smart bee (SB-ABC) to solve constrained problems which applies its historical memories for the solution. The numerical experiments show efficiency of the method.

ABC-BA is a hybrid algorithm presented by Tsai (2014) and Stanarevic *et al*. (2011) that integrates ABC and Bee Algorithm (BA). In this algorithm, individuals can perform as an ABC individual in ABC sub-swarm or a BA individual in the BA sub-swarm. In addition, the population size of the ABC and BA sub-swarms change stochastically based on current best fitness values achieved by the sub-swarms. Experimental results demonstrate that ABC-BA outperforms ABC and BA algorithm.

Constrained ABC algorithm was also applied to solve many real-world engineering problems in recent years. Brajevic *et al*. (2011) proposed a Constrained Artificial Bee Colony (SC-ABC). This method is tested on several engineering benchmark problems which contain discrete and continuous variables. The numerical results were then compared with results obtained from Simple Constrained Particle Swarm Optimization Algorithm (SiC-PSO) which show very good performance. Akay and Karaboga (2012) used ABC to solve large scale optimization problems as well as engineering design problems. The numerical results show that the performance of ABC algorithm is comparable to those of state of the art algorithms under consideration. Upgraded Artificial Bee Colony (UABC) algorithm for constrained optimization problems was presented by Brajevic and Tuba (2013) to improve modification rate parameter and applying modified scout bee phase of the ABC algorithm. This algorithm was tested on several engineering benchmark problems and the performance was compared with the performance of the Akay and Karaboga algorithm (Akay and Karaboga, 2012). The numerical results show that the proposed algorithm produces better results.

ENHANCED ARTIFICIAL BEE COLONY ALGORITHM

According to the literature in most of the constrained ABC algorithms, the role of population initialization is ignored. However, in order to have powerful algorithm initial solutions must be generated uniformly within the search space. The uniformly distributed initial solutions help to generate at least some points in the neighbourhood of global solution. However, by applying chaotic method finally global solution can be founded. In this study, we employed both chaotic mechanism and opposition-based learning method into population initialization to enhance diversity. Among available chaotic method, logistic is selected to be used in initialization step which can be equation as:

$$c_{k+1} = \alpha(1-c_k) \tag{5}$$

where c_k is the kth chaotic number, $c\in(0, 1)$ under the conditions that the initial $c_0\in(0, 1)$ and c_0 cannot get numbers from set $\{0.0, 0.25, 0.75, 0.5, 1.0\}$. The parameter is set as 4. The initialization process based on chaotic search mechanism and opposition learning method is coded in Algorithm 2.

Algorithm 2 (initialization approach):

```
Consider the maximum number of chaotic iteration
K = 300, the population
size SN and the counter i = 1, j = 1
for i = 1 to SN/2
  for j = 1 to d
    Randomly initialize variables c_{0,j}∈(0, 1) and set
iteration counter k = 0
    for k = 1 to K
      c_{k+1,j} = α(1-c_{kj})
    end
   x_{i,j} = x_{min,j}+x_{max,j}-x_{min,j}
    end
end
Set the individual counter i = 1 and j = 1
for i = SN/2 to SN
  for j = 1 to d
   OP_{i,j} = x_{min,j}+x_{max,j}-x_{min,j}
  end
end
Select SN individuals from the set P(SN) ∪O P(SN) as
initial population
```

After initialization the main loop consists of employed bees, onlooker bees and scout bees is subjected to repeated until the stopping criterion is met. In this algorithm the new search equation is proposed for employed bee phase using Eq. 6 to improve the exploitation behaviour of ABC:

$$v_{ij} = \begin{cases} x_{ij}+\gamma_{ij}(x_{ij}-x_{r_1j})+\mu_{ij}(x_{r_1j}-x_{r_2j}) & \text{if } R_j<MR \\ x_{ij} & \text{otherwise} \end{cases} \tag{6}$$

Where:

r_1, r_2 = Two different random integer indices selected from {1, 2, ..., SN}
γ_{ij} = A random number between [-1, 1]
μ_{ij} = Random number between [0, 1]
R_j = Uniformly distributed random number and MR is control parameter in range [0, 1]

In addition, x_{best} is the best solution found so far. In Eq. 6, the second and third terms enhance exploration capability.

Algorithm 3 (employed bee phase of EABC algorithm):

```
for i = 1: SN
  for j = 1: d
    Produce the new solution V_i for employed bee using Eq. 6
  end for
If no parameter is changed, change one random parameter of the solution
using
Evaluate the quality of V_i
Apply Deb's mechanism to select between V_i and X_i
If solution X_i does not improved trail_i = trail_i+1,
otherwise trail_i = 0
End if
```

After producing a new solution, EABC algorithm makes a selection using Deb's mechanism (Karaboga and Akay, 2009) instead of using greedy selection in unconstrained ABC. Applying Deb 's rules,the bee either memorizes the new solution by forgetting the current solution or keeps the current solution.

Deb's method uses a tournament selection mechanism where two solutions are compared at a time by applying following rules:

- Any feasible solution is preferred to any infeasible solution
- Among two feasible solutions, the one having better objective function value is preferred
- Among two infeasible solutions, the one having smaller constraint violation is preferred

After completion of the search by all employed bees they share the information of the solutions with the onlooker bees. In this probability selection mechanism (Liang *et al*., 2006) infeasible solutions are also allowed to participate in the colony. The probability values of feasible solutions are between 0.5 and 1 and for infeasible solution between 0 and 0.5. The probability method is defined as Eq. 7:

$$p_i = \begin{cases} 0.5 + \left(\dfrac{\text{Fitness}_i}{\sum_{j=1}^{SN} \text{Fitness}_j} \right) \times 0.5 & \text{if solution is feasible} \\ \left(1 - \dfrac{\text{Violation}_i}{\sum_{j=1}^{SN} \text{Violation}_j} \right) \times 0.5 & \text{if solution is in feasible} \end{cases} \quad (7)$$

Based on the probability selection mechanism, solutions are selected proportional to their fitness values if solutions are feasible and inversely proportional to their constraint violation values if solutions are infeasible. This process is simulated by the procedure given in Algorithm 4.

After receiving fitness values information from employed bees, onlooker bee selects a solution based on their probability values. Then, onlooker bees produce modification on the position of the selected solution using Eq. 8:

$$p_i = \begin{cases} x_{i1j} + \varphi_{ij}(x_{bestj} - x_{r2j}) + \Phi_{ij}(x_{bestj} - x_{r3j}) & \text{if } R_j < MR \\ x_{ij} & \text{otherwise} \end{cases} \quad (8)$$

Algorithm 4 (calculate probability for EABC algorithm):
for i = 1: SN
Calculate the probability values using Eq. 7
where violation_i is the constraint violation of solution
X_i and fitness_i is the fitness value of the solution X_i. The
fitness is defined by Eq. 9:

$$\text{Fitness}_i = \begin{cases} 1/(1+f_i) & \text{if } f_i \geq 0 \\ 1+\text{abs}(f_i) & \text{if } f_i < 0 \end{cases} \quad (9)$$

where f_i is the cost value of the solution X_i.
end for

where r_1, r_2 and r_3 are three different random integer indices selected from {1, 2, ..., SN}. $\varphi_{i,j}$ and Φ_{ij} is uniformly distributed random real number in the range [-1, 1].

As in the case of employed bees the Deb's rules are employed to compare current solution with new solution. If the new solution produces better result it remains in population and the old solution is removed Algorithm 5.

Algorithm 5 (onlooker bee phase for EABC algorithm):
t = 0, i = 1
repeat
if random < p_i then
t = t+1
for j = 1: d
Produce a new solution V_i for the onlooker bee of the solution X_i using Eq. 8
end for
Apply the selection process between V_i and X_i based on
Deb's Method
If solution X_i does not improve $\text{trail}_i = \text{trail}_i+1$,
otherwise $\text{trail}_i = 0$
end if
i = i+1
i = i mod (SN+1)
until t = SN

In Eq. 8, the first term improves the exploration ability and the second and third terms enhance exploitation capability.

After distribution of all onlooker bees, if a solution can not improve further through predetermined number of cycles (limit) is abandoned and replaced with a new solution discovered by scout bees. In EABC algorithm, a smart flight scout bee is proposed to enhance the exploitation ability of algorithm. Scout bee phase is defined as Eq. 10:

$$v_{ij} = x_{ij} + k_{ij}(x_{kj} - x_{ij}) - (1 - k_{ij})(x_{bestj} - x_{ij}) \quad (10)$$

Where:
k_{ij} = Uniformly real number in [-1, 1]
x_{bestj} = The best solution found so far

NUMERICAL EXPERIMENTS AND COMPARISONS

To evaluate and compare the performance of the proposed algorithms, 24 constrained benchmark functions form CEC 2006 (Stanarevic *et al*., 2011) are applied. MCABC and other constrained ABC algorithms under comparisons are coded in MALAB environment. The value of each parameters used are given in Table 1.

The numerical performance of proposed EABC algorithm was compared against constrained ABC (Karaboga and Basturk, 2007), MABC (Liang *et al*., 2006), M-ABC (Mezura-Montes *et al*., 2010), SF-ABC (Karaboga and Akay, 2011) and MO-ABC (Li *et al*., 2012) algorithms. Each algorithm are tested for 24 test function and after 30 independent runs of each algorithm the average solution is considered which can be seen in Table 2 and 3. The

Table 1: Parameters setting

Parameters	Symbols	Values
Solutions number	SN	20
Maximum cycle number	MCN	6000
Modification rate	MR	0.8
Population size	PS	40
Limit	Limit	150
Scout production period	SPP	150
Epsilon	ε	0.001

Table 2: Function values obtained by ABC, MABC, M-ABC, SF-ABC, MO-ABC and EABC

Problems	ABC	MABC	M -ABC	SF-ABC	MO-ABC	EABC
g01						
Best	-15.00000	-1500000	-15.00000	-15.00000	-15.00000	-15.00000
Mean	-15.00000	-15.00000	-15.00000	-14.16321	-15.00000	-15.00000
Worst	-15.00000	-15.00000	-15.00000	-12.52510	-15.00000	-15.00000
SD	0.000000	0.000000	0.000000	0.923125	0.000000	0.000000
g02						
Best	0.803567	0.803538	0.803614	-0.708944	-0.803610	-0.803618
Mean	-0.791744	-0.792927	-0.799450	-0.471249	-0.793510	-0.802736
Worst	-0.752924	-0.750302	-0.778176	-0.319535	-0.744582	-0.794656
SD	0.013292	0.011052	-0.006440	0.010832	0.016124	0.002685
g03						
Best	-1.004657	-1.004817	-1.000000	-1.000000	-1.000000	-1.005002
Mean	-1.000096	-1.001941	-1.000000	-1.000000	-1.000000	-1.004961
Worst	-0.979651	-0.989160	-1.000000	-1.000000	-1.000000	-1.004910
SD	0.005979	0.000375	0.0000	0.0000	0.0000	0.000026
g04						
Best	-30665.542	-30665.42	-30665.539	-30665.539	-30665.539	-30665.54
Mean	-30665.542	-30665.42	-30665.539	-30665.539	-30665.539	-30665.54
Worst	-30665.542	-30665.42	-30665.539	-30665.539	-30665.539	-30665.54
SD	0.0000000	0.0000000	0.00000	0.00000	0.00000	0.000000
g05						
Best	5126.489	5127.099	5126.734	5126.506	5126.657	5126.530
Mean	5177.239	5236.991	5178.178	5126.527	5162.506	5249.399
Worst	5307.988	5802.318	5317.183	5126.859	5229.119	5824.530
SD	57.86021	156.0343	56.000	0.0793	47.8203	202.4742
g06						
Best	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814
Mean	-6961.814	-6961.814	-6961.814	-6961.813	-6961.813	-6961.814
Worst	-6961.814	-6961.814	-6961.814	-6961.805	-6961.804	-6961.814
SD	0.0000000	0.0000000	0.000000	0.0002	0.0001	0.000000
g07						
Best	24.46138	24.47032	24.312235	24.16453	24.32325	24.31434
Mean	24.70718	24.68698	24.416402	24.65821	24.45653	24.38832
Worst	25.16577	25.36005	24.794032	25.55140	24.92938	24.70644
SD	0.181394	0.178641	0.12723	0.326021	0.135023	0.08267
g08						
Best	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504
Mean	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504
Worst	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504
SD	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000000
g09						
Best	680.6381	680.6371	680.6331	680.6325	680.6312	680.6324
Mean	680.6506	680.6515	680.6474	680.6450	680.6350	680.6493
Worst	680.6757	680.6760	680.6768	680.8584	680.6363	680.7374
SD	0.0080749	0.009610	0.054310	0.041251	0.004215	0.021527
g10						
Best	7160.63125	7220.5540	7051.7752	7049.5166	7053.3204	7117.8841
Mean	7364.94034	7347.8433	7233.8101	7116.8236	7167.8015	7447.8902
Worst	7691.30330	7924.1286	7604.1290	7362.7406	7418.3340	8034.5085
SD	129.8405	134.14103	101.325	82.12450	83.00825	236.67923
g11						
Best	0.7490003	0.7490001	0.7500000	0.7500000	0.7500000	0.7490000
Mean	0.7490022	0.7490032	0.7500000	0.7500000	0.7500000	0.7499839
Worst	0.7490101	0.7490140	0.7500000	0.7500000	0.7500000	0.7529146
SD	0.0000020	0.000003	0.000000	0.000000	0.000000	0.00110317
g12						
Best	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Mean	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Worst	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
SD	0.0000000	0.000000	0.000000	0.000000	0.000000	0.0000000
g13						
Best	0.5551238	0.4895965	0.0538901	0.0539860	0.4542041	0.184749
Mean	0.9497812	0.9576896	0.1577912	0.2638542	0.4560438	0.7331314
Worst	1.4929540	1.4375342	0.4419785	1.000000	0.4891204	1.000000
SD	0.1469151	0.1613582	0.0172430	0.2162045	0.0215840	0.232139

Table 3: Function values obtained by ABC, MABC, M-ABC, SF-ABC, MO-ABC and EABC

Problems	ABC	MABC	M-ABC	SF-ABC	MO-ABC	EABC
g14						
Best	-45.11878	-45.32082	-47.64541	-46.66514	-46.450835	-46.06784
Mean	-42.68215	-42.65421	-47.27156	-46.46824	-45.998013	-43.94703
Worst	-40.60165	-40.05962	-46.53698	-43.87123	-45.316798	-41.59535
SD	1.171236	1.195831	0.245762	0.520124	0.257	0.9756053
g15						
Best	941.21911	951.43752	961.71521	961.71511	961.71512	954.23572
Mean	958.84762	960.89221	961.71879	961.71553	961.88313	966.58910
Worst	972.95780	970.68460	961.79125	961.72013	964.33983	978.00521
SD	7.512742	4.878944	0.014319	000.159	0.54267	7.6150701
g16						
Best	-1.905155	-1.905155	-1.905155	-1.905155	-1.905155	-1.905155
Mean	-1.905155	-1.905155	-1.905155	-1.905155	-1.905155	-1.905155
Worst	-1.905155	-1.905155	-1.905155	-1.905155	-1.905155	-1.9050155
SD	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
g17						
Best	8886.685	8879.576	8866.5986	8927.598	8939.125	8860.575
Mean	9053.597	9053.567	8987.4589	8928.865	8946.172	8982.984
Worst	9249.174	9215.365	9165.2543	8938.617	8956.235	9249.278
SD	123.0898	122.6397	95.6532	3.12132	9.528253	109.1522
g18						
Best	-0.8405680	-0.8593651	-0.866023	-0.866025	-0.865976	-0.8660230
Mean	-0.6895726	-0.7107018	-0.795019	-0.740748	-0.767198	-0.8265902
Worst	-0.6616021	-0.6613345	-0.672223	-0.501205	-0.670714	-0.6713253
SD	0.05082904	0.06776626	0.093789	0.1453562	0.0960035	0.07813747
g19						
Best	36.774012	37.580864	33.254703	32.662712	33.7698315	32.9962814
Mean	39.297845	39.834920	34.265623	33.107137	35.3147859	33.6537523
Worst	42.701610	42.427351	35.736841	34.914012	37.3645831	35.5405501
SD	1.4571242	1.1743492	0.631240	0.51325	0.687514	0.527484
g23						
Best	-	-	159.7542	-350.12614	-	-1071.639
Mean	-	-	-35.28473	-121.37464	-	-327.1551
Worst	-	-	109.1275	276.00379	-	149.2050
SD	-	-	82.7698	157.895	-	325.5405
g24						
Best	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013
Mean	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013
Worst	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013	-5.508013
SD	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

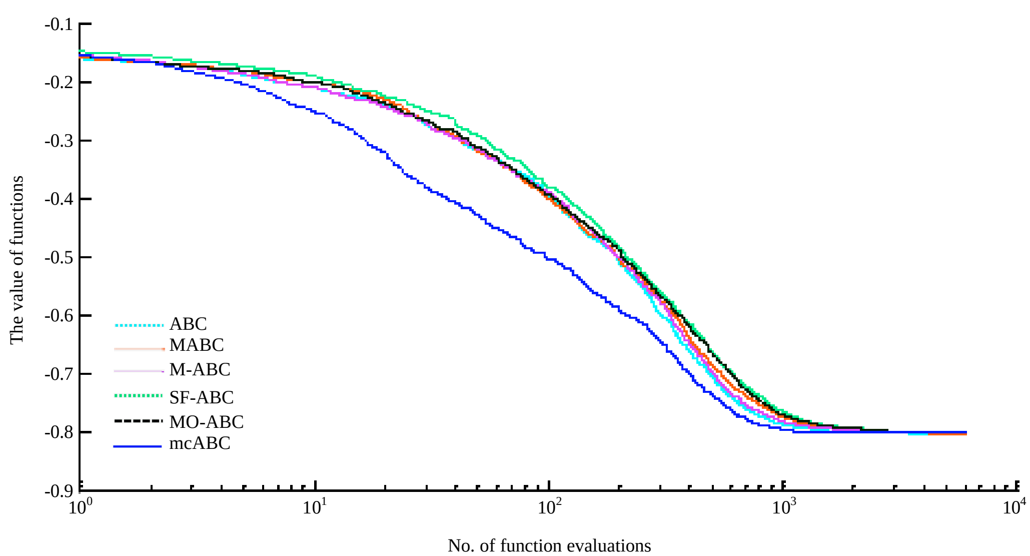

Fig. 1: Iterations to convergence for problem g02

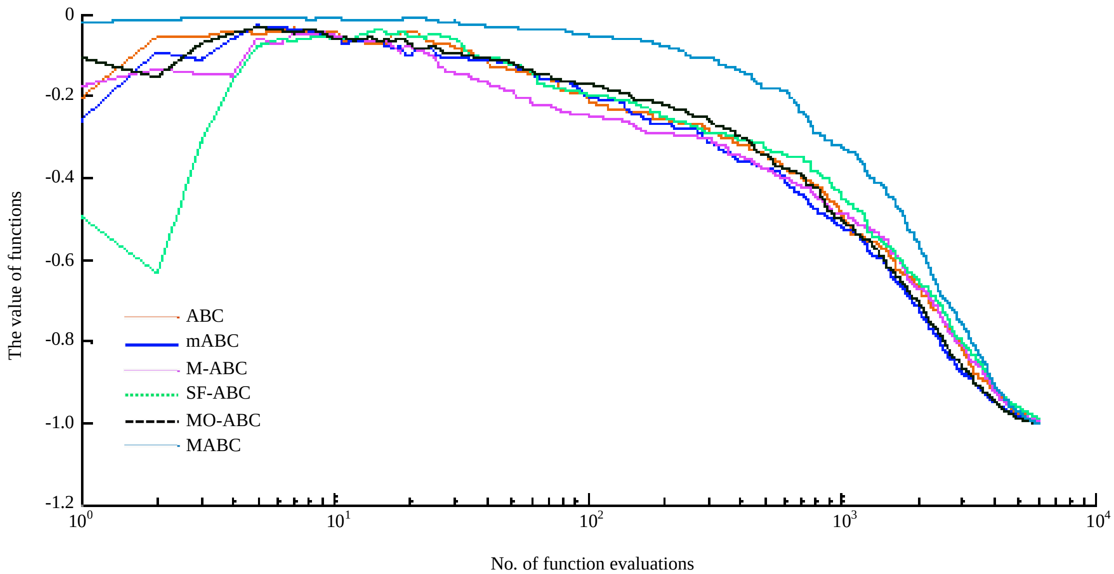

Fig. 2: Iterations to convergence for problem g03

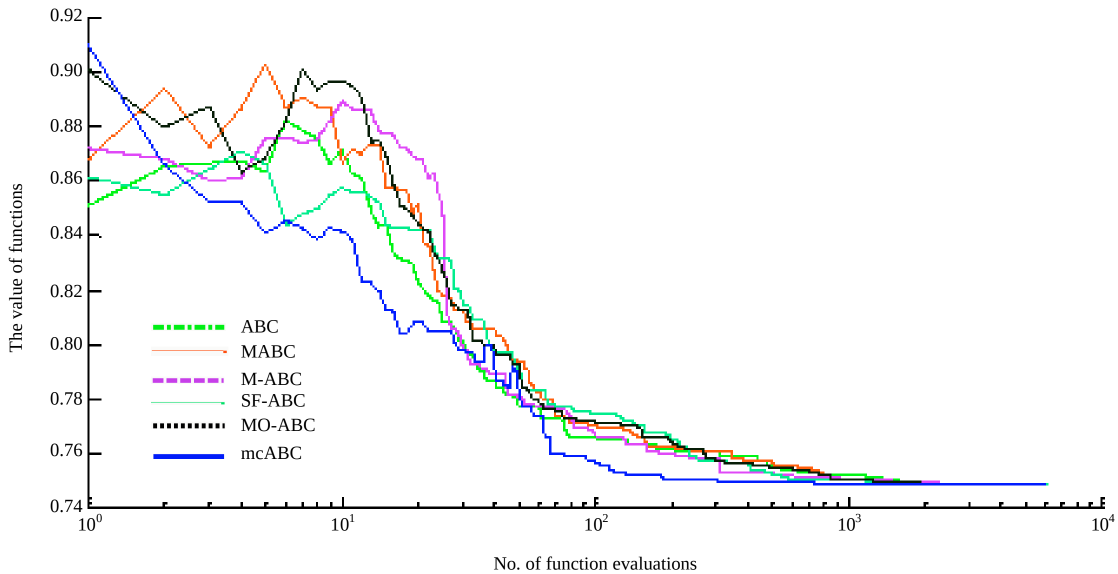

Fig. 3: Iterations to convergence for problem g11

problems g20, g21, g22 are not considered because no feasible solutions can be found for these problems by algorithms. The simulation results demonstrate that all algorithms under comparison obtained the same results for problems g06, g12, g16 and g24. The EABC is superior to other algorithms in problems g02, g03, g04, g08, g11, g17, g18 and g23. The SF-ABC algorithm in problems g05, g10, g13, g15, g19 has good performance compare with other algorithms. However, MO-ABC is outperformed in problems g09, g14.

The numerical performance showed that EABC provided comparable result with respect to other state of the art algorithms in comparison to solve COPs.

In order to compare the convergence ability of EABC with the other state of the art algorithms (Fig. 1-4) are presented which clearly show that EABC is able to converge faster than other algorithms. It confirms that the new search equations can accelerate the constrained ABC convergence.

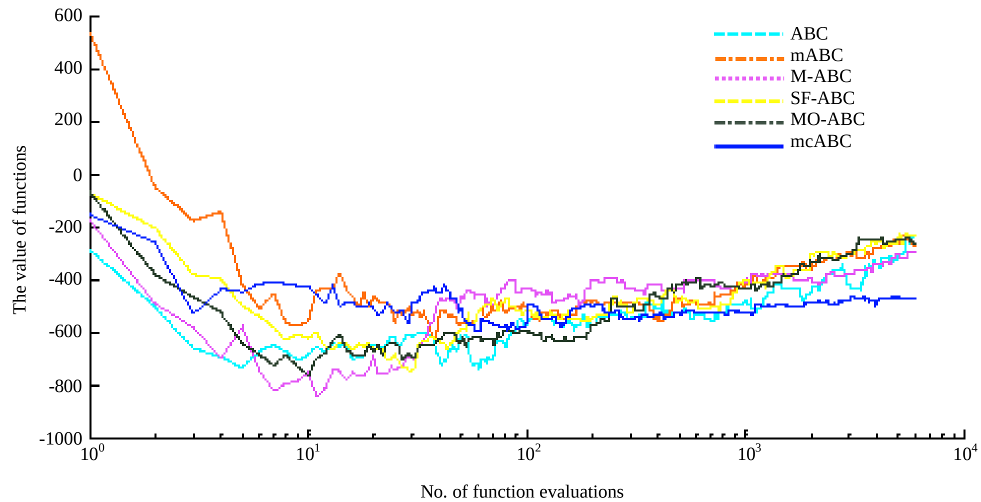

Fig. 4: Iterations to convergence for problem g23

DISCUSSION

In this study, we have introduced a modified constrained ABC called EABC algorithm to solve constrained optimization problems in which the initial population is generated using Chaotic Search Method along with opposition-based learning method. In addition, two new search equations are proposed for employed bee and onlooker bee phases to enhance the global convergence of ABC algorithm. Smart flight method is also applied into scout bee phases to improve the exploitation behaviour of algorithm. The experimental results were tested on 24 benchmark functions and show that ECABC is competitive with state of the art constrained ABC under comparison.

ACKNOWLEDGEMENT

Researchers would like to thank Universiti Teknologi Malaysia for the financial funding through grant RUG 08H47.

REFERENCES

Akay, B. and D. Karaboga, 2012. Artificial bee colony algorithm for large-scale problems and engineering design optimization. J. Intell. Manuf., 23: 1001-1014.

Aydin, D., S. Ozyon, C. Yasar and T. Liao, 2014. Artificial bee colony algorithm with dynamic population size to combined economic and emission dispatch problem. Int. J. Electr. Power Energy Syst., 54: 144-153.

Bacanin, N. and M. Tuba, 2012. Artificial Bee Colony (ABC) algorithm for constrained optimization improved with genetic operators. Stud. Inform. Control, 21: 137-146.

Brajevic, I. and M. Tuba, 2013. An upgraded Artificial Bee Colony (ABC) algorithm for constrained optimization problems. J. Intell. Manuf., 24: 729-740.

Brajevic, I., M. Tuba and M. Subotic, 2011. Performance of the improved artificial bee colony algorithm on standard engineering constrained problems. Int. J. Math. Comput. Simul., 5: 135-143.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. Comput. Methods Applied Mech. Eng., 186: 311-338.

Dorigo, M. and C. Blum, 2005. Ant colony optimization theory: A survey. Theor. Comput. Sci., 344: 243-278.

Gao, W.F., S.Y. Liu and L.L. Huang, 2014. Enhancing artificial bee colony algorithm using more information-based search equations. Inform. Sci., 270: 112-133.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. 1st Edn., University of Michigan Press, Ann Arbor, MI., USA., ISBN-13: 9780472084609, Pages: 183.

Karaboga, D. and B. Akay, 2009. A comparative study of artificial bee colony algorithm. Applied Math. Comput., 214: 108-132.

Karaboga, D. and B. Akay, 2011. A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. Applied Soft Comput., 11: 3021-3031.

Karaboga, D. and B. Basturk, 2007. Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems. Proceedings of the 12th International Fuzzy Systems Association World Congress, June 18-21, 2007, Springer, Berlin/Heidelberg, pp: 789-798.

Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri, Turkey, October 2005. http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf.

Kennedy, J., 2011. Particle Swarm Optimization. In: Encyclopedia of Machine Learning, Sammut, C. and G.I. Webb (Eds.). Springer Science and Business Media, New York, USA., ISBN: 9780387307688, pp: 760-766.

Koziel, S. and Z. Michalewicz, 1999. Evolutionary algorithms, homomorphous mappings and constrained parameter optimization. Evol. Comput., 7: 19-44.

Li, G., P. Niu and X. Xiao, 2012. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. Applied Soft Comput., 12: 320-332.

Liang, J.J., T.P. Runarsson, E. Mezura-Montes, M. Clerc, N. Suganthan, C.A.C. Coello and K. Deb, 2006. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. J. Applied Mech., 41: 2-24.

Mezura-Montes, E. and O. Cetina-Dominguez, 2012. Empirical analysis of a modified artificial bee colony for constrained numerical optimization. Applied Math. Comput., 218: 10943-10973.

Mezura-Montes, E., M. Damian-Araoz and O. Cetina-Domingez, 2010. Smart flight and dynamic tolerances in the artificial bee colony for constrained optimization. Proceedings of the IEEE Congress on Evolutionary Computation, July 18-23, 2010, Barcelona, pp: 1-8.

Stanarevic, N., M. Tuba and N. Bacanin, 2011. Modified artificial bee colony algorithm for constrained problems optimization. Int. J. Math. Models Methods Applied Sci., 5: 644-651.

Subotic, M., 2011. Artificial bee colony algorithm with multiple onlookers for constrained optimization problems. Proceeding of the European Computing Conference, April 28-30, 2011, Paris, France, pp: 251-256.

Tsai, H.C., 2014. Integrating the artificial bee colony and bees algorithm to face constrained optimization problems. Info. Sci., 258: 80-93.

Xiang, W.L. and M.Q. An, 2013. An efficient and robust artificial bee colony algorithm for numerical optimization. Comput. Operat. Res., 40: 1256-1265.