Research Journal of Applied Sciences 11 (1): 14-22, 2016
ISSN: 1815-932X
© Medwell Journals, 2016

An Improved Artificial Bee Colony Algorithm for Constrained Optimization

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

Abstract: Artificial Bee Colony algorithm (ABC) is one of the most popular swarm intelligence algorithms possessing few control parameters and being competitive with other population-based algorithms. However, there is still an insufficiency in this algorithm regarding its convergence behavior. This algorithm is good at exploration but poor at exploitation and yet tackling the issue becomes more challenging if the problem involves constraints. In this research, an improved constrained ABC (iABC) algorithm is proposed to address this class of optimization problems. The modifications that have been introduced in iABC include a novel chaotic approach to generate initial population and two new search equations to enhance exploitation ability of the algorithm. In addition, a new fitness mechanism, along with an improved probability selection scheme has been devised to exploit both feasible and informative infeasible solutions. The proposed algorithm has been tested using CEC2006 benchmark suites. The performance of the iABC algorithm has been compared against the state of the art constrained ABC algorithms. According to the experimental results the proposed algorithm demonstrates a comparative performance and in some cases superior to the algorithms under study.

Key words: Artificial bee colony, constrained optimization, nature inspired algorithms, swarm intelligence, iABC

INTRODUCTION

Almost all real-world optimization problems in science, engineering and industry involve with a number of constraints. In general, Constrained Optimization Problem (COP) can be formulated by Eq. 1:

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

where, $x = (x_1, x_2, \dots, x_n)$ and $l_i \leq x_i \leq u_i$ with l_j and u_j which are lower bound and upper bound, respectively.

It is well known that solving COPs is a challenging task. Optimization methods to solve constrained problems can be divided into two groups; Derivative-Based and Derivative-Free Methods. As there have always been real problems where derivatives are unavailable, derivative-based methods are often inefficient to solve these problems. However, derivative-free techniques do not have such limitation and can effectively apply for this class of optimization problems. As an important branch of derivative-free methods, Evolutionary Algorithms (EAs) have shown remarkable success to solve COPs problems. The most prominent EAs have been introduced in the literatures are Genetic Algorithm (GA) (Tang *et al*., 1996), Differential Evolution (DE) (Storn and Price, 1997), Ant Colony Optimization (ACO) (Dorigo and Birattari, 2010), Particle Swarm Optimization (PSO) (Kennedy, 2010), Artificial Bee Colony (ABC) (Karaboga, 2005), Bacterial Foraging Optimization (BFOA) (Passino, 2002), Biogeography-Based Optimization (BBO) (Simon, 2008), Artificial Immune Systems (AIS) (Farmer *et al*., 1986).

Among these algorithms ABC has been recently proposed. It is an effective algorithm for global optimization. This algorithm has the advantage of employing few control parameters. It has been shown that the performance of the ABC is competitive with other population-based optimization algorithms and the algorithm requires a fewer number of function evaluations to reach to an optimal solution (Karaboga and Basturk, 2007a, b, 2008; Karaboga and Akay, 2009). However, the solution search equations of the ABC algorithm are good at exploration but poor at exploitation which results in poor convergence. Moreover, for constrained optimization problems there is also the need to apply a good constraint handling method. Aiming at improving the performance of constrained ABC algorithm in this paper some modifications are proposed.

Artificial bee colony: ABC algorithm is a recently introduced population-based algorithm proposed by Karaboga (2005) which simulate the foraging behavior of

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

honey bee colonies. The colony of artificial bees includes three groups of bees: employed bees, onlooker bees and scout bees. Half of the colony consists of employed bees and the other half consists of onlooker bees. Employed bees first exploit the food source and gathering required information. Then, they carry the information about the position of food source back to the hive and share this information with onlooker bees.

Onlooker bees are waiting to the hive to received the information from employed bees and then choose food source with better quality using probability selection mechanism as a proportional of the quality of food source. Therefore, the food sources with good quality attract more onlooker bees compared to food source with lower quality. If the quality of the food source is not improved through a predetermined number of iteration, the food source will be abandoned by its employed bee and employed bee becomes a scout and starts to search for a new food source randomly in the neighborhood of the hive. Through the search process, scout bees are responsible for exploration while exploitation is done using employed and onlooker bees.

In ABC, each food source represents a possible solution to the problem and the nectar amount of each food source is the fitness value of the related solution. The number of employed bees or the onlooker bees is equal to the number of solutions SN in the population. At initialization step, a population of SN solutions are randomly generated using Eq. 2:

$$x_{i,j} = l_j + \text{rand}(0, 1)(u_j - l_j) \qquad (2)$$

Where:

i	=	1, 2, ..., SN
j	=	{1, 2, ..., d} and d is dimension of problem
l_j and u_j	=	The lower and upper bounds for the dimension j, respectively

After initialization, the population of solution is repeated in a cycle of the employed bees, onlookers and scouts. Each employed bee generates a new food source in their neighborhood using Eq. 3:

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

where, $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., d\}$ are randomly chosen indexes, k has to be different from i, $\phi_{i,j}$ is a random number in the range [-1, 1]. After each v_i is calculated the fitness value of this solution is evaluated and a greedy selection mechanism is applied comparing x_i and v_i. If the fitness of v_i is better than fitness value of x_i, then it will be replaced with x_i and x_i will be removed, otherwise x_i is retained in population.

After all employed bees complete their searches, they share their information about fitness and position of solutions with the onlooker bees. An onlooker bee chooses a solution using probability value associate with the solution where p_i is defined as follows:

$$p_i = \frac{\text{fit}_i}{\sum_{j=1}^{SN} \text{fit}_j} \qquad (4)$$

where, fit_i is the fitness value of solution i. The higher the value fit_i of has more probability that the ith solution is selected. Once the onlooker has selected solution x_i a modification is done on the solution using Eq. 3. If a new solution has better quality than the old solution, the old solution is replaced with new solution otherwise the old solution remained in the population.

If a solution cannot be improved further through a predetermined number of trials (limit), the solution is abandoned and the corresponding employed bee becomes a scout. The scout produces a solution randomly using Eq. 2. The detailed pseudo code of original ABC algorithm is presented in the 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
  Onlooker bee phase
  Scout bee phase
  Memorize the best solution achieved so far
  cycle = cycle+1
  Until cycle = maximum cycle number
```

In recent years the vast majority of unconstrained ABC algorithm have been presented and applied for practical problems (Akay and Karaboga, 2012; Gao *et al*., 2014; Banitalebi *et al*., 2015; Karaboga and Gorkemli, 2014; Kiran *et al*., 2015).

Literature review: ABC algorithm has been originally introduced to address unconstrained optimization problems (Karaboga, 2005). Then, this method is adapted to deal with constrained optimization problems. The presence of various constraints and interferences between them makes COPs more challenging than unconstrained optimization problems. In this study, we briefly present the available constrained ABC algorithms in the literature.

The first attempt to apply ABC algorithm to solve COPs is done by Karaboga and Bastruck (2007b). To cope with constraints, Deb (2000)'s mechanism is employed to be used instead of the greedy selection mechanism 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 alternative Deb's rules are employed to direct the solutions to feasible region of search space. In addition, the scout bee phase of the algorithm provides a diversity mechanism that allows new and probably infeasible individuals to be in the population. Scouts are generated at a predetermined period of cycles for discovering new solution randomly. This period is another control parameter called Scout Production Period (SPP). At each SPP cycle, it is controlled if there is an abandoned solution or not. If there is, a scout production process is executed. The numerical performance of the proposed ABC algorithm is evaluated and compared with constrained PSO and DE algorithms and results show that ABC algorithm can be effectively applied for solving constrained optimization problems.

Mezura-Montes *et al.* (2010) presented Smart Flight ABC (SF-ABC) algorithm to improve the performance of constrained ABC where smart flight operator is applied in scout bee phase to direct search towards promising region of the search space. Therefore, 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, the combinations of two dynamic tolerances are also applied into SF-ABC to transform the original COP into unconstrained optimization. The numerical results demonstrate the competitive performance of SF-ABC with constrained ABC (Karaboga and Basturk, 2007b).

A modified ABC was introduced by Karaboga and Akay (2011) to solve COPs. 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. To recognize this algorithm through this study the abbreviation MABC will be used.

Another modified constrained ABC was developed by Subotic where Multiple Onlooker bees (MO-ABC) are applied into original constrained ABC. In this algorithm, three trial solutions are applied to form a new solution. The numerical performance of the algorithm when compared with the original ABC shows comparative results.

Mezura-Montes and Cetina-Dominguez (2012) presented a Modified ABC (M-ABC). This algorithm consist of four modifications on the selection mechanism, the equality and boundary constraints and scout bee operators compared to the original constrained ABC. The mechanisms to handle equality and boundary constraints are enhanced with the aim to support a more appropriate approach to the feasible region of the search space. A binary tournament selection based on feasibility is supplanted with the fitness selection of solutions applied in the original ABC. In addition, smart flight operator is employed to be used in scout bee instead of the uniformly random approach in constrained ABC (Karaboga and Basturk, 2007a, b). The numerical results show that M-ABC provides comparable results with respect to algorithms under comparison.

An efficient constrained ABC (eABC) algorithm was suggested by Babaeizadeh and Rohanin where two new solution search equations was introduced, respectively for employed bee and onlooker bee to enhance the exploitation of algorithm.

A Genetic Inspired ABC algorithm (GI-ABC) was introduced to adopt GA in the process of replacement of exhausted solutions (Bacanin and Tuba, 2012). In this algorithm, uniform crossover and mutation operators from GAs are applied to improve the performance of ABC algorithm.

Stanarevic *et al.* (2011) suggested Smart Bee ABC algorithm (SB-ABC) to solve constrained problems. In this algorithm, smart bee is used to memorize the solutions and their fitness. Then, the best solution is replaced with a new random solution if the new solution is unfeasible or if the new solution is feasible but it does not have better fitness. The numerical experiments show efficiency of the method.

ABC-BA is a hybrid algorithm presented by Tsai (2014) that integrates ABC and Bee Algorithm (BA) to solve COPs. 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 algorithms, respectively.

Constrained ABC algorithm was also applied to solve many real-world engineering problems in recent years. Brajevic proposed a Constrained Artificial Bee Colony (SC-ABC) and applied on several standard engineering benchmark problems of discrete and continuous variables. The numerical results then were compared to results obtained from Simple Constrained Particle Swarm Optimization Algorithm (SiC-PSO) which show a 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 comparison. Upgraded Artificial Bee Colony (UABC) algorithm was also introduced for constrained optimization problems by Brajevic and Tuba to improve fine-tuning features of the modification rate parameter and applying modified scout bee phase of the ABC algorithm. This algorithm was then tested on several engineering benchmark problems and the performance was compared with the performance of the Akay and Karaboga (2012) algorithm. The numerical results show that the UABC produces better results. For more information see the recent survey on constrained ABC.

MATERIALS AND METHODS

Improved constrained artificial bee colony: The initial population plays an important role in efficient implementation of population-based algorithms as it affects the global convergence and the quality of the final solution. Simple random initialization is the most frequent method that has been applied to generate initial population. However, generating initial population using this method is not necessarily exhaustive and this affects the performance of the ABC algorithm. Babaeizadeh and Ahmad (2014a) applied chaotic search mechanism to initialize population for constrained ABC. Then, this method compared with the ABC (Karaboga and Basturk, 2007) and numerical results show the competitive results. A modified constrained ABC algorithm (mcABC) was proposed in which chaotic mechanism as well as opposition based method was applied into initialization phase of ABC algorithm to enhance the global convergence of algorithm. The numerical results demonstrate the effectiveness of the proposed method (Babaeizadeh and Ahmad, 2014b).

Algorithm 2 (Initialization approach):

```
for i = 1 to SN/2
   for j = 1 to d
      Generate a uniform random number $\theta$
      in [0, 1]
      Let $\psi = \theta - 1$
      Generate an integer random number k in [50, 100]
     for k = 1 to K
           $\theta = mad(\xi\theta, 1)$
           $\psi = (1-\theta)(1-2mod(k, 2))$
     end
            $x_{i,j} = l_i+0.5(1+\psi)(u_j-l_j)$
   end
 end
 Set the individual counter i = 1 and j = 1
 for i = SN/2 to SN
    for j = 1 to d
          $x_{i,j} = l_j+u_j-l_j$
    end
 end
```

Enhanced ABC (EABC) algorithm is also 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.

In this study, aiming to enhance the diversity of the initial population of iABC algorithm, a novel chaotic mechanism along with opposition-based learning is introduced. This procedure is described in Algorithm 2.

After initialization, the population of solutions is repeated in a cycle of the search procedures of the employed, the onlooker and scout bee phases. In order to enhance exploitation ability of algorithm a new solution search equation is proposed for employed bee phase as follows:

$$\nu_{ij} = \begin{cases} x_{ij}+\gamma_{ij}(x_{b,j}-x_{ij})+\mu_{ij}(x_{r1j}-x_{ij})+\sigma_{ij}(x_{r2j}-x_{ij}) & R_j<MR \\ x_{ij}+\tau_{ij}(x_j-x_{r1j}) & \text{otherwise} \end{cases} \quad (5)$$

where, r_1, r_2 are randomly chosen index has to be different form i and γ_{ij} is a random number between [0, 0.5], μ_{ij} and σ_{ij} between [-0.5, 0.5], τ_{ij} between [-1, 1]. R_j is uniformly distributed random number in the range [-1, 1] and MR is a control parameter which controls the number of parameters to be modified.

After generating a new solution, Deb's rules are applied in selection process. Using Deb's mechanism, either the new solution is memorized or old solution. The framework of employed bee phase is given in Algorithm 3. After all employed bees complete their searches, they share their information related to the fitness values and the positions of their solutions with the onlooker bees.

What makes COPs more difficult to handle is the presence of various constraints. As informative infeasible individuals can also provide useful information about the optimal direction, it is necessary to keep a number of infeasible solutions in the population. In probability selection mechanism a large number of individuals in the population are feasible.

Algorithm 3 (Employed bee phase for iABC):

```
for i = 1:SN
   for j = 1:d
  Produce a new solution $v_i$ using Eq. 5
  end for
    If no parameter is changed, choose a parameter randomly and change it
    form solution $x_i$ using Eq. 5
   Evaluate the solution $v_i$
   Apply the selection process between $v_i$ and $x_i$ based on Deb's Method
   If solution $x_i$ does not improve $trail_i$+1, otherwise $trail_i$ = 0
end if
```

Therefore, informative infeasible individuals in the population have little chance to survive into the next population. Motivated by the above-mentioned consideration, to improve the selection mechanism, a modified probability selection mechanism is proposed to balance the population of feasible individuals and infeasible individuals which are close to the feasible region.

The new probability selection mechanism is based on the following instruction to calculate the probability p_i for solution x_i, i = 1, 2, ..., SN as Eq. 6:

$$p_i = \begin{cases} 0.5+\dfrac{fit_i}{\sum_{j=1}^{SN} fit_i}+(0.5+2\varepsilon)-\varepsilon & \text{if solution is feasible} \\ 0.5-\dfrac{violation_i}{\sum_{j=1}^{SN} violation_i}+(0.5+2\varepsilon)+2\varepsilon & \text{if solution is infeasible} \end{cases} \quad (6)$$

Where:

$violation_i$ = The constraint violation of solution

x_i and fit_i = The fitness value of the solution x_i

The value of fit_i can be calculated as follow: if x_i is feasible:

$$fit_i = \begin{cases} \dfrac{1}{(1+f_i)} & f_i>0 \\ \dfrac{1}{|f_i|} & f_i<0 \\ 1 & f_i = 0 \end{cases} \quad (7)$$

Otherwise if it is infeasible:

$$fit_i = \begin{cases} \dfrac{1}{(f_i\,violation_i)} & f_i>0 \\ \dfrac{|f_i|}{violation_i} & f_i<0 \\ \dfrac{1}{violation_i} & f_i = 0 \end{cases} \quad (8)$$

and violation is defined as follow:

$$violation(x) = \sum_{i=1}^{p+q} violation_i(x) \quad (9)$$

The distance of a solution x from the jth constraint can be created as:

$$violation_i(x) = \begin{cases} \max\{0, g_i(x)\} & \text{if } 1 \le i \le p \\ \max\{0, |h_i(x)|-\delta\} & \text{if } 1 \le j \le q \end{cases} \quad (10)$$

where, δ is the tolerance value for the equality constraints. An onlooker bee then evaluates the information shared by employed bees and selects a solution with a probability associated with its nectar amount. After solution selection, onlooker bees produce modification on the position of the selected solution by taking advantage of the global best and random solution to guide the candidate solution toward promising region of search space using Eq. 11:

$$v_{ij} = \begin{cases} x_{ij}+\chi_{ij}(x_b-x_{j,r1})+\eta_{ij}(x_{ij}-x_{j,r2}) & R_j \le MR \\ x_{ij}+\rho_{ij}(x_{ij}-x_{j,r1}) & \text{otherwise} \end{cases} \quad (11)$$

where, x_{r1j} and x_{r2j} are uniformly random solution and $j \in \{1, 2, ..., d\}$, χ_{ij} is a random number in [-1, 1], η_{ij} and ρ_{ij} are random number in the range [0, 1]. Similar with employed bee phase after generating new solution using Eq. 11, the new solution is compared with current solution using Deb's rules. If the new solution has better equality it will remained in the population and the current solution removed otherwise the current solution is remained. The framework of onlooker bee phase is given in Algorithm 4.

After distribution of all onlooker bees, if a solution can not improve further through predetermined number of cycles (limit) it is abandoned and replaced with a new solution discovered by scout bees. The scout produces a new solution using smart flight operator defined in Eq. 12:

$$v_{ij} = x_{ij}+\omega_{ij}(x_{rj}-x_{ij})+(1-\omega_{ij})(x_{bj}-x_{ij}) \quad (12)$$

Table 1: The main characteristics of the test problems

Functions	Types	n	ρ	LI	NI	LE	NE	α
g01	Quadratic	13	0.0111	9	0	0	0	6
g02	Nonlinear	20	99.9971	1	1	0	0	1
g03	Polynomial	10	0	0	0	0	1	1
g04	Quadratic	5	52.1230	0	6	0	0	2
g05	Cubic	4	0	2	2	0	3	3
g06	Cubic	2	0.0066	0	5	0	0	2
g07	Quadratic	10	0.0003	3	2	0	0	6
g08	Nonlinear	2	0.8560	0	4	0	0	0
g09	polynomial	7	0.5121	0	3	0	0	2
g10	Linear	8	0.0010	3	0	0	0	3
g11	Polynomial	2	0	0	1	0	1	1
g12	Quadratic	3	4.7713	0	1	0	0	0
g13	Quadratic	5	0	0	0	0	3	3
g14	Nonlinear	10	0	0	0	3	0	3
g15	Quadratic	3	0	0	0	1	1	2
g16	Nonlinear	5	0.0204	4	34	0	0	4
g17	Nonlinear	6	0	0	0	0	4	4
g18	Quadratic	9	0	0	13	0	0	6
g19	Nonlinear	15	33.4761	0	5	0	0	0
g20	Linear	24	0	0	6	2	12	16
g21	Linear	7	0	0	1	0	5	6
g22	Linear	22	0	0	1	8	11	19
g23	Linear	9	0	0	2	3	1	6
g24	Linear	2	79.6556	0	2	0	0	2

where, ω_{ij} is random number in [-1, 1] and r is random index that have to be different from $i\in\{1, 2, ..., SN\}$, x_b is the best solution found so far.

Algorithm 4 (Onlooker bee phase for iABC):

```
t = 0, i = 1
Repeat
    if random <$p_i$ then
        t = t+1
     for j = 1:d
      Produce a new solution $v_i$ using Eq. 11
    end for
      If no parameter is changed, choose a parameter randomly and change
      it form solution $x_i$ using Eq. 11
      Evaluate the solution $v_i$
  Apply the selection process between $v_i$ and $x_i$ based on Deb's Method
  If solution $x_i$ does not improve trail$_i$ = +1, otherwise trail$_i$ = 0
 end if
  i = i+1
  i = i mod(SN+1)
  until t = SN
```

Numerical experiments and comparisons: In order to evaluate the performance of iABC algorithm and show the efficiency and superiority of the proposed algorithm, 24 well-known benchmark problems form CEC2006 (Liang *et al*., 2006) are applied.

The proposed algorithm is evaluated and compared with four state of the art constrained ABC algorithms. The iABC algorithms as well as other algorithms in comparison are coded in MALAB environment. Each problem runs 30 times and statistical results are provided including the best, median, mean, worst results and the standard deviation which can be seen in study.

Benchmark test problems and parameter settings: The main characteristics of 24 benchmark functions are shown in Table 1. Table 1 describes various kinds of these test functions (linear, non-linear, polynomial, quadratic and cubic) with different numbers of decision variables, different types (linear inequalities, linear equalities, nonlinear inequalities and nonlinear equalities) and numbers of constraints. In Table 1, ρ is the estimated ratio between the feasible region and the search space, LI is the number of linear inequality constraints, NI is the number of nonlinear inequality constraints, LE is the number of linear equality constraints, NE is the number of non-linear equality constraints, a is the number of constraints active at the optimal solution and n is the number of variables of the problem. However as all the algorithms considered in comparison were not able to obtain feasible solutions for g20, g21 and g22 we exclude these problems from our experiments. In addition, the value of each parameters used are given in Table 2.

Table 2: Parameters setting

Parameters	Symbols	Values
Solutions number	SN	20
Maximum cycle number	MCN	6000
Modification rate	MR	0.8
Delta	δ	0.0001
Epsilon	ε	0.01

RESULTS AND DISCUSSION

The numerical performance of iABC algorithm was compared with original ABC (Karaboga and Basturk, 2007a, b), MABC (Karaboga and Akay, 2011), M-ABC (Mezura-Montes and Cetina-Dominguez, 2012), SF-ABC (Mezura-Montes *et al*., 2010). From Table 3, it is obvious that the iABC algorithm in problems g02, g03, g07, g08, g09, g11, g16 and g23 outperforms compare with other algorithms. For g04, g10, g19, the performance of SF-ABC

Table 3: The numerical results obtained by ABC, MABC, M-ABC and iABC

Problems	Results	ABC	MABC	M -ABC	SF-ABC	iABC
g01	Best	-1.500000e+01	1.500000e+01	1.500000e+00	1.500000e+01	1.500000e+01
	Mean	1.500000e+01	1.500000e+01	1.500000e+01	-1.426341e+01	1.500000e+01
	Worst	1.500000e+01	1.500000e+01	1.500000e+01	-1.274809e+01	1.500000e+01
	SD	9.418509e-14	5.190673e-15	7.420126e-15	8.890347e-01	9.895836e-16
g02	Best	-8.035669e-01	-8.035383e-01	-8.036169e-01	-7.175021e-01	-8.036189e-01
	Mean	-8.017445e-01	-8.026779e-01	-7.994087e-01	-5.990304e-01	-8.014211e-01
	Worst	-7.929237e-01	-8.003014e-01	-7.780131e-01	-4.323677e-01	-7.861565e-01
	SD	3.002902e-03	1.049929e-03	5.790153e-03	2.497013e-02	5.211638e-03
g03	Best	-1.004657e+00	-1.004817e+00	-1.000803e+00	-1.000396e+00	-1.005003e+00
	Mean	-1.000096e+00	-1.001941e+00	-1.000438e+00	-1.000107e+00	-1.004983e+00
	Worst	-9.796509e-01	-9.891600e-01	-1.000164e+00	-1.000042e+00	-1.004926e+00
	SD	5.979115e-03	3.751844e-03	4.704529e-05	5.683220e-05	1.755578e-05
g04	Best	-3.066554e+04	-3.066554e+04	-3.066554e+04	-3.066553e+04	-3.066554e+04
	Mean	-3.066554e+04	-3.066554e+04	-3.066553e+04	-3.066553e+04	-3.066554e+04
	Worst	-3.066554e+04	-3.066554e+04	-3.066552e+04	-3.066553e+04	-3.066554e+04
	SD	3.981196e-11	7.744473e-11	4.901256e-11	2.067912e-11	1.519625e-11
g05	Best	5.126863e+03	5.127099e+03	5.126815e+03	5.126793e+03	5.126514e+03
	Mean	5.187239e+03	5.236991e+03	5.298769e+03	5.126604e+03	5.319742e+03
	Worst	5.437988e+03	5.802318e+03	5.509277e+03	5.126918e+03	5.723262e+03
	SD	5.696131e+01	1.560343e+02	5.617809e+01	5.046901e-01	2.082831e+02
g06	Best	-6.961814e+03	-6.961814e+03	-6.961814e+03	-6.961814e+03	-6.961814e+03
	Mean	-6.961814e+03	-6.961814e+03	-6.961814e+03	-6.961814e+03	-6.961814e+03
	Worst	-6.961814e+03	-6.961814e+03	-6.961814e+03	-6.961814e+03	-6.961814e+03
	SD	2.764179e-12	4.931216e-12	5.412767e-12	3.902387e-12	2.240565e-12
g07	Best	2.446138e+01	2.447030e+01	2.431439e+01	2.431643e+01	2.431173e+01
	Mean	2.470718e+01	2.468698e+01	2.445981e+01	2.465759e+01	2.446559e+01
	Worst	2.516577e+01	2.536005e+01	2.545217e+01	2.554426e+01	2.505395e+01
	SD	1.813943e-01	1.785620e-01	3.425503e-01	3.149533e-01	1.720010e-01
g08	Best	-9.582504e-02	-9.582504e-02	-9.582521e-02	-9.582531e-02	-9.58204e-02
	Mean	-9.582504e-02	-9.582504e-02	-9.582537e-02	-9.582546e-02	-9.582504e-02
	Worst	-9.582504e-02	-9.582504e-02	-9.582562e-02	-9.582549e-02	-9.582504e-02
	SD	2.823006e-17	2.823006e-17	5.013611e-17	1.992503e-17	1.945623e-17
g09	Best	6.80638e+02	6.806371e+02	6.806323e+02	6.806311e+02	6.806307e+02
	Mean	6.86506e+02	6.806515e+02	6.806378e+02	6.806429e+02	6.806406e+02
	Worst	6.806506e+02	6.806760e+02	6.806892e+02	6.808601e+02	6.806593e+02
	SD	8.074911e-03	9.561019e-03	2.349103e-02	5.115349e-02	7.625513e-03
g10	Best	7.160631e+03	7.205540e+03	7.051408e+03	7.049937e+03	7.050539e+03
	Mean	7.364940e+03	7.347843e+03	7.234266e+03	7.216495e+03	7.216535e+03
	Worst	7.691303e+03	7.924128e+03	7.470874e+03	7.370533e+03	7.522059e+03
	SD	1.298405e+02	1.341410e+02	2.356713e+02	4.802264e+02	1.109373e+02
g11	Best	7.490001e-01	7.490000e-01	7.501841e-01	7.520393e-01	7.490000e-01
	Mean	7.490021e-01	7.490031e-01	7.512236e-01	7.533215e-01	7.493012e-01
	Worst	7.490104e-01	7.490142e-01	7.514019e-01	7.534092e-01	7.507898e-01
	SD	2.030974e-06	3.620362e-06	3.490254e-05	2.641176e-05	5.063214e-04
g12	Best	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00
	Mean	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00
	Worst	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00	-1.000000e+00
	SD	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
g13	Best	5.551238e-01	4.895965e-01	2.213604e-01	2.285133e-01	4.495833e-01
	Mean	9.497812e-01	9.576896e-01	2.674431e-01	2.639674e-01	7.352076e-01
	Worst	1.492954e+00	1.437534e+00	5.301298e-01	6.190731e-01	9.118312e-01
	SD	1.469151e-01	1.613582e-01	1.627542e-01	2.437194e-01	1.488054e-01
g14	Best	-4.511878e+01	-4.532082e+01	-4.757941e+01	-4.690835e+01	-4.700910e+01
	Mean	-4.268215e+01	-4.265421e+01	-4.720886e+01	-4.646839e+01	-4.504622e+01
	Worst	-4.060165e+01	-4.005962e+01	-4.684037e+01	-4.377068e+01	-4.375761e+01
	SD	1.171236e+00	1.195831e+00	2.632415e-01	5.670931e-01	6.689353e-01
g15	Best	9.412191e+02	9.514375e+02	9.608947e+02	9.622345e+02	9.576714e+02
	Mean	9.588476e+02	9.608922e+02	9.617855e+02	9.632661e+02	9.694047e+02
	Worst	9.729578e+02	9.706846e+02	9.624628e+02	9.748746e+02	9.776851e+02
	SD	7.512742e+00	4.878944e+00	1.511032e-02	1.273942e-02	6.183477e+00
g16	Best	-1.905155e+00	-1.905155e+00	-1.905155e+00	-1.905155e+00	-1.905155e+00
	Mean	-1.905155e+00	-1.905155e+00	-1.905155e+00	-1.905155e+00	-1.905155e+00
	Worst	-1.905155e+00	-1.905155e+00	-1.905155e+00	-1.905155e+00	-1.905155e+00
	SD	1.068872e-15	6.307382e-16	2.791278e-16	3.091653e-14	7.318069e-16
g17	Best	8.886685e+03	8.879576e+03	8.866618e+03	8.895849e+03	8.867942e+03
	Mean	9.053597e+03	9.053567e+03	8.987459e+03	8.957531e+03	8.946634e+03
	Worst	9.249174e+03	9.215365e+03	9.1652190e+03	8.966835e+03	9.158921e+03
	SD	1.230898e+02	1.226397e+02	9.432612e+01	2.331451e+00	6.220550e+01

Table 3: Continous

Problems	Results	ABC	MABC	M -ABC	SF-ABC	iABC
g18	Best	-8.40568e-01	-8.593651e-01	-8.660074e-01	-8.660258e-01	-8.660020e-01
	Mean	-6.895726e-01	-7.107018e-01	-7.943215e-01	-7.377351e-01	-7.376434e-01
	Worst	-6.616021e-01	-6.613345e-01	-6.643098e-01	-5.171642e-01	-6.715862e-01
	SD	5.082904e-02	6.776626e-02	8.572117e-02	2.249256e-01	9.189952e-02
g19	Best	3.677401e+01	3.758086e+01	3.364751e+01	3.266260e+01	3.325666e+01
	Mean	3.929784e+01	3.983492e+01	3.432583e+01	3.310718e+01	3.442409e+01
	Worst	4.270161e+01	4.242735e+01	3.637352e+01	3.491403e+01	3.604190e+01
	SD	1.457124e+00	1.174349e+00	5.802467e-01	4.842309e-01	6.929925e-01
g23	Best	-	-	-	-3.622547e+02	-4.836472e+02
	Mean	-	-	-	-1.397621e+02	-8.491502e+01
	Worst	-	-	-	2.468019e+02	1.627247e+02
	SD	-	-	-	1.562291e+02	1.339717e+02
g24	Best	-5.508013e+00	-5.508013e+00	-5.508013e+00	-5.508013e+00	-5.508013e+00
	Mean	-5.508013e+00	-5.508013e+00	-5.5080130e+00	-5.508013e+00	-5.508013e+00
	Worst	-5.508013e+00	-5.508013e+00	-5.508013e+00	-5.508013e+00	-5.508013e+00
	SD	1.806812e-15	3.735596e-15	4.256902e-15	2.394062e-15	1.806724e-15

was superior to all other algorithms. However, M-ABC is superior in problems g13, g14, g15 and g16. The numerical performance shows that iABC provided comparable result with respect to other state of the art constrained ABC algorithms in solving COPs.

CONCLUSION

In this study, an improved Artificial Bee Colony algorithm (iABC) was proposed for the constrained non-linear optimization problems. The proposed iABC algorithm benefits from the advantages of possessing several new features. A novel chaotic technique has been devised and hybridized with opposition based learning method to diversify the initial population which in turn affects the global convergence of the algorithm. Moreover, two new search equations have been introduced into employed and onlooker bee phases to improve the balance of the exploration and exploitation of the algorithm. A probability selection mechanism along with a new fitness formulation has been designed to take into account the informative infeasible solutions as well as the feasible solutions. In addition, the smart flight operator is employed on the scout bee phase of this algorithm. The iABC has been comprehensively tested on the test problems of the CEC2006 benchmark suites. It has been compared with several state of the art algorithms, where the simulations revealed that iABC has a competitive performance and in some cases superior to the algorithms considered in this study.

ACKNOWLEDEGEMENT

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

REFERENCES

Akay, B. and D. Karaboga, 2012. A modified Artificial Bee Colony algorithm for real-parameter optimization. Inform. Sci., 192: 120-142.

Babaeizadeh, S. and R. Ahmad, 2014a. A modified artificial bee colony algorithm for constrained optimization problems. J. Convergence Inform. Technol., 9: 151-163.

Babaeizadeh, S. and R. Ahmad, 2014b. Modified artificial bee colony algorithm with chaotic search method for constrained optimization problems. Proceedings of the 2nd International Science Postgraduate Conference, March 10-12, 2014, Faculty of Science, Universiti Teknologi Malaysia.

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

Banitalebi, A., M.I.A. Aziz, A. Bahar and Z.A. Aziz, 2015. Enhanced compact artificial bee colony. Inform. Sci., 298: 491-511.

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

Dorigo, M. and M. Birattari, 2010. Ant Colony Optimization. In: Encyclopedia of Machine Learning, Sammut, C. and G.I. Webb (Eds.). Springer, US., pp: 36-39.

Farmer, J.D., N.H. Packard and A.S. Perelson, 1986. The immune system, adaptation and machine learning. Phys. D: Nonlinear Phenomena, 22: 187-204.

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.

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. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. J. Global Optim., 39: 459-471.

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. and B. Basturk, 2008. On the performance of Artificial Bee Colony (ABC) algorithm. Applied Soft Comput., 8: 687-697.

Karaboga, D. and B. Gorkemli, 2014. A quick Artificial Bee Colony (qABC) algorithm and its performance on optimization problems. Applied Soft Comput., 23: 227-238.

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., 2010. Particle Swarm Optimization. In: Encyclopedia of Machine Learning, Sammut, C. and G.I. Webb (Eds.). Springer, US., pp: 760-766.

Kiran, M.S., H. Hakli, M. Gunduz and H. Uguz, 2015. Artificial bee colony algorithm with variable search strategy for continuous optimization. Info. Sci., 300: 140-157.

Liang, J.J., T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.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. Technical Report, September 18, 2006.

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.

Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst., 22: 52-67.

Simon, D., 2008. Biogeography-based optimization. IEEE Trans. Evol. Comput., 12: 702-713.

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.

Storn, R. and K. Price, 1997. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim., 11: 341-359.

Tang, K.S., K.F. Man, S. Kwong and Q. He, 1996. Genetic algorithms and their applications. IEEE Signal Process. Magazine, 13: 22-37.

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