Modified Global Best Artificial Bee Colony for constrained optimization problems

Jagdish Chand Bansal[1,*], Susheel Kumar Joshi[2,]

Department of Mathematics, South Asian University, New Delhi, India

Harish Sharma[3,]

Department of Computer Science and Engineering, Rajasthan Technical University, Kota, India

---

Abstract

Artificial Bee Colony (ABC) is a well-known algorithm in the class of swarm-intelligence-based optimization algorithms. Recently, a variant of ABC, Gbest-guided ABC (GABC) was proposed. GABC was verified to perform better than ABC, in terms of efficiency and reliability. In the position update process of GABC, Gbest (the best individual in the swarm) individual influences the movement of the swarm. This movement may create a cluster around the Gbest individual which further leads to the premature convergence, particularly for constrained optimization problems. This paper presents a modification in GABC for constrained optimization problems. GABC is modified in both employed and onlooker bee phases by incorporating the concept of fitness probability based individual movement. The modified GABC is tested over 20 constrained benchmark problems and applied to solve 3 engineering design problems. Optimal power flow problem has also been solved using modified GABC to check the efficiency of the proposed algorithm.

Keywords: Artificial bee colony, Constrained optimization, Optimal power flow problem, Exploration, Exploitation

---

*Corresponding author

Email addresses: jcbansal@gmail.com (Jagdish Chand Bansal), sushil4843@gmail.com (Susheel Kumar Joshi), harish.sharma0107@gmail.com (Harish Sharma)

[1]Department of Mathematics, South Asian University, New Delhi, India

[2]Department of Mathematics, South Asian University, New Delhi, India

[3]Department of Computer Science and Engineering, Rajasthan Technical University, Kota, India

1. Introduction

Problems in various fields, like economics, engineering design, structural optimization are modeled as constrained optimization problems. Due to the presence of constraints, usually, the problem becomes difficult to solve. A general form of a constrained optimization problem (in minimization case) is:

$$\begin{aligned} \underset{\vec{x}}{\text{minimize}} \quad & f(\vec{x}), \\ \text{subject to} \quad & g_j(\vec{x}) \leq 0, \quad \forall j = 1, 2, ...q \\ & h_j(\vec{x}) = 0, \quad \forall j = 1, 2, ...r \end{aligned} \tag{1}$$

Here, f($\vec{x}$) is the objective function defined on a search space $\mathbb{S}$. $\vec{x} = (x_1, ..., x_n) \in \mathbb{S} \subset \mathbb{R}^n$ is a n-dimensional vector bounded by its lower and upper limits i.e. $l_i \leq x_i \leq u_i, \forall i = 1, 2, ...n$. R^n is the n-dimensional field of real numbers, $g(\vec{x})$ is the set of q inequality constraints and $h(\vec{x})$ is the set of r equality constraints. The solution of problem (1) is a vector $\vec{x}$ defined in the search space $\mathbb{S}$ that minimizes f($\vec{x}$) such that the set of given equality and inequality constraints are satisfied.

Many traditional, mostly deterministic optimization methods are available in the literature to solve the constrained optimization problems. For solving real world constrained optimization problems, deterministic methods like generalized reduced gradient methods [1], sequential quadratic programming methods [2] etc require many mathematical conditions, e.g. convexity, continuity or differentiability to be satisfied. A real world problem rarely follows all required conditions. In this scenario, deterministic methods become infeasible to solve a real world problem, in their original form.

On the contrary, nature-inspired optimization algorithms have merits over deterministic methods, such as derivative free mechanism, having higher ability to avoid local optima, flexible in terms of the applicability, and being simple for implementation.

Therefore nature-inspired optimization algorithms like, particle swarm optimization (PSO) [3], differential evolution (DE) [4] and artificial bee colony (ABC) [5] are preferred to solve such complex constrained optimization problems.

Several new variants of these algorithms are developed in order to make their performance more powerful over constrained optimization problems. In this context, we observe artificial bee colony (ABC) which was invented by Karaboga et al. [6] and is inspired by the foraging behavior of honey bees. Similar to other swarm based optimization algorithms, ABC possesses a swarm of candidate solutions, which are the food sources of honey bees. The nectar amount (or quality) of food source represents fitness. There are three categories of honey bees in the hive with respect to their work assignments, namely employed bees, onlooker bees and scout bees. Employed and onlooker bees collect nectar from the food sources. The bee associated to abandoned food source becomes scout bee. The scout bee searches new food sources in different directions and creates fluctuations in the search process. Thus the search space is exploited by the onlooker and employed bees, while the exploration of the search space is done by scout bees.

The dominance of random components in the position update strategy of ABC tends to explore the search space at the expense of the possibility of skipping true solution. Researchers are continuously trying to establish a proper trade-off between the exploitation and exploration capabilities to improve the performance of ABC algorithm.

To improve the exploitation property of search space, Wei-feng Gao et al. [7] proposed a new search strategy called ABC/best/1, associated with a novel chaotic initialization technique. In this search strategy, solution updates itself in the neighborhood of the previous best solution. To improve the convergence characteristics of ABC, Anan Banharnsakun et al. [8] proposed a modified search equation for onlooker bees. In this equation, onlooker bees follow the direction of the best-so-far solution in a biased way, instead of a randomly selected neighbor one. Li et al. [9] improved the search process of ABC by using best-so-far solution, inertia weight, and acceleration coefficients. In this ABC model, a pool of different search equations is used to produce multiple new solutions, in which the best one is selected by the greedy approach. Alatas [10] proposed chaotic ABC, in which the parameter adaptation of ABC is done by chaotic maps and scout bee uses the chaotic search to explore new regions of search space. To improve the exploitation ability of ABC, Wei-feng Gao et al. [11] used Powell's method as a local search tool. Karaboga and Gorkemli [12] introduced a more accurate behavior of onlooker bees to improve the local search ability of ABC. Kiran et al. [13] used five search strategies and counters for updating the solutions of ABC. Sharma et al. [14] introduced Levy flight random walk inspired search strategy as a local search to improve the exploitation ability of ABC.

To solve constrained optimization problems, D. Karaboga and B. Akay [15] proposed a new variant of ABC, namely Modified ABC (MABC) in which the selection scheme of ABC is replaced by Deb's selection rules [16]. Mezura-Montes et al. [5] proposed elitist artificial bee colony. In the proposed approach, equality constraints are controlled by dynamic tolerance mechanism to promote the exploration of search space. Bravejic et al. [17] presented an improved version of ABC, in which the very first swarm of candidate solutions is initialized randomly, after that all further swarms contain the best solution of their previous swarm. Mezura-Montes et al. [18] presented a novel algorithm in which the exploitation of search space is done in the vicinity of the best solution of the current swarm and constraints are handled by ϵ- constrained approach. In order to improve the efficiency of ABC in constrained search space, Brajevic [19] proposed a crossover-based ABC in which dynamic tolerance is used to handle the equality constraints and improved boundary constraint-handling method is employed.

In 2010, Zhu and Kwong [20] proposed an improved ABC algorithm, namely Gbest-guided ABC (GABC) in which Gbest (the best individual in the swarm) is incorporated into the position update equation to improve the exploitation ability of the search space. Since all the solutions which are going to be updated, move towards the Gbest solution, there is enough chance to trap in local optima. On the other hand, the performance of GABC is not good enough to establish its competitiveness to solve the constrained optimization problems.

To overcome this deficiency of GABC and make it efficient for constrained optimization problems, a new variant of GABC namely, Modified GABC (MGABC) is introduced.

Rest of the paper is organized as follows. Section 2 describes the standard ABC. A brief description of MGABC is introduced in Section 3. Section 4 describes the properties of considered benchmark functions, adopted experimental setting, constraint handling and experiment results with a comparative study. In section 5 optimal power flow problem with three different objective functions has been solved by MGABC to verify its applicability for solving the real world problems. Finally, the conclusion is given in section 6.

2. Standard ABC

Artificial bee colony algorithm is inspired from forging behavior of honey bees. In ABC, the position of a food source corresponds to a possible solution to the optimization problem and the quality (nectar amount) of each food source represents its fitness of the associated solution.

Initially, ABC generates a uniformly distributed initial swarm of SN food sources (potential solutions), where the dimension of each food source x_i $(i = 1, 2, ...SN)$ is D, which is equal to the number of variables in the optimization problem. Each solution is initialized as follows:

$$x_{ij} = x_{minj} + rand\,(0,1)\,(x_{maxj} - x_{minj}) \tag{2}$$

Where x_i is the i^{th} solution in the swarm. x_{minj} and x_{maxj} are the lower and upper bounds of x_i in the j^{th} dimension and rand $(0,1)$ is a uniformly distributed random number in the range $[0,1]$.

These randomly initialized solutions are updated in employed bee phase as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{3}$$

Where $j \in \{1,2,...,D\}$ and $k \in \{1,2,...,SN\}$. k is picked randomly and different from i. ϕ_{ij} is a random number in the range $[-1,1]$. After producing new position, bee replaces its position with the position having better fitness between the solution x_{ij} and newly generated solution v_{ij}. Employed bees measure the fitness (fit_i) of each food source (x_i) using its objective value (f_i) as follows:

$$fit_i = \begin{cases} \frac{1}{1+f_i}, & f_i \geqslant 0 \\ 1 + abs(f_i), & f_i < 0 \end{cases} \tag{4}$$

After exploiting the food sources, the employed bees return to the hive and share the information about the position and quality of food sources with onlooker bees through various dances at various parts of the hive.

In onlooker bee phase, this information is evaluated by Onlooker bees and they select a food source x_i with a probability $prob_i$, based on its fitness. There can be several approaches to calculate $prob_i$ but in every approach, it should be a function of fitness. Here $prob_i$ is calculated as:

$$prob_i = \frac{0.9 \times fit_i}{maxfit} + 0.1 \tag{5}$$

Here, maxfit is the maximum fitness of the swarm. Onlooker bee selects a food source based upon its probability and modifies its position using same position update equation (3) as the employed bee used. The onlooker bee memorizes that position which obtained after applying the greedy approach between previous and new solution. The number of the employed bees and onlooker bees is same and equal to the number of food sources.

If any individual food source is not getting exploited any more up to a predefined number of cycles (also known as the limit), then that food source is considered to be abandoned. The bee involved with this abandoned food source turns into scout bee. The scout bee reinitializes the abandoned food source inside the predefined search space using equation (2).

3. Modified Gbest based Artificial bee colony

The search strategy of ABC is determined by the position update equation (3), in which, the solution moves towards or away from another randomly selected solution to produce a new solution. The solution will move towards an inferior or superior solution, depending upon the quality of a randomly selected solution. Now since the probability of this randomly selected solution x_{kj} being good or bad in equation (3) is equal, therefore the selection of random solution plays an important role in the success of ABC [20]. Thus, the presence of random coefficient ϕ_{ij} and randomly selected solution x_{kj} make, ABC search process more random than required. Hence a balanced random search in ABC may improve its performance.

To overcome this weakness, Zhu and Kwong [20] proposed a new variant of ABC, namely Gbest-guided ABC (GABC) by introducing the global best term in the position update equation (3) of ABC which, in fact, is inspired by particle swarm optimization (PSO) [3]. The position update equation of GABC is

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(y_j - x_{ij}) \tag{6}$$

Here ψ_{ij} is a normally distributed number in $[0, K]$, where K is a non-negative constant. y_j is the element of the Gbest solution in j^{th} dimension. According to equation (6), every individual which is going to be updated, moves towards the Gbest solution found in the current swarm which improves the exploitation ability of the search space but this ability comes at a cost, it may cause premature convergence and thus stagnation.

In the onlooker bee phase of GABC, position is updated according to the fitness based probability. Therefore, all the individuals having better fitness converge around the Gbest solution which may ultimately, cause premature convergence. Although GABC improves the efficiency of ABC algorithm, significantly but fails to establish its competitiveness to deal with the constrained optimization problems.

Therefore, to avoid the possibilities of premature convergence and to make GABC robust for constrained optimization problems, a new variant of GABC, namely Modified GABC (MGABC) is proposed.

In the proposed MGABC, instead of one position update equation, two new position update equations are introduced for employed bee phase and onlooker bee phase as follows:

1. Position update process of MGABC for employed bee phase is shown in Algorithm 1

Algorithm 1 Position update in Employed bee phase of MGABC

---

for $i = 1, \ldots, SN$ **do**
  for $j = 1, \ldots, D$ **do**

$$v_{ij} = x_{ij} + \sigma_i \times (x_{ij} - x_{kj}) + \psi_{ij} \times (y_j - x_{ij}) \tag{7}$$

  end for
end for

   Here, σ_i is a uniform random number between $[-2 + prob_i, 2 - prob_i]$, ψ_{ij} is same as defined in equation (6) and rest of the symbols have their usual meaning.

2. Position update process of MGABC for onlooker bee phase, is shown in Algorithm 2:

Algorithm 2 Position update in Onlooker bee phase of MGABC

---

for $i = 1, \ldots, SN$ **do**

$$v_{ij} = x_{ij} + \xi \times (x_{ij} - x_{kj}) + \eta_i \times (y_j - x_{ij}) \tag{8}$$

end for

   Here, ξ and η_i are uniformly distributed number in $[-1, 1]$ and $[0, 2 - prob_i]$, respectively.

In the first modification (Algorithm 1), the second term in the right-hand side of the equation (7), is responsible for enhancing the exploration ability of the search process. The solution x_{ij} moves towards or away from the randomly selected solution x_{kj} with a step size scaled by σ_i. A solution x_i having less fitness will also have less fitness probability ($prob_i$). A solution having less fitness probability will have a large interval for choosing σ_i, it is more probable that σ_i takes large values (either positive or negative) in the interval. This value will force the solution (less fit solution) to move with large step size towards or away from the randomly selected solution. On the contrary, a solution having large fitness probability will have a small interval for σ_i. Due to the small value of σ_i, the solution will move with a small step size towards or away from the randomly selected solution. Therefore the update equation (7) will produce large changes in less fit solutions and small changes in more fit solutions. Therefore the search process produced by employed bee update equation (7) will have a better exploration ability. Further, in this modification, solution updates itself in all possible dimensions, this improves its perturbation rate and hence the exploration ability of search process. The third term maintains the convergence speed of the algorithm as in GABC. Thus in MGABC convergence speed with exploration capability is better maintained.

In the second modification (Algorithm 2), the third term in the right-hand side of equation (8) improves the exploitation of the search process in which a solution moves towards the Gbest solution with a step size scaled by η_i. A solution having less fitness probability will have a large interval for η_i. Due to this comparatively large interval, the possibility for η_i to attain large values is increased. Since this interval contains only non-negative real numbers, therefore, the solution will move towards the Gbest solution with large step size. On the other hand, a solution having large fitness probability will have a small interval for η_i, implies that it will move with a small step size towards the Gbest solution. Therefore the improved search strategy of onlooker bee phase will have a better exploitation ability. In short, the proposed MGABC reduces the demerits of GABC while maintaining the merits of GABC.

4. Experimental results and discussion

4.1. Benchmark Problems under consideration

To verify the performance of proposed algorithm MGABC, it is tested over 3 engineering design problems and 19 test constrained problems of CEC-2006 [21] along with one constrained optimization problem [22]. The properties and characteristics of considered engineering design problems (Φ_1-Φ_3) and CEC-2006 test bed (g_1-g_{19}) including one constrained optimization problem (g_{20}) are given in Tables 1 and 2, respectively.

Table 1

Engineering design problems (f^* denotes optimal value and AE denotes Acceptable Error)

Test Problem	Decision Variable	f^*	AE
Coil Compression Spring (Φ_1) [14]	3	2.6254	1.0E-04
Pressure Vessel (Φ_2) [14]	4	7197.729	1.0E-05
Welded Beam (Φ_3) [14]	4	1.724852	1.0E-01

4.2. Experimental setting

To prove the efficiency of MGABC, it is compared with basic ABC and recent variants of ABC, named Best-So-Far ABC (BSFABC) [8], Gbest-guided ABC (GABC) [20], Modified ABC (MABC) [23] and Levy flight ABC (LFABC) [14] over the benchmark problems given in section 4.1, with the following parameter setting:

4.2.1. Parameters setting for engineering design problems (Φ_1-Φ_3)

- Swarm size NP =50,
- Number of food sources SN = NP/2,
- limit=Dimension×Number of food sources=D×SN [14, 23],
- The number of simulations/run =100,
- Maximum number of function evaluations =200000,
- The stopping criteria is either maximum number of function evaluations is reached or $|f - f^*| \leq$ Acceptable Error, where f is the obtained value of an algorithm for any given problem and f^* is given in Table 1.
- Other than these parameters, other parameters are set as per their original papers.

Table 2

CEC-2006 constrained benchmark functions (g_1-g_{19}) [21] and constrained optimization problem (g_{20}) [22] (N_{eq} denotes the number of equality constraints, N_{ieq} denotes the number of inequality constraints and N_a is the number of active constraints)

Test problem	**Dim**	**Type of function**	N_{eq}	N_{ieq}	**Optimal functional value**	N_a
g_1	13	Quadratic	0	9	-15.000000	6
g_2	20	Nonlinear	0	2	-0.8036191042	1
g_3	10	Polynomial	1	0	-1.000500100	1
g_4	5	Quadratic	0	6	-30665.53867178	2
g_5	4	Cubic	3	2	5126.4967140071	3
g_6	2	Cubic	0	2	-6961.81387558	2
g_7	10	Quadratic	0	8	24.3062090681	6
g_8	2	Nonlinear	0	2	-0.0958250415	0
g_9	7	Polynomial	0	4	680.6300573745	2
g_{10}	8	Linear	0	6	7049.2480205286	6
g_{11}	2	Quadratic	1	0	0.749900000	1
g_{12}	3	Quadratic	0	1	-1.000000	0
g_{13}	5	Nonlinear	3	0	0.0539415140	3
g_{14}	10	Nonlinear	3	0	-47.7648884595	3
g_{15}	3	Quadratic	2	0	961.7150222899	2
g_{16}	5	Nonlinear	0	38	-1.905155258	4
g_{17}	6	Nonlinear	4	0	8853.539675	4
g_{18}	9	Quadratic	0	13	-0.866025	6
g_{19}	2	linear	0	2	-5.508013	2
g_{20}	2	Quadratic	1	1	1.3935	2

4.2.2. Parameters setting for constrained problems (g_1-g_{20})

- Swarm size NP =100
- Number of food sources SN = NP/2,
- The number of simulations/run =30 [21],
- Maximum number of function evaluations =240000 [21].
- All the other setting of considered algorithms are same as for engineering design problems.

4.2.3. Constraint handling

To solve constrained optimization problems, constraint (either equality or inequality) handling plays a vital role to obtain the feasible solution. In this paper, adaptive penalty function approach of constraint handling is used for all experiments.

If $\mathbb{S}$ is the feasible search space and $\vec{x}$ is the solution vector obtained by the proposed approach then the objective function F in the constrained optimization problem (1) is written as:

$$F(\vec{x}) = \begin{cases} f(\vec{x}), & \text{if } \vec{x} \in \mathbb{S} \\ f(\vec{x}) + c\,(g_j(\vec{x}) \text{ or } h_j(\vec{x})), & \text{if } \vec{x} \notin \mathbb{S} \end{cases} \tag{9}$$

Here, c is a penalty value which is multiplied by the violated constraint j. Usually, c is assigned a very large value, in the case of minimizing the objective function.

4.3. Results and statistical Analyses of Experiments

4.3.1. Engineering design problems (Φ_1-Φ_3)

Table 3 presents the experimental results of proposed and considered algorithms over the engineering design problems. In this table, a comparative analysis is made in terms of standard deviation (*SD*), mean error (*ME*), average number of function evaluations (*AFEs*) and success rate (*SR*). *ME*, *SR* and *AFEs* exhibit the accuracy, reliability and efficiency of an algorithm, respectively. The superiority of an algorithm over others based on a particular criterion is presented by the bold entry. It is clear that out of 3 problems, in 2 problems MGABC is better than any other algorithm on all criteria. While in one problem it is more reliable and efficient than any other algorithm. Statistical visualization through boxplots has also been carried out.

Table 3

Results: Engineering Design Problems (**TP** denotes Test Problem under consideration)

TP	Algorithm	SD	ME	AFEs	SR
Φ_1	MGABC	**2.68E-03**	**1.96E-03**	**105926.96**	**75**
	ABC	1.04E-02	1.31E-02	194782.77	7
	BSFABC	5.14E-03	3.06E-02	198120.28	1
	GABC	6.03E-03	5.52E-03	173978.67	31
	MABC	4.71E-03	4.24E-03	171571.53	30
	LFABC	1.92E-02	1.84E-02	164327.6	22
Φ_2	MGABC	**3.69E-05**	**3.20E-05**	**89659.43**	**60**
	ABC	9.28E+00	1.59E+01	200025.31	0
	BSFABC	2.69E+01	3.07E+01	200036.44	0
	GABC	4.13E+00	6.89E+00	200027.81	0
	MABC	9.32E+00	1.52E+01	200021.94	0
	LFABC	6.05E-02	3.25E-02	198352.4	3
Φ_3	MGABC	4.32E-03	9.50E-02	**2895.67**	**100**
	ABC	6.89E-02	1.93E-01	195217.6	5
	BSFABC	4.99E-03	**9.46E-02**	48955.61	**100**
	GABC	8.63E-03	9.64E-02	99808.18	83
	MABC	**4.17E-03**	9.53E-02	34054.72	99
	LFABC	6.04E-03	9.87E-02	39570.51	99

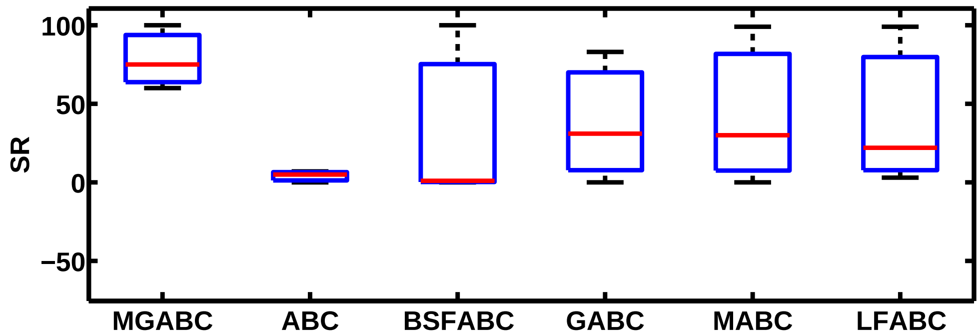


Fig. 1. Boxplots (Success Rate (SR))

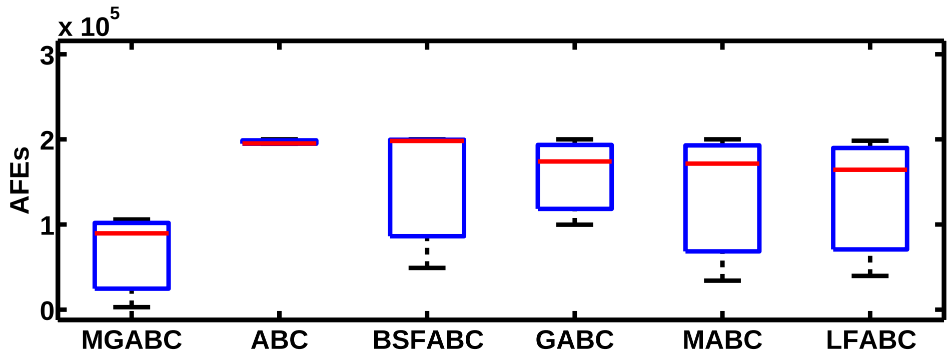


Fig. 2. Boxplots (Average Number of Function Evaluations (AFEs))

Boxplot analyses [14] has been done for *SR*, *AFEs* and *ME*. The boxplots for MGABC, ABC, BSFABC, GABC, MABC and LFABC are shown in Fig. 1, Fig. 2 and Fig. 3 with respect to *SR*, *AFEs* and *ME*, respectively. High interquartile range and median in boxplot analysis of *SR* (Fig. 1) and low interquartile range and median in boxplot analysis of *AFEs* and *ME* (Fig. 2 and 3) verify that the proposed algorithm is more reliable, cost-effective and accurate as compared to other considered algorithms.

It is clear from the boxplots graphs (Fig. 1, Fig. 2 and Fig. 3) that the data corresponding to considered algorithms are not normally distributed. Therefore, a non-parametric statistical test Mann-Whitney U rank sum [14] is applied to evaluate the significant difference among these data. In this study, this test is performed on the data of *AFEs* at 5% level of significance ($\alpha = 0.05$)

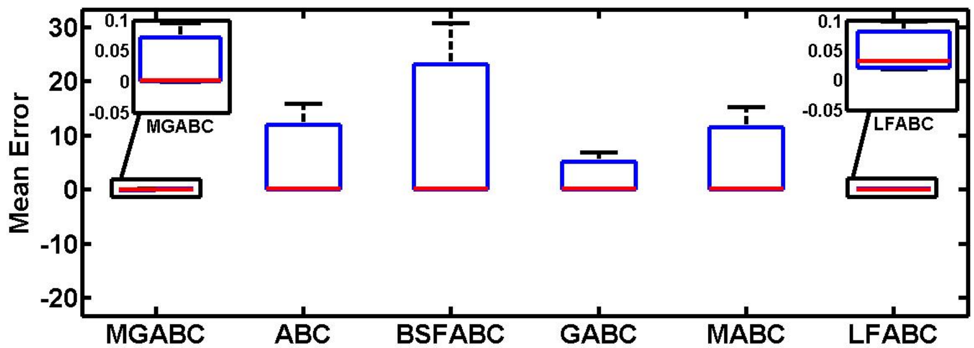


Fig. 3. Boxplots (Mean Error (ME))

with the null hypothesis, 'There is no significant difference in the data', between MGABC-ABC, MGABC-BSFABC, MGABC-GABC, MGABC-MABC, and MGABC-LFABC.

Table 4 presents the results of Mann-Whitney U rank sum test for *AFEs* of 100 runs. In Table 4, '-' or '+' sign shows that MGABC has more (worse performance) or less (better performance) *AFEs* than considered algorithms, while '=' sign shows that there is no significant difference between compared algorithms. In Table 4, 15 '+' sign out of 15 comparisons assure that MGABC is better than other considered algorithms.

As far as the convergence speed is concerned, the proposed algorithm is compared with the considered algorithms using Acceleration Rate (AR) criterion [14]. In AR, the comparison is made by the ratio between the AFEs of proposed (MGABC) and considered algorithm (ALGO), and defined as:

$$AR = \frac{AFEs_{ALGO}}{AFEs_{MGABC}}, \tag{10}$$

Here, ALGO $\in$ {ABC, BSFABC, GABC, MABC, LFABC} and $AR > 1$ means MGABC is faster than the compared algorithm (ALGO). Table 5 presents a comparison between MGABC and ABC, MGABC and BSFABC, MGABC and GABC, MGABC and MABC, and MGABC and LFABC in terms of AR. These results confirm that MGABC has better convergence speed as compared to other considered algorithms.

Table 4

Comparison based on *AFEs* of 100 runs using Mann-Whitney U rank sum test at $\alpha = 0.05$ significance level, TP: Test Problem.

TP	U rank sum test with MGABC Vs				
	ABC	BSFABC	GABC	MABC	LFABC
Φ_1	+	+	+	+	+
Φ_2	+	+	+	+	+
Φ_3	+	+	+	+	+

Table 5

Acceleration Rate (AR) of MGABC as compared to ABC, BSFABC, GABC, MABC and LFABC, TP: Test Problems

TP	ABC	BSFABC	GABC	MABC	LFABC
Φ_1	1.84	1.87	1.64	1.62	1.55
Φ_2	2.23	2.23	2.23	2.23	2.21
Φ_3	67.42	16.90	34.47	11.76	13.67

4.3.2. Constrained problems (g_1-g_{20}):

Table 6

Minimization results for constrained problems (g_1-g_{20}) (**TP** denotes Test Problem under consideration and **SD** symbolizes Standard Deviation)

TP	Algorithm	Mean Value	Best Value	Worst Value	SD
g_1	MGABC	-13.55354	**-15**	-9	1.636944
	ABC	**-15.000000**	**-15.000000**	**-15.000000**	**0.000000**
	BSFABC	**-15.000000**	**-15.000000**	**-15.000000**	**0.000000**
	GABC	**-15.000000**	**-15.000000**	**-15.000000**	**0.000000**
	MABC	**-15.000000**	**-15.000000**	**-15.000000**	**0.000000**
	LFABC	**-15.000000**	**-15.000000**	**-15.000000**	**0.000000**
g_2	MGABC	-0.7890629	**-0.8036108**	-0.7604863	0.01196791
	ABC	-0.635357	-0.673116	-0.589790	0.022018
	BSFABC	-0.629568	-0.665515	-0.589329	0.017605
	GABC	-0.661051	-0.721057	-0.629680	0.024026
	MABC	**-0.802104**	-0.803250	**-0.797467**	**0.001217**
	LFABC	-0.654621	-0.702064	-0.602961	0.018694
g_3	MGABC	**-1.000383**	**-1.0004**	**-1.000258**	**4.10816E-05**
	ABC	-0.171238	-0.413404	-0.039678	0.079140
	BSFABC	-0.082431	-0.258418	-0.002342	0.070550
	GABC	-0.291008	-0.660630	-0.130695	0.110260
	MABC	-0.899948	-0.966073	-0.758138	0.049395
	LFABC	-0.478242	-0.778276	-0.243808	0.134006
g_4	MGABC	**-30665.54**	**-30665.54**	**-30665.54**	**1.04809E-11**
	ABC	-30465.140000	-30611.570000	-30255.230000	100.883800
	BSFABC	-30603.010000	-30660.500000	-30496.130000	38.142130
	GABC	-30591.390000	-30655.920000	-30511.290000	37.555720
	MABC	-30665.010000	-30665.450000	-30662.530000	0.520758
	LFABC	-30621.73	-30665.53	-30529.16	36.556420
g_5	MGABC	5467.756	**5126.497**	6112.169	330.8681
	ABC	5159.486000	5133.072000	5232.318000	22.938990
	BSFABC	5145.052000	5127.879000	5198.292000	19.156530

Table 6 Continued:

TP	Algorithm	Mean Value	Best Value	Worst Value	SD
	GABC	5159.314000	5130.135000	5250.816000	30.480630
	MABC	5151.738000	5128.447000	5220.837000	22.838680
	LFABC	**5129.588**	5126.499	**5148.278**	**6.224214**
g_6	MGABC	-6959.489	**-6961.803**	-6957.123	1.176998
	ABC	-6910.779000	-6959.298000	-6546.435000	89.944270
	BSFABC	-6871.868000	-6955.590000	-6737.955000	46.218550
	GABC	-6955.523000	-6960.822000	-6938.135000	4.362150
	MABC	-6945.104000	-6955.760000	-6925.760000	7.988857
	LFABC	**-6961.11800**	-6961.799	**-6958.094**	**0.986294**
g_7	MGABC	24.78064	**24.32653**	25.09927	0.3122748
	ABC	27.613380	25.266610	29.614430	1.013112
	BSFABC	29.826280	26.545150	36.323910	2.480659
	GABC	26.278930	24.887590	28.239230	0.670455
	MABC	**24.663930**	24.405660	**25.029870**	**0.125392**
	LFABC	26.524650	24.906590	28.673970	0.983646
g_8	MGABC	**-0.095825**	**-0.095825**	**-0.095825**	**0.000000**
	ABC	**-0.095825**	**-0.095825**	**-0.095825**	**0.000000**
	BSFABC	**-0.095825**	**-0.095825**	**-0.095825**	**0.000000**
	GABC	**-0.095825**	**-0.095825**	**-0.095825**	**0.000000**
	MABC	**-0.095825**	**-0.095825**	**-0.095825**	**0.000000**
	LFABC	**-0.095825**	**-0.095825**	**-0.095825**	**0.000000**
g_9	MGABC	**680.6309**	**680.6302**	**680.6322**	**0.000512664**
	ABC	684.069300	681.980200	684.949000	0.767071
	BSFABC	684.343100	682.668500	685.263000	0.522194
	GABC	683.142200	681.626500	684.641400	0.954408
	MABC	680.752200	680.652300	680.822400	0.047177
	LFABC	682.503700	681.118100	684.404900	0.918807

Table 6 Continued:

TP	Algorithm	Mean Value	Best Value	Worst Value	SD
g_{10}	MGABC	**7357.461**	**7104.006**	**7504.944**	**121.9031**
	ABC	10546590.000000	7194.580000	316161000.000000	56751180.000000
	BSFABC	44148.990000	7380.079000	1075328.000000	191487.200000
	GABC	7669.448000	7167.390000	8264.672000	320.780000
	MABC	7461.740000	7199.169000	8264.599000	214.768100
	LFABC	7761.153	7237.139	7364.458	557.801600
g_{11}	MGABC	**0.750025**	**0.749995**	0.750127	0.000034
	ABC	0.750551	0.750002	0.752860	0.000767
	BSFABC	0.750630	0.750015	0.754634	0.000971
	GABC	0.750031	**0.749995**	0.750194	0.000050
	MABC	0.750587	0.750026	0.751924	0.000512
	LFABC	0.750027	**0.749995**	**0.750062**	**0.000013**
g_{12}	MGABC	**-1.000000**	**-1.000000**	**-1.000000**	**0.000000**
	ABC	**-1.000000**	**-1.000000**	**-1.000000**	**0.000000**
	BSFABC	**-1.000000**	**-1.000000**	**-1.000000**	**0.000000**
	GABC	**-1.000000**	**-1.000000**	**-1.000000**	**0.000000**
	MABC	**-1.000000**	**-1.000000**	**-1.000000**	**0.000000**
	LFABC	**-1.000000**	**-1.000000**	**-1.000000**	**0.000000**
g_{13}	MGABC	0.171074	**0.05394861**	0.4377867	0.1746232
	ABC	0.075318	0.058870	0.105832	0.012775
	BSFABC	0.085440	0.056302	0.158985	0.028497
	GABC	**0.060251**	0.054121	0.080656	0.006275
	MABC	0.162756	0.066506	0.307423	0.063837
	LFABC	0.060283	0.055041	**0.072824**	**0.005037**
g_{14}	MGABC	-47.246220	-47.675860	-46.465260	0.285642
	ABC	-47.023740	-47.704730	-46.084640	0.400683
	BSFABC	-46.839350	-47.597790	-45.654510	0.500343
	GABC	**-47.506650**	**-47.740360**	**-47.201370**	**0.140430**
	MABC	-46.112750	-47.286820	-45.202270	0.487249

Table 6 Continued:

TP	Algorithm	Mean Value	Best Value	Worst Value	SD
	LFABC	-47.314140	-47.625000	-46.499870	0.297742
g_{15}	MGABC	962.173700	**961.715100**	965.208600	0.777151
	ABC	963.296200	961.881200	967.319000	1.043270
	BSFABC	962.261100	961.740500	963.848500	0.569401
	GABC	961.903400	961.724900	962.899900	0.266831
	MABC	963.834800	961.752800	967.112100	1.615972
	LFABC	**961.788800**	961.715500	**962.1733300**	**0.113430**
g_{16}	MGABC	**-1.905155**	**-1.905155**	**-1.905155**	**0.000000**
	ABC	-1.629697	-1.864421	-1.483864	0.087453
	BSFABC	-1.680447	-1.852251	-1.481733	0.106812
	GABC	-1.864507	-1.896252	-1.828660	0.018879
	MABC	-1.889544	-1.902206	-1.856813	0.010544
	LFABC	-1.829377	-1.904851	-1.607071	0.064475
g_{17}	MGABC	8915.998	**8853.53**	9241.82	70.82186
	ABC	8905.314	8861.847	8944.38	20.33432
	BSFABC	8913.008	8859.945	8952.708	24.43911
	GABC	**8872.048**	8853.679	**8930.157**	**17.45531**
	MABC	8940.914	8885.975	8987.502	25.58631
	LFABC	8873.469000	8853.89700	8943.6800	23.110250
g_{18}	MGABC	**-0.8657735**	**-0.8660253**	**-0.8648695**	**0.000299329**
	ABC	-0.8417745	-0.855895	-0.7976417	0.01479391
	BSFABC	-0.8367012	-0.8626372	-0.7439228	0.02714512
	GABC	-0.8610056	-0.8653894	-0.8519819	0.003983532
	MABC	-0.8576567	-0.8646378	-0.826246	0.007096925
	LFABC	-0.860475	-0.865095	-0.846507	0.003888
g_{19}	MGABC	**-5.508013**	**-5.508013**	**-5.508013**	**1.77636E-15**
	ABC	-5.504064	-5.507353	-5.497827	0.002439545
	BSFABC	-5.493959	-5.506947	-5.466542	0.009757688
	GABC	-5.505612	-5.507465	-5.502027	0.001440591

Table 6 Continued:

TP	Algorithm	Mean Value	Best Value	Worst Value	SD
	MABC	-5.507967	-5.508011	-5.507868	3.32596E-05
	LFABC	-5.507860	-5.508007	-5.506985	0.000238
	MGABC	**1.394359**	**1.393571**	**1.399163**	**1.19E-03**
	ABC	1.419499	1.398581	1.449173	1.46E-02
	BSFABC	1.467479	1.395309	1.662487	6.64E-02
g_{20}	GABC	1.407121	1.394780	1.442717	9.76E-03
	MABC	1.495488	1.425206	1.661920	5.25E-02
	LFABC	1.404975	1.393600	1.426981	1.02E-02

The proposed algorithm MGABC is re-evaluated over these problems based on CEC 2006 criteria of comparison. The algorithms are compared based on the mean, best, worst and SD of the objective function values. Table 6 presents the experimental results. Bold entries represent the superiority of the algorithm.

As shown in Table 6, for 9 problems (g_3, g_4, g_9, g_{10}, g_{11}, g_{16}, g_{18}, g_{19} and g_{20}) MGABC outperforms others in terms of mean value. According to best value criteria, MGABC has superior results than others in 15 problems (g_2, g_3, g_4, g_5, g_6, g_7, g_9, g_{10}, g_{13}, g_{15}, g_{16}, g_{17}, g_{18}, g_{19} and g_{20}). Among all the criterion of comparison, MGABC proves its supremacy on 8 problems (g_3, g_4, g_9, g_{10}, g_{16}, g_{18}, g_{19} and g_{20}). While for g_8 and g_{12}, MGABC obtains the same results as others.

Furthermore, To find the significant difference in these results, Mann-Whitney rank sum test is carried out with the same setting as section 4.3.1. The data sets are the best value of each run. Table 7 shows the results of the aforesaid test for the best value of 30 runs. Out of 100 comparisons, 63 '+' signs confirm that MGABC is significantly better than other considered algorithms.

5. Application of MGABC in solving Optimal Power Flow Optimization Problem

5.1. Optimal Power Flow (OPF) optimization problem

OPF problem is a well-known constrained optimization problem in the field of electrical engineering, in which the control variables of a power system are required to be optimized while all physical and operational restrictions are satisfied such that the fuel cost of the system is minimized

Table 7

Comparison based on best value of 30 runs for constrained problems (g_1-g_{20}) using Mann-Whitney U rank sum test at $\alpha = 0.05$ significance level, TP: Test Problem.

TP	U rank sum test with MGABC Vs				
	ABC	BSFABC	GABC	MABC	LFABC
g_1	-	-	-	-	-
g_2	+	+	+	-	+
g_3	+	+	+	+	+
g_4	+	+	+	+	+
g_5	-	-	-	-	-
g_6	+	+	+	+	-
g_7	+	+	+	=	+
g_8	=	=	=	=	=
g_9	+	+	+	+	+
g_{10}	+	+	+	+	+
g_{11}	+	+	=	+	=
g_{12}	=	=	=	=	=
g_{13}	-	-	=	-	-
g_{14}	+	+	-	+	-
g_{15}	+	+	=	+	+
g_{16}	+	+	+	+	+
g_{17}	=	=	-	+	+
g_{18}	+	+	+	+	+
g_{19}	+	+	+	+	=
g_{20}	+	+	+	+	+

[24].

Mathematically, the general OPF problem can be defined as:

$$MinJ(\alpha, \beta) \tag{11}$$

subject to:

$$D(\alpha, \beta) = 0 \tag{12}$$

$$U^{min} \leq U(\alpha, \beta) \leq U^{max} \tag{13}$$

Here, J is the objective function of two variables α and β. α is the vector for control variables (independent variables) defined as:

$$\alpha = [M_{g2}....M_{gng}, N_{g1}....N_{gng}, W_{c1}....W_{cnc}, P_1....P_{nt}] \tag{14}$$

Where,

- (M_g) is the generator real powers except at slack bus,
- (N_g) is the generator bus voltages,
- (W_c) is the shunt VAR compensation,
- (P) is the transformer tap settings,
- nc is the number of shunt VAR compensators,
- nt is the number of regulating transformers.

β is the vector of state variables (dependent variables) defined as:

$$\beta = [M_{g1}, N_{l1}....N_{lnl}, W_{g1}....W_{gng}, R_{l1}....R_{lNl}] \tag{15}$$

Where,

- (M_{g1}) is the generator active power at slack bus,
- (N_l) is the load bus voltages,

- (W_g) is the generator reactive powers,
- (R_l) is the transmission line loading (line flow),
- nl is the number of load buses,
- Nl is the number of transmission lines.

ng denotes the number of generators in both variables α and β.

$D(\alpha, \beta)$ is the set of equality constraints representing load flow equations as follows [24]:

$$M_{gi} - M_{di} - N_i \sum_{k=1}^{nb} N_k \; (D_{ik} \; \cos\theta_{ik} + B_{ik} \; \sin\theta_{ik}) = 0 \tag{16}$$

and

$$W_{gi} - W_{di} - N_i \sum_{k=1}^{nb} N_k \; (D_{ik} \; \sin\theta_{ik} + B_{ik} \; \cos\theta_{ik}) = 0 \tag{17}$$

Where,

- M_{gi} is the active power of i^{th} generator,
- W_{gi} is the reactive power of i^{th} generator,
- M_{di} is the active power demand of i^{th} bus,
- W_{di} is the reactive power demand of i^{th} bus,
- D_{ik} is the transfer conductance between buses i and k,
- B_{ik} is the transfer susceptance between buses i and k,
- θ_{ik} is the phase angle difference between the voltages at buses i and k,
- nb is the maximum number of bus bars.

$U(\alpha, \beta)$ is the set of system operational limiting constraints which includes following inequality constraints:

- Generator constraints:

$$M_{gi}^{min} \leq M_{gi} \leq M_{gi}^{max}, \text{ for } i = 1, 2, .., ng \tag{18}$$

$$W_{gi}^{min} \leq W_{gi} \leq W_{gi}^{max}, \text{ for } i = 1, 2, .., ng \tag{19}$$

$$N_{gi}^{min} \leq N_{gi} \leq N_{gi}^{max}, \text{ for } i = 1, 2, .., ng \tag{20}$$

- Security constraints:

$$N_{li}^{min} \leq N_{li} \leq N_{li}^{max}, \text{ for } i = 1, 2, .., nl \tag{21}$$

$$R_{li} \leq R_{li}^{max}, \text{ for } i = 1, 2, .., Nl \tag{22}$$

- Transformer constraints:

$$P_{i}^{min} \leq P_{i} \leq P_{i}^{max}, \text{ for } i = 1, 2, .., nt \tag{23}$$

- Shunt VAR compensator constraints:

$$W_{ci}^{min} \leq W_{ci} \leq W_{ci}^{max}, \text{ for } i = 1, 2, .., nc \tag{24}$$

Penalty function approach is applied for constrained handling [24]. Therefore, the modified objective function is written as:

$$\begin{aligned} \text{Min } J_{mod} = J(\alpha, \beta) + \lambda_m (M_{g1} - M_{g1}^{lim})^2 + \\ \lambda_n \sum_{i=1}^{nl} (N_{li} - N_{li}^{lim})^2 + \lambda_w \sum_{i=1}^{ng} (W_{gi} - W_{gi}^{lim})^2 \\ + \lambda_r \sum_{i=1}^{Nl} (R_{li} - R_{li}^{lim})^2 \end{aligned} \tag{25}$$

Here, λ_m, λ_n, λ_w and λ_r represent the penalty factors. a^{lim} denotes the limit value of a, where a can be any dependent variable from to M_{g1}, N_{li}, W_{gi} or R_{li}.

Table 8

Cost coefficients of generator in Case 1.

Cost coefficients	Bus Number					
	1	2	5	8	11	13
a	0.00	0.00	0.00	0.00	0.00	0.00
b	2.00	1.75	1.00	3.25	3.00	3.00
c	0.00375	0.01750	0.06250	0.00834	0.02500	0.02500

5.2. Test system under consideration

In order to validate the the wide applicability of MGABC algorithm, it is applied on standard IEEE 30-bus system. The configuration of IEEE 30-bus system is considered from [24] and defined as follows:

- Six generators at buses 1, 2, 5, 8, 11 and 13,
- Four transformers with off-nominal tap ratios in lines $6-9$, $6-10$, $4-12$ and $28-27$,
- Nine shunt VAR compensation buses at buses 10, 12, 15, 17, 20, 21, 23, 24 and 29.

The proposed MGABC has been applied to solve the OPF problem for three cases regarding to three different objective functions.

Case 1: Quadratic fuel cost function

In this case, generator cost characteristic is represented by a quadratic cost function of generator power output. Therefore, in equation(25), $J(\alpha, \beta)$ is taken as:

$$J(\alpha, \beta) = \sum_{i=1}^{ng} f_i(M_{gi}) = \sum_{i=1}^{ng}(a_i + b_i M_{gi} + c_i M_{gi}^2) \tag{26}$$

Where f_i denotes the fuel cost of i^{th} generator. The variables a_i, b_i and c_i represent the cost coefficients of i^{th} generator. The values of these variables are given in Table 8.

Table 9

Cost coefficients of generator in Case 2.

Bus Number	From MW	To MW	Cost coefficients		
			a	b	c
1	50	140	55.00	0.70	0.0050
	140	200	82.50	1.05	0.0075
2	20	55	40.00	0.30	0.0100
	55	80	80.00	0.60	0.0200

Case 2: Piecewise quadratic fuel cost functions

For this case, the generators associated with buses 1 and 2 possess two fuel options. In order to model different fuels, the cost characteristics of these generators are expressed as a piece-wise quadratic cost function.

$$f_i(M_{gi}) = \begin{cases} a_{i1} + b_{i1}M_{gi} + c_{i1}M_{gi}^2 \ , & M_{gi}^{min} \leq M_{gi} \leq M_{gi1} \\ a_{i2} + b_{i2}M_{gi} + c_{i2}M_{gi}^2 \ , & M_{gi1} \leq M_{gi} \leq M_{gi2} \\ \ldots\ldots\ldots\ldots & \\ a_{ik} + b_{ik}M_{gi} + c_{ik}M_{gi}^2 \ , & M_{gik-1} \leq M_{gi} \leq M_{gi}^{max} \end{cases} \tag{27}$$

Where a_{ik}, b_{ik} and c_{ik} are cost coefficients of the i^{th} generator for fuel type k. Therefore, in (25), $J(\alpha, \beta)$ is taken as:

$$J(\alpha, \beta) = \sum_{i=1}^{ng} f_i(M_{gi}) = \sum_{i=1}^{2} f_i(M_{gi}) + \sum_{i=3}^{ng}(a_i + b_iM_{gi} + c_iM_{gi}^2) \tag{28}$$

Where equation (27) is used to select the values of $f_i(M_{gi})$ for generators 1 and 2. The values of cost coefficients are given in Table 9, whereas the other generators take same cost coefficients as of case 1.

Case 3: Quadratic fuel cost function with valve-point effects

In this case, the generators of buses 1 and 2 are considered to have the valve point effects on their

Table 10

Cost coefficients of generator in Case 3.

Bus Number	M_{gi}^{min}	Cost coefficients				
		a	b	c	d	e
1	50	150.00	2.00	0.0016	50.00	0.0630
2	20	25.00	2.50	0.0100	40.00	0.0980

characteristics. The cost coefficients of these generating buses are taken from [24] and given in Table 10. The cost coefficients for other generators are same as of case 1.

The cost characteristics of generators associated with 1 and 2 are described as:

$$f_i(M_{gi}) = a_i + b_i M_{gi} + c_i M_{gi}^2 + |d_i \sin(e_i(M_{gi}^{min} - M_{gi}))| \quad \text{, where } i = 1 \text{ and } 2. \tag{29}$$

where a_i, b_i, c_i, d_i and e_i are cost coefficients of the i^{th} generating unit. Therefore, $J(\alpha, \beta)$ in the objective function (equation (25) is designed for this case as:

$$\begin{aligned} J(\alpha, \beta) = \sum_{i=1}^{ng} f_i(M_{gi}) = \sum_{i=1}^{2} (a_i + b_i M_{gi} + c_i M_{gi}^2 + \\ |d_i \sin(e_i(M_{gi}^{min} - M_{gi}))|) \\ + \sum_{i=3}^{ng} (a_i + b_i M_{gi} + c_i M_{gi}^2) \end{aligned} \tag{30}$$

5.3. Solving Optimal Power Flow problem using MGABC

To optimize the objective functions mentioned in section 5.2, MGABC initializes a swarm of SN food sources where each food source is defined as:
$\alpha = (M_{g2}....M_{gng}, N_{g1}....N_{gng}, W_{c1}....W_{cnc}, P_1....P_{nt})$. Each control variable of the individual food source α is bounded within predefined limits given in Table 11. In case of violation of limits of any control variable, modified objective function (equation 25) is considered for optimization [25].

Table 11

The bounds of control variables

Control variables	Min	Max	Control variables	Min	Max
M_1	50	200	P_{11}	0.9	1.1
M_2	20	80	P_{12}	0.9	1.1
M_5	15	50	P_{15}	0.9	1.1
M_8	10	35	P_{36}	0.9	1.1
M_{11}	10	30	W_{c10}	0	5
M_{13}	12	40	W_{c12}	0	5
N_1	0.95	1.1	W_{c15}	0	5
N_2	0.95	1.1	W_{c17}	0	5
N_5	0.95	1.1	W_{c20}	0	5
N_8	0.95	1.1	W_{c21}	0	5
N_{11}	0.95	1.1	W_{c23}	0	5
N_{13}	0.95	1.1	W_{c24}	0	5
			W_{c29}	0	5

Table 12

Load data.

Bus Number	Load		Bus Number	Load		Bus Number	Load	
	M	W		M	W		M	W
1	0.000	0.000	11	0.000	0.000	21	0.175	0.112
2	0.217	0.127	12	0.112	0.075	22	0.000	0.000
3	0.024	0.012	13	0.000	0.000	23	0.032	0.016
4	0.076	0.016	14	0.062	0.016	24	0.087	0.067
5	0.942	0.190	15	0.082	0.025	25	0.000	0.000
6	0.000	0.000	16	0.035	0.018	26	0.035	0.023
7	0.228	0.109	17	0.090	0.058	27	0.000	0.000
8	0.300	0.300	18	0.032	0.009	28	0.000	0.000
9	0.000	0.000	19	0.095	0.034	29	0.024	0.009
10	0.058	0.020	20	0.022	0.007	30	0.106	0.019

5.3.1. Parameters Setting for OPF Problem

To solve OPF problem using MGABC and ABC, the following setting is adopted:

- Colony size NP =50 and number of food sources SN= NP/2,
- The number of simulation/run=100,
- Maximum number of generations/iterations = 1000,
- Other parameters of MGABC and ABC are same as the given in Section 4.2,
- Tables 8, 9 and 10 represent the cost coefficients of case 1, 2 and 3, respectively.
- The bounds for control variables are adopted from [24] and are given in Table 11
- The load data are selected from [24] and listed in Table 12,
- The line data are adopted from [24],
- The penalty factors λ_m, λ_n, λ_w and λ_r are taken as 10^5.

5.3.2. Results Analysis and Discussion

The optimal control variables of IEEE 30 bus system with all three cases achieved by MGABC are given in Table 13. A fair comparison of MGABC with state-of-art algorithms and other nature-inspired algorithms has been presented in Tables 14-16 regarding minimum fuel cost and average minimum fuel cost over 100 runs. After analyzing the results of all three cases, it is clear that MGABC is more robust than ABC and other algorithms which have been taken into consideration. As far as the convergence speed is concerned, a comparative study is made (refer Fig. 4) between MGABC and ABC. It is clear from Fig. 4a, 4b, and 4c that MGABC has higher convergence rate than ABC to solve OPF problem for all three cases. Table 17 presents a comparison of MGABC with recent work for solving different cases of optimal power flow problem.

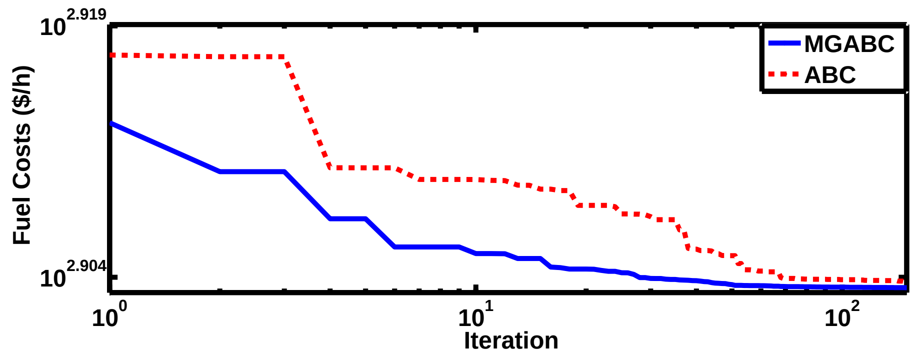
6. Conclusion

In this paper, two modified position update strategies have been proposed for employed and onlooker bee phases of GABC to make it more efficient for constrained optimization problems. These strategies produce an individual movement in the solution which is based upon its fitness probability.

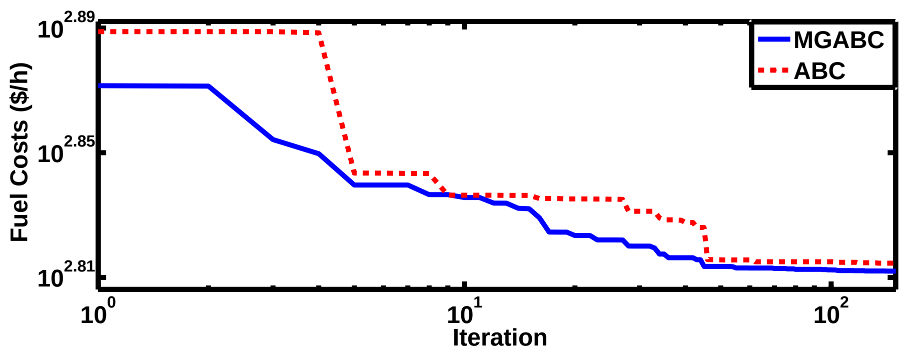
Table 13

Optimal values of control variables obtained by MGABC in all three cases.

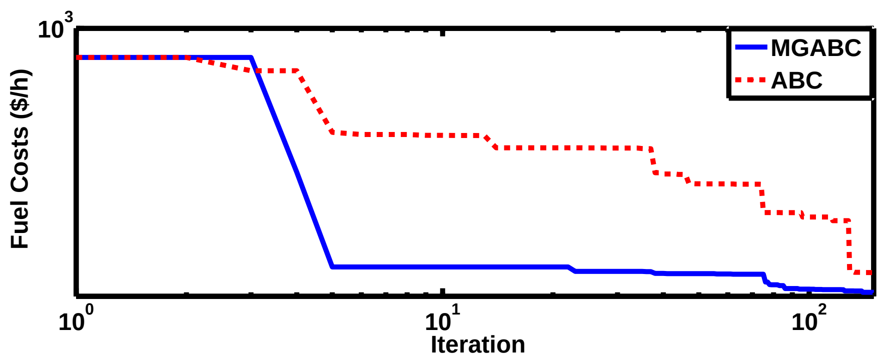
Control variables	Case 1	Case 2	Case 3
M_1	177.0576	139.982	199.5956
M_2	48.6906	55	20
M_5	21.3816	24.1716	21.9925
M_8	21.3193	34.99	26.6493
M_{11}	11.9632	18.4592	12.7639
M_{13}	12	17.527	12.0176
N_1	1.08372	1.07326	1.08395
N_2	1.04325	1.03865	1.0099
N_5	1.03308	1.03175	1.03018
N_8	1.03724	1.03951	1.04729
N_{11}	1.0787	1.06266	1.07703
N_{13}	1.04974	1.06294	1.05292
P_{11}	1.01432	0.993451	1.05282
P_{12}	0.957501	0.96414	0.908546
P_{15}	0.970988	0.997858	0.97705
P_{36}	0.973029	0.974064	0.973815
W_{c10}	5	5	4.89258
W_{c12}	2.90361	4.0749	0.179152
W_{c15}	3.2368	5	4.47957
W_{c17}	5	5	5
W_{c20}	4.35918	5	4.9673
W_{c21}	5	1.03988	4.99997
W_{c23}	4.86991	4.37257	3.80944
W_{c24}	5	5	4.99267
W_{c29}	2.66776	2.96206	2.60403
Fuel cost ($/h)	**800.4533**	**646.5207**	**918.7265**



(a) Case 1



(b) Case 2



(c) Case 3

Fig. 4. Convergence speed for MGABC and ABC over Case 1, Case 2 and Case 3

Table 14

Minimization results of MGABC along with different optimization methods for Case 1 of IEEE 30-bus system over 100 runs

Methods	Fuel cost ($/h)	
	Min	Average
ITS [24]	804.5560	-
EP [24]	802.6300	803.5100
IEP [24]	802.4650	802.5210
DE-OPF [24]	802.3940	-
MDE-OPF [24]	802.3760	802.3820
TS [24]	802.5020	-
TS/SA [24]	802.7880	-
SADE-ALM [24]	802.4040	-
Enhanced GA [24]	802.0600	-
PSO [24]	800.4890	-
ABC-OPF [24]	802.9086	803.6341
BBO [25]	800.4852	-
DBBO [25]	800.4564	-
MGABC	**800.4533**	**800.5765**

Table 15

Minimization results of MGABC along with different optimization methods for Case 2 of IEEE 30-bus system over 100 runs

Methods	Fuel cost ($/h)	
	Min	Average
ITS [24]	654.8740	-
EP [24]	647.7900	649.7000
IEP [24]	649.3120	650.2170
DE-OPF [24]	648.3840	-
MDE-OPF [24]	647.8460	648.3560
TS [24]	651.2460	-
TS/SA [24]	654.3780	-
PSO [24]	647.6900	-
GSA [24]	646.8480	646.8962
BBO [24]	647.7430	647.7645
ABC-OPF [24]	648.9124	649.4393
MGABC	**646.5207**	**646.6735**

Table 16

Minimization results of MGABC along with different optimization methods for Case 3 of IEEE 30-bus system over 100 runs

Methods	Fuel cost ($/h)	
	Min	Average
ITS [24]	969.1090	-
EP [24]	955.5090	959.3630
IEP [24]	953.5730	956.4600
DE-OPF [24]	931.0850	-
MDE-OPF [24]	930.793	942.5010
TS [24]	956.0000	-
TS/SA [24]	959.5630	-
EADDE [24]	930.745	-
SADE-ALM [24]	944.031	-
BBO [24]	919.7647	919.8389
GSA [24]	929.7260	930.9240
ABC-OPF [24]	930.4153	931.2629
MGABC	**918.7265**	**918.7912**

Table 17

Comparison of MGABC with recent work for solving different cases of optimal power flow problem

Algorithm	Case 1	Case 2	Case 3
Optimal Power Flow Using Differential Search Algorithm [26]	799.0943	647.9215	923.4573
ALC-PSO [27]	**797.1457**	——	**915.3243**
Dragonfly Algorithms [28]	800.6594	——	——
ABCGLN [24]	800.4464	**646.4461**	918.8439
TLBO [29]	799.0715	647.9202	923.4147
MGABC	800.4533	646.5207	918.7265

This movement rectifies the possibility of premature convergence caused by the formation of a cluster around the Gbest individual in GABC. The experimental results tested over 3 constrained engineering problems and 20 constrained benchmark problems show the effectiveness of the modified update strategies in MGABC. MGABC significantly outperforms other considered algorithms on most of the test problems, with higher accuracy, reliability and efficiency.
Additionally, MGABC is tested over OPF problem with three different cases. The outcome verifies the wide applicability of proposed modification in ABC.

References

[1] G. Gabriele, K. Ragsdell, The generalized reduced gradient method: A reliable tool for optimal design, Journal of Engineering for Industry 99 (2) (1977) 394–400.

[2] P. E. Gill, E. Wong, Sequential quadratic programming methods, in: Mixed integer nonlinear programming, Springer, 2012, pp. 147–224.

[3] C.-l. Sun, J.-c. Zeng, J.-s. Pan, An improved vector particle swarm optimization for constrained optimization problems, Information Sciences 181 (6) (2011) 1153–1163.

[4] E. Mezura-Montes, M. E. Miranda-Varela, R. del Carmen Gómez-Ramón, Differential evolu-

tion in constrained numerical optimization: an empirical study, Information Sciences 180 (22) (2010) 4223–4262.

[5] E. Mezura-Montes, R. E. Velez-Koeppel, Elitist artificial bee colony for constrained real-parameter optimization, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE, 2010, pp. 1–8.

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

[7] W.-f. Gao, S.-y. Liu, A modified artificial bee colony algorithm, Computers & Operations Research 39 (3) (2012) 687–697.

[8] A. Banharnsakun, T. Achalakul, B. Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, Applied Soft Computing 11 (2) (2011) 2888–2901.

[9] G. Li, P. Niu, X. Xiao, Development and investigation of efficient artificial bee colony algorithm for numerical function optimization, Applied soft computing 12 (1) (2012) 320–332.

[10] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Systems with Applications 37 (8) (2010) 5682–5687.

[11] W.-f. Gao, S.-y. Liu, L.-l. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, Cybernetics, IEEE Transactions on 43 (3) (2013) 1011–1024.

[12] D. Karaboga, B. Gorkemli, A quick artificial bee colony (qabc) algorithm and its performance on optimization problems, Applied Soft Computing 23 (2014) 227–238.

[13] M. S. Kiran, H. Hakli, M. Gunduz, H. Uguz, Artificial bee colony algorithm with variable search strategy for continuous optimization, Information Sciences 300 (2015) 140–157.

[14] H. Sharma, J. C. Bansal, K. Arya, X.-S. Yang, Lévy flight artificial bee colony algorithm, International Journal of Systems Science 47 (11) (2016) 2652–2670.

[15] D. Karaboga, B. Akay, A modified artificial bee colony (abc) algorithm for constrained optimization problems, Applied Soft Computing 11 (3) (2011) 3021–3031.

[16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, Evolutionary Computation, IEEE Transactions on 6 (2) (2002) 182–197.

[17] I. Brajevic, M. Tuba, M. Subotic, Improved artificial bee colony algorithm for constrained problems, in: Proceedings of the 11th WSEAS international conference on nural networks and 11th WSEAS international conference on evolutionary computing and 11th WSEAS international conference on Fuzzy systems, World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 185–190.

[18] E. Mezura-Montes, O. Cetina-Domínguez, Empirical analysis of a modified artificial bee colony for constrained numerical optimization, Applied Mathematics and Computation 218 (22) (2012) 10943–10973.

[19] I. Brajevic, Crossover-based artificial bee colony algorithm for constrained optimization problems, Neural Computing and Applications 26 (7) (2015) 1587–1601.

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

[21] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. C. Coello, K. Deb, Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization, Journal of Applied Mechanics 41 (2006) 8.

[22] Z. W. Geem, J. H. Kim, G. Loganathan, A new heuristic optimization algorithm: harmony search, simulation 76 (2) (2001) 60–68.

[23] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences 192 (3) (2012) 120–142.

[24] J. C. Bansal, S. S. Jadon, R. Tiwari, D. Kiran, B. K. Panigrahi, Optimal power flow using artificial bee colony algorithm with global and local neighborhoods, International Journal of System Assurance Engineering and Management (2014) 1–12.

[25] J. C. Bansal, P. Farswan, A novel disruption in biogeography-based optimization with application to optimal power flow problem, Applied Intelligence (2016) 1–26.

[26] H. R. E.-H. Bouchekara, M. A. Abido, Optimal power flow using differential search algorithm, Electric Power Components and Systems 42 (15) (2014) 1683–1699.

[27] R. P. Singh, V. Mukherjee, S. Ghoshal, Particle swarm optimization with an aging leader and challengers algorithm for the solution of optimal power flow problem, Applied Soft Computing 40 (2016) 161–177.

[28] T. K. Bashishtha, L. Srivastava, Nature inspired meta-heuristic dragonfly algorithms for solving optimal power flow problem, Nature.

[29] Optimal power flow using teaching-learning-based optimization technique, Electric Power Systems Research 114 (2014) 49 – 59.

Jagdish Chand Bansal is an assistant professor (senior grade) at South Asian University, New Delhi, India. He has obtained his Ph.D. from IIT Roorkee. He is the editor-in-chief of International Journal of Swarm Intelligence (IJSI) published by Inderscience. His primary area of interest is nature-inspired optimization techniques.

Susheel Kumar Joshi received his M.Tech degree in Systems Analysis and Computer Applications from National Institute of Technology, Surathkal, Karnataka, India in 2013. Currently, he is perusing Ph.D. from South Asian University, New Delhi, India. His area of research is nature-inspired optimization algorithms and soft computing.

Harish Sharma is working as an associate professor at Rajasthan Technical University, India. He has obtained his Ph.D. from ABV-Indian Institute of Information Technology and Management, Gwalior, India. His area of research is nature-inspired optimization algorithms and soft computing.