

# Optimal power flow using artificial bee colony algorithm with global and local neighborhoods

Jagdish Chand Bansal · Shimpi Singh Jadon ·  
Ritu Tiwari · Deep Kiran · B. K. Panigrahi

Received: 14 November 2014

© The Society for Reliability Engineering, Quality and Operations Management (SREQOM), India and The Division of Operation and Maintenance, Lulea University of Technology, Sweden 2014

**Abstract** Optimal power flow (OPF) is one of the most requisite tools for power system operation analysis. This problem has a complex mathematical formulation which is relatively hard to solve. This paper presents a swarm intelligence-based approach to solve the OPF problem. The proposed approach describes the use of a modified artificial bee colony (ABC) algorithm called ABC with global and local neighborhoods (ABCGLN) to determine the optimal settings of OPF control variables. ABCGLN is a recent modified version of basic ABC algorithm that can handle non-differentiable, non-linear, and multi modal objective functions. The ABCGLN approach is tested here on the standard IEEE 30-bus test system with three different objective functions for minimizing quadratic fuel cost function, piecewise quadratic cost function and quadratic cost function with valve point effects. The simulation results demonstrate the potential of ABCGLN algorithm of finding effective and robust quality solutions to solve OPF problem with various objective

functions for the considered system as compared to those available in the literature.

**Keywords** Optimal power flow · Artificial bee colony · Optimization · Swarm intelligence

## 1 Introduction

In past two decades, optimal power flow (OPF) problem has got much attention. OPF is a latest interest of many utilities as being one of the main tools for optimal operation and planning of modern power systems. Therefore, the OPF is the basic tool which facilitates electric utilities to determine secure and economic operating conditions for an electric power system. An OPF problem focuses on determining the optimal setting of power system control variables in order to optimize a selected objective function for example fuel cost while satisfying some physical and operational constraints. These constraints include equality constraints in the form of power flow equations and inequality constraints in the form of limits on control variables and operational limits on system dependent variables. The control variables for the problem are generator real powers, generator bus voltages, the reactive powers for shunt VAR compensations and transformer tap settings while the system dependent variables are the load bus voltages, transmission line loadings and generators reactive powers. In general, the OPF problem is a large-scale, highly constrained nonlinear and non-convex optimization problem.

Dommel and Tinney (1968), was first to present the formulation of OPF problem. Since then it is been used by many research scientists. Many traditional mathematics-

---

J. C. Bansal  
South Asian University, New Delhi, India  
e-mail: jcbansal@gmail.com

S. S. Jadon (✉) · R. Tiwari  
ABV-Indian Institute of Information Technology and  
Management, Gwalior, India  
e-mail: shimpisingh2k6@gmail.com

R. Tiwari  
e-mail: tiwariritu2@gmail.com

D. Kiran · B. K. Panigrahi  
Indian Institute of Technology, New Delhi, India  
e-mail: dkiran5@gmail.com

B. K. Panigrahi  
e-mail: bkpanigrahi@ee.iitd.ac.in

based techniques like linear programming (Mota-Palomino and Quintana 1986; Al-Muhawesh and Qamber 2008), non-linear programming (Habibollahzadeh et al. 1989), interior point method (Granville 1994; Wei et al. 1998) and Newton-based method (Sun et al. 1984) have been applied to solve the different OPF problems. Generally, these classical techniques are applied with assumption that the fuel cost characteristic of a generating unit is a smooth and convex function. Moreover, these techniques are very sensitive towards initial solution and even they may fail to converge due to inappropriate starting values of variables. The main disadvantage with these techniques is that these are not possible to solve practical systems because of nonlinear characteristics such as prohibited operating zones, valve point effects and piecewise quadratic cost functions. Now, it becomes mandatory to develop such optimization techniques which can overcome these drawbacks and tolerate these complexities.

With the emergence of computers having tremendous capabilities, the meta-heuristic techniques such as particle swarm optimization (Abido 2002a), genetic algorithm (Bakirtzis et al. 2002), tabu search (Abido 2002b), differential evolution (Sayah and Zehar 2008), simulated annealing (Roa-Sepulveda and Pavez-Lazo 2003) and evolutionary programming (Ongsakul and Tantimaporn 2006; Yuryevich and Wong 1999), etc., have been implemented to solve various OPF problems. The reported results of these meta heuristic techniques in literature are promising and encourage the researchers for further research in the same direction.

Karaboga (2005) proposed the recent addition to this category known as artificial bee colony (ABC) optimization algorithm. The ABC algorithm mimics the foraging behavior of honey bees while searching food for them. ABC is a simple population-based optimization algorithm which is effective for both continuous and discrete optimization problems. Collaborative trial and error method is the main concept behind ABC algorithm which enables its algorithmic procedure to find the solution. The ABC population consists of possible solutions in terms of food sources for honey bees whose fitness is regulated in terms of nectar amount which the food source contains. The swarm updating in ABC is due to two processes namely, the variation process and the selection process which are responsible for exploration and exploitation, respectively. The ABC algorithm has successfully been tested on almost all domains of science and engineering like electronics engineering (Chidambaram and Lopes 2009; Kaviani et al. 2012), electrical engineering (Jones and Bouffet 2008; Nayak et al. 2009; Sulaiman et al. 2012), computer science engineering (Karaboga and Cetinkaya 2011; Lam et al. 2012; Lei et al. 2010), mechanical engineering (Banharnsakun et al. 2012; Pawar et al. 2008; Xu and Duan 2010), civil engineering (Akay and Karaboga 2012; Li

et al. 2011; Mandal et al. 2012), medical pattern classification and clustering problems (Karaboga et al. 2008) and mathematical graph problems (Xing et al. 2007; Singh 2009; Yeh and Hsieh 2011). Many of the recent modifications and applications of ABC algorithm can be studied in Bansal et al. (2013b).

However the ABC achieves a good solution but, like the other optimization algorithms, it also has problem of premature convergence and stagnation. On the other hand, it is also required to tune the ABC control parameters based on problem. Also literature says that basic ABC itself has some drawbacks like stop proceeding toward the global optimum even though the population has not converged to a local optimum (Karaboga and Akay 2009) and it is observed that the position update equation of ABC algorithm is inefficient to balance exploration and exploitation (Zhu and Kwong 2010). Therefore these drawbacks require a modification in position update equation of ABC in order to make it capable to balance exploration and exploitation. These drawbacks have also addressed in earlier research (Banharnsakun et al. 2011; Bansal et al. 2013a, c; Gao and Liu 2012; Jadon et al. 2014b; Karaboga and Akay 2011; Sharma et al. 2013). Based on neighborhood concept, Jadon et al. (2014a) proposed a recent modified ABC variant called ABC with global and local neighborhoods (ABCGLN). In ABCGLN, authors proposed a new position update equation for employed bees in ABC which linearly incorporate two components namely local and global components in its swarm updating process in order to set a trade off between the exploration and exploitation capabilities of ABC. Here, local component is responsible for explorative moves and has better chance of locating the minima of test function. On the other hand, global component is responsible for exploitive moves and rapidly converges to a minimum of the test function.

In this paper, we used this ABC variant ABCGLN to solve OPF problem with three different objective functions and compared it with various results available in literature.

Rest of the paper is organized as follows: Section 2 describes the basic mathematical formulation of OPF problem. Section 3 explains ABCGLN algorithm. Section 4 gives statistical analysis to compare the performance of the ABCGLN strategy on OPF problem with other meta-heuristic algorithms. Finally, in Section 5, paper is concluded.

## 2 OPF formulation

The aim of solving an OPF problem is to provide an optimal setting of power system's control variables while satisfying some physical and operational restrictions in terms of equality and inequality constraints. Mathematical formulation of a general OPF problem can be seen as:

$$\text{Min } F(u, x), \tag{1}$$

subject to:

$$G(u, x) = 0, \tag{2}$$

$$H^{\min} \leq H(u, x) \leq H^{\max}, \tag{3}$$

where  $F$  is the objective function to be optimized,  $u$  is the vector of control variables (independent variables) which includes generator bus voltages ( $V_g$ ), generator real powers ( $P_g$ ) except at slack bus, transformer tap settings ( $T$ ) and shunt VAR compensation ( $Q_c$ ) and  $x$  is the vector of state variables (dependent variables) which includes generator active power at slack bus ( $P_{g1}$ ), load bus voltages ( $V_l$ ), generator reactive powers ( $Q_g$ ) and transmission line loading (line flow;  $S_l$ ). Hence,  $u$  and  $x$  can be expressed as:

$$u = [P_{g2}, \dots, P_{gng}, V_{g1}, \dots, V_{gng}, Q_{c1}, \dots, Q_{cnc}, T_1, \dots, T_{nt}], \tag{4}$$

$$x = [P_{g1}, V_{l1}, \dots, V_{lnl}, Q_{g1}, \dots, Q_{gng}, S_{l1}, \dots, S_{lnl}], \tag{5}$$

where  $ng$ ,  $nc$ ,  $nt$ ,  $nl$ ,  $Nl$  are the number of generators, number of shunt VAR compensators, number of regulating transformers, number of load buses and number of transmission lines, respectively.

$G(u, x)$  is the set of equality constraints which represents typical load flow equations:

$$P_{gi} - P_{di} - V_i \sum_{k=1}^{nb} V_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) = 0, \tag{6}$$

and

$$Q_{gi} - Q_{di} - V_i \sum_{k=1}^{nb} V_k (G_{ik} \sin \theta_{ik} + B_{ik} \cos \theta_{ik}) = 0, \tag{7}$$

where  $P_{gi}$  and  $Q_{gi}$  are the active and reactive powers at  $i$ th generators,  $P_{di}$  and  $Q_{di}$  are the active and reactive power demands at  $i$ th bus,  $G_{ik}$  and  $B_{ik}$  are the transfer conductance and susceptance between buses  $i$  and  $k$ , respectively,  $\theta_{ik}$  is the phase angle difference between the voltages at buses  $i$  and  $k$  and  $nb$  is the total number of bus bars.

$H(u, x)$  is the set of system operational limiting constraints which includes following inequality constraints:

– *Generator constraints*

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max}, \quad \text{for } i = 1, 2, \dots, ng, \tag{8}$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}, \quad \text{for } i = 1, 2, \dots, ng \tag{9}$$

$$V_{gi}^{\min} \leq V_{gi} \leq V_{gi}^{\max}, \quad \text{for } i = 1, 2, \dots, ng. \tag{10}$$

– *Security constraints*

$$V_{li}^{\min} \leq V_{li} \leq V_{li}^{\max}, \quad \text{for } i = 1, 2, \dots, nl, \tag{11}$$

$$S_{li} \leq S_{li}^{\max}, \quad \text{for } i = 1, 2, \dots, nl. \tag{12}$$

– *Transformer constraints*

$$T_i^{\min} \leq T_i \leq T_i^{\max}, \quad \text{for } i = 1, 2, \dots, nt. \tag{13}$$

– *Shunt VAR compensator constraints*

$$Q_{ci}^{\min} \leq Q_{ci} \leq Q_{ci}^{\max}, \quad \text{for } i = 1, 2, \dots, nc. \tag{14}$$

There are various techniques available in the literature to handle constraints in optimization algorithm. In this article, penalty function method is adopted to handle constraints with respect to dependent variables i.e., if any dependent variable violates its bound then square of that violation amount multiplied by a fix penalty factor is added to its corresponding fitness function so that infeasible solutions can be rejected. On the other hand, the constraints corresponding to control variables are handled by generating them between the given bounds in its initialization phase. In this way, the modified objective function for OPF is of the following form:

$$\begin{aligned} \text{Min } F_{\text{mod}} = & F(u, x) + \lambda_p (P_{g1} - P_{g1}^{\text{lim}})^2 \\ & + \lambda_v \sum_{i=1}^{nl} (V_{li} - V_{li}^{\text{lim}})^2 + \lambda_q \sum_{i=1}^{ng} (Q_{gi} - Q_{gi}^{\text{lim}})^2 \\ & + \lambda_s \sum_{i=1}^{Nl} (S_{li} - S_{li}^{\text{lim}})^2, \end{aligned} \tag{15}$$

where  $\lambda_p$ ,  $\lambda_v$ ,  $\lambda_q$  and  $\lambda_s$  are the penalty factors (all are set to  $10^5$  in this paper) and  $a^{\text{lim}}$  is the limit value (may be min or max limit depending upon which limit dependent variable  $a$  has crossed).

### 3 ABC algorithm with global and local neighborhoods

The ABC algorithm is a population based relatively recent swarm intelligence based algorithm which is inspired by foraging behavior of honey bees. In ABC, each solution is known as food source of honey bees whose fitness is determined in terms of the quality of the food source. ABC is made of three group of bees: employed bees, onlooker bees and scout bees. The number of employed and onlooker bees is equal. The employed bees search the food source in the environment and store the information, like the quality and the distance of the food source from the hive. Onlooker bees wait in the hive for employed bees and after collecting information from them, they start searching in neighborhood of that food sources which are having better nectar. If any food source is abandoned then scout bee finds new food source randomly in the search space. While searching the solution of any optimization problem,

ABC algorithm first initializes ABC parameters and swarm then it requires the repetitive iterations of the three phases namely employed bee phase, onlooker bee phase and scout bee phase.

For an efficient optimization algorithm, two concepts exploration and exploitation should be well balanced which are actually contradictory to each other in nature. Exploration means the ability of exploring diverse regions of the search space and exploitation means refining already explored regions. In ABC, at any instance, a solution is updated through information flow from other solutions of the swarm. This position updating process uses a linear combination of current position of the potential solution which is going to be updated and position of a randomly selected solution as step size with a random coefficient  $\phi_{ij} \in [-1, 1]$ . This process plays an important role to decide the quality of new solution. If the current solution is far from randomly selected solution and absolute value of  $\phi_{ij}$  is also high then the change will be large enough to jump the true optima. On the other hand, small change will decrease the convergence rate of whole ABC process. Further, it is also suggested in literature (Karaboga and Akay 2009; Zhu and Kwong 2010) that basic ABC itself has some drawbacks, like stop proceeding toward the global optimum even though the population has not converged to a local optimum. Karaboga and Akay (2009) also analyzed the various variants of ABC and found that the ABC in its current form shows poor performance and remains inefficient to balance the exploration and exploitation capabilities of the search space. Consequently, convergence speed of ABC is also deteriorated.

To overcome above shortcomings, Jadon et al. (2014a) proposed a modified ABC by incorporating local neighborhood and global neighborhood strategies in its position update equation of employed bee phase. In local neighborhood strategy each member is updated through best solution found so far in its sub population (i.e., local neighborhood), while in the global neighborhood strategy each member takes the advantage of best solution found so far in the whole population. In ABCGLN, the neighborhood structure of any solution is static and has been defined on the set of indices of the solutions. Neighborhood of a solution is the set of other solutions to which it is connected. For global neighborhood strategy, neighborhood structure is the star topology i.e., each solution is connected to each other solution and therefore cardinality of neighborhood of any solution in this strategy is equal to the number of food sources. On the other hand, for local neighborhood strategy, ring topology is adopted as neighborhood structure and cardinality of neighborhood is considered as 10% of the number of food sources i.e., local neighborhood of each solution includes 10% solutions based on their indices as

5% from the forward side and 5% from the backward side. Here, it should be noticed that solutions belonging to a local neighborhood are not necessarily local in the sense of their geographical nearness or similar fitness values but, the overlapping neighborhoods based on the indices of the swarm members have been considered in ABCGLN. For more details about the neighborhood structure adopted, readers are advised to go through the original article ABCGLN (Jadon et al. 2014a). In ABCGLN, for each employed bee, two components a local and a global component are created. The local component for a bee is created based on best solution position in that bee's local neighborhood and two other solutions selected randomly from this neighborhood while global component is created based on best solution of whole swarm and two other solutions selected randomly from the whole swarm i.e., from the global neighborhood. The mathematical equations for these components are as follows:

$$\begin{aligned} L_{ij} &= x_{ij} + (prob_i)(x_{lj} - x_{ij}) + \phi_{ij}(x_{r1j} - x_{r2j}), \\ G_{ij} &= x_{ij} + (1 - prob_i)(x_{gj} - x_{ij}) + \phi_{ij}(x_{R1j} - x_{R2j}), \end{aligned} \quad (16)$$

where each  $j$  represents the  $j$ th dimension of the each position,  $L_i$  and  $G_i$  are the local and global components, respectively.  $x_l$  and  $x_g$  are respectively the best positions in local and global neighborhoods of  $i$ th solution.  $x_{r1}$ ,  $x_{r2}$  are the two neighbors chosen randomly from the local neighborhood of  $i$ th solution such that  $r1 \neq r2 \neq i$ .  $x_{R1}$ ,  $x_{R2}$  are the two neighbors chosen randomly from the global neighborhood i.e., from whole swarm such that  $R1 \neq R2 \neq i$ .  $\phi_{ij}$  is random number in  $[-1, 1]$  and  $prob_i$  is the probability of  $i$ th solution as in Eq. (19). Here in both equations the third term i.e., difference between two randomly selected neighbors is added so that individual member should not follow only the best position in order to prevent the swarm from getting trapped in local minima. Also second term in both the equations is multiplied by their probabilities which is a function of fitness. It is clear that solution with better fitness will give more weightage to its local best than the global best in order to prevent the swarm to converge too quickly as global best may be a local optima in initial stages. On the other hand the solution with low fitness will give priority to global best more than the local best as there surroundings are not good enough so they follow better directions. Finally both the local and the global components together guide the new direction for employed bees as follows: for  $i$ th employed bee, the new position is,

$$v_{ij} = (prob_i) \times G_{ij} + (1 - prob_i) \times L_{ij}. \quad (17)$$

Now based on greedy selection, employed bee selects one between the old and new positions. In this way, the

involvement of weighted linear sum of both local and global best components, is expected to balance the exploration and exploitation capabilities of ABC algorithm as the local best components of respective bees explore the search space or tries to identify the most promising search space regions, while the global best component will exploit the identified search space. The pseudo code of implementation of this whole neighborhood concept in employed bee phase of ABCGLN is explained in Algorithm 1. Algorithm 1 introduced a new parameter  $cr$  which is set to a low value 0.3 (Jadon et al. 2014a).

Now whole ABCGLN algorithmic procedure can be understood in following way. In ABCGLN, first initialization of the solutions is done like in basic ABC as:

### 3.1 Initialization of the swarm

If  $D$  is the number of variables in the optimization problem then each food source  $x_i$  ( $i = 1, 2, \dots, SN$ ) is a  $D$ -dimensional vector among the  $SN$  food sources and is generated using a uniform distribution as:

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}), \quad (18)$$

here  $x_i$  represents the  $i$ th food source in the swarm,  $x_{minj}$  and  $x_{maxj}$  are bounds of  $x_i$  in  $j$ th dimension and  $rand[0, 1]$  is a uniformly distributed random number in the range  $[0, 1]$ . After initialization phase ABCGLN requires the cycle of the three phases namely employed bee phase, onlooker bee phase and scout bee phase to be executed.

### 3.2 Employed bees phase

In this phase,  $i$ th candidate's position is updated using Eq. (17): after generating new position, the position with better fitness between the newly generated and old one is selected.

### 3.3 Onlooker bees phase

In this phase, employed bees share the information associated with their food sources like quality (nectar) and position of the food source with the onlooker bees in the hive. Onlooker bees evaluate the available information about the food source and based on its fitness they select solutions with probability  $prob_i$ . Here  $prob_i$  can be calculated as function of fitness (there may be some other method):

$$prob_i(G) = \frac{0.9 \times fitness_i}{maxfit} + 0.1, \quad (19)$$

here  $fitness_i$  is the fitness value of the  $i$ th solution and  $maxfit$  is the maximum fitness among all the solutions. Based on this probability, onlooker selects a solution and modifies it using the ordinary ABC position update Eq. (20). Again by applying greedy selection, if the fitness is higher than the previous one, the onlooker bee stores the new position in its memory and forgets the old one.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}). \quad (20)$$

### 3.4 Scout bees phase

If for a predetermined number of cycles, any bee's position is not getting updated then that food source is taken to be abandoned and this bee becomes scout bee. In this phase, the abandoned food source is replaced by a randomly chosen food source within the search space. The number of cycles after which a particular food source becomes abandoned is known as *limit* and is a crucial control parameter. In this phase the abandoned food source  $x_i$  is replaced by a randomly chosen food source within the search space using the Eq. (18) as in initialization phase.

---

#### Algorithm 1 employed bee phase for ABCGLN:

---

```

Input employed bee  $i$ ,  $prob_i$  and best solution  $g$  found so far in whole swarm;
Find best solution  $l$  in local neighborhood of  $i^{th}$  bee;
Select two neighbors  $x_{r1}, x_{r2}$  randomly from the local neighborhood of  $i^{th}$  bee such that  $r1 \neq r2 \neq i$ ;
Select two neighbors  $x_{R1}, x_{R2}$  randomly from whole swarm such that  $R1 \neq R2 \neq i$ ;
for (each dimension  $j$ ) do
  if ( $rand < cr$ ) then { # rand is a uniform random number between 0 and 1 }
    Generate local component  $L_{ij}$  and global component  $G_{ij}$  using Equation (16);
    Generate new position  $v_{ij}$  using Equation (17);
  else
     $v_{ij} = x_{ij}$ ;
  end if
end for
Return better of old position  $x_{ij}$  and new position  $v_{ij}$  for employed bee  $i$ ;

```

---

The pseudo-code of the ABCGLN algorithm is shown in Algorithm 2 (Jadon et al. 2014a ).

---

**Algorithm 2** ABC with Global and Local neighborhoods (ABCGLN):

---

```

Initialize the population and control parameters;
while Termination criteria is not satisfied do
    Memorize the best food source found so far;
    Calculate  $prob_i$  using Equation (19) for each bee;
    Employed bee phase: apply Algorithm (1) to generate new food sources for each employed bee.
    Onlooker bees phase: use Equation (20) to update the food sources for particular bees selected based on  $prob_i$ ;
    Scout bee phase: use Equation (18) to determine the new food sources for exhausted food sources.
end while
Return the best solution found so far.

```

---

**Table 1** The upper and lower limits of control variables

Control variables	Min	Max	Control variables	Min	Max
$P_1$	50	200	$T_{11}$	0.9	1.1
$P_2$	20	80	$T_{12}$	0.9	1.1
$P_5$	15	50	$T_{15}$	0.9	1.1
$P_8$	10	35	$T_{36}$	0.9	1.1
$P_{11}$	10	30	$Q_{c10}$	0	5
$P_{13}$	12	40	$Q_{c12}$	0	5
$V_1$	0.95	1.1	$Q_{c15}$	0	5
$V_2$	0.95	1.1	$Q_{c17}$	0	5
$V_5$	0.95	1.1	$Q_{c20}$	0	5
$V_8$	0.95	1.1	$Q_{c21}$	0	5
$V_{11}$	0.95	1.1	$Q_{c23}$	0	5
$V_{13}$	0.95	1.1	$Q_{c24}$	0	5
			$Q_{c29}$	0	5

---

### 3.5 ABCGLN implementation for OPF

ABCGLN first initializes each individual member of the population which represents a potential solution. The structure of each individual member is  $u = (P_{g2}, \dots, P_{gng}, V_{g1}, \dots, V_{gng}, Q_{c1}, \dots, Q_{cnc}, T_1, \dots, T_m)$  where each involved variable is self constrained i.e., each control variable of individual member  $u$  is initialized randomly within allowable its given limits given in Table 1. If any control variable violates its lower or upper limits then that is adjusted to the corresponding violated limit. To handle the inequality constraints of dependent variables, the modified objective function [Eq. (15)] is considered to be optimized. The flow chart of OPF problem's solution through ABCGLN is presented in Fig. 1. This is clear from Flow chart 1 that whenever a new solution is generated during initialization or employed or onlooker or scout phase, then power flow algorithm is run for that individual solution  $u$  to determine the state variables and to evaluate its fitness.

## 4 Experimental results and discussion

### 4.1 Test system under consideration

To validate the effectiveness and robustness of the ABCGLN algorithm, it is applied on standard IEEE 30-bus system. This test system consists of 6 generating units at buses 1, 2, 5, 8, 11 and 13 interconnected with 41 transmission lines with a total load of 283.4 MW and 126.2 MVAR and 4 transformers with off-nominal tap ratios in lines 6–9, 6–10, 4–12 and 28–27. The bus data and the branch data are taken from Alsac and Stott (1974). The shunt injections are provided at buses 10, 12, 15, 17, 20, 21, 23, 24 and 29 as given in Abido (2002a). The maximum and minimum limits on control variables are mentioned in Table 1. Three different objective functions to minimize the total quadratic fuel cost, piece wise quadratic fuel cost and quadratic fuel cost with valve point effects are considered to demonstrate the effectiveness of ABCGLN algorithm. To check the robustness of the ABCGLN algorithm, 100 runs were executed for solving the OPF problem in each case.

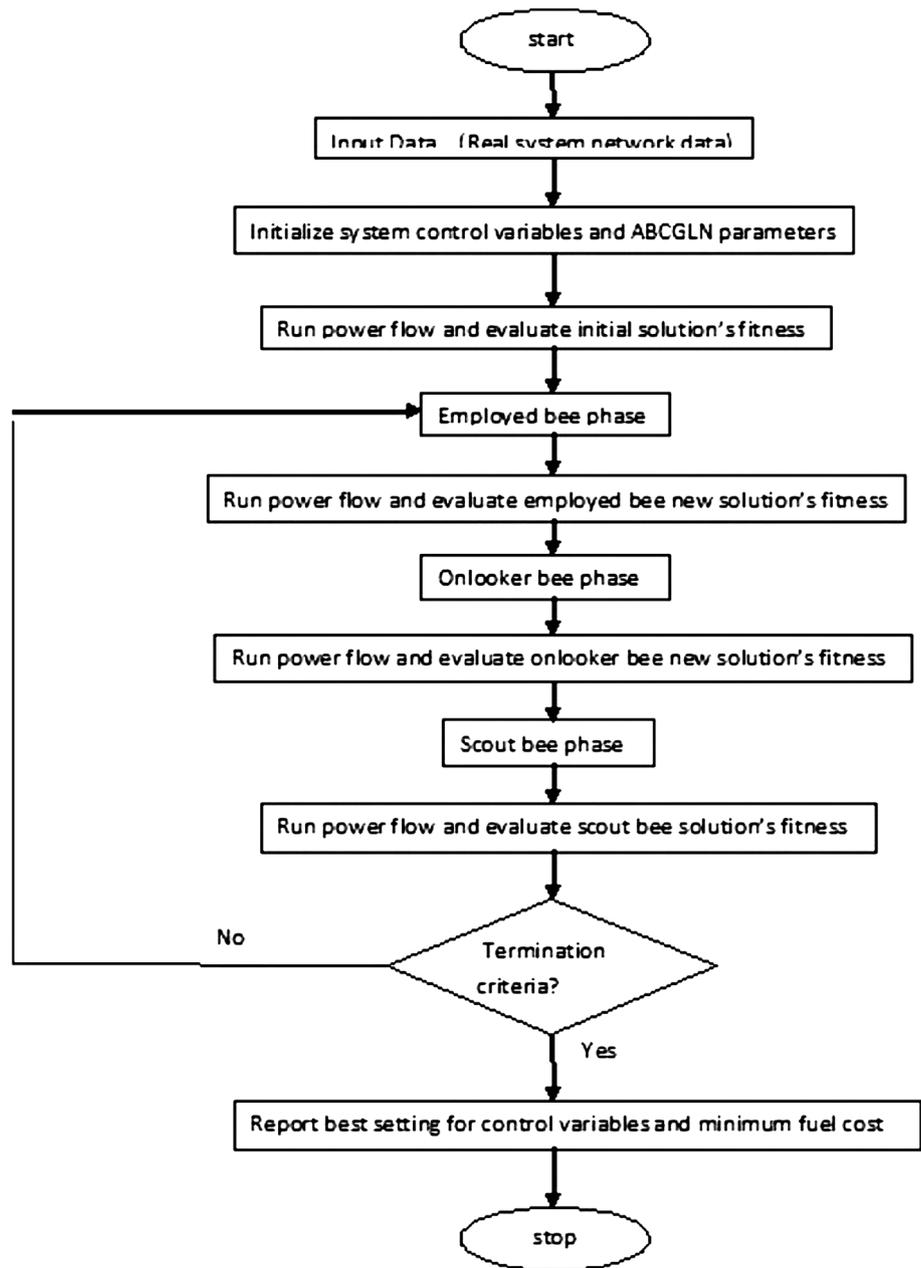
#### 4.1.1 Case 1: quadratic fuel cost function

The generator cost characteristic are defined as quadratic cost function of generator power output and Eq. (15) is selected as objective function to be optimized. In Eq. (15), the  $F(u, x)$  is designed for this case as:

$$F(u, x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^{ng} (a_i + b_i P_{gi} + c_i P_{gi}^2), \quad (21)$$

where  $f_i$  and  $P_{gi}$  are fuel cost and active power of  $i$ th generator, respectively. The variables  $a_i$ ,  $b_i$  and  $c_i$  represent the cost coefficients of  $i$ th generator whose values have been taken from standard IEEE 30-bus system and is given here in Table 2.  $ng$  is the total number of generators in the system.

**Fig. 1** ABCGLN algorithm flow for OPF problem



**Table 2** Generator cost coefficients for case 1

Cost coefficients	Bus nos.					
	1	2	5	8	11	13
<i>a</i>	0.00	0.00	0.00	0.00	0.00	0.00
<i>b</i>	2.00	1.75	1.00	3.25	3.00	3.00
<i>c</i>	0.00375	0.01750	0.06250	0.00834	0.02500	0.02500

4.1.2 Case 2: piece-wise quadratic fuel cost functions

Practically, in a power system, various fuel sources like natural gas, coal and oil are supplied to thermal generating units.

The fuel cost functions of these units may be dissevered as piece-wise quadratic fuel cost functions for different fuel types (Abou El Ela et al. 2010). In this case, the objective of the problem is to minimize fuel cost among the available fuel

**Table 3** Generator cost coefficients for case 2

Bus nos.	From MW	To MW	Cost coefficients		
			<i>a</i>	<i>b</i>	<i>c</i>
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

of each unit and satisfy the system constraints. The problem becomes a non-convex optimization problem with discontinues value at each boundary which may result in many local optima and hence limit the application of the traditional mathematical methods (Dieu and Ongsakul 2006). The cost characteristics of generating units connecting at buses 1 and 2 are represented by a piece-wise quadratic cost function to model different fuels, which are defined as follows:

$$f_i(P_{gi}) = \begin{cases} a_{i1} + b_{i1}P_{gi} + c_{i1}P_{gi}^2, & P_{gi}^{min} \leq P_{gi} \leq P_{gi1} \\ a_{i2} + b_{i2}P_{gi} + c_{i2}P_{gi}^2, & P_{gi1} \leq P_{gi} \leq P_{gi2} \\ \dots \\ a_{ik} + b_{ik}P_{gi} + c_{ik}P_{gi}^2, & P_{gik-1} \leq P_{gi} \leq P_{gi}^{max}, \end{cases} \quad (22)$$

where *a<sub>ik</sub>*, *b<sub>ik</sub>* and *c<sub>ik</sub>* are cost coefficients of the *i*th generator for fuel type *k*. So, here Eq. (15) is selected as objective function to be optimized where the *F(u, x)* is designed for this case as:

$$F(u, x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^2 f_i(P_{gi}) + \sum_{i=3}^{ng} (a_i + b_iP_{gi} + c_iP_{gi}^2), \quad (23)$$

where *f<sub>i</sub>(P<sub>gi</sub>)* for generating units 1 and 2 are selected based on Eq. (22) and the cost coefficients for these units are given in Table 3 and the cost coefficients of other generators have the same values as of case 1.

4.1.3 Case 3: quadratic fuel cost function with valve point effects

For this case, the generating units of buses 1 and 2 are considered to have the valve point effects on their

**Table 4** Generator cost coefficients for case 3

Bus nos.	<i>P<sub>gi</sub><sup>min</sup></i>	Cost coefficients				
		<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
1	50	150.00	2.00	0.0016	50.00	0.0630
2	20	25.00	2.50	0.0100	40.00	0.0980

**Table 5** Load data

Bus nos.	Load		Bus nos.	Load		Bus nos.	Load	
	<i>P</i>	<i>Q</i>		<i>P</i>	<i>Q</i>		<i>P</i>	<i>Q</i>
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

**Table 6** Best control variable settings achieved by ABCGLN algorithms for different cases

Control variables	Case 1: Quadratic fuel cost function	Case 2: Piece-wise quadratic fuel cost function	Case 3: Quadratic fuel cost function with valve point effects
<i>P</i> <sub>1</sub>	177.0220	139.9909	199.6030
<i>P</i> <sub>2</sub>	48.6853	55.0000	20.0000
<i>P</i> <sub>5</sub>	21.3817	24.0789	22.1228
<i>P</i> <sub>8</sub>	21.3059	35.0000	25.8842
<i>P</i> <sub>11</sub>	12.0071	18.1608	13.4462
<i>P</i> <sub>13</sub>	12.0000	17.8988	12.0000
<i>V</i> <sub>1</sub>	1.0839	1.0758	1.0796
<i>V</i> <sub>2</sub>	1.0335	1.0094	1.0595
<i>V</i> <sub>5</sub>	1.0340	1.0313	1.0319
<i>V</i> <sub>8</sub>	1.0380	1.0495	1.0370
<i>V</i> <sub>11</sub>	1.0666	1.0505	1.0767
<i>V</i> <sub>13</sub>	1.0481	1.0574	1.0597
<i>T</i> <sub>11</sub>	1.0616	1.0294	1.0890
<i>T</i> <sub>12</sub>	0.9089	0.9291	0.9000
<i>T</i> <sub>15</sub>	0.9676	0.9873	0.9831
<i>T</i> <sub>36</sub>	0.9728	0.9736	0.9728
<i>Q</i> <sub>c10</sub>	5.0000	4.9999	5.0000
<i>Q</i> <sub>c12</sub>	5.0000	5.0000	4.2523
<i>Q</i> <sub>c15</sub>	4.3082	5.0000	1.4910
<i>Q</i> <sub>c17</sub>	5.0000	4.9999	5.0000
<i>Q</i> <sub>c20</sub>	4.5056	4.9693	5.0000
<i>Q</i> <sub>c21</sub>	5.0000	5.0000	4.7040
<i>Q</i> <sub>c23</sub>	3.4272	3.9497	5.0000
<i>Q</i> <sub>c24</sub>	5.0000	5.0000	5.0000
<i>Q</i> <sub>c29</sub>	2.4945	2.8101	1.6957
Fuel cost (\$/h)	800.4464	646.4461	918.8439

characteristics. The cost coefficients of these generating buses are adopted from Vaisakh and Srinivas (2011) and are given in Table 4. The cost coefficients for

**Table 7** Results comparison of 100 runs among different methods for IEEE 30-bus system: case 1

Optimization methods	Fuel cost (\$/h)		Execution time (s)
	Min	Average	
ITS (Ongsakul and Tantimaporn 2006)	804.5560	–	–
EP (Yuryevich and Wong 1999)	802.6300	803.5100	51.4
IEP (Ongsakul and Tantimaporn 2006)	802.4650	802.5210	99.0
DE-OPF (Sayah and Zehar 2008)	802.3940	–	36.4
MDE-OPF (Sayah and Zehar 2008)	802.3760	802.3820	24.0
TS (Ongsakul and Tantimaporn 2006)	802.5020	–	–
TS/SA (Ongsakul and Tantimaporn 2006)	802.7880	–	–
SADE-ALM (Vaisakh and Srinivas 2011)	802.4040	–	–
Enhanced GA (Bakirtzis et al. 2002)	802.0600	–	–
PSO (Abido 2002a)	800.4890	–	–
ABC-OPF	802.9086	803.6341	27.0
ABCGLN	<b>800.4464</b>	<b>800.5607</b>	18.0

**Table 8** Results comparison of 100 runs among different methods for IEEE 30-bus system: case 2

Optimization methods	Fuel cost (\$/h)		Execution time (s)
	Min	Average	
ITS (Ongsakul and Tantimaporn 2006)	654.8740	–	–
EP (Yuryevich and Wong 1999)	647.7900	649.7000	51.6
IEP (Ongsakul and Tantimaporn 2006)	649.3120	650.2170	100.4
DE-OPF (Sayah and Zehar 2008)	648.3840	–	37.2
MDE-OPF (Sayah and Zehar 2008)	647.8460	648.3560	37.1
TS (Ongsakul and Tantimaporn 2006)	651.2460	–	–
TS/SA (Ongsakul and Tantimaporn 2006)	654.3780	–	–
PSO (Abido 2002a)	647.6900	–	–
GSA (Duman et al. 2012)	646.8480	646.8962	11.0
BBO (Bhattacharya and Chattopadhyay 2011)	647.7430	647.7645	12.0
ABC-OPF	648.9124	649.4393	29.0
ABCGLN	<b>646.4461</b>	<b>646.6944</b>	22.0

other generating units are same as of case 1. The cost characteristics of generating units 1 and 2 are defined as:

**Table 9** Results comparison of 100 runs among different methods for IEEE 30-bus system: case 3

Optimization methods	Fuel cost (\$/h)		Execution time (s)
	Min	Average	
ITS (Ongsakul and Tantimaporn 2006)	969.1090	–	–
EP (Vaisakh and Srinivas 2011)	955.5090	959.3630	57.7
IEP (Ongsakul and Tantimaporn 2006)	953.5730	956.4600	93.6
DE-OPF (Sayah and Zehar 2008)	931.0850	–	45.0
MDE-OPF (Sayah and Zehar 2008)	930.793	942.5010	41.9
TS (Ongsakul and Tantimaporn 2006)	956.0000	–	–
TS/SA (Ongsakul and Tantimaporn 2006)	959.5630	–	–
EADDE (Vaisakh and Srinivas 2011)	930.745	–	3.98
SADE-ALM (Vaisakh and Srinivas 2011)	944.031	–	–
BBO (Bhattacharya and Chattopadhyay 2011)	919.7647	919.8389	11.9
GSA (Duman et al. 2012)	929.7260	930.9240	10.0
ABC-OPF	930.4153	931.2629	34.0
ABCGLN	<b>918.8439</b>	<b>919.1928</b>	27.0

$$f_i(P_{gi}) = a_i + b_i P_{gi} + c_i P_{gi}^2 + \left| d_i \sin \left( e_i \left( P_{gi}^{min} - P_{gi} \right) \right) \right|, \quad \text{where } i = 1 \text{ and } 2, \quad (24)$$

where  $a_i, b_i, c_i, d_i$  and  $e_i$  are cost coefficients of the  $i$ th generating unit. So, here Eq. (15) is selected as objective function to be optimized where the  $F(u, x)$  is designed for this case as:

$$F(u, x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^2 \left( a_i + b_i P_{gi} + c_i P_{gi}^2 + \left| d_i \sin \left( e_i \left( P_{gi}^{min} - P_{gi} \right) \right) \right| \right) + \sum_{i=3}^{ng} \left( a_i + b_i P_{gi} + c_i P_{gi}^2 \right). \quad (25)$$

#### 4.2 Experimental setting

The results obtained from the proposed ABCGLN are stored in the form of minimum objective function value, mean objective function value in 100 runs and execution time to reach at minimum. The results are compared with the results available from other optimization algorithms in literature. The following parameter setting is adopted while implementing ABCGLN and basic ABC algorithms to solve the various OPF problems:

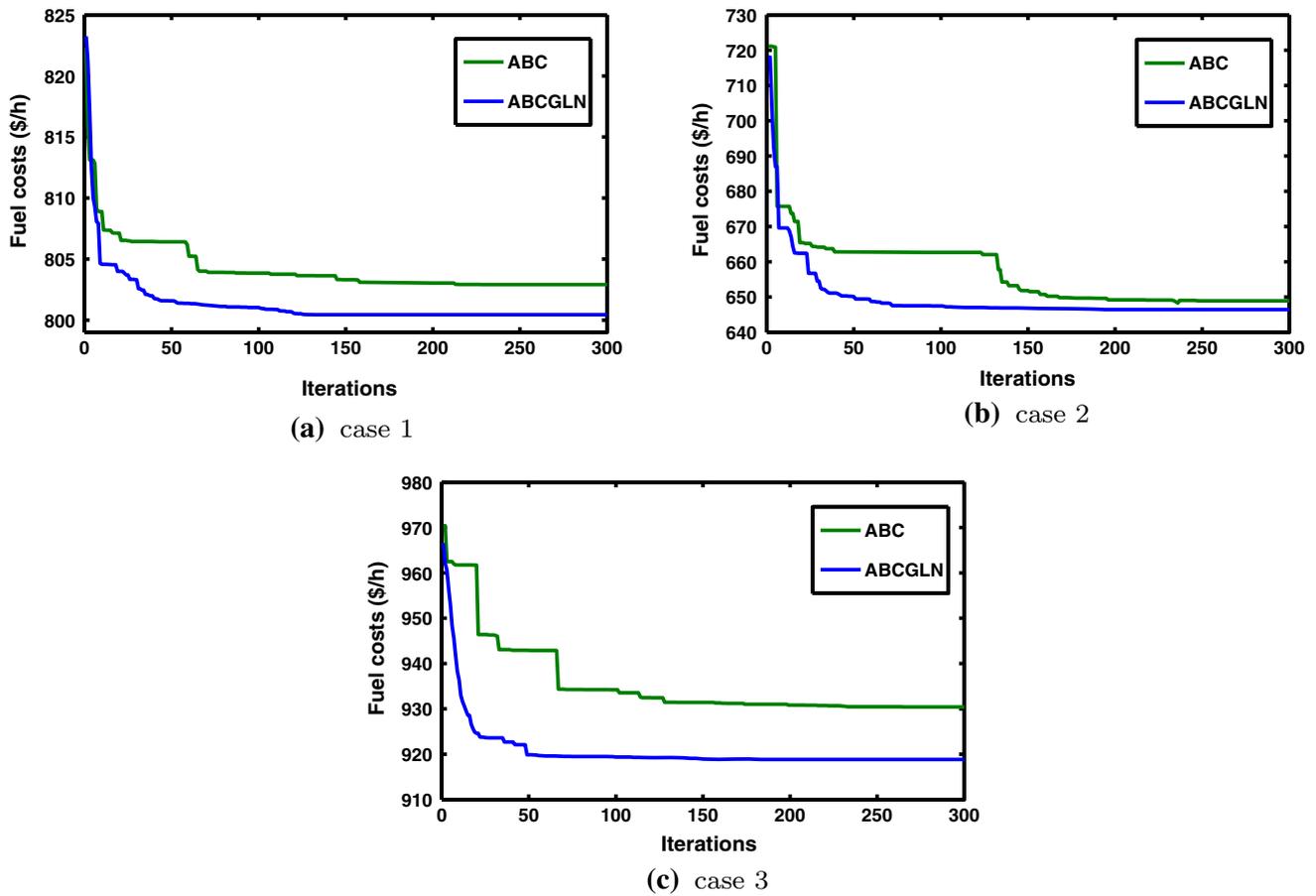


Fig. 2 Convergence speed for ABCGLN and ABC

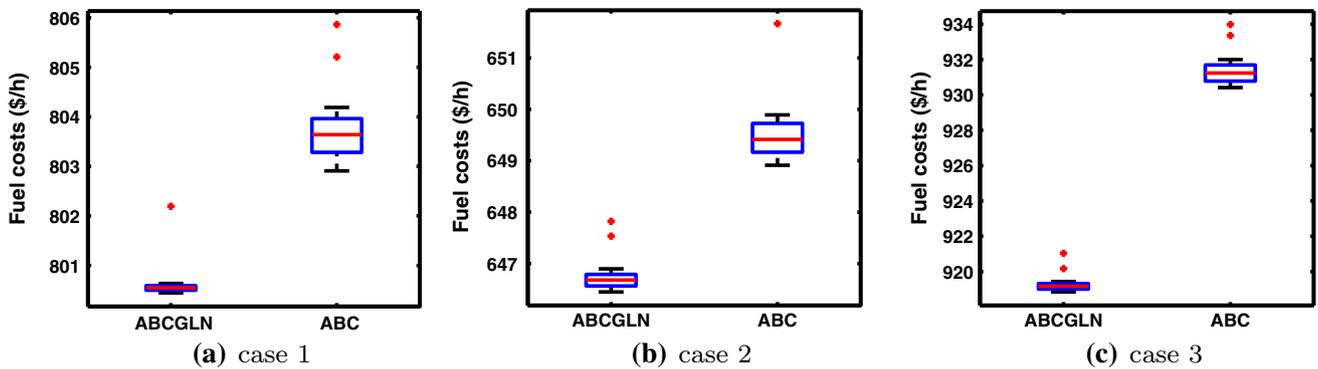


Fig. 3 Box-plots for ABCGLN and ABC for minimum fuel cost for 100 runs

- The number of simulations/run =100,
- Colony size  $NP = 50$  (Diwold et al. 2011; El-Abd 2011) and number of food sources  $SN = NP/2$ ,
- $\phi_{ij} = rand[-1, 1]$  and limit = dimension  $\times$  number of food sources =  $D \times SN$  (Karaboga and Akay 2011; Akay and Karaboga 2010),
- The parameter  $cr$  in Algorithm 1 is set to 0.3 (Jadon et al. 2014a),
- The cost coefficients for all cases are given in Tables 2, 3 and 4,
- The upper and lower limits for control variables is taken from Lee et al. (1985) and is given in Table 1,

- The load data is adopted from Alsac and Stott (1974) and Lee et al. (1985) and given in Table 5,
- The line data for 30-bus system is taken from Alsac and Stott (1974) and Lee et al. (1985),
- The penalty factors  $\lambda_p$ ,  $\lambda_v$ ,  $\lambda_q$  and  $\lambda_s$  are set to  $10^5$ .

### 4.3 Results analysis of experiments

Tables 6, 7, 8 and 9 present the numerical results comparison of all considered algorithms for cases 1–3 with the experimental settings shown in Sect. 4.2. Table 6 shows the optimal control parameter settings achieved by the ABCGLN algorithm for OPF problem in each case. The results comparison of the ABCGLN with other algorithms in terms of minimum fuel cost, average minimum fuel cost over 100 runs and total execution time (in seconds) to reach at achieved minimum by the algorithm, is presented in Tables 7, 8 and 9. The bold entries in Tables 7–9 represent the best value among all entries. It can be observed from these tables that most of the time, ABCGLN is more reliable as it has achieved minimum fuel cost than all the considered algorithms in all three cases. However, it spent more time than the some algorithms in some cases but in those cases also, it achieved better minima. In terms of average minimum fuel cost in all runs, ABCGLN is also reliable than the others as it gave less mean fuel cost for all cases. If ABCGLN is compared to basic ABC only then also it shows a clear superiority of ABCGLN in every terms of accuracy, reliability and efficiency.

Further, to compare convergence speed of ABCGLN with respect to ABC, we calculated the convergence rate in Fig. 2a–c. It is clear from the these figures that ABCGLN has higher convergence rate than the ABC to solve OPF problem for all three cases. Since the empirical distribution of results can efficiently be represented by box-plot (Williamson et al. 1989), the box-plots for minimum fuel cost for 100 runs for ABCGLN and basic ABC algorithms have been represented in Fig. 3 to analyze the algorithms output more intensively. Figure 3 shows that ABCGLN is cost effective as the interquartile range and median are very low for ABCGLN.

## 5 Conclusion

This paper presents the solution of OPF problem of IEEE 30-bus system using a recent version of ABC algorithms, named ABCGLN. The ABCGLN is tested and compared on standard IEEE 30-bus system with various objective functions which minimizes the fuel cost function, non-smooth piece-wise quadratic cost function and quadratic cost function with valve point effects. The reported results

show the efficiency and robustness of ABCGLN in solving OPF. Results of ABCGLN are also compared with other algorithms' results available in the literature and comparisons confirm the superiority of ABCGLN over other meta-heuristic algorithms to find feasible and more accurate solution to OPF problems without any restrictions on the shape of the fuel cost curves.

## References

- Abido MA (2002a) Optimal power flow using particle swarm optimization. *Int J Electr Power Energy Syst* 24(7):563–571
- Abido MA (2002b) Optimal power flow using tabu search algorithm. *Electr Power Compon Syst* 30(5):469–483
- Abou El Ela AA, Abido MA, Spea SR (2010) Optimal power flow using differential evolution algorithm. *Electr Power Syst Res* 80(7):878–885
- Akay B, Karaboga D (2010) A modified artificial bee colony algorithm for real-parameter optimization. *Inf Sci* 192:120–142. doi:10.1016/j.ins.2010.07.015
- Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J Intell Manuf* 23(4):1001
- Al-Muhawesh TA, Qamber IS (2008) The established mega watt linear programming-based optimal power flow model applied to the real power 56-bus system in eastern province of Saudi Arabia. *Energy* 33(1):12–21
- Alsac O, Stott B (1974) Optimal load flow with steady-state security. *IEEE Trans Power Appar Syst PAS-93(3):745–751*
- Bakirtzis AG, Biskas PN, Zoumas CE, Petridis V (2002) Optimal power flow by enhanced genetic algorithm. *IEEE Trans Power Syst* 17(2):229–236
- Banharnsakun A, Achalakul T, Sirinaovakul B (2011) The best-so-far selection in artificial bee colony algorithm. *Appl Soft Comput* 11(2):2888–2901
- Banharnsakun A, Sirinaovakul B, Achalakul T (2012) Job shop scheduling with the best-so-far ABC. *Eng Appl Artif Intell* 25(3):583–593
- Bansal JC, Sharma H, Arya KV, Nagar A (2013a) Memetic search in artificial bee colony algorithm. *Soft Comput* 17(10):1911–1928
- Bansal JC, Sharma H, Jadon SS (2013b) Artificial bee colony algorithm: a survey. *Int J Adv Intell Paradigms* 5(1):123–159
- Bansal JC, Sharma H, Nagar A, Arya KV (2013c) Balanced artificial bee colony algorithm. *Int J Artif Intell Soft Comput* 3(3):222–243
- Bhattacharya A, Chattopadhyay PK (2011) Application of biogeography-based optimisation to solve different optimal power flow problems. *IET Gener Transm Distrib* 5(1):70–80
- Chidambaram C, Lopes HS (2009) A new approach for template matching in digital images using an artificial bee colony algorithm. In: *World congress on nature and biologically inspired computing, 2009. NaBIC 2009. IEEE, Coimbatore*, pp 146–151
- Dieu VN, Ongsakul W (2006) Enhanced augmented Lagrange Hopfield network for economic dispatch with piecewise quadratic cost functions. *Fuel* 2(2):2–2
- Diwold K, Aderhold A, Scheidler A, Middendorf M (2011) Performance evaluation of artificial bee colony optimization and new selection schemes. *Memet Comput* 1–14
- Dommel HW, Tinney WF (1968) Optimal power flow solutions. *IEEE Trans Power Appar Syst PAS-87(10):1866–1876*

- Duman S, Güvenç U, Sönmez Y, Yörükeren N (2012) Optimal power flow using gravitational search algorithm. *Energy Convers Manag* 59:86–95
- El-Abd M (2011) Performance assessment of foraging algorithms vs. evolutionary algorithms. *Inf Sci* 182(1):243–263
- Gao W, Liu S (2012) A modified artificial bee colony algorithm. *Comput Oper Res* 39(3):687–697
- Granville S (1994) Optimal reactive dispatch through interior point methods. *IEEE Trans Power Syst* 9(1):136–146
- Habibollahzadeh H, Luo G-X, Semlyen A (1989) Hydrothermal optimal power flow based on a combined linear and nonlinear programming methodology. *IEEE Trans Power Syst* 4(2):530–537
- Jadon SS, Bansal JC, Tiwari R, Sharma H (2014a) Artificial bee colony algorithm with global and local neighborhoods. *Int J Syst Assur Eng Manag* 5:1–13
- Jadon SS, Bansal JC, Tiwari R, Sharma H (2014b) Expedited artificial bee colony algorithm. In: *Proceedings of the third international conference on soft computing for problem solving*. Springer, India, pp 787–800
- Jones KO, Bouffet A (2008) Comparison of bees algorithm, ant colony optimisation and particle swarm optimisation for PID controller tuning. In: *Proceedings of the 9th international conference on computer systems and technologies and workshop for PhD students in computing*. ACM, New York, pp IIIA-9
- Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report TR06. Erciyes University Press, Erciyes
- Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132
- Karaboga D, Akay B (2011) A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Appl Soft Comput* 11(3):3021–3031
- Karaboga N, Cetinkaya MB (2011) A novel and efficient algorithm for adaptive filtering: artificial bee colony algorithm. *Turk J Electr Eng Comput Sci* 19:175–190
- Karaboga B, Akay D, Ozturk C (2008) Training neural networks with ABC optimization algorithm on medical pattern classification. In: *International conference on multivariate statistical modelling and high dimensional data mining*, Kayseri, Turkey, June 1923
- Kavian YS, Rashedi A, Mahani A, Ghassemlooy Z (2012) Routing and wavelength assignment in optical networks using artificial bee colony algorithm. *Optik Int J Light Electron Opt*
- Lam SSB, Raju ML, Ch S, Srivastav PR et al (2012) Automated generation of independent paths and test suite optimization using artificial bee colony. *Procedia Eng* 30:191–200
- Lee KY, Park YM, Ortiz JL (1985) A united approach to optimal real and reactive power dispatch. *IEEE Trans Power Appar Syst* 5:1147–1153
- Lei X, Huang X, Zhang A (2010) Improved artificial bee colony algorithm and its application in data clustering. In: *IEEE fifth international conference on bio-inspired computing: theories and applications (BIC-TA)*. IEEE, Changsha, pp 514–521
- Li HJ, Li JJ, Kang F (2011) Artificial bee colony algorithm for reliability analysis of engineering structures. *Adv Mater Res* 163:3103–3109
- Mandal SK, Chan FTS, Tiwari MK (2012) Leak detection of pipeline: an integrated approach of rough set theory and artificial bee colony trained SVM. *Expert Syst Appl* 39(3):3071–3080
- Mota-Palomino R, Quintana VH (1986) Sparse reactive power scheduling by a penalty function—linear programming technique. *IEEE Trans Power Syst* 1(3):31–39
- Nayak SK, Krishnanand KR, Panigrahi BK, Rout PK (2009) Application of artificial bee colony to economic load dispatch problem with ramp rate limits and prohibited operating zones. In: *World congress on nature and biologically inspired computing, 2009. NaBIC 2009*. IEEE, Coimbatore, pp 1237–1242
- Ongsakul W, Tantimaporn T (2006) Optimal power flow by improved evolutionary programming. *Electr Power Compon Syst* 34(1):79–95
- Pawar P, Rao R, Shankar R (2008) Multi-objective optimization of electro-chemical machining process parameters using artificial bee colony (ABC) algorithm. In: *Advances in mechanical engineering (AME-2008)*, Surat, India
- Roa-Sepulveda CA, Pavez-Lazo BJ (2003) A solution to the optimal power flow using simulated annealing. *Int J Electr Power Energy Syst* 25(1):47–57
- Sayah S, Zehar K (2008) Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. *Energy Convers Manag* 49(11):3036–3042
- Sharma H, Bansal JC, Arya KV (2013) Opposition based lévy flight artificial bee colony. *Memet Comput* 5(3):213–227
- Singh A (2009) An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Appl Soft Comput* 9(2):625–631
- Sulaiman MH, Mustafa MW, Shareef H, Abd Khalid SN (2012) An application of artificial bee colony algorithm with least squares support vector machine for real and reactive power tracing in deregulated power system. *Int J Electr Power Energy Syst* 37(1):67–77
- Sun DI, Ashley B, Brewer B, Hughes A, Tinney WF (1984) Optimal power flow by Newton approach. *IEEE Trans Power Appar Syst PAS-103(10):2864–2880*
- Vaisakh K, Srinivas LR (2011) Evolving ant direction differential evolution for OPF with non-smooth cost functions. *Eng Appl Artif Intell* 24(3):426–436
- Wei H, Sasaki H, Kubokawa J, Yokoyama R (1998) An interior point nonlinear programming for optimal power flow problems with a novel data structure. *IEEE Trans Power Syst* 13(3):870–877
- Williamson DF, Parker RA, Kendrick JS (1989) The box plot: a simple visual method to interpret data. *Ann Intern Med* 110(11):916
- Xing F, Fenglei L, Haijun D (2007) The parameter improvement of bee colony algorithm in TSP problem. *Sci Pap Online* November (2007)
- Xu C, Duan H (2010) Artificial bee colony (ABC) optimized edge potential function (EPF) approach to target recognition for low-altitude aircraft. *Pattern Recognit Lett* 31(13):1759–1772
- Yeh WC, Hsieh TJ (2011) Solving reliability redundancy allocation problems using an artificial bee colony algorithm. *Comput Oper Res* 38(11):1465–1473
- Yuryevich J, Wong KP (1999) Evolutionary programming based optimal power flow algorithm. *IEEE Trans Power Syst* 14(4):1245–1250
- Zhu G, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl Math Comput* 217(7):3166–3173