Control Engineering Practice 61 (2017) 163–172

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

An improved artificial bee colony optimization algorithm based on orthogonal learning for optimal power flow problem

Wenlei Bai[a,*], Ibrahim Eke[b], Kwang Y. Lee[a]

[a] Baylor University, Waco, TX 76706 USA
[b] Kirikkale University, Kirikkale 71000 Turkey

ARTICLE INFO

Keywords:
Modern heuristic optimization
Optimal power flow (OPF)
Artificial bee colony (ABC)
Orthogonal learning (OL)
Nonlinear optimization

ABSTRACT

The increasing fuel price has led to high operational cost and therefore, advanced optimal dispatch schemes need to be developed to reduce the operational cost while maintaining the stability of grid. This study applies an improved heuristic approach, the improved Artificial Bee Colony (IABC) to optimal power flow (OPF) problem in electric power grids. Although original ABC has provided robust solutions for a range of problems, such as the university timetabling, training neural networks and optimal distributed generation allocation, its poor exploitation often causes solutions to be trapped in local minima. Therefore, in order to adjust the exploitation and exploration of ABC, the IABC based on the orthogonal learning is proposed. Orthogonal learning is a strategy to predict the best combination of two solution vectors based on limited trials instead of exhaustive trials, and to conduct deep search in the solution space. To assess the proposed method, two fuel cost objective functions with high non-linearity and non-convexity are selected for the OPF problem. The proposed IABC is verified by IEEE-30 and 118 bus test systems. In all case studies, the IABC has shown to consistently achieve a lower cost with smaller deviation over multiple runs than other modern heuristic optimization techniques. For example, the quadratic fuel cost with valve effect found by IABC for 30 bus system is 919.567 $/hour, saving 4.2% of original cost, with 0.666 standard deviation. Therefore, IABC can efficiently generate high quality solutions to nonlinear, nonconvex and mixed integer problems.

1. Introduction

In electric power grids, the optimal power flow (OPF) problem is of great importance for power system operators (SO) to maintain a reliable and economic power system operation. The main goals of OPF are to optimize the fuel cost, power losses, voltage stability, and emission cost, while satisfying system constraints. Traditional OPF involving conventional fossil-fuel power plants is a highly nonlinear, nonconvex and mixed integer problem (Adaryani & Karami, 2013; Bai, Abedi, & Kwang, 2016). For example, the cost function of a fossil-fuel power plant can be quadratic or in other nonlinear form when the valve effect is considered. An overview of OPF can be found in (Cain, O'Neill, & Castillo, 2012; Gan, Thomas, & Zimerman, 2000; Momoh, Koessler, Bond, & Stott, 1997).

In all, the OPF is a non-linear, non-convex optimization problem due to the cost functions and constraints of a large number of power plants integrated into the power grid. A wide range of traditional optimization techniques such as quadratic programming, nonlinear programming, interior point method, mixed integer programming (Alsac & Stott, 1974; Burchett, Happ, & Vierath, 1984; Hua, Sasaki, Kubokawa, & Yokoyama, 1998; Shoults & Sun, 1982) have already been implemented in this field. Some of the techniques have even been adopted by industry because of their fast convergence and robustness. However, those approaches linearize the OPF problem first, and fail to consider the non-smooth, non-differentiable and non-convex properties of the system.

To circumvent such problem, various modern heuristic optimization algorithms have been developed for power system optimization (Lee & El-Sharkawi, 2008) because such techniques tackle the original problem without modifying it. In general, heuristic algorithms are developed based on two categories which are single-solution based and population based approaches. Several examples of single-solution based approach are tabu search and simulated annealing (Abido, 2002; Soares, Vale, Morais, & Faria, 2011), while population based approaches include particle swarm optimization (PSO), gravitational search algorithm (GSA), differential evolution (DE), genetic algorithm (GA), harmony search, and artificial bee colony (Abou, A, Abido, & Spea, 2010; Adaryani & Karami, 2013; Bakirtzis, Biskas, Zoumas, &

* Corresponding author.
E-mail addresses: Wenlei_Bai@baylor.edu (W. Bai), eke@kku.edu.tr (I. Eke), Kwang_Y_Lee@baylor.edu (K.Y. Lee).

http://dx.doi.org/10.1016/j.conengprac.2017.02.010
Received 20 May 2016; Received in revised form 19 February 2017; Accepted 20 February 2017
Available online 04 March 2017
0967-0661/

Petridis, 2002; Park, Jeong, Shin, & Lee, 2010; Sivasubramani & Swarup, 2011). In addition to these original heuristic methods, enhanced approaches based on the original ones have been developed for more efficient search. The authors in (Bakirtzis et al., 2002) improved basic GA to solve the OPF by introducing an advanced and problem-specific genetic operator. Such operator includes the fitness scaling and elitism features, and the algorithm was tested on IEEE-30 and IEEE RTS-96 system. Reference (Park et al., 2010) proposed an improved PSO to tackle the problem considering the valve point effect on the regular quadratic fuel cost function.

This study focuses on the artificial bee colony (ABC) method reported by Karaboga in 2005 (Karaboga, 2005). The ABC falls into the category of population-based optimization algorithms which have been demonstrated competitive to other methods because the ABC controls fewer parameters and is robust (Karaboga & Akay, 2009; Pan, Tasgetiren, Suganthan, & Chua, 2011; Zhan, Zhang, Li, & Shi, 2011). The balance between exploration and exploitation is an important issue for modern heuristic optimization techniques. The former is the capability of investigating various unknown regions in search space, and the latter is the ability to make the best decision given current information (Crepinsek, Liu, & Mernik, 2013). In reality, the two aspects are contradictory to each other and therefore a well balanced approach needs to be found. The search process of ABC performs well for exploration because the searching scheme is random enough for exploration; however, it performs poorly for exploitation and thus causes poor convergence (Gao, Liu, & Huang, 2013).

In order to enhance the ability of exploitation, researchers proposed a search mechanism which utilizes the information of current best solution inspired by differential evolution (DE). In such search mechanism, onlooker bees only search around the best solution formed in the previous iteration according to a predefined probability (Gao & Liu, 2012). Gao and Liu (2011) improved the initialization phase in that the chaotic system was utilized, and modified the search mechanism using the information of current best solution. Such work is able to improve the exploitation.

The search equation of the original ABC randomly selects a dimension of the solution vector and performs mutation with the same dimension of another solution vector. Here the dimension refers to the number of control variables in a solution vector. For example, if the solution vector consists of 24 control variables, it is interpreted as 24 dimensions in such solution vector. However, this search scheme falls short of effectiveness because one solution vector may contain useful information on some dimensions while the other solution may contain good information on its other dimensions. In other words, merely concentrating on a specific dimension of the solution will be likely to lose other useful information for solution improvement. Therefore, in order to update the solution considering all the information of each dimension from two candidate solutions, inspired by the orthogonal experimental design (OED) Gao (2013) proposed an orthogonal learning (OL) technique to obtain better exploitation. The OED is utilized to determine the best combination out of two vectors via a relatively small number of experimental tests instead of exhaustive trials (Zhan et al., 2011; Zhang & Leung, 1999). The OL strategy is implemented with the help of OED, and details of such strategy will be described later.

In all, the previous works on OPF have either fallen short of the ability to tackle original problem without approximation or the balance in exploration and exploitation in modern heuristic techniques. Thus far, to the best knowledge of authors, the application of ABC based on orthogonal learning on power system operation problems has not been documented in the literature yet. Here, we first propose this method to handle the OPF problem, which is to be our main contribution. With that, better optimization solution can be found by improving the balance in exploration and exploitation. The performance was tested on modified IEEE 30 and 118 bus test systems and comparative analysis was conducted with other methods. For the case of minimizing fuel cost considering valve effect, the total cost can be reduced by 4.2% compared with the original ABC. Power system is a highly non-linear system and therefore many control and optimization become hard-to-solve problems without linearizing system. IABC is to possibly further improve the solutions of those problems such as controlling the Flexible Alternating Current Transmission System (FACTS) devices, optimizing the placement of distributed generators.

2. Problem formulation

2.1. Traditional OPF problem

The objective of traditional optimal power flow (OPF) is to minimize fuel cost for power generation by determining a setting of control variables while satisfying network constraints and operational requirements. Its mathematical formulation is:

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

$$s.\ t.\ \ g(x, u) = 0 \tag{2}$$

$$h(x, u) \le 0 \tag{3}$$

where vector u represents control variables and it includes generator real power P_G except at slack bus, generator bus voltage V_G, transformer tap TP (discrete variable), and shunt compensator Q_C (discrete variable) at selected buses; vector x represents state variables and it includes real power P_{G1} at slack bus, voltages V_L at load bus, reactive power Q_G at generator bus, and loadings S_L of transmission lines.

The objective functions f from (1) considered in the study are the real power losses and total fuel cost. Two different fuel cost functions are considered here, quadratic cost functions with and without the valve point loading (Park et al., 2010):

$$f_1 = a_i P_{Gi}^2 + b_i P_{Gi} + c_i \tag{4}$$

$$f_2 = a_i P_{Gi}^2 + b_i P_{Gi} + c_i + |d_i \sin(e_i(P_{Gi,\min} - P_{Gi}))| \tag{5}$$

where a_i, b_i, c_i, e_i, and P_{Gi} denote for the fuel cost coefficients and real power of the i-th unit. Fig. 1 shows the effect of valve point loading on a quadratic cost function. In a real power plant, steam is controlled by valves to enter the turbine through separate nozzle groups. The best efficiency is achieved when each nozzle group operates at full output (Decker & Brooks, 1958). Therefore in order to achieve highest possible efficiency for given output, valves are opened in sequence and this results in a rippled cost curve, as shown in Fig. 1.

Resistance and reactance in transmission lines cause real power loss, and minimizing real power loss is one of the major concerns for system operation. The mathematical formation of the objective function is shown as follows:

$$f_3 = \sum_{k=1}^{N_l} \frac{r_k}{r_k^2 + x_k^2} [V_i^2 + V_j^2 - 2V_i V_j \cos(\delta_i - \delta_j)] \quad \forall\, i, \forall\, j \tag{6}$$

where N_l is the number of transmission lines, r_k and x_k represents the

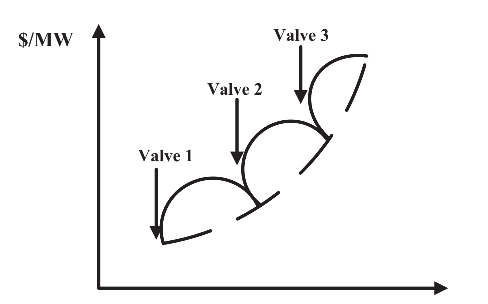

Fig. 1. Effect of valve point loading on a quadratic cost function.

resistance and reactance of the transmission line k that links bus i and j; V_i, V_j, δ_i and δ_j are the voltage magnitudes and angles at bus i and j, respectively. In all, f_1, f_2 and f_3 are the three cost functions considered for case studies.

The equality constraint (2) is the AC power flow balance equation at each bus representing that the power flowing into that specific bus is equal to the power flowing out, and is defined as:

$$P_i = V_i \sum_{j=1}^{N} V_j Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij})$$
$$Q_i = V_i \sum_{j=1}^{N} V_j Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}) \quad \forall\, i, \forall\, j \tag{7}$$

where N is the total number of buses, P_i and Q_i are the injected real and reactive power of bus i; δ_i and δ_j are the angle at bus i and j; Y_{ij} and θ_{ij} are the Y-bus admittance matrix elements.

Inequality constraints h in (3) are listed as generator limits, tap position of transformers, shunt capacitor constraints, security constraints, load bus voltage and transmission line flows.

2.1.1. Generator limits

$$P_{Gi,\max} \le P_{Gi} \le P_{Gi,\max}$$
$$Q_{Gi,\max} \le Q_{Gi} \le Q_{Gi,\max}$$
$$V_{Gi,\max} \le V_{Gi} \le V_{Gi,\max} \quad i \in N_G \tag{8}$$

where N_G is the number of generators, and the minimum/maximum real, reactive power and voltage limits of unit i are denoted by $P_{Gi,\ min}$, $P_{Gi,\ max}$, $Q_{Gi,\ min}$, $Q_{Gi,\ max}$, $V_{Gi,\ min}$, and $V_{Gi,\ max}$.

2.1.2. Tap positions of transformers

$$TP_{i,\min} \le TP_i \le TP_{i,\max} \quad i \in N_T \tag{9}$$

where N_T is the number of tap-changing transformers; $TP_{i,\ min}$, and $TP_{i,\ max}$, are the limits of transformers.

2.1.3. Shunt capacitors constraints

$$Q_{ci,\min} \le Q_{ci} \le Q_{ci,\max} \quad i \in N_c \tag{10}$$

where N_c is the number of shunt capacitors; $Q_{ci,\ min}$ and $Q_{ci,\ max}$ are the limits of shunt capacitors.

2.1.4. Security constraints on the limits of load bus voltage and transmission line flows

$$V_{Li,\min} \le V_{Li} \le V_{Li,\max} \quad i \in N_{pq}$$
$$S_{Li} \le S_{Li,\max} \quad i \in N_l \tag{11}$$

where N_{pq} is the number of PQ bus, $V_{Li,\ min}$, $V_{Li,\ max}$ and $S_{Li,\ max}$ are the limits of voltage magnitudes at PQ buses and maximum line flow of transmission line i respectively. There are three types of buses in power system: slack bus, PV bus and PQ bus. Slack bus is to balance the real and reactive power in the system while performing load flow calculations, it is also known as reference bus. PV bus is the node where real power P and voltage magnitude V are specified, it is also known as generator bus. PQ bus is the node where real and reactive power is specified, known as load bus.

2.2. Incorporation of inequality constraints

The control variables (real power generation of PV buses, voltage at all generator buses, transformer tap settings, and shunt compensators) are randomly initialized within the feasible domain, while a penalty function is introduced in order to force the state variables into the direction of the feasible domain as well. In other words, penalty functions are utilized to handle the inequality constraints. The penalty cost function is defined as:

$$p(x_i) = \begin{cases} (x_i - x_{i,\max})^2 & if \quad x_i > x_{i,\max} \\ (x_{i,\min} - x_i)^2 & if \quad x_i < x_{i,\min} \\ 0 & if \quad x_{i,\min} \le x_i \le x_{i,\max} \end{cases} \tag{12}$$

where p is the penalty function of state variable x_i at bus i. The penalty cost increases with a quadratic form when state variables are exceeding the limits and the cost is zero if the constraints are not violated. For example, if one of the PQ bus voltages exceeds the limit, a certain amount of penalty will be added, which leads to the increase of total cost, and eventually this solution will be abandoned.

Thus, the augmented objective function by adding the penalty function of the PQ bus voltage, reactive power generation, slack bus, and transmission line capacity is described as:

$$F = f + C_p p(P_{G1}) + C_q \sum_{i=1}^{N_g} p(Q_{Gi}) + C_v \sum_{i=1}^{N_{pq}} p(V_{Li}) + C_s \sum_{i=1}^{N_l} p(S_{Li}) \tag{13}$$

where f is the original fuel cost function (f_1, f_2, or f_3 in this paper), C_p, C_q, C_v and C_s respectively denote penalty factors of real power generation of slack bus, reactive power output of the generator buses, and PQ bus voltage and transmission line capacity.

3. Design iabc based on orthogonal learning

3.1. Original ABC algorithm

In the original ABC by Karaboga, initial artificial bees are spread out randomly in a multidimensional search space. Each artificial bee has the ability to store current information and communicate with neighbours. Mimicking the foraging behaviours of natural honey bee swarms, ABC has been addressed in various applications (Fong, Asmuni, & McCollum, 2015; Karaboga & Akay, 2012; Karaboga, Alay, & Ozturk, 2007; Singh, 2009).

There are employed bees, onlookers, and scouts in the population. The merit of population-based algorithm is that every agent works collaboratively to search for solutions. Thus in ABC each artificial bee communicates and cooperates with one another to explore food sources by evaluating the quality of the food sources, called nectars. Employed bees first share nectars information to onlooker bees. Then, onlooker bees find food sources based on the nectars information and more profitable sources are more likely to be chosen by onlookers. If a source is not worth exploiting anymore, the source will be abandoned by bees and the employed bee of that source will become a scout to randomly search the environment.

When ABC is applied to the optimal power flow problem, the food source means feasible solutions, nectars mean the fitness values of cost functions, and more profitable sources mean the solutions that correspond with high fitness values. The process of onlooker bees finding more profitable sources is actually to update the current solution to another candidate solution.

At initialization, each solution vector $X_i = \{X_{i,1}, X_{i,2}, \ldots, X_{i,D}\}$ is formed randomly within the limits of the control variables as follows:

$$X_{i,j} = X_{i,j\ \min} + rand(0, 1) \times (X_{i,j\ \max} - X_{i,j\ \min}) \tag{14}$$

where X_{i,j_min} and X_{i,j_max} are the lower and upper bounds for dimension j; i is from 1 to SN, and j is a random number from 1 to D, and SN is the number of employed bees and onlooker bees, D is the number of control variables and $rand(0,1)$ is a uniformly distributed random number in (0,1).

On employed bee phase, each bee searches for rich artificial food sources via updating current solutions based on their neighbourhood's information and assess the nectar of new solutions. The search equation which is adopted to update a candidate solution V_i is defined as:

$$V_{i,j} = X_{i,j} + \Phi_{i,j} \times (X_{i,j} - X_{k,j}) \tag{15}$$

Table 1
(Zhan et al., 2011) Chemical reaction experiment.

Factors			
Levels	A Temp. °C	B Oxygen (cm^3)	C Water (%)
1 *L1*	80	90	5
2 *L2*	85	120	6
3 *L3*	90	150	7

where k is an integer different from i, uniformly chosen from [1, SN], $\Phi_{i,j}$ is a random number from [−1,1]. Such searching scheme is random enough for exploration. If the updated solution has higher fitness value than the old one, employed bee will memorize the new solution and discard the old one; otherwise they will keep the old solutions. This particular process is called 'greedy selection'. As mentioned previously, the search scheme (15) of basic ABC only focuses on one dimension of the solution (j is a randomly selected dimension) to update the food source, which leads to the deficiency of exploiting the solution space, while other dimensions may contain good information for improving food sources as well.

When all the employed bees finish their updating process, on looker bees obtain the fitness information of food from the employed bees. The onlooker bees will continue exploiting for good food source according to the probability which is proportional to each solution's quality. The process of choosing food by onlooker bees is based on a scheme called roulette wheel selection, which is similar to the genetic algorithm. The roulette wheel selection scheme, as defined in the following, is adopted to mimic the fact that onlooker bees tend to update the food sources which have higher nectar:

$$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \tag{16}$$

where fit_i and P_i are the fitness value and probability associated with solution i, respectively. By adopting such a scheme, solutions with higher nectar will be assigned with higher probability and in turn will have higher chance to be selected by onlooker bees. The onlooker bee updates the selected solution using (15) as the employed bees do and memorize the solution by greedy selection. This process will continue until every onlooker bee finishes its search.

After a predefined number of searching cycles, food sources become exhausted (inactive solution) if their quality could not be improved anymore, then the employed bee will become a scout bee to start a random direction to search for new food source. This process is to avoid local optima. In the original ABC, only one scout in each cycle is allowed to occur (Karaboga, 2005). After finding a new food source, the scout bee will turn itself back to employed bee.

Even though the basic ABC has proven its robustness to various applications, however like many metaheuristic techniques, there are still some identified weaknesses in the balance of exploitation and exploration (Gao & Liu, 2011, 2012; Zhu & Kwong, 2010). In reality, the exploitation and exploration is in contradiction and the key for improving the solution of optimization problems is to balance exploitation and exploration (Al-Betar, Khader, & Zaman, 2012). The deficiency of basic ABC is inherited from each phase and explained as following: in the employed bee phase, the update of current solution is based on neighbourhood' search which lead to poor exploitation ability because of the limited information that neighbourhood can provide. The same issue arises to the onlooker bee phase because the same updating scheme is utilized. In addition, the roulette wheel selection scheme may lead to imbalance between exploitation and exploration because this scheme merely emphasizes on exploiting the solutions with high nectars; however, those with low nectars may also contain useful information to improve solutions. Lastly, in the scout bee phase, inactive food sources will be abandoned and the scout will search a new food source to replace the abandoned one randomly. This can help avoid local minimal and yet the random search of new food will further decrease the exploitation and convergence speed.

3.2. Orthogonal learning strategy

The OL method, which is analogous to orthogonal experimental design (OED), can obtain the best candidate solution with few searching combinations. The OED was first introduced by R. A. Fisher in the 1920's to study the effect of multi-factors to the experimental output. As a powerful statistical tool, the OED was utilized to discover how much rain, water, fertilizer, sunshine, etc., were required to produce the best crop (Roy, 2011). To illustrate the concept of OED the following simple chemical reaction experiment is considered as shown in Table 1 (Zhan et al., 2011).

In this experiment, there are three *factors*: temperature (A), amount of Oxygen (B) and percentage of water (C) determining a chemical conversion rate. In addition, each factor contains three *levels*. For instance, the water can be 5%, 6%, or 7%. Thus there are 3^3=27 total number of combinations that need to be experimented to find the best conversion rate. However, with the help of OED, the best combination can be predicted by only testing few representative combinations, thus reducing total testing cost. The following describes

Table 2
Best combination levels BY OED.

Comb.	A: Temp. (°C)	B: Oxygen (cm^3)	C: Water (%)	Results (reaction rate)
*Cb*1	(1) 80	(1) 90	(1) 5	f_1=31
*Cb*2	(1) 80	(2) 120	(2) 6	f_2=54
*Cb*3	(1) 80	(3) 150	(3) 7	f_3=38
*Cb*4	(2) 85	(1) 90	(2) 6	f_4=53
*Cb*5	(2) 85	(2) 120	(3) 7	f_5=49
*Cb*6	(2) 85	(3) 150	(1) 5	f_6=42
*Cb*7	(3) 90	(1) 90	(3) 7	f_7=57
*Cb*8	(3) 90	(2) 120	(1) 5	f_8=62
*Cb*9	(3) 90	(3) 150	(2) 6	f_9=64

levels	Factor Analysis		
L1	H_{A1}=(f_1+f_2+f_3)/3=41	H_{B1}=(f_1+f_4+f_7)/3=47	H_{C1}=(f_1+f_6+f_8)/3=45
L2	H_{A2}=(f_4+f_5+f_6)/3=48	H_{B2}=(f_2+f_5+f_8)/3=55	H_{C2}=(f_2+f_4+f_9)/3=57
L3	H_{A3}=(f_7+f_8+f_9)/3=61	H_{B3}=(f_3+f_6+f_9)/3=48	H_{C3}=(f_3+f_5+f_7)/3=48
OED Results	*A*3	*B*2	*C*2

the definition of orthogonal array and factor analysis, which leads to a comprehensive understanding of OL.

1) *Orthogonal Array*: First, we use '$L_N(s^k)$' to denote an array with s levels per factor for k factors, and L and N respectively represent an array and the total number of combinations. For example, in the chemical reaction experiment given in Table 1 we define an array $L_9(3^3)$ with 3 factors, 3 levels per factor, and 9 combinations, as follows:

$$L_9(3^3) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 3 \\ 3 & 2 & 1 \\ 3 & 3 & 2 \end{bmatrix} \tag{17}$$

An $N\times k$ array A is defined as an orthogonal array (OA) which has *index* λ with *strength* t on $0\le t\le k$ when each $N\times t$ sub-array of A contains all the combinations of t-tuple exactly λ times as a row (Hedayat, Sloane, & Stufken, 1999). Eq. (17) gives an example of a 9×3 OA with strength 2 and index 1. The reason why such OA is strength 2 and index 1 is because tuples (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3) appear in any two columns one time. Note that an array with strength 3 and index 1 yields the full 27 combinations of triplets.

An OA is a predefined table for the OED method to work on. As mentioned earlier, the benefit of utilizing OED is to obtain the best combination by conducting only few experiments. The total nine experiments specified by the $L_9(3^3)$ are presented in Table 2. For instance, the first row is [1 1 1], which means that the factors A (Temperature), B (Oxygen), and C (Water) are all designed to the first level (80°, 90 cm^3, and 5%, respectively). The last column shows the results of the experiment for each combination.

2) *Factor Analysis*: Factor analysis (FA) is the tool to assess the effects of each factor on the experimental results in order to determine the best combination of levels. With all N cases of experimental results of OA known, the FA is conducted to determine the best combination. The process of FA is described as:

To determine the effect of each level for each factor, H_{ks} is evaluated as the average effect of level s (s = 1,2,3) for the k-th factor (k = A, B, C),

$$H_{ks} = \frac{\sum_{n=1}^{9} f_n \times z_{nks}}{\sum_{n=1}^{9} z_{nks}} \tag{18}$$

where f_n is the experimental result of the n-th (n=1,2…,9) combination, z_{nks} is 1 if in the n-th (n=1,2…,9) combination, the level of the k-th factor (k= A, B, C) is s (s=1,2,3), otherwise is 0. For instance if we want to evaluate the effect of level 1 in factor B (B1), by inspection from the 3-rd column of Table 2 we find that combinations Cb1, Cb4 and Cb7 involve all the experiments of level 1 for factor B, with the corresponding experimental results f_1=31, f_4=53 and f_7=57, and the average effect H_{B1}=47. After computing the effect of all levels for each factor, the most effective level for each factor can be determined by selecting the highest quantity of H_{ks} for each factor. The FA results can be found in Table 2 and the details of FA are explained in (Zhan et al., 2011; Zhang & Leung, 1999). From Table 2, the best combination determined by FA is (A3, B2, C2). Note that this combination (90 °C, 120 cm^3, 6%) is not one of the nine tested combinations. The OL will be implemented in the ABC algorithm in order to obtain the best

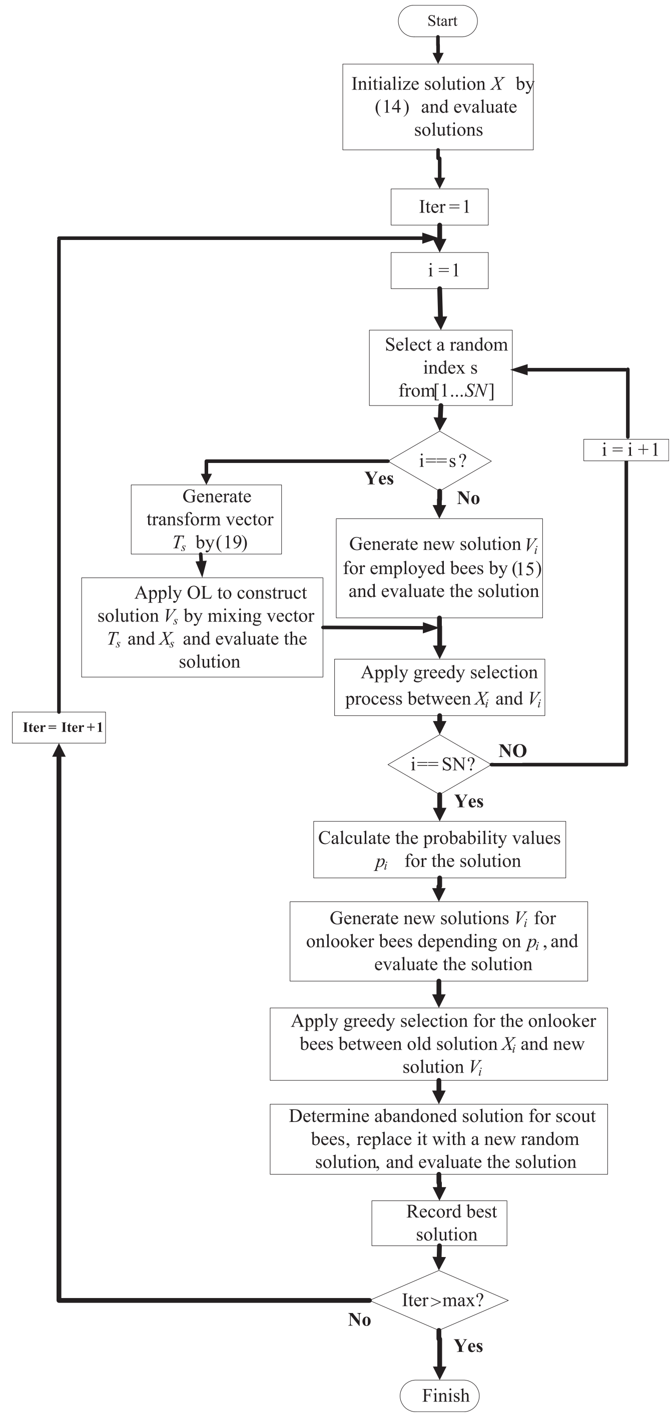

Fig. 2. The overall structure of IABC.

candidate solution efficiently with few searching combinations by the analogy of OED. However, there is no guarantee that OL can be fit for any dimension of control variables problem because only fixed size of orthogonal array can be generated.

3.3. Improved ABC with orthogonal learning

As mentioned previously, the original ABC has poor efficiency on exploitation, and to overcome these issues, the OL strategy is proposed to find an efficient candidate solution. Considering the aforementioned discussion, the process of implementing OL into ABC is described below. First, a transmission vector T_s is formed whenever the index i (indicating the current solution index) is equal to a random integer s:

$$T_s = X_k + rand(0, 1) \times (X_{best} - X_k)$$
$$k \neq s \in [1, SN] \quad (19)$$

where X_{best} is the best individual which has the best fitness value in current iteration, X_k is one of the *SN* feasible solutions different than current solution X_i. The best candidate solution V_s is formed by combining the information of T_s and X_s; in other words, OL is applied to predict the best candidate solution by combining T_s and X_s with few tests as the analogy of the OED experiment in current iteration. It is worthy to mention that OL is applied whenever the index *i* is equal to a random integer *s* at each iteration; hence, OL has only been applied once at each iteration to save computational cost. The overall structure of the IABC algorithm is given in Fig. 2. The procedure of forming the candidate solution V_s by OL is described after Fig. 2

An individual employed bee is randomly chosen to use OL strategy to create a candidate solution, while other employed bees employ (15) to generate a candidate solution. The idea of adopting OL is that we want to formulate a solution vector by combining the good information of every dimension of two solution vectors. Instead of conducting exhaustive tests, OL is implemented to predict the best combination of dimensions based on two candidate solutions. Construction of a candidate solution V_s is summarized as following steps:

1) Generate a 2-level OA $L_N(2^k)$, with $N=2^{\lceil \log_2(D+1) \rceil}$, where *N* denotes for the total combination numbers for an OA and *D* is the dimension of the problem. ('⌈ ⌉' is the ceiling bracket, meaning round the number to the integer closer to ∞). The reason why 2-level OA is developed is because there are only two candidate solutions (one is the current solution X_s, and the other one is the transmission vector T_s) used for OL. Thus by choosing either level, the values from vector X_s or T_s will be used to combine the best solution.
2) Fill the OA $L_N(2^k)$ by the information of T_s and X_s. The OA is a 2 level, denoted by '1' and '2' and *D* factors (control variables) OA, and in such OA the value of T_s is chosen when the entry of OA is '1', and that of X_s is chosen otherwise.
3) Obtain *N* test solutions Z_n ($1 \le n \le N$) with the corresponding value of T_s (19) and X_s according to a 2-level OA $L_N(2^k)$.
4) Evaluate every test solution Z_n ($1 \le n \le N$), $f(Z_n)$, and record the best solution Z_b according to fitness values.
5) For each factor (control variable) conduct FA to obtain the best level.
6) With the best levels determined in step (4), predict the best combination solution Z_p, and evaluate Z_p.
7) If Z_p has better fitness value than Z_b, it is adopted as the candidate solution vector V_s.

The AC power flow was implemented in the project and was calculated by Newton-Raphson method. Numerical solutions and statistical analysis of different cases are presented as in the following section.

4. Case studies

In order to verify the effectiveness of proposed algorithm, both IABC and ABC were implemented in the modified IEEE 30 and 118 test systems. The numerical results are assessed and comparison is made with other modern heuristic methods. The computer used for simulations work has 3.4 GHz Intel core i7 Processor and 8 GB RAM. The power flow was calculated by MATPOWER package (Zimmerman, Murillo-Sanchez, & Thomas, 2011). Four case studies are presented: The first one is the benchmark case. The second one is for testing the algorithm for a more complex model. The third one is the most typical case in practice. The fourth one is also for verifying the algorithm whether it can work on a large-scale system.

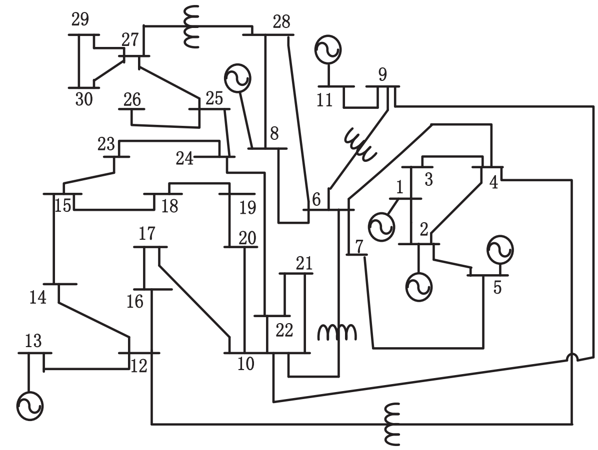

Fig. 3. IEEE-30 bus system.

4.1. Case1: minimizing fuel cost for IEEE 30-bus system

Case 1 is the standard OPF problem with quadratic cost function. Reference (Shoults & Sun, 1982) gives the data of IEEE 30-bus test system, and control variables limits can be found from (Lee, Park, & Ortiz, 1985). There are total 24 control variables which consist of five real power output control at PV bus and voltage magnitudes control of all six generator buses, nine shunt compensators control for injecting reactive power and four transformer tap controls. The six generators are shown in Fig. 2. Shunt compensators are installed on buses 10, 12, 15, 17, 20, 21, 23, 24 and 29. In addition, lines 4–12, 6–9, 6–10, and 28-27 are equipped with tap-changing transformers as shown in Fig. 3. The system is at 100 MVA base with active power demand of 2.834 p.u. and reactive power demand of 1.262 p.u. Fuel cost coefficients were taken from (Lee et al., 1985).

The objective in this case is to minimize the total generator fuel cost (4). The simulation was run 30 times in order to conduct statistical analysis. The minimum total cost from IABC is 799.321 $/h, with the maximum 799.322 $/h, the average 799.321 $/h, and zero standard deviation. Results are compared with the results from other methods such as basic ABC, gravitational search algorithm (GSA), linearly decreasing inertia weight particle swarm optimization (LDI-PSO), enhanced genetic algorithm (EGA), modified differential evolution (MDE), and modified shuffle-frog leaping algorithm (MSFA) (Bakirtzis et al., 2002; Duman, Güvenç, Sönmez, & Yörükeren, 2012; Khorsandi, Hosseinian, & Ghazanfari, 2013; Pan et al., 2011; Sayah & Zehar, 2008). The comparison including execution time is given in Table 3. Fig. 4 shows the convergence properties of ABC, and IABC algorithms.

Table 3 shows that the IABC approach found the minimum solution of 799.321 $/h, less than all other methods in the literature, and faster

Table 3
Comparison for fuel cost minimization in IEEE 30-bus system.

METHOD	Fuel cost ($/h)				
	Min	Avg.	Max	Std. Dev. (σ)	t (s)
IABC	799.321	**799.321**	799.322	**0.000**	56.8
ABC	800.834	800.944	801.518	0.162	39.8
GSA (Duman et al., 2012)	805.175	812.194	827.459	N/A	10.8
LDI-PSO (Khorsandi et al., 2013)	800.734	801.557	803.869	N/A	N/A
EGA (Bakirtzis et al., 2002)	802.060	N/A	802.140	N/A	N/A
MDE (Sayah & Zehar, 2008)	802.376	802.382	802.404	N/A	23.3
MSFLA (Niknam, Narimani, Jabbari, & Malekpour, 2011)	802.287	802.414	802.509	N/A	N/A

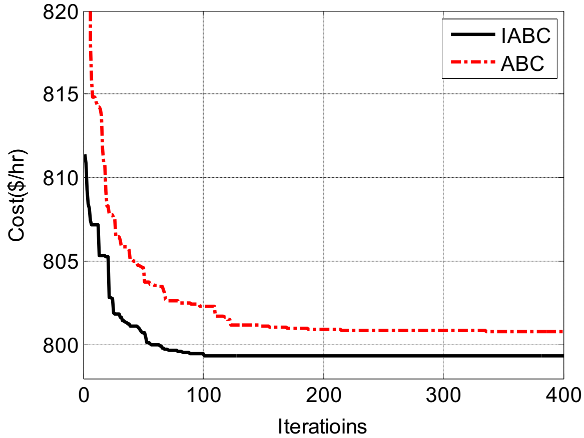

Fig. 4. Convergence performance in case 1 for IEEE-30 bus system.

Table 4
Comparison for valve-pont loading effect in IEEE 30-bus system.

Method	Fuel cost ($/h)				
	Min	Avg.	Max	Std. Dev. (σ)	t (s)
IABC	918.167	**919.567**	921.458	**0.662**	96.2
ABC	945.450	960.565	973.599	8.547	74.6
GSA (Zhan and Leung, 1999)	929.724	930.925	932.049	N/A	N/A
MDE (Karaboga et al., 2007)	930.793	942.501	954.073	N/A	N/A

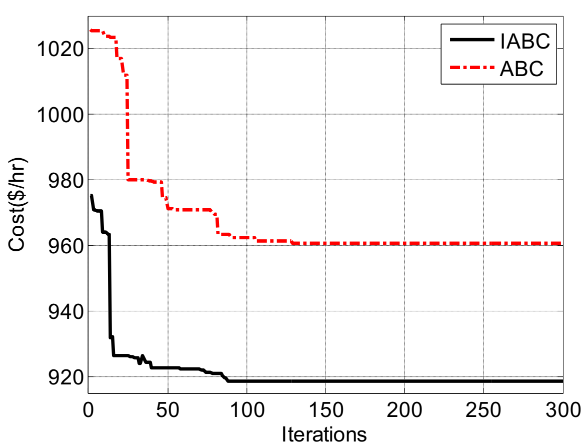

Fig. 5. Convergence performance in Case 2 for IEEE-30 bus system.

Table 5
Comparison for Total power loss in IEEE 30-bus.

Method	Total real power loss (MW)				
	Min	Avg.	Max	Stand. Dev. (σ)	t (s)
IABC	3.084	**3.086**	3.100	**0.003**	104.2
ABC	3.206	3.212	3.227	0.006	70.8
EGA (Bakirtzis et al., 2002)	N/A	3.201	N/A	N/A	N/A

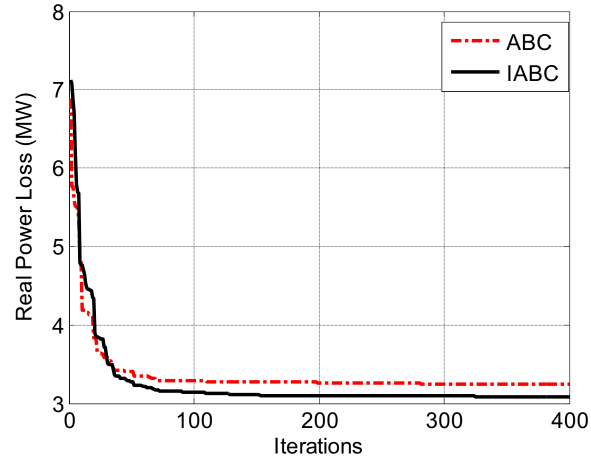

Fig. 6. Convergence performance in Case 3 for IEEE-30 bus system.

convergence of IABC is demonstrated in Fig. 3. The standard deviation can be omitted compared to other methods. It is worth mentioning that for the standard quadratic fuel cost function, the improvement seems not significant because the cost function is not complex enough to show the improvement.

4.2. Case 2: fuel cost with valve-point effect

In this case, bus 1 and bus 2 have units with the fuel cost function with valve-point effect (5). Table 4 gives the comparison with the results obtained from other methods. Note that since this case is not a typical bench mark problem, less studies are found for comparison. The same issue applies to the remaining two cases. Fig. 5 gives the convergence characteristics from the basic ABC and IABC method.

Simulation was run 30 times again for meaningful statistical results. The maximum total fuel cost from IABC, the average and the minimum cost are 921.458 $/h, 919.567 $/h, and 918.167 $/h respectively and the standard deviation is 0.662 $/h. As shown in Table 5, the IABC approach found the minimum solution of 918.167 $/h, less than all other methods in the literature, and better convergence property is shown in Fig. 5. IABC proves to be more robust because of the small standard deviation (Fig. 6).

4.3. Case 3: loss minimization for IEEE 30-bus system

The goal is to minimize the total real power loss as defined in (6). The control and state variables are identical with the previous two cases and the fuel cost function for this case is in the regular quadratic form. The results are compared with original ABC and EGA from reference (Bakirtzis et al., 2002).

The minimum total real power loss from IABC is 3.084 MW, the average is 3.086 MW, the maximum is 3.100 MW and with the standard deviation of 0.003. In this case the standard deviations for both cases are small.

Table 6
Case 4 comparison.

Method	Fuel cost ($/h)				
	Min	Avg.	Max	Std. Dev. (σ)	t (s)
IABC	129,862	**129,895**	129,941	**40.8**	4157.8
ABC	130,210	130,321	130,410	90.5	4037.5

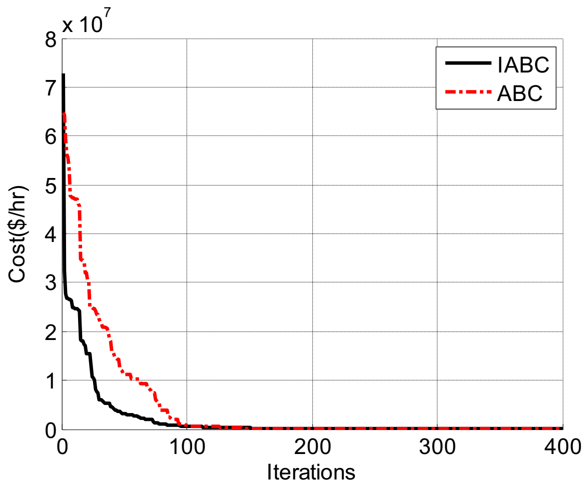

Fig. 7. Convergence performance in Case 4 for IEEE-118 bus system.

4.4. Case 4: minimizing fuel cost for IEEE 118-bus system

Assessing the developed algorithm on large-scale power system is necessary to demonstrate its effectiveness and robustness. Thus, a large-scale power system, IEEE-118-bus test system is adopted. For this case, there are total 130 control variables including 9 transformer tap controls, 14 shunt compensator controls, 53 real power output controls and voltage magnitude control of all 54 generators buses. Note that one of the generator bus is slack bus and thus is not considered as control variable. Data for IEEE 118-bus system can be found in (Power System Test Case Archive). The cost function is typical quadratic cost by (4). The minimal cost found is 129,862 $/h. The IABC was compared with regular ABC in Table 6. Fig. 7 gives the convergence property.

From Table 6, the minimum, average and maximum cost found by IABC are 129,862, 129,895 and 129.941 respectively. IABC simulation has the standard deviation of 40.8 which is much less than the original ABC. The IABC takes longer execution time than other methods, because OL is implemented at each iteration to conduct deep search. Parameters that used in all three cases have been specified in Table 7.

4.5. Statistical analysis

In order to draw convincing conclusions, statistical analysis over all cases were conducted. Figs. 8–11 present the box plot for IABC and ABC algorithms and Table 8 gives the one-tail paired *t*-test results.

The box plot for Case 1 showed the 1st quartile, median, 3rd quartile, minimum and maximum values out of 30 simulation runs and it is obvious that the results from IABC is very consistent (1st quartile, median, 3rd quartile, minimum and maximum values are almost the same number) and better fuel cost is found.

The same conclusion is drawn from Case 2 as it is from Case1: optimization performance has less deviation and better fuel cost by IABC.

The box plots for Case 3 and Case 4 demonstrate the robustness and effectiveness of IABC algorithm because of its performance of smaller deviation and cost values.

From the *t*-test, it is seen that the IABC outperforms ABC in all four cases at 0.05 confidence level in terms of the total generation cost and

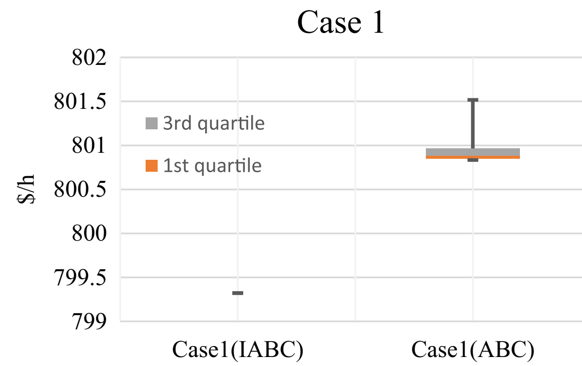

Fig. 8. Box plot for case 1.

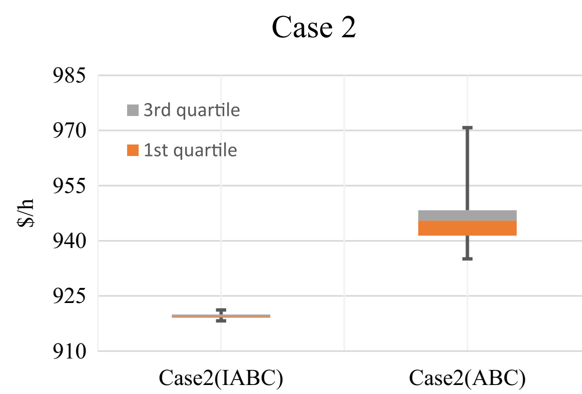

Fig. 9. Box plot for case 2.

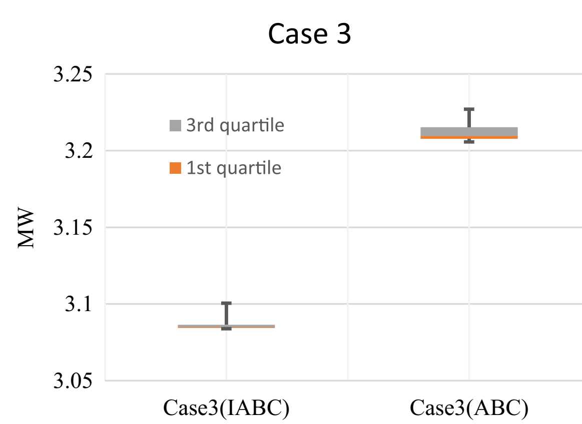

Fig. 10. Box plot for case 3.

Table 7
Algorithms parameters.

	ABC				IABC			
	Colony Size	Food Number	Limited Trials	Max Cycles	Colony Size	Food Number	Limited Trials	Max Cycles
Case 1	100	50	100	400	100	50	100	400
Case 2	100	50	100	300	100	50	100	300
Case 3	100	50	100	400	100	50	100	400
Case 4	300	150	100	400	300	150	400	400

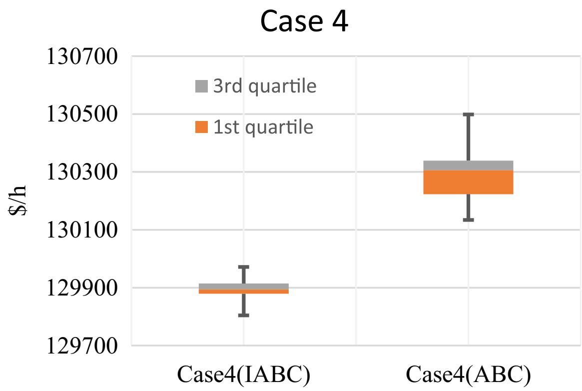

Fig. 11. Box plot for case 4.

Table 8
Paired statistical T test for IABC and ABC.

Cases	ABC		IABC		P-value
	Best	Avg.	Best	Avg.	
1	800.834	800.944	799.321	799.321	5.20e−33
2	945.450	960.565	918.167	919.567	3.81e−18
3	3.206	3.212	3.084	3.086	5.79e−41
4	130,210	130,321	129,862	129,895	6.26e−23

power losses.

The null hypothesis (H_0) is defined as that there is no differences between two algorithms and the alternative hypothesis (H_1) is that the performance of IABC is better than the original ABC. Since all the p-values are smaller than 0.05, we can draw conclusion that there is significant difference between two algorithms; in other words, H_0 is rejected and H_1 is accepted.

It is worth to point out that it takes longer time for running IABC, because on employed bee deep search on solution space was performed in order to find promising solution. Meanwhile the computing time is not well reported for other techniques in the literature, thus making the full comparison on computing time impossible. In order to reduce the uncertainty of simulations, the focus can be put on choosing algorithm parameters (the number of bees, the maximum trails number, etc.), depending on different problems, if proper algorithm parameters are determined the robustness will increase.

5. Conclusion

The paper proposed an improved ABC algorithm based on OL to tackle the non-smooth, non-linear and non-convex OPF problem. Orthogonal learning is implemented to predict the best combination of two solution vectors based on limited trials instead of exhaustive trials to conduct deep search in the solution space. Such a process improved the exploitation of ABC and led to promising results. In order to verify the effectiveness of proposed algorithm, IABC and basic ABC were tested and the results were compared with other modern heuristic methods. Better feasible solutions were found. For example, the cost considering valve effect was reduced by 4.2%. Different case studies and statistical analysis have demonstrated that the IABC is effective, accurate and robust with better optimization performance. In addition, IABC can be applied to large-scale power systems. It is obvious that in order to gain better solution, longer computation time is required; however the computational burden can be further reduced with the help of parallel computing. Future work can focus on extending the applications of IABC in power system such as optimizing the placement of distributed generators.

Acknowledgment

İ.Eke is supported by The Scientific and Technical Research Council of Turkey (TUBITAK) Turkey, through the postdoctoral research program 2219, under application number 1059B191300593. and Scientific Research Projects Coordination Unit (BAP - project number 2012/112) of Kirikkale University (TUBiTAK).

References

Abido, M. A. (2002). Optimal power flow using Tabu search algorithm. *Electric Power Component and Systems*, *30*, 469–483.
Abou, E. E., A, A., Abido, M. A., & Spea, S. R. (2010). Optimal power flow using differential evolution algorithm. *Electric Power System Research*, *80*, 878–885.
Adaryani, M. R., & Karami, A. (2013). Artificial bee colony algorithm for solving multi-objective optimal power flow problem. *Electrical Power & Energy Systems*, *53*, 219–230.
Al-Betar, M. A., Khader, A. T., & Zaman, M. (2012). University course timetabling using a hybrid harmony search metaheuristic algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Applications and Reviews*, *42*(*5*), 664–681.
Alsac, O., & Stott, B. (1974). Optimal load flow with steady-state security. *IEEE Transactions on Power Apparatus and Systems*, *93*, 745–751.
Bai, W., Abedi, R., & Lee, K. Y. (2016). Distributed generation system control strategies with PV and fuel cell in microgrid operation. *Control Engineering Practice*, 53, 184–193.
Bakirtzis, A. G., Biskas, P. N., Zoumas, C. E., & Petridis, V. (2002). Optimal power flow by enhanced genetic algorithm. *IEEE Transactions on Power and Systems*, *17*, 229–236.
Burchett, R. C., Happ, H. H., & Vierath, D. R. (1984). Quadratically convergent optimal power flow. *IEEE Transactions on Power Apparatus and Systems*, vol PAS*103*, 3267–3275.
Cain, M.B., O'Neill R.P., and Castillo A. (2012). History of optimal power flow and formulations. *Federal Energy Regulatory Commission Report* available:⟨http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf⟩.
Crepinsek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, *45*(3), 35–68.
Decker, G. L., & Brooks, A. D. (1958). Valve point loading turbines. *Electrical Engineering*, *77*(6), 501.
Duman, S., Güvenç, U., Sönmez, Y., & Yörükeren, N. (2012). Optimal power flow using gravitational search algorithm. *Energy Conversion and Management*, *59*, 86–95.
Fong, C. W., Asmuni, H., & McCollum, B. (2015). A hybrid swarm-based approach to university timetabling. *IEEE Transactions on Evolutionary Computation*, *19*(6), 870–884.
Gan, D., Thomas, R. J., & Zimerman, R. D. (2000). Stability-constrained optimal power flow. *IEEE Transactions on Power Apparatus and Systems*, *15*(2), 535–540.
Gao, W. F., Liu, S. Y., & Huang, L. L. (2013). A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Transactions on Cybernetics*, *43*(3), 1011–1024.
Gao, W., & Liu, S. (2012). A modified artificial bee colony algorithm. *Computers & Operations Research*, *39*, 687–697.
Gao, W., & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, *111*(17), 871–882.
Hedayat, A. S., Sloane, N. J. A., & Stufken, J. (1999). *Orthogonal arrays: Theory and applications* (ISNB: 987-0387987668) New York: Springer.
Hua, W., Sasaki, H., Kubokawa, J., & Yokoyama, R. (1998). An interior point nonlinear programming for optimal power flow problems with a novel data structure. *IEEE Transactions on Power Systems*, *13*, 870–877.
Karaboga, D (2005). *An idea based on honey bee swarm for numerical optimization.* Erciyes University, Kayseri, Turkey, Tech. Rep.- TR06.
Karaboga, D., & Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, *214*, 108–132.
Karaboga, D., Alay, B., & Ozturk, C. (2007). Artificial Bee Colony (ABC) optimization algorithm for training feed-forward neural networks. *Modeling Decisions for Artificial Intelligence*, 318–329.
Karaboga, D., & Akay, B. (2012). Artificial bee colony algorithm for large scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, *23*, 1001–1104.
Khorsandi, A., Hosseinian, S. H., & Ghazanfari, A. (2013). Modified artificial bee colony algorithm based on fuzzy multi-objective technique for optimal power flow problem. *Electric Power Systems Research*, *95*, 206–213.
Lee, K. Y., & El-Sharkawi, M. A. (Eds.). (2008). Modern heuristic optimization techniques with applications to power systems (Power System Test Case Archive. Available). New York: Wiley.
Lee, K. Y., Park, Y. M., & Ortiz, J. (1985). A united approach to optimal real and reactive power dispatch. *IEEE Transactions on Power Apparatus and Systems*, *104*, 1147–1153.
Momoh, J. A., Koessler, R. J., Bond, M. S., & Stott, B. (1997). Challenges to optimal power flow. *IEEE Transactions on Power Systems*, *12*(1), 444–447.
Niknam, T., Narimani, M. R., Jabbari, M., & Malekpour, A. R. (2011). A modified shuffle frog leaping algorithm for multi-objective optimal power flow. *Energy*, *36*, 6420–6432.
Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial

bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, *181*, 2455–2468.
Park, J. B., Jeong, Y. W., Shin, J. R., & Lee, K. Y. (2010). An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Transactions on Power Systems*, *25*(1), 156–166 (Power System Test Case Archive. Available) ⟨https://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm⟩.
Roy, R. K. (2011). *Design of experiments using the taguchi approach* (ISBN: 9780471361015). Wiley.
Sayah, S., & Zehar, K. H. (2008). Modified differential evolution algorithm for optimal power flow with non-smooth cost function. *Energy Conversion and Management*, *49*, 3036–3042.
Shoults, R. R., & Sun, D. T. (1982). Optimal power flow based upon P-Q decomposition. *IEEE Transactions on Power Apparatus and Systems*, *vol. PAS-101*, 397–405.
Sivasubramani, S., & Swarup, K. S. (2011). Multi-objective harmony search algorithm for optimal power flow problem. *International Journal of Electrical Power & Energy Systems*, *33*, 745–752.
Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, *9*(2), 625–631.
Soares, J., Vale, Z. A., Morais, H., & Faria, P. (2011). Simulated annealing metaheuristic to solve the optimal power flow. *IEEE Pesera General meeting*, 1–8.
Zhang, Q. F., & Leung, Y. W. (1999). An orthogonal genetic algorithm for multimedia multicast routing. *IEEE Transactions on Evolutionary Computation*, *3*(1), 53–62.
Zhan, Z. H., Zhang, J., Li, Y., & Shi, Y. H. (2011). Orthogonal learning particle swarm optimization. *IEEE Trans Evol Comput*, *15*(6), 832–847.
Zimmerman, R. D., Murillo-Sanchez, C. E., & Thomas, R. J. (2011). MATPOWER steady-state operations, planning and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, *26*(1), 12–19.
Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics*, *217*(7), 3166–3173.