Electrical Power and Energy Systems 83 (2016) 283–297

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

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

Transient stability constrained optimal power flow using oppositional krill herd algorithm

Aparajita Mukherjee [a,*], Provas Kumar Roy [b], V. Mukherjee [a]

[a] Department of Electrical Engineering, Indian School of Mines, Dhanbad, Jharkhand, India
[b] Department of Electrical Engineering, Jalpaiguri Govt. Engineering College, Jalpaiguri, West Bengal, India

ARTICLE INFO

Article history:
Received 28 April 2014
Received in revised form 16 March 2016
Accepted 29 March 2016
Available online 22 April 2016

Keywords:
Evolutionary algorithm
Oppositional krill herd algorithm
Optimal power flow
Transient stability

ABSTRACT

Transient stability constrained optimal power flow (TSCOPF) is becoming an effective tool for many problems in power systems since it simultaneously considers economy and dynamic stability of system operations. It is increasingly important because modern power systems tend to operate closer to the stability boundaries due to the rapid increase of electricity demand and the deregulation in power sector. TSCOPF is, however, a nonlinear optimization problem with both algebraic and differential equations which is difficult to solve even for small power network. In order to solve the TSCOPF problem efficiently, a relatively new optimization technique, named as krill herd algorithm (KHA), is employed in this paper. KHA simulates the herding behavior of krill swarms in response to specific biological and environmental processes to solve multi-dimensional, linear and nonlinear problems with appreciable efficiency. To accelerate the convergence speed and to improve the simulation results, opposition based learning (OBL) is also incorporated in the basic KHA method. The simulation results, obtained by the basic KHA method and the proposed oppositional KHA (OKHA) algorithm, are compared to those obtained by using some other recently developed methods available in the literature. In this paper, case studies conducted on 10 generator New England 39-bus system and 17 generator 162-bus system indicate that the proposed OKHA approach is much more, computationally, efficient than the other reported popular state-of-the-art algorithms including the basic KHA and the proposed method is found to be a promising tool to solve the TSCOPF problem of power systems.

Introduction

The systematic interconnection of power systems that took place in the second half of the twentieth century was an attempt to strengthen the networks and to facilitate the transmission of electricity. This effort has brought new operational challenges that could not be faced by power engineers unless the state of the network was properly monitored in real-time [1]. The modern deregulated environment has driven utilities around the world to operate their power systems closer to their stability boundary for better use of transmission networks. The optimal power flow (OPF) has been an important tool in power system operation for the past four decades. The concept of OPF was first introduced by Carpentier [2]. It is a very powerful tool to find out an accurate balance between economics and security. The main goal of the OPF problem is to determine the optimal operating state of the power system by optimizing a particular objective function while satisfying certain specified physical and operating constraints [3]. For secured and economical operation of power system, it is required to be stable under some severe disturbances, *i.e.* system operation must satisfy transient stability constraints.

OPF is a nonlinear, non-convex, large-scale, static optimization problem with both continuous and discrete control variables [4]. The careful and intelligent scheduling of the generating units can not only reduce the operating cost significantly but also assure higher reliability and security of power system. Many successful techniques [5–9] have been developed in the intervening decades that focus on overcoming the limitations of OPF study in terms of flexibility and reliability for practical applications. In case of conventional OPF formulation, transient stability constraints are not considered. The operating point of the system often fails to maintain transient stability, when subjected to a credible contingency. The ordinary OPF problems have been, extensively, studied in [10]. Several conventional optimization techniques have been applied to solve the OPF problem, such as linear programming, interior point method (IPM) and Newton method [11]. But as the

* Corresponding author. Tel.: +91 0326 2235644; fax: +91 0326 2296563.
E-mail addresses: mukherjeeaparajita29@gmail.com (A. Mukherjee), roy_provas@yahoo.com (P.K. Roy), vivek_agamani@yahoo.com (V. Mukherjee).

http://dx.doi.org/10.1016/j.ijepes.2016.03.058
0142-0615/

power industry is growing faster and moves into a competitive environment, transient stability is becoming an important factor to be considered while dealing with OPF study [12].

Transient stability constrained OPF (TSCOPF) is, however, a nonlinear optimization problem with both algebraic and differential equations in the time domain. It considers optimal and stable operations simultaneously. As a special requirement of the system, the initial or feasible operating point should withstand the disturbance and can move to a new stable equilibrium state after the clearance of the fault without disturbing the equality and the inequality constraints. Due to huge dimension of TSCOPF problem (especially, for system dealing with detailed machine models), it is really a tough exercise to deal with this type of problem. For a given power system configuration, although the number of possible contingencies are numerous, there are few critical contingencies that may cause instability. After analyzing and filtration, the major contingency is selected and the TSCOPF procedure is applied to find out the optimal operating point.

Various optimization techniques have been evolved in the last two decades to solve the TSCOPF problem. An improved genetic algorithm (GA) was proposed by Chan et al. [13] to solve multi-contingency TSCOPF problem where generator rotor angle constraints were additionally considered. An IPM method was introduced by Xia et al. [14] to efficiently perform the TSCOPF. But, in case of large scale TSCOPF problem, IPM suffers from the curse of dimensionality, unacceptable memory consumption and enough computational time [14]. To overcome these drawbacks, a new approach (called as reduced space IPM (RSIPM)) was introduced in [15] to reduce the memory usage as well as CPU time. In order to improve further the computational efficiency, an enhanced numerical discretization method was proposed in [16] where truncation error of specific numerical integration algorithm was considered and discretization occurs in inequality constraints instead of equality constraints that has enabled to reduce the prime dual linear systems dimension to 50% from more than 80–90%. Afterward, Pizano-Martinez et al. have proposed an efficient and practical approach in [17] to reduce the huge dimension of the TSCOPF problem where the sets of dynamic and transient stability constraints were reduced to one single stability constraint and this method was implemented, efficiently, on WSCC 3-machine 9-bus system and the Mexican 46-machine 190-bus system. For achieving more effective and flexible computation of TSCOPF, combination of classical deterministic programming technique and evolutionary algorithm was introduced by Xu et al. [18]. Geng and Jiang have proposed two-level parallel decomposition approach [19] based on the RSIPM method to solve TSCOPF problem. Moreover, in the recent past, various other nature-inspired optimization algorithms have been also designed and applied to solve the TSCOPF problem of power system. These include evolutionary programming [20], particle swarm optimization (PSO) [21,22], differential evolution (DE) [23], GA [24], etc.

However, this paper represents a new bio-inspired swarm intelligence algorithm, called krill herd algorithm (KHA) (proposed by Gandomi and Alavi [25]), to solve TSCOPF problem of power system. The main motive of using this technique is to show that the clustering may be also applied to the natural intelligent techniques. Clustering also shows the swarm behavior and, thus, it is used with the nature inspired algorithms to help for finding the food for the krill individuals. For determining the time dependent position of an individual krill, three important actions are considered in [25]. These are (a) motion induced by other krills, (b) foraging activity and (c) random diffusion.

Recently, KHA methodology is becoming very much popular in different fields of research interests like, structural optimization problem [26], portfolio optimization problem [27], numerical optimization problem [28], phase stability and phase equilibrium calculations [29], production scheduling problem [30], optimum design of truss structures [31] and so on. Moreover, a few new variant of KHA have been also proposed by the researchers such as opposition based KHA with cauchy mutation and position clamping [32], biogeography based optimization (BBO) embedded with KHA [33], and quantum-behaved PSO based KHA [34]. Also, variants of KHA like simulated annealing (SA) based KHA [35], chaotic PSO based KHA [36], harmony search algorithm (HSA) based KHA [37], GA based KHA incorporating stud selection and crossover operator [38], DE based KHA [39] and cuckoo search based KHA [40] have been proposed in the literature for global numerical optimization problem.

Moreover, to accelerate the convergence rate and to improve the simulation results, oppositional based learning (OBL) [41] is integrated in this paper with the conventional KHA technique. The main idea behind the OBL is the simultaneous consideration of an estimate and its corresponding opposite estimate in order to achieve a better approximation for the current candidate solution. So, OBL is proved to be an effective technique to improve the performance of various basic optimization approaches like PSO [42], ant colony optimization (ACO) [43], BBO [44], HSA [45–47], gravitational search algorithm (GSA) [48] and bat algorithm [49].

In this article, the concept of OBL is embedded with the basic KHA technique (termed as OKHA) for achieving improved response and better convergence characteristics. Two test power systems (*i.e.* 10 generator New England 39-bus system and 17 generator 162-bus system) are chosen in this work to solve the TSCOPF problem of power system. A single objective function *i.e.* quadratic fuel cost (without considering valve point effect) is considered here. The results obtained from the proposed OKHA method are compared with other three algorithms (*viz.* GSA, BBO and KHA) reported in the recent state-of-the-art literature.

The rest of this paper is organized as follows: 'Problem formulation' Section analyzes the mathematical problem formulation part of OPF problem. Basic aspects of transient stability analysis are assessed in Section 'Transient stability assessment'. Section, 'Algorithms employed' describes about the basic attributes of some algorithms like GSA, BBO, KHA and OKHA. 'OKHA applied to TSCOPF problem' Section elaborates the implementation part of the proposed OKHA for TSCOPF problem. Simulation results and discussion part are elucidated in 'Simulation results and discussion' part while 'Conclusion' Section draws the conclusion of the present work.

Problem formulation

The main goal of the OPF problem is to minimize the total fuel cost while satisfying all the equality and the inequality constraints. Mathematically, OPF problem may be represented as

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

$$\text{subject to}: \left.\begin{matrix} g(u) = 0 \\ h(u) \leqslant 0 \end{matrix}\right\} \tag{2}$$

The objective function may be written as [23]

$$f(u) = C_T = \sum_{x=1}^{N_G} (a_x P_{G_x}^2 + b_x P_{G_x} + c_x) \tag{3}$$

where C_T is the total generating cost, a_x, b_x, c_x are the fuel cost coefficients of the generator x, P_{G_x} is the active power generation of the unit x, N_G is the total number of generator buses.

Constraints

The OPF problem has two categories of constraints (*viz.* the equality constraint and the inequality constraint). These two types of constraints are, sequentially, described below.

Equality constraints

The following power flow equations form the equality constraints:

$$\begin{cases} P_{G_x} - P_{D_x} - \sum_{y=1}^{N_B} V_x V_y (G_{xy} \cos \delta_{xy} + B_{xy} \sin \delta_{xy}) = 0 \\ Q_{G_x} - Q_{D_x} - \sum_{y=1}^{N_B} V_x V_y (G_{xy} \sin \delta_{xy} - B_{xy} \cos \delta_{xy}) = 0 \end{cases} \quad (4)$$

where V_x, V_y are the voltages of the buses x and y, respectively, P_{G_x}, Q_{G_x} are the active and the reactive power, respectively, of the generator x, P_{D_x}, Q_{D_x} are the active and the reactive power demand at bus x, $G_{xy}, B_{xy}, \delta_{xy}$, respectively, are the conductance, the susceptance and the phase difference of voltage, between buses x and y and N_B is the number of buses.

Inequality constraints

These constraints represent the system operating limits as follow:

(i) *Generator constraints*

Generator voltage, active power outputs and reactive power outputs of bus x should lie between their respective lower and upper limits, as follows:

$$\left.\begin{aligned} V_{G_x}^{min} \leqslant V_{G_x} \leqslant V_{G_x}^{max}, \quad x = 1, 2, \ldots, N_G \\ P_{G_x}^{min} \leqslant P_{G_x} \leqslant P_{G_x}^{max}, \quad x = 1, 2, \ldots, N_G \\ Q_{G_x}^{min} \leqslant Q_{G_x} \leqslant Q_{G_x}^{max}, \quad x = 1, 2, \ldots, N_G \end{aligned}\right\} \quad (5)$$

where $V_{G_x}^{min}, V_{G_x}^{max}$ are the minimum and the maximum generator voltage, respectively, of bus x, $P_{G_x}^{min}, P_{G_x}^{max}$ are the minimum and the maximum active power output, respectively, of bus x and $Q_{G_x}^{min}, Q_{G_x}^{max}$ are the minimum and the maximum reactive power output, respectively, of bus x.

(ii) *Transformer constraints*

Transformer tap settings are bounded by their respective upper and lower limits as:

$$T_x^{min} \leqslant T_x \leqslant T_x^{max}, \quad x = 1, 2, \ldots, N_T \quad (6)$$

where T_x^{min}, T_x^{max} are the minimum and the maximum tap setting limits, respectively, of transformer x and N_T is the number of regulating transformers.

(iii) *Shunt compensator constraints*

Reactive power injections at buses are restricted by their respective maximum and minimum limits as:

$$Q_{C_x}^{min} \leqslant Q_{C_x} \leqslant Q_{C_x}^{max}, \quad x = 1, 2, \ldots, N_C \quad (7)$$

where $Q_{C_x}^{min}, Q_{C_x}^{max}$ are the minimum and the maximum VAR injection limits, respectively, of the shunt compensator x and N_C is the number of shunt compensators.

(iv) *Security constraints*

These include transmission line loadings and voltages at load buses as follows:

$$V_{L_x}^{min} \leqslant V_{L_x} \leqslant V_{L_x}^{max}, \quad x = 1, 2, \ldots, N_L \quad (8)$$

$$S_{L_x} \leqslant S_{L_x}^{max}, \quad x = 1, 2, \ldots, N_{TL} \quad (9)$$

where $V_{L_x}^{min}, V_{L_x}^{max}$ are the minimum and the maximum load voltage, respectively, of load bus x, $S_{L_x}, S_{L_x}^{max}$ are the apparent power flow and the maximum apparent power flow limit, respectively, through branch x, N_L is the number of load buses and N_{TL} is the number of transmission lines.

Transient stability assessment

The term '*transient stability*' refers to the ability of a synchronous machine of an interconnected power system to return to normal or stable operation after being subjected to some form of disturbances and the instability, usually, appears in the form of increasing angular swings of some generators leading to their loss of synchronism with respect to the other generators [50].

Mathematically, transient stability problem is described by a set of differential–algebraic equations. For describing the transient behavior of a system incorporating transient stability, the generator rotor angles are used here with respect to the initial center of all the generators. So, the position of center of inertia (COI) is defined as

$$\delta_{COI} = \frac{\sum_{x=1}^{N_G} M_x \delta_x}{\sum_{x=1}^{N_G} M_x} \quad (10)$$

where δ_x is the rotor angle for the generator x and M_x is the inertia constant of the generator x for an N_G generator system.

In view of inequality constraints, it may be put as

$$\delta_{min} \leqslant \delta_x - \delta_{COI} \leqslant \delta_{max}, \quad x \subseteq S_G \quad (11)$$

where $\delta_{\min}, \delta_{\max}$ are, respectively, the lower and the upper limits of generator rotor angles and S_G is the set of synchronous generators.

Algorithms employed

In this article, a new algorithm, named as KHA [25], is tested to solve the TSCOPF problem of power system. Moreover, to improve the convergence speed and the quality of the simulation results, the idea of OBL is integrated with the basic KHA technique. To check the compatibility and the effectiveness of the proposed OKHA, it is compared with the basic KHA and other two recently developed evolutionary algorithms, namely, GSA [51,52] and BBO [53,54]. The details of these algorithms are explained below.

GSA

GSA, proposed by Rashedi et al. [51], is based on the law of gravity and motion. In this algorithm [48,51,52], agents are considered as objects and their performance is measured by their masses. The gravitational force, by which the objects attract each other, causes a global movement of all the objects toward the objects with heavier masses. The objects with heavier masses through which gravitational force cooperate by using a direct form of communication, guarantee the exploitation step of the algorithm and move more slowly than the lighter ones. In GSA, active gravitational mass, passive gravitational mass, position and inertial mass are the four specified masses or agents. Using a fitness function, its gravitational and inertial masses are determined while the position of the mass corresponds to a solution [48,51,52].

GSA could be considered as an isolated system of masses. Now, this masses present optimum solution in the search space. More precisely, masses obey the following laws:

Law of gravity

It states that any two bodies in the universe attract each other with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them.

Law of motion

The current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

The position of the ith agent may be defined by considering a system of K agents (masses).

$$P_i = (p_i^1, \ldots, p_i^d, \ldots, p_i^n), \quad i = 1, 2, 3, \ldots, N_p \tag{12}$$

where p_i^d presents the position of the agent i in the dimension d and n is the total dimension of the search space.

Based on GSA, mass of each agent may be calculated after computing the fitness of current population by using (13) and (14) [48]

$$m_i(t) = \frac{fitness_i(t) - worst(t)}{best(t) - worst(t)} \tag{13}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{i=1}^{N_p} m_i(t)} \tag{14}$$

where $M_i(t)$ and $fitness_i(t)$ symbolize, respectively, the mass and the fitness value of the agent i at time t while $worst(t)$ and $best(t)$ are defined (for a minimization problem) [48] as follows:

$$best(t) = min_{i \in \{1,\ldots,N_p\}} fitness_i(t) \tag{15}$$

$$worst(t) = max_{i \in \{1,\ldots,N_p\}} fitness_i(t) \tag{16}$$

Now, based on the law of gravity, the total forces applied on an agent among a set of heavier masses may be calculated by using (17) which is followed by an another equation (refer (18)) that shows the calculation of acceleration based on the law of motion. After that, the velocity and position of agent may be updated, respectively, by (19) and (20) [48]. The updated velocity of an agent is considered to the sum of the fraction of its current velocity added to its acceleration.

$$X_i^n(t) = \sum_{\substack{j=k_{best}, \\ j \neq i}}^{K} rand_j \times G(t) \times \frac{M_i(t) \times M_j(t)}{H_{ij}(t) + \chi} \times (p_j^n(t) - p_i^n(t)) \tag{17}$$

$$acc_i^d(t) = \frac{X_i^n(t)}{M_i(t)} = \sum_{\substack{j=k_{best}, \\ j \neq i}}^{K} rand_j \times G(t) \times \frac{M_j(t)}{H_{ij}(t) + \chi} \times (p_j^n(t) - p_i^n(t)) \tag{18}$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + acc_i^d(t) \tag{19}$$

$$P_i^d(t+1) = P_i^d(t) + u_i^d(t+1) \tag{20}$$

where $rand_i$ and $rand_j$ are two random numbers defined in the interval [0, 1], χ is a small constant, k_{best} is the set of first K agents consist of best fitness value and bigger mass that may be considered as a function of time and $H_{ij}(t)$ is the Euclidian distance between the two agents i and j, defined by (21).

$$H_{ij}(t) = \|p_i(t), p_j(t)\|_2 \tag{21}$$

Here, the gravitational constant G may be considered as a function of the initial value (G_o) and time (t) (*i.e.* decreased with time to control the search accuracy) formulated by (22) [48]

$$G(t) = G_o \times e^{-\lambda\left(\frac{Iter}{Iter_{max}}\right)} \tag{22}$$

where G_o value is chosen as 100 [48], λ is preferred as 10 [48], $Iter$ and $Iter_{\max}$ are the present and the total number of iterations, respectively.

Computational procedure of GSA algorithm

The different steps of the GSA algorithm are shown in Algorithm 1 [48] while its flowchart may be found in Fig. 1 [51].

Algorithm 1: Computational procedure of GSA [48]

Step 1	Search space identification.
Step 2	Generate initial population between minimum and maximum values.
Step 3	Evaluate the fitness of each agent.
Step 4	Update the mass of the object $M_i(t)$ using (13) and (14), $best(t)$ based on (15), $worst(t)$ based on (16). Also, update the gravitational constant $G(t)$ using (22).
Step 5	Calculate the total force using (17) for different directions.
Step 6	Calculate the acceleration using (18) and the velocity of an agent using (19).
Step 7	Update the position of an agent by (20).
Step 8	Check for constraints of the problem.
Step 9	Repeat Step 3 to Step 8 until the stopping criteria is satisfied.

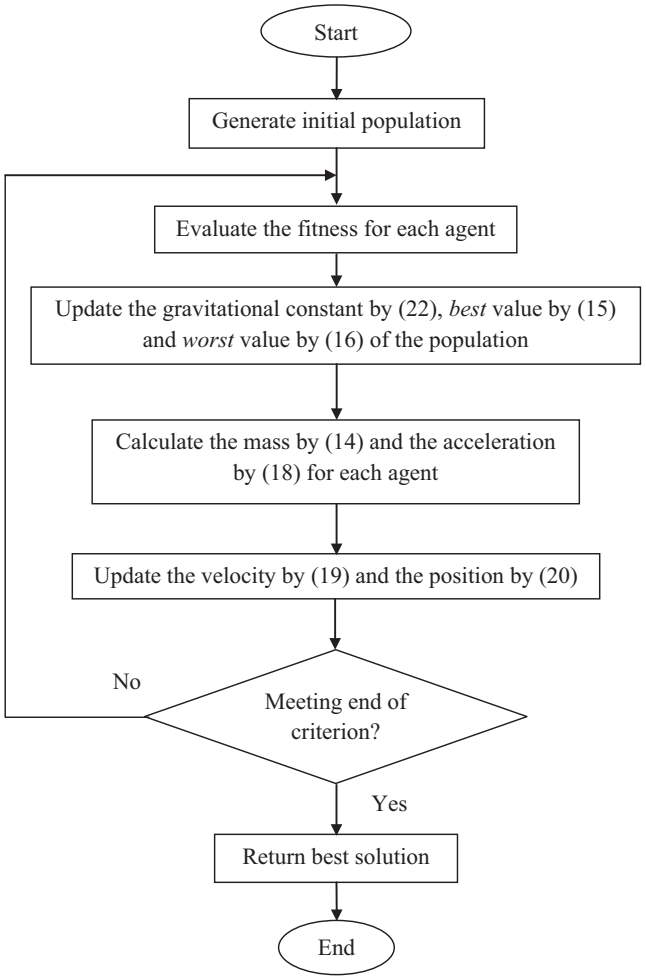

Fig. 1. Flowchart of GSA [51].

BBO

Mathematical model of biogeography [53] describes the evolution of new species (*i.e.* the migration of species (like animals, fish, birds, or insects) between islands and the extinction of these species). A habitat is defined as an island (area) that is, geographically, isolated from the other islands. Islands that are friendly to life are said to have a high habitat suitability index (HSI). The features that correlate with the HSI values include rainfall, vegetative diversity, topographic diversity, land area, temperature and others. The index variables that qualify the suitability are termed as suitability index variables (SIVs). Islands with a high HSI can support many species while islands with a low HSI can support only few numbers of species. Islands with a high HSI have many species that emigrate to nearby habitats because of the large populations and the large numbers of species that they host. So, the emigration process may be defined as the migration of some species from a habitat to an exterior habitat whereas an entry of some species into one habitat from an outside habitat is known as immigration process. Islands with a high HSI not only have a high emigration rate, but they also have a low immigration rate because they already support many species. Species that migrate to such islands will tend to die in spite of the high HSI of the preferred island, because there is too much competition for resources from other species. Islands with a low HSI have a high immigration rate because of their low populations.

Comparing the concept of BBO with any other optimization technique, it may be noted that a higher value of HSI corresponds to a good solution while a lower value of HSI corresponds to a poor solution. Poor solutions accept much more new features from good solutions that help to raise the quality of poor solutions with low HSI.

Mathematically, the whole process of immigration and emigration may be expressed by a probabilistic model. If $P_s(t)$ denotes the probability of a habitat at time t containing exactly s species, then at time $(t + \Delta t)$, the probability will be given by (23) [53,54]

$$P_s(t+\Delta t) = P_s(t)(1-\lambda_s\Delta t-\mu_s\Delta t) + P_{(s-1)}\lambda_{(s-1)}\Delta t + P_{(s+1)}\mu_{(s+1)}\Delta t \tag{23}$$

If Δt time is taken as very small, then the probability of more than one immigration or emigration may be ignored. Taking the limit as $\Delta t \to 0$, the species count probability may, mathematically, be expressed as follows [53,54]:

$$\dot{P}_s = \begin{cases} -(\lambda_s+\mu_s)P_s + \mu_{s+1}P_{s+1}, & S=0 \\ -(\lambda_s+\mu_s)P_s + \lambda_{s-1}P_{s-1} + \mu_{s+1}P_{s+1}, & 1 \leqslant S \leqslant S_{max}-1 \\ -(\lambda_s+\mu_s)P_s + \lambda_{s-1}P_{s-1}, & S=S_{max} \end{cases} \tag{24}$$

where P_{s-1} is the probability of habitat containing $(s-1)$ species, P_{s+1} is the probability of habitat containing $(s+1)$ species, λ_s and μ_s are the immigration and emigration rate, respectively, for habitat containing s species, λ_{s-1} and μ_{s-1} are the immigration and emigration rate, respectively, for habitat containing $(s-1)$ species, λ_{s+1} and μ_{s+1} are the immigration and emigration rate, respectively, for habitat containing $(s+1)$ species and $S_{\max}$ is the maximum count of species in the habitat.

Immigration rate (λ_s) and emigration rate (μ_s) may be represented by (25) and (26), in order, as follows [53,54]

$$\lambda_s = I\left(1-\frac{S}{S_{max}}\right) \tag{25}$$

$$\mu_s = \frac{E\times S}{S_{max}} \tag{26}$$

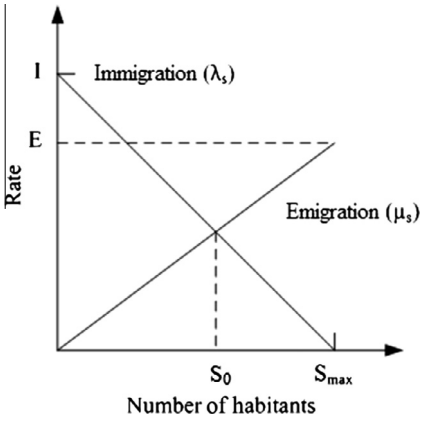

Fig. 2. Species model of a single habitat BBO [53].

where I is the maximum immigration rate and E is the maximum emigration rate, respectively.

Fig. 2 describes the model of habitants' profusion in a single habitat. The two functions, μ_s and λ_s, are the number of habitants in the habitat. From this figure, it may be seen that the presence of zero habitant in the habitat results to the maximum possible immigration rate of I (in case of immigration curve). Due to the increase of number of species, the habitat becomes more crowded and a few number of species are able to successfully endure and, as a result, the immigration rate decreases. The largest possible number of habitants is $S_{\max}$ where the immigration rate becomes zero. On the other side, in case of emigration curve, the emigration rate must be zero for the presence of zero habitants. When the habitat is becoming more crowded, the emigration rate increases. In Fig. 2, S_0 is the equilibrium point, where immigration and emigration rate become equal [53,54].

BBO technique is, mainly, based on migration and mutation. The basic concept of migration and mutation are given below.

Migration

In this algorithm, population of candidate solutions is represented as vector of real numbers. For sharing the information between habitats, emigration and immigration rates of individuals are used. Here, each solution may be modified based on other solution using habitat modification probability. If any individual solution is selected to be modified, then immigration rate (λ_s) is used to probabilistically decide whether or not to modify any SIV in that solution. After selection of any SIV for modification, emigration rates (μ_s) of other solutions are applied, to probabilistically select among the population set for migration.

Mutation

In BBO, species count probabilities are used to determine mutation rates. Species count probability of each individual may be calculated by using (24). Each population member relates to an associated probability that indicates the likelihood with which it exists as a solution for a given problem. If the probability has lower value, then the chance of mutation with other solution is higher but if the probability has a higher value, then it has very little chance to mutate. Mutation rate of each individual solution may be defined by (27) [53,54]:

$$m(s) = m_{max}\left(\frac{1-Ps}{P_{max}}\right) \tag{27}$$

where $m(s)$ is the mutation rate for habitat containing s species, $m_{\max}$ is the maximum mutation rate and $P_{\max}$ is the maximum probability.

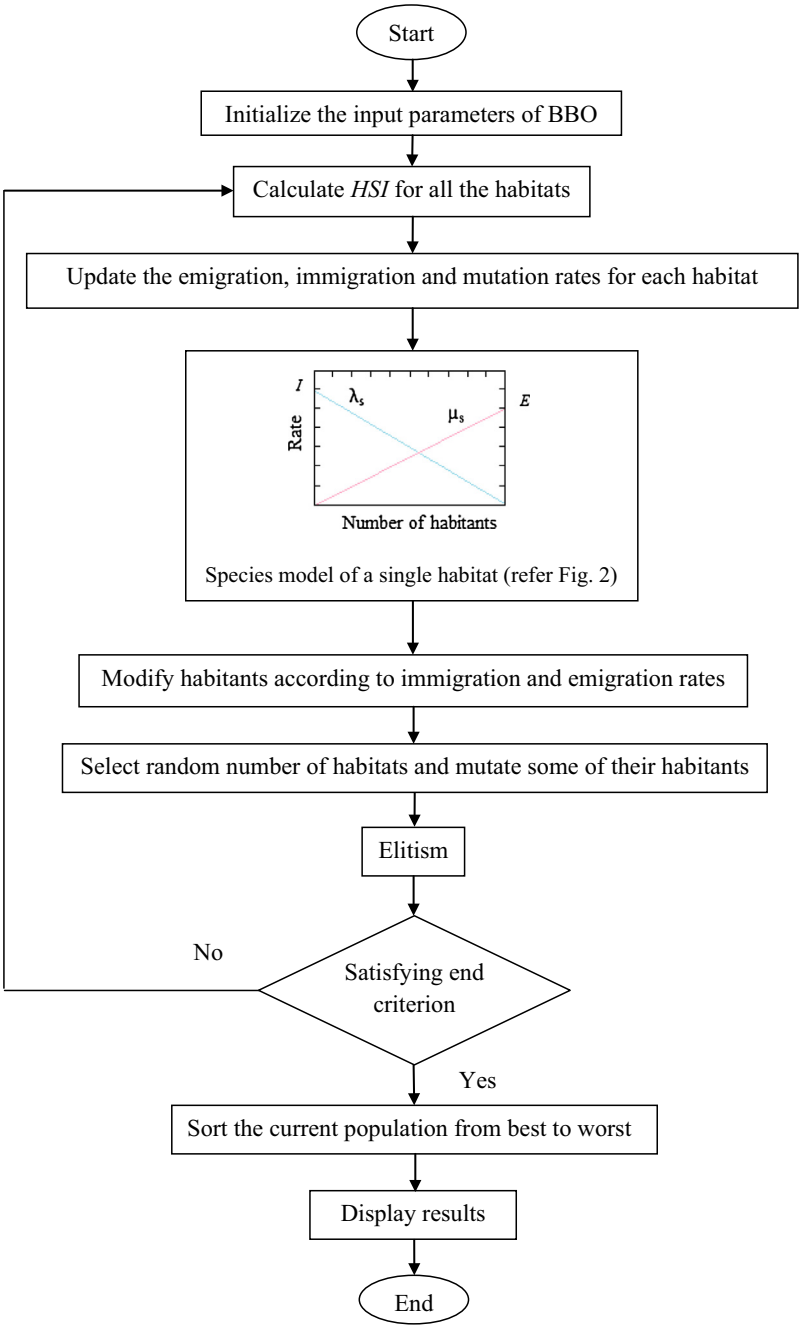

Fig. 3. Flowchart of BBO [54].

Computational procedure of BBO

Algorithm 2 [53] describes the computational procedure of the studied BBO algorithm while Fig. 3 portrays its flowchart [54].

Algorithm 2: Computational procedure of BBO [53]

Step 1	Initialize the BBO parameters.
Step 2	Initialize the random set of habitats where each habitat is analogous to the potential solution to the particular problem.
Step 3	For each habitat, update the mutation rate, the immigration rate and the emigration rate.
Step 4	Adjust each non-elite habitat by probabilistically using immigration and emigration rates.
Step 5	Update the probability of species count for each habitat using (24). Transform each non-elite habitat depending upon its probability and calculate each HSI.
Step 6	Sort the population from the best to the worst.
Step 7	Go to Step 3 for next iteration.
Step 8	Stop iterations after reaching a certain number of iterations.

KHA

KHA is a novel meta-heuristic swarm intelligence based optimization method for solving optimization problems. It is based on the herding behavior of krill swarms in response to the specific biological and environmental processes. It has been first introduced by Gandomi and Alavi in 2012 [25]. In this optimization algorithm, the objective function for the krill movement is supposed to be a combination of the least distances of the position of the food and the highest density of the herd. The KHA repeats the implementation of the three movements and follows the search directions that enhance the objective function value. The movement of each individual krill is determined by three main processes. These are:

(a) movement induced by other krill individuals,
(b) foraging activity,
(c) random diffusion.

Basic KHA technique may be expressed by Lagrangian model in an n dimensional decision space, as shown in (28) [25]

$$\frac{du_p}{dt} = W_p^{new} + W_{F_p}^{new} + W_{D_p}^{new} \tag{28}$$

where W_p^{new} is the motion induced by other krill individuals, $W_{F_p}^{new}$ is the foraging motion and $W_{D_p}^{new}$ is the physical diffusion of the krill individuals, respectively.

Motion induced by other krill individuals

In this process, the krill individuals try to maintain a high density while the velocity of each individual is influenced by the movement of the others. The direction of motion induced (ψ_p) is, approximately, evaluated by the three effects, namely, (a) local effect, (b) target effect and (c) repulsive effect. For a krill individual p, this motion may be formulated as

$$W_p^{new} = \psi_p W_p^{max} + u_w W_p^{old} \tag{29}$$

$$\psi_p = \sum_{q=1}^{n_s} \left[\frac{v_p - v_q}{v_w - v_b} \times \frac{u_p - u_q}{|u_p - u_q| + rand\ (0,1)} \right] + 2\left[rand\ (0,1) + \frac{z}{z_{max}} \right] v_p^{best}\ X_p^{best} \tag{30}$$

where W_p^{max} is the maximum induced motion, u_w is the inertia weight of the motion induced in the range [0, 1], W_p^{old} is the previous induced motion of the pth krill individuals, v_w and v_b are the worst and the best position among all the krill individuals of the population, respectively, v_p and v_q are the fitness values of the pth and the qth individuals, respectively, n_s is the number of krill individuals other than the particular krill, z and $z_{\max}$ are, respectively, the number of current iteration and maximum number of iterations and u represents the related positions.

For the determination of the distance between the individual krills and the neighbors, a parameter named as sensing distance (D_s) is used. It may be formulated by (31)

$$D_s = \frac{1}{5N_p} \sum_{k=1}^{N_p} |u_p - u_k| \tag{31}$$

where N_p is the total number of the krill individual and u_x, u_k are the position of the xth and kth krill, respectively. It is noted here that if the distance between the two individual krill has lesser value than the sensing distance, then they are treated as neighbors.

Foraging activity

Foraging activity is based upon two main factors. First is the present food location and second is the information about the previous food location. The foraging velocity may be expressed for the pth krill individual by (32)

$$W_{F_p}^{new} = 0.02 \times \left[2 \times \left(1 - \frac{z}{z_{max}} \right) \times v_p \times \frac{\sum_{k=1}^{n_s} \frac{u_k}{v_k}}{\sum_{k-1}^{n_s} \frac{1}{v_q}} + v_p^{best} \times X_p^{best} \right] + u_{w_F} \times W_{F_p}^{old} \tag{32}$$

where u_{w_F} is the inertia weight of the foraging motion, $W_{F_p}^{new}$ and $W_{F_p}^{old}$ are the foraging motions of the new and the old pth krill, respectively.

Random diffusion

The random diffusion process of the krill individual is, mainly, considered to enhance the population diversity. It may be expressed as follows:

$$W_{D_p}^{new} = \alpha \times W_D^{max} \tag{33}$$

where W_D^{max} is the maximum diffusion speed and α is the random directional vector lies between $[-1, 1]$.

Position update

In this process, the individual krill alters its current positions and moves to better positions based on induction motion, foraging motion and random diffusion motion. According to the three above discussed motions, the updated position of the pth krill individual may be expressed by (34)

$$u_p^{new'} = u_p^{new} + (W_p^{new} + W_{F_p}^{new} + W_{D_p}^{new}) \times P_C \sum_{q=1}^{n_d} (U_q - L_q) \tag{34}$$

where n_d is the total number of variables, U_q and L_q are the upper and the lower limits of the qth variables ($q = 1, 2, \ldots, n_d$), respectively and P_C is the position constant number between [0, 2].

In order to improve the performance of the optimization problem and to speed up the convergence property, the crossover and the mutation process of DE algorithm is incorporated with KHA.

(a) *Crossover*

Crossover process is, mainly, controlled by a parameter, called as crossover probability (CR). To update the position of own, each krill individual interacts with others. In this process, the qth component of the pth krill may be formulated by (35) and (36)

$$u_{p,q} = \begin{cases} u_{k,q} & \text{if} \quad rand \leqslant CR \\ u_{p,q} & \text{if} \quad rand < CR \end{cases} \quad \text{where } k = 1, 2, \ldots, N_p \tag{35}$$

$$CR = 0.2 v_p^{best} \tag{36}$$

(b) *Mutation*

Mutation process is, mainly, controlled by a parameter called as mutation probability (MR). This process may be formulated by (37)

$$u_{p,q} = u_{best,q} + \gamma (u_{m,q} - u_{n,q}) \tag{37}$$

where $u_{best,q}$ is the global best vector, $u_{m,q}$ and $u_{n,q}$ are the two randomly selected vectors and γ is a scalar number between 0 and 1.

The modified value of $u_{p,q}$ may be calculated by using (38)

$$u_{p,q}^{mod} = \begin{cases} u_{p,q}^{new} & \text{if} \quad rand \leqslant MR \\ u_{p,q} & \text{if} \quad rand > MR \end{cases} \tag{38}$$

Computational procedure of KHA technique

The computational steps of KHA technique are presented in Algorithm 3 [25] and the corresponding flowchart is drawn in Fig. 4 [25].

Algorithm 3: Computational procedure of KHA [25]

Step 1 Initialization of the algorithm parameters.

Step 2 Afterward, depending upon the population size, initial feasible population set are generated and it may be expressed by (39).

$$X^{N_p} = \begin{bmatrix} X^1_{g_1}, & X^1_{g_2}, & \ldots, & X^1_{g_p}, & \ldots, & X^1_{g_n} \\ X^2_{g_1}, & X^2_{g_2}, & \ldots, & X^2_{g_p}, & \ldots, & X^2_{g_n} \\ X^p_{g_1}, & X^p_{g_2}, & \ldots, & X^p_{g_p}, & \ldots, & X^p_{g_n} \\ . & . & \ldots, & . & \ldots & . \\ X^{N_p}_{g_1}, & X^{N_p}_{g_2}, & \ldots, & X^{N_p}_{g_p}, & \ldots, & X^{N_p}_{g_n} \end{bmatrix} \quad (39)$$

Step 3 Determination of the fitness value of each individual according to its position using (34).

Step 4 Evaluation of motion induced by other krill individuals, foraging motion and random diffusion motion using (29), (32) and (33), in order.

Step 5 Modification of the position of each krill individual using (34).

Step 6 Implementation of crossover and mutation process to modify the position of each individual krill in the search space using (35) and (37), respectively.

Step 7 Stop if maximum number of iteration is achieved, otherwise repeat from Step 2.

OKHA

Opposition based learning

OBL is, basically, a machine intelligence strategy which was expressed by Tizhoosh [41]. It considers the current individual and its opposite individual simultaneously in order to get a better approximation at the same time for a current candidate solution. It has been also proved that an opposite candidate solution has a greater opportunity to be closer to the global optimum solution than a random candidate solution [55]. So, the concept of OBL may be utilized to enhance the performance of the population-based algorithms [56].

The general OBL concept has been, successfully, applied in DE [57], GA [58], reinforcement learning [59], ACO [60], window memorization [61], SA [62], PSO [63,64], fuzzy sets [65], BBO [66,67], teaching learning based optimization (TLBO) [68] and so on.

In proposing this technique, two definitions (*i.e.* opposite point and opposite number) should be clearly defined as follows.

Opposite number

Let $p \in [u, v]$ be a real number. The opposite number of p^* is defined by (40)

$$p^* = u + v - p \quad (40)$$

Opposite point

Let, $P = (p_1, p_2, \ldots, p_d)$ be a point in d-dimensional space, where $p_m \in [u_m, v_m]$ and $m = \{1, 2, \ldots, d, \ldots, n\}$. The opposite point is defined by (41)

$$p^*_m = u_m + v_m - p_m \quad (41)$$

Opposition based population initialization

By utilizing opposite points, a suitable starting candidate solution may be obtained even when there is not a priori knowledge about the solution. The main steps of the proposed approach are listed below:

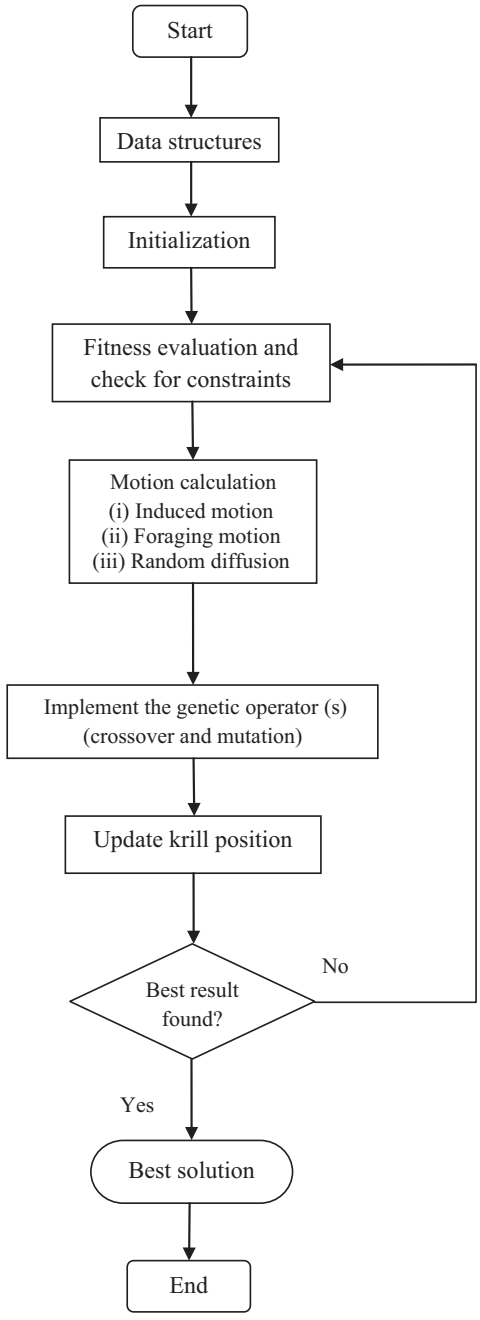

Fig. 4. Flowchart of KHA [25].

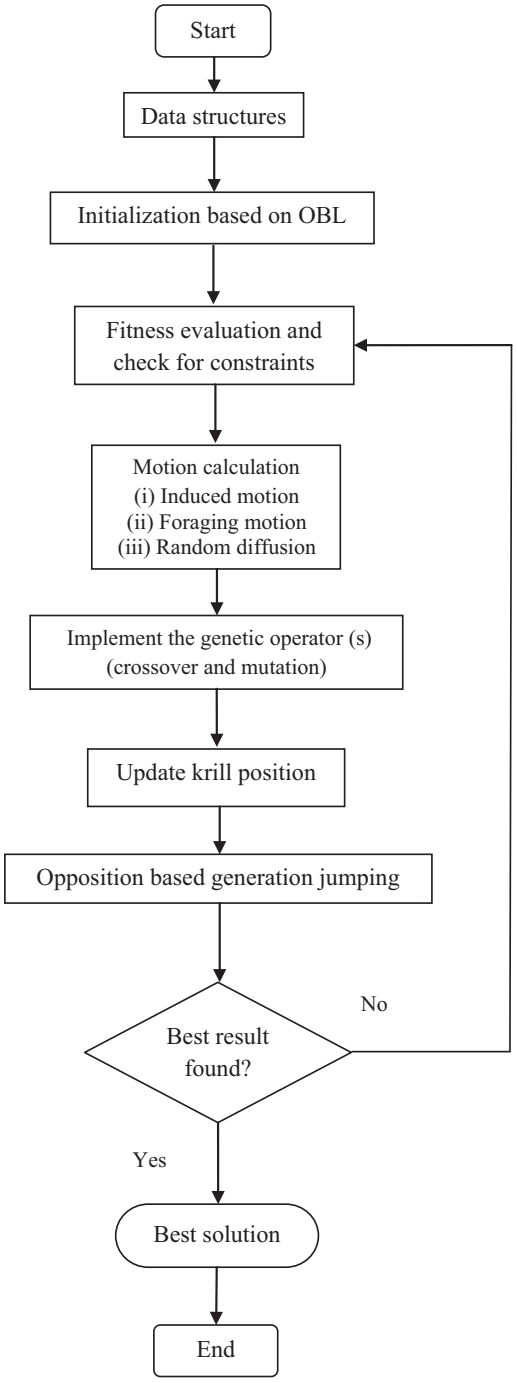

Fig. 5. Flowchart of the proposed OKHA.

Step 1	Initialize the population set $P(N_p)$ in a random manner.
Step 2	Calculate opposite population by $OP_{m,n} = u_n + v_n - P_{m,n}$ $m = 1,2,\ldots,N_p$ and $n = 1,2,\ldots,d$ (42) where $P_{m,n}$ and $OP_{m,n}$ denote the n th variable of the m th vector of the population and opposite population, respectively.
Step 3	Select the fittest individuals from $\{P \cup OP\}$ as initial population.

Opposition based generation jumping

If the similar type of approach (as mentioned above) is applied to the current population, the whole evolutionary process can be forced to jump to a new candidate solution which is more suitable than the current one. Based on a jumping rate (J_R), after following the induction, the foraging action and the random diffusion processes of KHA, the new population is generated and opposite population is calculated. From this comparison, the fittest individuals are selected. In each generation, search space is reduced to calculate the opposite points *i.e.*

$$OP_{m,n} = Min_n^p + Max_n^p - P_{m,n} \quad m = 1,2,\ldots,N_p \text{ and } n = 1,2,\ldots,d \tag{43}$$

where $\lfloor Min_n^p, Max_n^p \rfloor$ is the current interval in the population which is becoming increasingly smaller than the corresponding initial range $[u_n, v_n]$.

Flowchart of OKHA

The flow chart of the proposed OKHA algorithm is shown in Fig. 5.

OKHA applied to TSCOPF problem

The main steps of the proposed OKHA approach, as applied to the TSCOPF problem of the present work, is described in Algorithm 4.

Algorithm 4: Implementation of the proposed OKHA for TSCOPF problem

Step 1	Initialize all the individual population within their effective real operating limits in a random manner. The voltages of generator buses, active power of the entire generator buses except slack bus, tap settings of regulating transformers, reactive power injections are generated randomly satisfying the inequality constraints defined in (5)–(7). Afterward, Newton Raphson based load flow [69] is run to check if the constraints are within the limits or not. Moreover, the transient stability criterion, defined in (11), is verified. If any one of these does not satisfy the inequality constraints, then that particular set has to be discarded and initialised again. Now, several initial set (P), depending upon the population size, are generated. A feasible solution set represents the position of different krill individuals. Depending upon the population size, initial krill position matrix is created by (39).
Step 2	For determining the opposite population (OP) set using opposition based learning, again the voltages of generator buses, active power of the entire generator buses except slack bus, tap settings of regulating transformers, reactive power injections are evaluated by using (42) and Newton Raphson based load flow method [69] is run to satisfy the equality constraints. Moreover, the transient inequality criterion is verified. If any one of these violates the limits, then the corresponding set is reinitialised.
Step 3	Select the fittest individuals from $\{P \cup OP\}$.
Step 4	Sort the selected fittest vectors from best to worst.
Step 5	Choose few elite solutions based on the fitness value.

(*continued on next page*)

Step 6 The position of krill individuals of non-elite population set are modified using (34). Afterward, the position of each krill individual is further modified applying crossover and mutation operation of DE (defined in (35) and (37), in order).

Step 7 Run load flow [69] to evaluate the dependent variables of TSCOPF and calculate the fitness value of the population set.

Step 8 Based on jumping rate (J_R), the opposite population of the current population and corresponding fitness values are calculated.

Step 9 Fittest individuals are determined from $\{P \cup OP\}$.

Step 10 Stop if maximum number of iteration is achieved, otherwise repeat from Step 6.

Simulation results and discussion

In this section, KHA and the proposed OKHA algorithms are applied for the solution of the TSCOPF problem of power system. Two different test systems (*viz.* 10 generator New England 39-bus system and 17 generator 162-bus system) are considered for verifying their applicability for TSCOPF study. For both the systems, a classical generator model is used as the synchronous generators and a constant impedance model is used for the loads. All simulations are carried out using MATLAB 2008a and implemented on a personal computer with Intel Pentium Dual-Core 1.73 GHz processor with 2GB-RAM. For implementing the techniques in TSCOPF problems, maximum number of iterations of 100 (for both KHA and OKHA), integration time step of 0.01 s, total population size of 50 (for both KHA and OKHA), jumping rate (J_R) of 0.3 are considered. Total simulation period is 5.0 s for all the simulation works. Also, input parameters of KHA are taken from [25], which are as follows: $W_p^{max} = 0.01$, $W_D^{max} = 0.005$, $P_C = 0.5$ and the values of initial inertia weights (u_w and u_{w_F}) are equal to 0.9 for increasing exploration and these values are decreased linearly to 0.1 at the end for encouraging exploitation. In order to demonstrate the robustness, statistical comparison such as best, worst and mean results over 50 independent trial runs are reported. To indicate the superior optimization capability of the proposed OKHA, the results of interest are **bold faced** in all the respective tables.

Test case 1: 10 generator New England 39-bus system

This system comprises of 10 generator buses and 19 load buses. Bus 39 is taken here as the slack bus. The system data are available in [69]. The cost co-efficient data and power generation limits of different generators of this system are adopted from [18] and are given in Table 1. A 3-phase to ground fault occurs at bus 29 and between lines 28–29 in the system. The fault is cleared by opening the contacts of the circuit breakers by 100 ms. The simulation results of the proposed OKHA technique is shown in Table 2 and its results are compared to the other obtained results offered by generator classical model (CM) [70], generator detailed model

Table 1
Cost co-efficient data and power generation limits of 10 generator New England 39-bus system [18].

Generator	a_x	b_x	c_x	P_{min} (MW)	P_{max} (MW)
G_1	0.0193	6.9	0	0	350
G_2	0.0111	3.7	0	0	650
G_3	0.0104	2.8	0	0	800
G_4	0.0088	4.7	0	0	750
G_5	0.0128	2.8	0	0	650
G_6	0.0094	3.7	0	0	750
G_7	0.0099	4.8	0	0	750
G_8	0.0113	3.6	0	0	700
G_9	0.0071	3.7	0	0	900
G_{10}	0.0064	3.9	0	0	1200

Table 2
Simulation results obtained by different methods for fault at bus number 29 of 10 generator New England 39-bus system.

Control variables	DSA [71]	CM [70]	DM [70]	GSA [Studied]	BBO [Studied]	KHA [Studied]	OKHA [Proposed]
PG_{30} (p.u.)	2.4783	2.4873	2.49447	2.3940	2.7508	2.4599	2.4656
PG_{31} (p.u.)	5.7723	5.7784	5.78359	5.7649	5.6827	5.6758	5.6849
PG_{32} (p.u.)	6.5341	6.5447	6.54356	6.4892	6.6363	6.2331	6.4291
PG_{33} (p.u.)	6.4328	6.4500	6.4176	5.6787	6.4231	6.4613	6.3341
PG_{34} (p.u.)	5.1778	5.1882	5.1741	5.4076	5.1307	5.1490	4.9784
PG_{35} (p.u.)	6.6246	6.6432	6.60731	6.2537	6.7498	6.0591	6.3682
PG_{36} (p.u.)	5.6959	5.7137	5.68181	5.4954	5.4800	5.5764	5.4440
PG_{37} (p.u.)	5.4388	5.4781	5.47695	5.1126	5.5535	5.2100	5.3074
PG_{38} (p.u.)	7.7454	7.5202	7.54618	8.4546	7.6663	8.4106	8.4546
PG_{39} (p.u.)	10.0035	9.9560	10.03181	10.3041	9.2867	10.1462	9.8996
V_{30} (p.u.)	0.9840	1.0150	1.0150	1.0928	1.0977	1.0824	1.0964
V_{31} (p.u.)	1.0740	1.0870	1.0870	1.0873	1.0912	1.0447	1.0934
V_{32} (p.u.)	1.0080	1.0290	1.0290	1.0860	1.0766	1.0518	1.0870
V_{33} (p.u.)	1.0140	1.0160	1.0160	1.0931	1.0974	1.0763	1.0993
V_{34} (p.u.)	1.0190	1.0220	1.0220	1.0967	1.0938	1.0435	1.0967
V_{35} (p.u.)	1.0670	1.0620	1.0620	1.0820	1.0986	1.0800	1.0980
V_{36} (p.u.)	1.0870	1.0900	1.0900	1.0999	1.0989	1.0820	1.0999
V_{37} (p.u.)	1.0120	1.0470	1.0470	1.0974	1.0956	1.0679	1.0974
V_{38} (p.u.)	1.0510	1.0380	1.0380	1.0988	1.0922	1.0962	1.0988
V_{39} (p.u.)	1.0190	1.0530	1.0530	1.0948	1.0923	1.0844	1.0973
Cost ($/h)	61799.68	61600.76	61597.76	60969.18	60957.87	60943.69	**60883.55**

Table 3
Statistical comparison (out of 50 independent trial runs) among various methods for 10 generator New England 39-bus system.

Parameter	DSA [71]	CM [70]	DM [70]	GSA [Studied]	BBO [Studied]	KHA [Studied]	OKHA [Proposed]
Best cost ($/h)	61799.68	61600.76	61597.76	60969.18	60957.87	60943.69	**60883.55**
Worst cost ($/h)	NR*	NR*	NR*	61015.13	61008.42	60985.19	60892.25
Mean cost ($/h)	NR*	NR*	NR*	60989.15	60985.17	60962.24	60885.77

NR* means not reported in the referred literature.

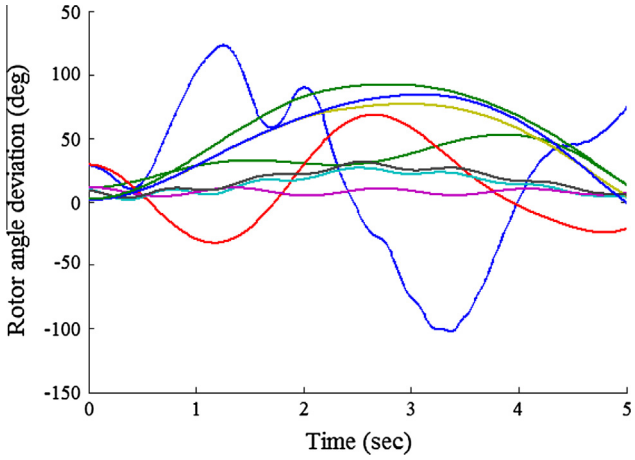

Fig. 6. Relative rotor angle deviation of different generators with respect to slack bus for 10 generator New England 39-bus system using GSA.

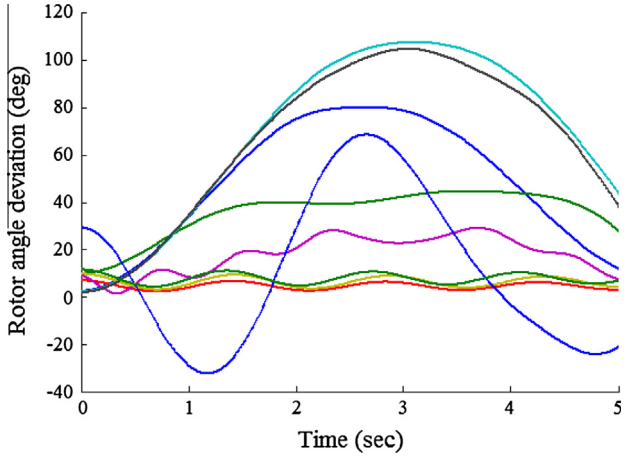

Fig. 7. Relative rotor angle deviation of different generators with respect to slack bus for 10 generator New England 39-bus system using BBO.

(DM) [70], dynamic simulation algorithm (DSA) [71], GSA, BBO and KHA techniques. It is found that the total cost from OKHA is less than that obtained by all other discussed algorithms.

Best, mean and worst fuel costs obtained by GSA, BBO, KHA and the proposed OKHA are presented in Table 3. From these statistical results, it is clear that the worst cost value of OKHA algorithm is even less as compared to the best fuel costs of other existing techniques. This observation clearly suggests the higher computational efficiency of OKHA than any other methods in terms of quality of solution and robustness. The relative rotor angle deviation curves of different generators with respect to slack bus are shown in Figs. 6–9, sequentially, (using GSA, BBO, KHA and proposed OKHA technique). From this rotor angle deviation curves, it is seen that all the curves are within limits and maintains stability. The convergence characteristics obtained by different techniques of this test

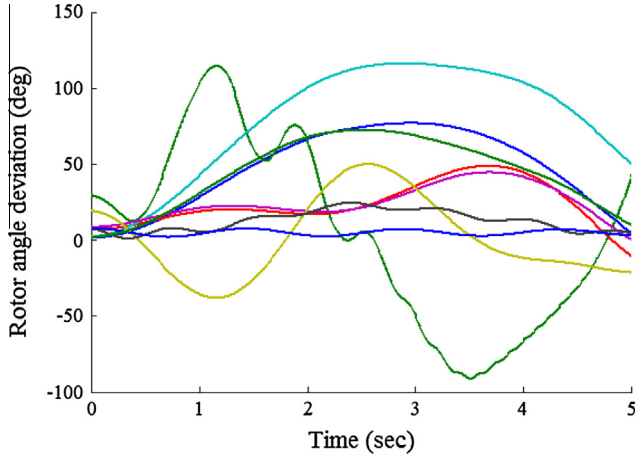

Fig. 8. Relative rotor angle deviation of different generators with respect to slack bus for 10 generator New England 39-bus system using KHA.

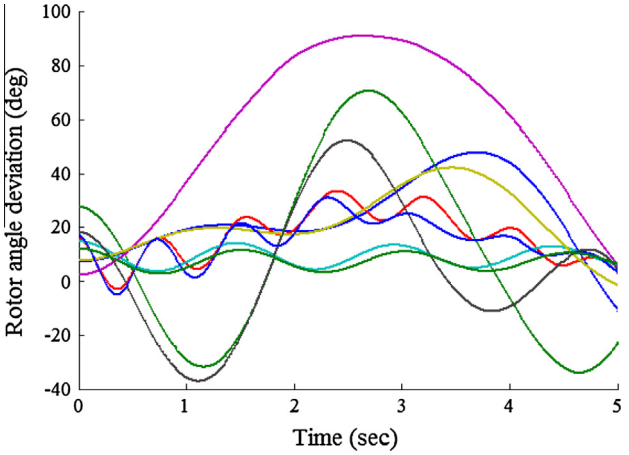

Fig. 9. Relative rotor angle deviation of different generators with respect to slack bus for 10 generator New England 39-bus system using OKHA.

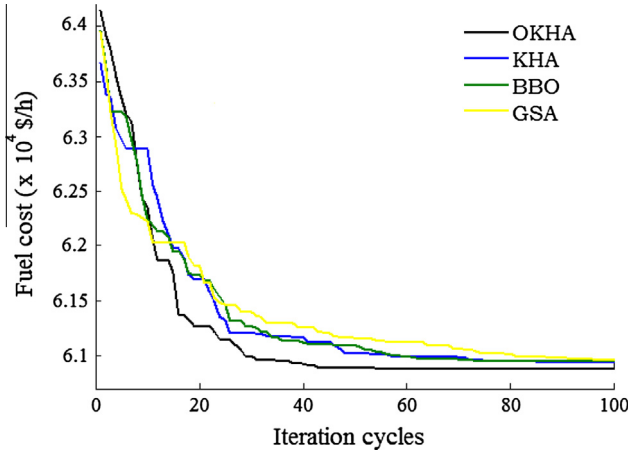

Fig. 10. Comparative convergence characteristic of fuel cost for 10 generator New England 39-bus system.

Table 4
Cost co-efficient data and power generation limits of 17 generator 162-bus system [72].

Generator	a_x	b_x	c_x	P_{min} (MW)	P_{max} (MW)
G_3	0.00064	0.50	0.0	1000	2300
G_6	0.00098	0.30	0.0	500	1094
G_{15}	0.00076	0.50	0.0	1000	1800
G_{27}	0.00076	0.50	0.0	1000	1800
G_{73}	0.00150	0.20	0.0	200	747
G_{76}	0.00088	0.30	0.0	500	1355
G_{99}	0.00200	0.40	0.0	0	450
G_{101}	0.00200	0.40	0.0	0	382
G_{108}	0.00084	0.30	0.0	0	1200
G_{114}	0.00200	0.40	0.0	0	431
G_{118}	0.00200	0.40	0.0	0	473
G_{121}	0.00150	0.30	0.0	200	920
G_{124}	0.00640	0.52	0.0	1000	2851
G_{125}	0.00640	0.67	0.0	1000	2688
G_{126}	0.00640	0.42	0.0	1000	2767
G_{130}	0.00150	0.30	0.0	200	755
G_{131}	0.00150	0.30	0.0	200	875

Table 5
Simulation results obtained by different methods for fault at bus number 1 of 17 generator 162-bus system.

Control variables	GSA [Studied]	BBO [Studied]	KHA [Studied]	OKHA [Proposed]
PG_3 (p.u.)	20.8952	21.7302	19.3818	19.9436
PG_6 (p.u.)	10.4926	10.8603	10.7574	10.8517
PG_{15} (p.u.)	16.1527	16.2511	17.1541	16.9445
PG_{27} (p.u.)	17.5148	17.9683	17.6595	17.9750
PG_{73} (p.u.)	7.4503	6.2834	7.3883	6.9504
PG_{76} (p.u.)	9.5115	9.6565	9.5236	9.9380
PG_{99} (p.u.)	4.1530	3.4393	4.4420	4.4194
PG_{101} (p.u.)	3.7575	3.7544	3.6922	3.6567
PG_{108} (p.u.)	9.7397	9.5216	9.5115	9.9905
PG_{114} (p.u.)	4.2481	4.2173	4.2053	4.3053
PG_{118} (p.u.)	4.3501	4.6996	4.6884	4.7010
PG_{121} (p.u.)	8.5450	9.0452	8.8195	8.1637
PG_{124} (p.u.)	10.8524	10.0329	10.0178	10.7940
PG_{125} (p.u.)	10.6664	11.0298	10.4218	10.0368
PG_{126} (p.u.)	10.2292	10.3578	11.3468	10.4579
PG_{130} (p.u.)	7.5071	7.5193	7.2521	7.2489
PG_{131} (p.u.)	8.6936	8.4358	8.5284	8.1677
V_3 (p.u.)	1.0533	1.0591	1.0518	1.0569
V_6 (p.u.)	1.0548	1.0597	1.0532	1.0558
V_{15} (p.u.)	1.0515	1.0495	1.0535	1.0414
V_{27} (p.u.)	1.0446	1.0537	1.0434	1.0502
V_{73} (p.u.)	1.0510	1.0565	1.0510	1.0593
V_{76} (p.u.)	1.0412	1.0545	1.0587	1.0585
V_{99} (p.u.)	1.0548	1.0542	1.0562	0.9943
V_{101} (p.u.)	1.0509	1.0574	1.0475	1.0567
V_{108} (p.u.)	1.0567	1.0526	1.0431	1.0376
V_{114} (p.u.)	1.0462	1.0436	1.0485	1.0230
V_{118} (p.u.)	1.0336	1.0546	1.0521	1.0484
V_{121} (p.u.)	1.0537	1.0598	1.0594	1.0568
V_{124} (p.u.)	1.0585	1.0585	1.0535	1.0514
V_{125} (p.u.)	1.0457	1.0523	1.0569	1.0481
V_{126} (p.u.)	1.0358	1.0590	1.0511	1.0497
V_{130} (p.u.)	1.0552	1.0528	1.0447	1.0483
V_{131} (p.u.)	1.0570	1.0580	1.0244	1.0476
Cost ($/h)	45234.08	45220.62	45162.77	**44666.16**
Loss (p.u.)	6.2126	6.2562	6.2439	5.9985

Table 6
Statistical comparison (out of 50 independent trial runs) among various methods for 10 generator 162-bus system.

Parameters	GSA [Studied]	BBO [Studied]	KHA [Studied]	OKHA [Proposed]
Best cost ($/h)	45234.08	45220.62	45162.77	**44666.16**
Worst cost ($/h)	45282.93	45263.57	45204.42	44676.71
Mean cost ($/h)	45259.65	45241.49	45186.59	44674.54

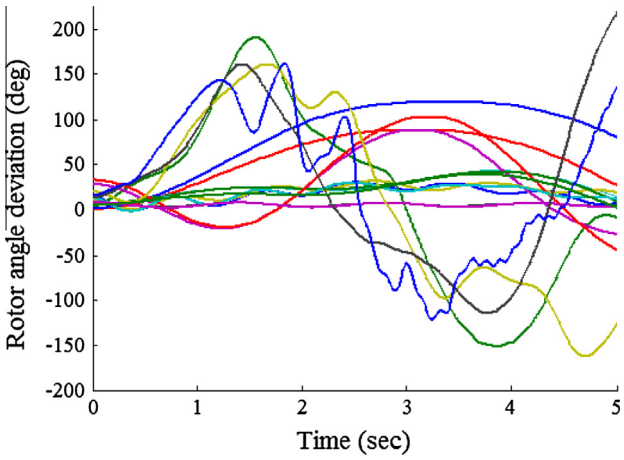

Fig. 11. Relative rotor angle deviation of different generators with respect to slack bus for 17 generator 162-bus system using GSA.

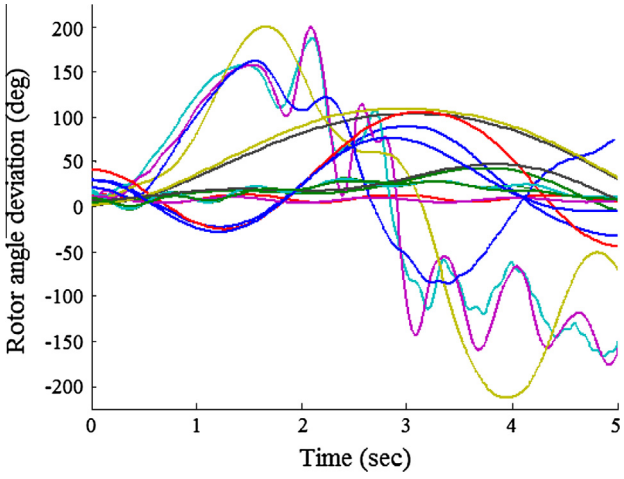

Fig. 12. Relative rotor angle deviation of different generators with respect to slack bus for 17 generator 162-bus system using BBO.

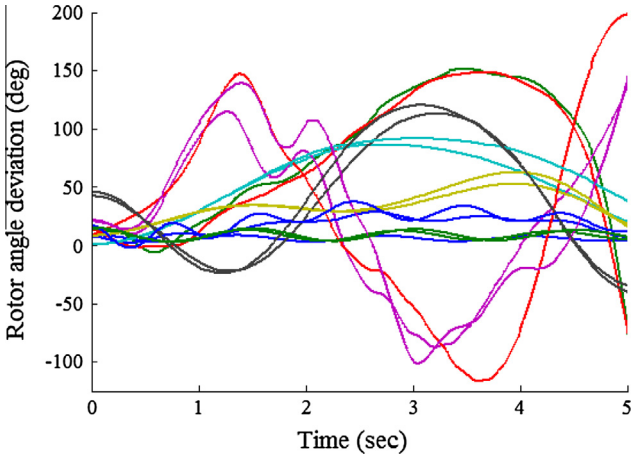

Fig. 13. Relative rotor angle deviation of different generators with respect to slack bus for 17 generator 162-bus system using KHA.

system are illustrated in Fig. 10. From the comparative convergence characteristics of the fuel cost, it is fascinating to observe that the cost function value converges smoothly at lesser iteration cycles for the proposed OKHA than GSA, BBO and KHA techniques.

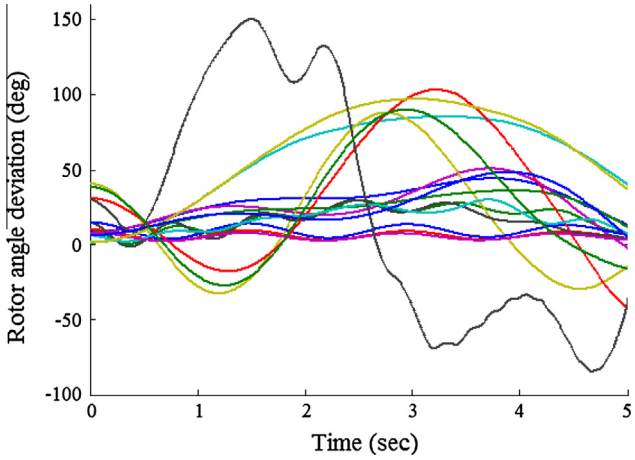

Fig. 14. Relative rotor angle deviation of different generators with respect to slack bus for 17 generator 162-bus system using OKHA.

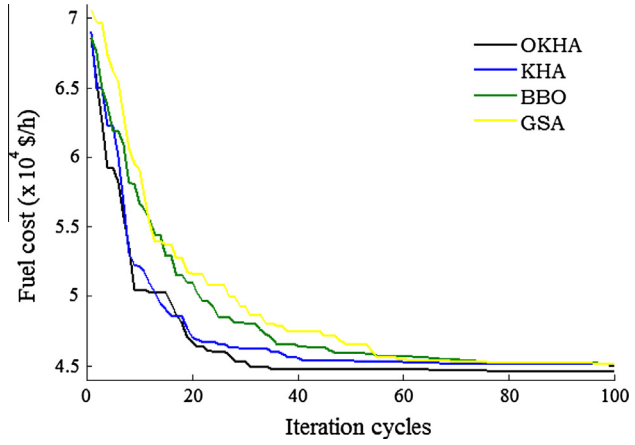

Fig. 15. Comparative convergence characteristics of fuel cost for 17 generator 162-bus system.

Test case 2: 17 generator 162-bus system

In order to further investigate the superiority and robustness of the proposed OKHA method, it is, finally, applied on a large system consisting of 162 buses. The upper and lower limits of bus voltage magnitudes are 0.94–1.06 p.u., respectively. System data of this test system is available in [72]. The fuel cost parameters and corresponding generator ratings are given in Table 4. A 3-phase to ground fault occurs at bus 1 and within lines 1–4. The fault is cleared by tripping the line at 0.240 s. The proposed algorithm is used to minimize the same objective function as used in Test case 1 and its performance is compared to that of GSA, BBO and KHA. The simulation results of the proposed OKHA technique along with GSA, BBO and KHA are given in Table 5. From Table 5, it is observed that, in minimizing the fuel cost, OKHA, GSA, BBO and KHA do not violate any operating constraint limits. The comparison shows that OKHA provides the minimum cost among all the comparative algorithms which demonstrate that the proposed method outperforms the other techniques in terms of optimal solutions.

Moreover, the statistical comparisons of best, worst and mean fuel cost values of different algorithms are listed in Table 6. From the statistical results of Table 6, it is clear that the difference among the best, worst and mean objective cost values, as obtained by OKHA, are very much insignificant. Moreover, it is observed that the worst value of OKHA is even much better than the best values offered by the other indicated methods. It clearly suggests that the proposed OKHA method produces similar results in most of the trials and, thus, the robustness of the proposed method may be proved. Figs. 11–14 are pertaining to the relative rotor angle deviation curves of different generators with respect to the slack bus of this test system (in the order of employing GSA, BBO, KHA and the proposed OKHA techniques). From rotor angle deviation curves, it may be inferred that all the curves are within safe limits and maintains stability. To illustrate the convergence property of the different algorithms, fuel cost values over 100 iterations are plotted in Fig. 15. This comparative convergence graph shows that cost function value converges smoothly by all the methods without any abrupt oscillations and OKHA converges at lesser iteration cycles than all other adopted methods.

Constraint-handling mechanism

Swarm intelligence behavior based nature-inspired evolutionary algorithms are unable to deal with the constraint-handling mechanism of constraints for solving numerical optimization problems. Recently, different researches are going on for successful handling of these constraints to solve nature inspired algorithms. The most significant and delegate techniques in this direction are presented and discussed in [73]. Penalty factor approach may be also used to find out the violated results in optimization problem. Ensemble of constrained handling techniques has been considered in [74]. For further interest on this topic, the works presented in [73,74] may be referred. However, ensemble of constraint-handling technique [74] is adopted in the present work.

Conclusion

To enhance the convergence behavior and to speed up the search process, OBL concept is applied on basic KHA algorithm and this new variant of KHA is termed as OKHA. OKHA is implemented, successfully, to solve TSCOPF problem of power systems. To check the superiority of the proposed method, it is implemented on two different standard test systems like 10 generator New England 39-bus system and 17 generator 162-bus systems. In addition to the basic KHA and the proposed OKHA, two recently developed algorithms like GSA and BBO are also adopted for the sake of comparison. The simulation test results, supported by statistical analysis, show the efficiency, the robustness, the stability and the enhanced convergence rate of the proposed OKHA technique over the other compared algorithms. This comparison results confirm that the proposed OKHA may be very much promising and encouraging tool for future engineering optimization task.

References

[1] Gomez-Exposito A, Conejo AJ, Canizares C. Electric energy systems: analysis and operation. London: CRC Press; 2008.
[2] Carpentier J. Optimal power flows. Int J Electr Power Energy Syst 1979;1(1):3–15.
[3] Dommel HW, Tinney WF. Optimal power flow solutions. IEEE Trans Power Apparat Syst 1968;PAS-87(10):1866–76.
[4] Momoh JA, Koessler RJ, Bond MS, et al. Challenges to optimal power flow. IEEE Trans Power Syst 1997;12(1):444–55.
[5] Bakirtzis AG, Biskas PN, Zoumas CE, et al. Optimal power flow by enhanced genetic algorithm. IEEE Trans Power Syst 2002;17(2):229–36.
[6] Shoults RR, Sun DT. Optimal power flow based upon P-Q decomposition. IEEE Trans Power Apparat Syst 1982;PAS-101(2):397–405.
[7] Sun DI, Ashley B, Brewer B, et al. Optimal power flow by Newton approach. IEEE Trans Power Apparat Syst 1984;PAS-103(10):2864–80.
[8] Saha TN, Maitra A. Optimal power flow using the reduced Newton approach in rectangular coordinates. Int J Electr Power Energy Syst 1998;20(6):383–9.
[9] Bai X, Wei H, Fujisawa K, et al. Semi definite programming for optimal power flow problems. Int J Electr Power Energy Syst 2008;30(6–7):383–92.

[10] Huneault M, Galiana FD. A survey of the optimal power flow literature. IEEE Trans Power Syst 1991;6(2):762–70.
[11] Momoh JA, El-Hawary ME, Adapa R. A review of selected optimal power flow literature to 1993, Part II: Newton, linear programming and interior point methods. IEEE Trans Power Syst 1999;14(1):105–11.
[12] Xue Y, Pavella M. Extended equal-area criterion: an analytical ultra-fast method for transient stability assessment and preventive control of power systems. Int J Electr Power Energy Syst 1989;11(2):131–49.
[13] Chan KY, Chan KW, Ling SH, et al. Solving multi-contingency transient stability constrained optimal power flow problems with an improved GA. In: Proc IEEE congress on evolutionary computation (CEC 2007), Singapore, September 25–28. p. 2901–8.
[14] Xia Y, Chan KW, Liu M. Direct nonlinear primal-dual interior-point method for transient stability constrained optimal power flow. IEE Proc Gener Transm Distrib 2005;152(1):11–6.
[15] Jiang Q, Geng G. A reduced-space interior point method for transient stability constrained optimal power flow. IEEE Trans Power Syst 2010;25(3):1232–40.
[16] Jiang Q, Huang Z. An enhanced numerical discretization method for transient stability constrained optimal power flow. IEEE Trans Power Syst 2010;25(4):1790–7.
[17] Pizano-Martinez A, Fuerte-Esquivel CR, Ruiz-Vega D. A new practical approach to transient stability-constrained optimal power flow. IEEE Trans Power Syst 2011;26(3):1686–96.
[18] Xu Y, Dong ZY, Meng K, et al. A hybrid method for transient stability-constrained optimal power flow computation. IEEE Trans Power Syst 2012;27(4):1769–77.
[19] Geng G, Jiang Q. A two-level parallel decomposition approach for transient stability constrained optimal power flow. IEEE Trans Power Syst 2012;27(4):2063–73.
[20] Tangpatiphan K, Yokoyama A. Adaptive evolutionary programming with neural network for transient stability constrained optimal power flow. In: Proc 15th int conf on intelligent system applications to power systems (ISAP '09), Curitiba, November 8–12. p. 1–6.
[21] Mo N, Zou ZY, Chan KW, et al. Transient stability constrained optimal power flow using particle swarm optimization. IET Gener Transm Distrib 2007;1(3):476–83.
[22] Rautray SK, Choudhury S, Mishra S, et al. A particle swarm optimization based approach for power system transient stability enhancement with TCSC. Procedia Technol 2012;6:31–8.
[23] Cai HR, Chung CY, Wong KP. Application of differential evolution algorithm for transient stability constrained optimal power flow. IEEE Trans Power Syst 2008;23(2):719–28.
[24] Kilic U, Ayan K. Transient stability constrained optimal power flow solution of ac-dc systems using genetic algorithm. In: Proc 3rd int conf on electric power and energy conversion systems (EPECS-2013), Istanbul, October 2–4. p. 1–6.
[25] Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 2012;17(12):4831–45.
[26] Gandomi AH, Alavi AH, Talatahari S. Structural optimization using krill herd algorithm. In: Yang XS et al., editors. Swarm intelligence and bio-inspired computation: theory and applications. Elsevier; 2013. p. 335–49 [Chapter 15].
[27] Bacanin N, Pelevic B, Tuba M. Krill herd (KH) algorithm for portfolio optimization. Math Comp Bus Manuf Tour. p. 39–44. ISBN: 978-960-474-332-2.
[28] Guo L, Wang G, Gandomi AH, et al. A new improved krill herd algorithm for global numerical optimization. Neurocomputing 2014;138:392–402.
[29] Moodley K, Rarey J, Ramjugernath D. Application of the bio-inspired krill herd optimization technique to phase equilibrium calculations. Comput Chem Eng 2015;74:75–88.
[30] Puongyeam H, Pongcharoen P, Vitayasak S. Application of krill herd (KH) algorithm for production scheduling in capital goods industries. In: Proc int conf on challenges IT, engineering and technology (ICCIET'2014), Phuket (Thailand), July 17–18. p. 67–72.
[31] Gandomi AH, Talatahari S, Tadbiri F, et al. Krill herd algorithm for optimum design of truss structures. Int J Bio-Inspired Comput 2013;5(5):281–8.
[32] Wang GG, Deb S, Gandomi AH, et al. Opposition-based krill herd algorithm with Cauchy mutation and position clamping. Neurocomputing 2016;177:147–57.
[33] Wang GG, Gandomi AH, Alavi AH. An effective krill herd algorithm with migration operator in biogeography-based optimization. Appl Math Modell 2014;38(9–10):2454–62.
[34] Wang GG, Gandomi AH, Alavi AH, et al. A hybrid method based on krill herd and quantum-behaved particle swarm optimization. Neural Comput Appl 2015:1–18.
[35] Wang GG, Guo L, Gandomi AH, Alavi AH, Duan H. Simulated annealing-based krill herd algorithm for global optimization. Abstr Appl Anal 2013;2013:1–11.
[36] Wang GG, Gandomi AH, Alavi AH. A chaotic particle-swarm krill herd algorithm for global numerical optimization. Kybernetes 2013;42(6):962–78.
[37] Wang GG, Guo L, Wang H, et al. Incorporating mutation scheme into krill herd algorithm for global numerical optimization. Neural Comput Appl 2014;24(3):853–71.
[38] Wang GG, Gandomi AH, Alavi AH. Stud krill herd algorithm. Neurocomputing 2014;128:363–70.
[39] Wang GG, Gandomi AH, Alavi AH, et al. Hybrid krill herd algorithm with differential evolution for global numerical optimization. Neural Comput Appl 2014;25(2):297–308.
[40] Wang GG, Gandomi AH, Yang XS, et al. A new hybrid method based on krill herd and cuckoo search for global optimization tasks. Int J Bio-Inspired Comput; 2016 [in press].
[41] Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In: Proc int conf on computational intelligence for modelling control and automation, 2005 and int conf on intelligent agents, web technologies and internet commerce, Vienna, November 28–30. p. 695–701.
[42] Wang H, Wu ZJ, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning. Inf Sci 2011;181(20):4699–714.
[43] Ma H, Ruan X, Jin B. Oppositional ant colony optimization algorithm and its application to fault monitoring. In: Proc 29th Chinese control conference (CCC), Beijing, July 29–31. p. 3895–8.
[44] Ergezer M, Simon D, Du D. Oppositional biogeography-based optimization. In: Proc int conf on systems, man and cybernetics (SMC 2009), San Antonio (TX), October 11–14. p. 1009–14.
[45] Upadhyay P, Kar R, Mandal D, et al. A novel design method for optimal IIR system identification using opposition based harmony search algorithm. J Franklin Inst 2014;351(5):2454–88.
[46] Banerjee A, Mukherjee V, Ghoshal SP. An opposition-based harmony search algorithm for engineering optimization problems. Ain Shams Eng J 2014;5(1):85–101.
[47] Singh RP, Mukherjee V, Ghoshal SP. The opposition-based harmony search algorithm. J Inst Eng (India): Ser B 2014;94(4):247–56.
[48] Shaw B, Mukherjee V, Ghoshal SP. Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm. Int J Electr Power Energy Syst 2014;55:29–40.
[49] Saha SK, Kar R, Mandal D, et al. A new design method using opposition-based BAT algorithm for IIR system identification problem. Int J Bio-Inspired Comput 2013;5(2):99–132.
[50] Kundur P, Paserba J, Ajjarapu V, et al. Definition and classification of power system stability IEEE/CIGRE joint taskforce on stability terms and definitions. IEEE Trans Power Syst 2004;19(3):1387–401.
[51] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179(13):2232–48.
[52] Roy PK, Mandal B, Bhattacharya K. Gravitational search algorithm based optimal reactive power dispatch for voltage stability enhancement. Electr Power Comp Syst 2012;40(9):956–76.
[53] Simon D. Biogeography-based optimization. IEEE Trans Evol Comput 2008;12(6):702–13.
[54] Mirjalili S, Mirjalili SM, Lewis A. Let a biogeography-based optimizer train your multi-layer perceptron. Inf Sci 2014;269:188–209.
[55] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition versus randomness in soft computing techniques. Appl Soft Comput 2008;8(2):906–18.
[56] Wang H, Li H, Liu Y, et al. Opposition based particle swarm algorithm with Cauchy mutation. In: Proc IEEE congress of evolutionary computation (CEC 2007), Singapore, September 25–28. p. 4750–6.
[57] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition-based differential evolution. IEEE Trans Evol Comput 2008;12(1):64–79.
[58] Lin ZY, Wang LL. A new opposition-based compact genetic algorithm with fluctuation. J Comput Inf Syst 2010;6(3):897–904.
[59] Tizhoosh HR. Opposition-based reinforcement learning. J Adv Comput Intell Inf 2006;10(4):578–85.
[60] Malisia AR, Tizhoosh HR. Applying opposition-based ideas to the ant colony system. In: Proc IEEE swarm intelligence symposium (SIS 2007), Honolulu (HI), April 1–5. p. 182–9.
[61] Khalvati F, Tizhoosh HR, Aagaard MD. Opposition-based window memorization for morphological algorithms. In: Proc IEEE symposium on computational intelligence in image and signal processing (CIISP 2007), Honolulu (HI), April 1–5. p. 425–30.
[62] Ventresca M, Tizhoosh HR. Simulated annealing with opposite neighbors. In: Proc IEEE symposium on foundations of computational intelligence (FOCI 2007), Honolulu (HI), April 1–5. p. 186–92.
[63] Han L, He X. A novel opposition-based particle swarm optimization for noisy problems. In: Proc 3rd international conference on natural computation (ICNC 2007), Haikou (China), August 24–27. p. 624–9.
[64] Shahzad F, Rauf Baig A, Masood S, et al. Opposition-based particle swarm optimization with velocity clamping (OVCPSO). Adv Comput Intell 2009;116:339–48.
[65] Tizhoosh HR. Opposite fuzzy sets with applications in image processing. In: Proc joint 2009 int fuzzy systems association world congress and 2009 European society of fuzzy logic and technology conf, Lisbon (Portugal), July 20–24. p. 36–41.
[66] Roy PK, Mandal D. Quasi-oppositional biogeography-based optimization for multi-objective optimal power flow. Electr Power Compon Syst 2011;40(2):236–56.
[67] Bhattacharya A, Chattopadhyay PK. Solution of economic power dispatch problems using oppositional biogeography-based optimization. Electr Power Compon Syst 2010;38(10):1139–60.
[68] Roy PK, Paul C, Sultana S. Oppositional teaching learning based optimization approach for combined heat and power dispatch. Int J Electr Power Energy Syst 2014;57:392–403.
[69] Pai MA. Energy function analysis for power system stability. Norwell, MA: Kluwer; 1989.
[70] Ahmadi H, Ghasemi H, Haddadi AM, et al. Two approaches to transient stability-constrained optimal power flow. Int J Electr Power Energy Syst 2013;47:181–92.

[71] Tu X, Dessaint LA, Duc HN. Transient stability constrained optimal power flow using independent dynamic simulation. IET Gener Transm Distrib 2013;7 (3):244–53.
[72] Power system test case archiveAvailable from: <http://www.ee.washington.edu/research/pstca/>2014 [accessed on: 13 September 2014].
[73] Mezura-Montesa E, Coello Coello CA. Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 2011;1 (4):173–94.
[74] Mallipeddi R, Jeyadevi S, Suganthan PN, et al. Efficient constraint handling for optimal reactive power dispatch problems. Swarm Evol Comput 2012;5: 28–36.