Electrical Power and Energy Systems 55 (2014) 378–391

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

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

Oppositional Real Coded Chemical Reaction Optimization for different economic dispatch problems

Kuntal Bhattacharjee [a,1], Aniruddha Bhattacharya [b,*], Sunita Halder nee Dey [c,2]

[a] B.C. Roy Engineering College, Durgapur, West Bengal 713206, India
[b] National Institute of Technology-Agartala, Tripura 799055, India
[c] Department of Electrical Engineering, Jadavpur University, Kolkata, West Bengal 700 032, India

ARTICLE INFO

Article history:
Received 8 June 2013
Received in revised form 18 September 2013
Accepted 25 September 2013

Keywords:
Chemical reaction
Economic Load Dispatch
Optimization
Oppositional Real Coded Chemical Reaction Optimization
Real Coded Chemical Reaction Optimization

ABSTRACT

This paper proposes an effective oppositional Real Coded Chemical Reaction algorithm (ORCCRO) to solve Economic Load Dispatch (ELD) problems involving different equality and inequality constraints. Effects of valve-point loading, multi-fuel options of large-scale thermal plants are also studied. System transmission loss has also been considered in few cases. Chemical Reaction Optimization (CRO) imitates the interaction of molecules in a chemical reaction to reach from a higher energy unstable state to a low energy stable state. A real coded version of it, known as Real-coded chemical reaction optimization (RCCRO). Oppositional based RCCRO (ORCCRO) have been used here to improve the effectiveness and quality of solutions in minimum time. The proposed opposition-based RCCRO (ORCCRO) of the present work employs opposition-based learning for population initialization and also for generation wise update operation. In the present work, quasi-opposite numbers have been utilized instead of pseudo random numbers to improve the convergence rate of the RCCRO. Simulation results establish that the proposed approach outperforms several other existing optimization techniques in terms quality of solution obtained and computational efficiency. Results also prove the robustness of the proposed methodology to solve ELD problems.

1. Introduction

The method Economic Load Dispatch determines the most efficient, reliable and low cost operation of a power system by dispatching the power generation resources to supply the load on the scheme. To minimize the total cost of generation while satisfying the operational constraints is its main objective. Moreover the use of highly nonlinear fuel cost characteristics of modern thermal power plants, the practical Economic Load Dispatch problem contains many local optimum solutions and need to consider a huge number of complex constraints. Therefore, the classical calculus-based methods [1] are unable to perform very well in solving ELD problems, as these techniques need smooth, differentiable objective function. Though Linear programming method [2] is fast and reliable it has some drawbacks related with the piecewise linear cost approximation. Therefore, Dynamic Programming (DP) approach was proposed by Wood and Wollenberg [3] to solve ELD problems. Though this technique does not impose any restriction on the nature of the cost curves, but suffers from dimensionality and larger simulation time.

In recent years, several attempts have been made to solve ELD with useful and effective techniques, such as genetic algorithm (GA) [4], evolutionary programming (EP) [5], simulated annealing (SA) [6], particle swarm optimization (PSO) [7], Ant Colony Optimization [8], Differential Evolution (DE) [9], Artificial Immune System (AIS) [10], Bacterial Foraging Algorithm (BFA) [11], Biogeography-based Optimization (BBO) [12] etc.

The SA method is usually slower than the GA method since the GA has parallel search capabilities, which imitate natural genetic operations. However, the limitation of GA of getting attentive in local minima and high computational time forced the researchers to search for more efficient optimization techniques. PSO inspired by social behaviour of bird flocking population based optimization and is computationally faster than GA and also required less memory for its implementation. A closer examination on the operation of PSO indicates that once inside the optimum region, the algorithm process get slower due to its inability to adjust the velocity step size to continue the search at an optimum grain. So for multi-modal function, particles sometimes fail to reach global optimal point. DE has been found to yield better and faster solution, satisfying all the constraints, both for uni-modal and multi-modal system, using its different crossover strategies. But when system complexity and size increases, DE method is unable to map its entire

* Corresponding author. Tel.: +91 9474188660.
E-mail addresses: kunti_10@yahoo.com (K. Bhattacharjee), ani_bhatta2004@rediffmail.com (A. Bhattacharya), sunitaju@yahoo.com (S.H. Dey).
[1] Tel.: +91 9832832822.
[2] Tel.: +91 9836315191.

0142-0615/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijepes.2013.09.033

unknown variables together in a better way. In DE all variables are changed together during the crossover operation. The individual variable is not tuned separately. So in starting stage, the solutions moves very fast towards the optimal point but at later stage when fine tuning operation is required, DE is unsuccessful to give better performance. Due to increase in number of operations, and larger size of population, convergence speed of AIS is much slower than DE or PSO. The optimization methodologies [11,12] have been developed to solve ED problem, the complexity of the task reveals the necessity for development of efficient algorithms to accurately locate the optimum solution. Moreover, exploration ability of BBO is excellent, but exploitation ability is not very significant. These methods do not always guarantee global best solutions; rather they often achieve a near global optimal solution.

Recently, different hybridization and modification of GA, EP, PSO, DE, BBO like improved GA with multiplier updating (IGA-MU) [13], directional search genetic algorithm (DSGA) [14], improved fast evolutionary programming (IFEP) [15], new PSO with local random search (NPSO_LRS) [16], adaptive PSO (APSO) [17], self-organizing hierarchical PSO (SOH-PSO) [18], improved coordinated aggregation based PSO (ICA-PSO) [19], shuffled DE (SDE) [20], DE with generator of chaos sequences and sequential quadratic programming (DEC-SQP) [21], variable scaling hybrid differential evolution (VSHDE) [22], bacterial foraging with Nelder–Mead algorithm (BF-NM) [23], hybrid differential evolution with biogeography-based optimization (DE/BBO) [24] etc. have been adopted to solve different types of ELD problems.

Evolutionary algorithms, swarm intelligence and bacterial foraging are all population based bio-inspired algorithm. However, the common disadvantages of these algorithms are complicated computation, using many parameters. For that reason it is also difficult to understand these algorithms for beginners.

In recent times, a new optimization technique based on the concept of chemical reaction, called chemical reaction optimization (CRO) has been proposed by Lam and Li [25]. In a chemical reaction, the molecules of initial reactants stay in high-energy unstable states and undergo a sequence of collisions either with walls of the container or with other molecules. The reactants pass through some energy barriers, reach in low-energy stable states and become the final products. CRO captures this phenomenon of driving high-energy molecules to stable, low energy states, through various types of on-wall or inter-molecular reactions. CRO has been proved to be a successful optimization algorithm in discrete optimization.

Basically, the CRO is designed to work in the discrete domain optimization problems. To make this newly developed technique suitable for continuous optimization domain, Lam et al. [26] has developed a real-coded version of CRO, known as real-coded CRO (RCCRO). RCCRO involves 4 numbers of steps. These are On-Wall Ineffective Collision, Decomposition, Intermolecular Ineffective Collision and Synthesis. However, all these steps are not executed in each iteration simultaneously. First the algorithm will check randomly whether an elementary reaction which will be performed that is unimolecular or intermolecular. If it is a unimolecular reaction then it will check the decomposition criteria. If decomposition criteria will be satisfied then decomposition will be performed otherwise on wall ineffective collision will be performed. Instead of unimolecular reaction if intermolecular reaction will be selected, the algorithm will check whether synthesis criteria is satisfied or not. If synthesis criteria will be satisfied then synthesis will be performed otherwise intermolecular ineffective collision will be performed. That means at any single iteration any one of the 4 steps will be executed either to explore the search space or to exploit the previously developed best solution to find much better solution. Therefore, number of steps executed in any single iteration is comparatively less. To increase the scope of searching in other regions, the molecule splits into two (or more) molecules during decomposition. A molecule with too little *KE* lacks the ability to transform to a new molecule with higher function value and gets stuck to a local minimum. When two (or more) such molecules collide, synthesis takes place and results in a single molecule with a solution far away from the original solutions. The resultant molecule can have higher *KE* due to the combination of energy from multiple molecules. It allows the exploration of a new region of the solution space. In whole "life cycle" of a molecule, it searches a region of the solution space for a certain period and then jumps to another region to continue the search. This process can repeat since the excessive energy of some molecules is recycled through *buffer*. If the searching time is not restricted, CRO can explore every possible region of the solution space and eventually find the global minimum. Moreover, the steps of elementary reactions are very simple in case of RCCRO.

As RCCRO have both good exploration and exploitation ability, therefore it can reach to optimal solution within very small number of iterations. So, total simulation time required by RCCRO to reach to optimal solution for any test system is quite less. It has been observed that the performance of RCCRO is quite superior compared to many previously developed soft computing techniques, when applied to solve continuous benchmark optimization problems.

Opposition-Based Learning (OBL) was proposed by Tizhoosh in [27]. OBL was first utilized to improve learning and back propagation in neural networks by Ventresca and Tizhoosh [28], and since then, it has been applied to many EAs, such as DE [29], PSO by Wang et al. [30], and ant colony optimization by Malisia [31]. OBL maps this theory to machine learning and proposes to use opposite instead of random numbers to evolve the population quickly. The main principle of OBL is to utilize opposite numbers to approach the solution. The inventors of OBL claim that a number's opposite is probably closer than a random number to a solution. Thus, by comparing a number to its opposite, a smaller search space is needed to converge to the right solution. Simon et al. [32] proved that a quasi-opposite number is usually closer than a random number to the solution. It has also been proven that a quasi-opposite number is usually closer than an opposite number to the solution. The improved computational efficiency of quasi-opposition based learning concept has motivated the present authors to incorporate this concept in RCCRO (ORCCRO) to accelerate the convergence speed of RCCRO to a larger extent by comparing the fitness of a solution estimate to its opposite and keeping the fitter one in the randomly selected population set. This newly developed algorithm is applied to solve different non-convex complex ELD problems in search for superior quality solutions in a computationally efficient way.

Section 2 of the paper provides a brief description and mathematical formulation of different types of ELD problems. Section 3 describes the proposed RCCRO algorithm shortly. Section 4 designs the oppositional based learning technique and a short description of the ORCCRO algorithm and it used in ELD problems. Simulation studies are presented and discussed in Section 5. The conclusion is drawn in Section 6.

2. Mathematical modeling of the ELD problem

Four different types of ELD problems have been formulated and solved by ORCCRO approach. These are:

2.1. ELD with valve-point effects and transmission loss

The overall objective function F_T of ELD problem considering valve-point effect [33] may be written as

$$F_T = \left(\sum_{i=1}^{N} Fi(Pi)\right) = \left(\sum_{i=1}^{N} ai + biPi + ciPi^2 + |\, ei \times \sin\{fi \times (Pi^{\min} - Pi)\}\,|\right) \tag{1}$$

where, $F_i(P_i)$, is cost function of the ith generator, and is usually expressed as a quadratic polynomial; a_i, b_i and c_i are the cost coefficients of the ith generator, e_i and f_i are the coefficients of the ith generator reflecting the valve-point effects; N is the number of committed generators; P_i is the power output of the ith generator. The ELD problem consists in minimizing F_T subject to following constraints:

2.1.1. Real power balance constraint

$$\sum_{i=1}^{N} P_i - (P_D + P_L) = 0 \tag{2}$$

where, P_L is the total transmission loss; P_D is the total system active power demand. Calculation of P_L using the B-matrix loss coefficients is expressed as:

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \tag{3}$$

2.1.2. The generating capacity constraint

The power generated by each generator shall be within their lower limit $Pi^{\min}$ and upper limit $Pi^{\max}$. So that

$$P_i^{\min} \leqslant P_i \leqslant P_i^{\max} \tag{4}$$

where $P_i^{\min}$ and $P_i^{\max}$ are the minimum and the maximum power outputs of the ith unit.

2.2. ELD with valve-point effects and transmission loss

The objective function of this type of ELD problem is same as mentioned in (1). The objective function F_T is to be minimized subject to the constraints of (2), (4). Transmission loss is considered here and P_L can be find out using the B-matrix loss coefficients which is expressed in (3).

2.3. ELD with quadratic cost function

The overall objective function F_T of ELD problem in this case, may be written as

$$F_T = \min \sum_{i=1}^{N} F_i(P_i) = \min \sum_{i=1}^{N} (a_i + b_i P_i + c_i P_i^2) \tag{5}$$

where, $F_i(P_i)$, is cost function of the ith generator, and is usually expressed as a quadratic polynomial; a_i, b_i and c_i are the cost coefficients of the ith generator; N is the number of committed generators; P_i is the power output of the ith generator. The objective function F_T is to be minimized subject to the constraints of (2), (4). Transmission loss is not considered.

2.4. ELD with non-smooth cost functions with multiple fuels and valve-point effects

For a power system with N generators and n_F fuel options for each unit, the cost function of the generator with valve-point loading is expressed as:

$$F_{ip}(P_i) = a_{ip} + b_{ip} P_i + c_{ip} P_i^2 + |e_{ip} \times Sin\{f_{ip} \times (P_{ip}^{\min} - P_i)\}|$$

$$\text{if } P_{ip}^{\min} \leqslant P_i \leqslant P_{ip}^{\max} \text{for fuel option } p; p = 1, 2, \ldots, n_F \tag{6}$$

where, $Pip^{\min}$ and $Pip^{\max}$ are the minimum and maximum power generation limits of ith generator with fuel option p, respectively; aip, bip, cip, eip and fip are the fuel-cost coefficients of ith generator for fuel option p. Considering N numbers of generators, the objective function is to be minimized subject to the constraints of (2), (4), without transmission loss.

2.5. Calculation for slack generator

Let N committed generating units deliver their power output subject to the power balance constraint (2) and the respective capacity constraints of (4). Assuming the power loadings of first $(N-1)$ generators are known, the power level of Nth generator (Slack Generator) is given by

2.5.1. Without transmission loss

$$P_N = P_D - \sum_{i=1}^{(N-1)} P_i \tag{7}$$

2.5.2. With transmission loss

$$P_N = P_D + P_L - \sum_{i=1}^{(N-1)} P_i \tag{8}$$

Using Eqs. (3) and (8), the modified form of equation is:

$$B_{NN} P_N^2 + P_N \left(2\sum_{i=1}^{N-1} B_{Ni} P_i + \sum_{i=1}^{N-1} B_{0N} - 1\right) + \left(P_D + \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} P_I B_{ij} P_J + \sum_{i=1}^{N-1} B_{0i} P_i - \sum_{i=1}^{N-1} P_i + B00\right) = 0 \tag{9}$$

The solution procedure of (9) to calculate Nth generator output, P_N is same as mentioned in [24].

3. Real-Coded Chemical Reaction Algorithm

This section presents an interesting new optimization algorithm called chemical reaction optimization (CRO) which has been recently proposed in [25,26].

CRO loosely imitates what happens to molecules in a chemical reaction system. Every chemical reaction tends to release energy; therefore, products generally have less energy than the reactants. In terms of stability, the lower the energy of the substance, the more stable it is. In a chemical reaction, the initial reactants in the high-energy unstable states undergo a sequence of collisions, pass through some energy barriers, and become the final products in low-energy stable states. It is not difficult to discover the correspondence between optimization and chemical reaction. Both of them aim to seek the global optimum with respect to different objectives and the process evolves in a stepwise fashion. With this, the chemical-reaction-inspired meta-heuristic, called chemical reaction optimization (CRO) [25] has been developed by Lam et al. in 2010.

CRO has been already proved to be a successful optimization algorithm with different applications [25], most of which are discrete optimization problems. In order to make this optimization technique suitable for both continuous and discrete optimization problems, Lam et al. presented a modified version of CRO in 2012, which is termed as real-coded chemical reaction optimization (RCCRO) [26].

3.1. Major components of RCCRO

3.1.1. Molecules

The manipulated agents involved in a reaction are known as molecules. Three main properties of each molecule are: (1) the molecular structure X; (2) current potential energy (PE); (3) current kinetic energy (KE); and some optional attributes which can be used to construct other versions of CRO for particular problems. The meanings of the attributes in the profile are already provided in [26].

3.2. Elementary reactions

In CRO, numerous collisions occur. These collisions occur either between the molecules or between the molecules and the walls of the container. Depending upon the type of collisions, distinct elementary reactions occur. There are four types of elementary reactions. These are: (1) on-wall ineffective collision; (2) decomposition; (3) intermolecular ineffective collision; and (4) synthesis. In the term of optimization, different elementary reactions explore the solution space in search for better solutions. Different types of elementary reactions are briefly described below:

3.2.1. On wall ineffective collision

When a molecule hits a wall and bounces back, a small change occurs to its molecular structure and *PE*. As the collision is not so vigorous, the resultant molecular structure is not too different from the original one. If X and X' represents the molecular structure before and after the on-wall collision respectively, then on-wall ineffective collision tries to transform X to X', in the close neighbourhood of X, that is

$$X' = X + \Delta \tag{10}$$

where, Δ is a perturbation for the molecule. There are many probability distributions which can be used to produce probabilistic perturbations. In this paper, Gaussian distribution based mutation operation has been utilized, to transform X to X', in the close neighbourhood of X. By the change of molecular structure, *PE* and *KE* also change from PE_X to $PE_{X'}$ and KE_X to $KE_{X'}$.

This change will happen only if Eq. (11) is satisfied.

$$PE_X + KE_X \geqslant PE_{X'} \tag{11}$$

If Eq. (11) does not hold, the change is not allowed and the molecule retains its original X, *PE* and *KE*. Due to interaction with a wall of the container, a certain portion of molecules' *KE* will be extracted and stored in the central energy buffer (*buffer*) when the transformation is complete. The stored energy can be used to support decomposition. The size of *KE* loss depends on a random number $a1 \in [KELossRate, 1]$, where *KELossRate* is a parameter of CRO. Updated *KE* and *buffer* is represented as

$$KE_{X'} = (PE_X - PE_{X'} + KE_X) \times a1 \tag{12}$$

$$buffer = buffer + (PE_X + KE_X - PE_{X'}) \times (1 - a1) \tag{13}$$

3.2.2. Decomposition

In decomposition, one molecule hits the wall and breaks into two or more molecule e.g., X'_1 and X'_2. Due to change of molecular structure, their *PE* and *KE* also changes from PE_X to $PE_{X'_1}$ and $PE_{X'_2}$, and KE_X to $KE_{X'_1}$ and $PE_{X'_2}$. This change is allowed, if the original molecule has sufficient energy (*PE* and *KE*) to endow the *PE* of the resultant ones, that is

$$PE_X + KE_X \geqslant PE_{X'_1} + PE_{X'_2} \tag{14}$$

Let $temp1 = PE_X + KE_X - PE_{X'_1} - PE_{X'_2}$

Then,

$$KE_{X'_1} = k \times temp1 \text{ and } KE_{X'_2} = (1 - k) \times temp1 \tag{15}$$

where, k is a random number uniformly generated from the interval [0,1]. Eq. (14) holds only when KE_X is large enough. Due to the conservation of energy, X sometimes may not have enough energy (both *PE* and *KE*) to sustain its transformation into X'_1 and X'_2. To encourage decomposition, a certain portion of energy, stored in the central buffer (*buffer*) can be utilized to support the change. In that case modified condition is

$$PE_X + KE_X + buffer \geqslant PE_{X'_1} + PE_{X'_2} \tag{16}$$

The new values of *KE* for resultant molecules and *buffer* are

$$KE_{X'_1} = (temp1 + buffer) \times m1 \times m2 \tag{17}$$

$$KE_{X'_2} = (temp1 + buffer) \times m3 \times m4 \tag{18}$$

$$buffer = buffer + temp1 - KE_{X'_1} - KE_{X'_2} \tag{19}$$

where, values of *m1, m2, m3* and *m4* are random Nos. generated in between [0, 1]. To generate X'_1 and X'_2, any mechanism which creates X'_1 and X'_2 quite different from X, is acceptable. However, in this paper, to generate X'_1 and X'_2, the same procedure mentioned in section IIIB of [26] is followed.

3.2.3. Intermolecular ineffective collision

An intermolecular ineffective collision describes the situation when two molecules collide with each other and then bounce away. The effect of energy change of the molecules is similar to that in an on-wall ineffective collision, but unlike on-wall ineffective collision this elementary reaction involves more than one molecule and no *KE* is drawn to the central energy *buffer*. Similar to the on-wall ineffective collision, this collision is not vigorous; therefore the new molecular structures are generated in the neighbourhood of previous molecular structures. In this paper, new molecular structures are created using the same concept mentioned in on-wall ineffective collision. Suppose, the original molecular structures are X_1 and X_2 are transformed after collision and two new molecular structures are X'_1 and X'_2 respectively. The two *PE* are changed from PE_{X_1} and PE_{X_2} to $PE_{X'_1}$ and $PE_{X'_2}$. The two *KE* are changed from KE_{X1} and KE_{X_2} to $KE_{X'_1}$ and $KE_{X'_2}$

The changes to the molecules are acceptable only if

$$PE_{X_1} + PE_{X_2} + KE_{X_1} + KE_{X_2} \geqslant PE_{X'_1} + PE_{X'_2} \tag{20}$$

The new values of *KE* are calculated as

$$KE_{X'_1} = (PE_{X_1} + PE_{X_2} + KE_{X_1} + KE_{X_2} - PE_{X'_1} - PE_{X'_2}) \times aaa1 \tag{21}$$

$$KE_{X'_2} = (PE_{X_1} + PE_{X_2} + KE_{X_1} + KE_{X_2} - PE_{X'_1} - PE_{X'_2}) \times (1 - aaa1) \tag{22}$$

where, $aaa1$ is a random number uniformly generated in the interval [0, 1]. If the condition of Eq. (20) fails, the molecules maintain the original X_1, X_2, PE_{X_1}, PE_{X_2}, KE_{X1} and KE_{X_2} .

3.2.4. Synthesis

Synthesis is a process when two or more molecules (in present paper two molecules X_1andX_2)collide to each other and combine to form a single molecule X'. The change is vigorous and the resultant molecular structure X' is greatly different from X_1 and X_2 As in decomposition, any mechanism which combines two molecules to form a single molecule may be used. In this paper, procedure mentioned in section IIIB of [26] is used to create X'. The two *PE* are changed from PE_{X_1} and PE_{X_2} to $PE_{X'}$. The two *KE* are change from

KE_{X_1} and KE_{X_2} to $PE_{X'}$. The modification is acceptable if following condition holds

$$PE_{X_1} + PE_{X_2} + KE_{X_1} + KE_{X_2} \geqslant PE_{X'} \tag{23}$$

The new value of *KE* of the resultant molecule is

$$KE_{X'} = PE_{X_1} + PE_{X_2} + KE_{X_1} + KE_{X_2} - PE_{X'} \tag{24}$$

If condition of Eq. (23) is not satisfied, X_1, $X_2$and their related *PE* and *KE* are preserved, instead of X', $PE_{X'}$ and $KE_{X'}$. The pseudo codes for all above-mentioned elementary reaction steps are available in [25].

4. Opposition based learning

Opposition-based learning (OBL) is developed by Tizhoosh [27] to improve computational efficiency and to accelerate the convergence rate of different optimization techniques. OBL has been proposed to improve candidate solution by considering current population as well as its opposite population at the same time. Many researchers successfully applied this learning process into different soft computing techniques [34–36].

Here, opposite and quasi-opposite numbers are defined in one-dimensional space. These definitions can easily be extended to higher dimensions.

If x be any real number between $[qa, qb]$, its opposite number x_0, is defined as

$$x_0 = qa + qb - x \tag{25}$$

If x be any real number between $[qa,qb]$, Its quasi-opposite point, x_{qo} is defined as

$$x_{qo} = rand(qc, x_o) \tag{26}$$

where, qc is the centre of the interval $[qa, qb]$ and can be calculated as $(qa + qb)/2$ and $rand(qc, x_o)$ is a random number uniformly distributed between qc and x_0. The same logic can be applied to reflect the quasi-opposite point x_{qo}, and therefore to obtain its quasi-reflected point x_{qr}. If x be any real number between $[qa, qb]$. Then the quasi-reflected point, x_{qr} is defined as

$$x_{qr} = rand(qc, x) \tag{27}$$

where, $rand(qc, x)$ is a random number uniformly distributed between qc and x.

4.1. Sequential steps of ORCCRO algorithm

The stepwise ORCCRO are mentioned below:

(1) In initialization stage, configure the initial settings for the molecules and the parameters (i.e., *PopSize*, *KELossRate*, *MoleColl*, *buffer*, *InitialKE*, α, and β). Specify the No. of unknown variables (n), lower and upper bounds of unknown variables of the given problem.
(2) Create each molecule set, after generating all the unknown variables of the problem randomly within their effective lower and upper bounds, satisfying different constraints. Each molecule set represents a potential solution of the problem. Generate several molecule set to create Molecular matrix, size of which is ($PopSize \times n$). In a same way, create quasi-opposite molecular matrix (QOM) using Eq. (26) after satisfying all the feasible constraints of their upper and lower limit bounds.
(3) Calculate each *PEs* for molecule set and Quasi-opposite molecular set.
(4) Set a new *PopSize* size of *PEs* by comparing each *PEs* for molecule set and Quasi-opposite molecular set. Set their *KE* values as *InitialKE* from the initialization in step 2.
(5) During iterative process, first check the type of reaction to be held because one molecular collision held in an iteration. Create a random number $b \in [0, 1]$. If b is greater than *MoleColl* or there is only one molecule left, the next reaction is a uni-molecular reaction; else it is an intermolecular reaction.
(6) For each uni-molecular reaction, choose one molecule randomly and check whether it satisfies the decomposition criterion: (number of hits − minimum hit number) > α. where, α is the tolerance of duration for the molecule without obtaining any new local minimum solution. If so, perform decomposition steps; else perform on-wall ineffective collision steps. For decomposition if Eq. (14) or Eq. (16) is satisfied, modify *KE* and *buffer* using Eq. (15) or Eqs. (17)–(19)respectively. Similarly for on wall ineffective collision if Eq. (11) is satisfied then modify *KE* and *buffer* using Eqs. (14) and (13) respectively. For both the cases, modify the *PE* of each molecule set using their objective function value.
(7) For each intermolecular reaction, select two (or more) molecule sets randomly from the molecular matrix and test the synthesis criterion: ($KE \leqslant \beta$). Where, β is the minimum *KE* a molecule should have. If the condition is satisfied, perform the synthesis; otherwise, perform different steps of an intermolecular ineffective collision. For synthesis if Eq. (23) is satisfied, modify *KE* using Eq. (24). For intermolecular collision, if Eq. (20) is satisfied, modify *KE* using Eq. (21) and (24).
(8) Select a new parameter '*jumping rate*'(J_r) within [0, 1]. Form quasi-opposite molecule set (QOM) from the newly developed molecular set generated in previous steps as per following procedure:

```
if rand < J_r
  for i = 1: PopSize
    for j = 1:n
      QOM(i, j) = rand(qc(j), x_o);
    end
  end
end
```

Newly created quasi-opposite molecule set (*QOM*) must satisfy the feasible constraints of their lower and upper bounds.

(1) Calculate each *PEs* for molecule set and Quasi-opposite molecular set.
(2) Set a new *PopSize* size of *PEs* by comparing each *PEs* for molecule set and Quasi-opposite molecular set.
(3) If the maximum No. of iterations is reached or specified accuracy level is achieved, terminate the iterative process, otherwise go to step 5 for continuation.

4.2. ORCCRO algorithm for Economic Load Dispatch problem

In this subsection, the procedure to implement the ORCCRO algorithm for solving the ELD problems has been described. The flow chart of ORCCRO for solving ELD problem is shown in Fig. 1. The detail sequential steps of the ORCCRO algorithm applied to solve ELD problem are presented below:

(1) *Representation of the Molecular Structure X and Quasi-opposite Molecular Structure OX*: Since the assessment variables for ELD problem are real power output of the generators, they are used to represent the individual molecular structure.

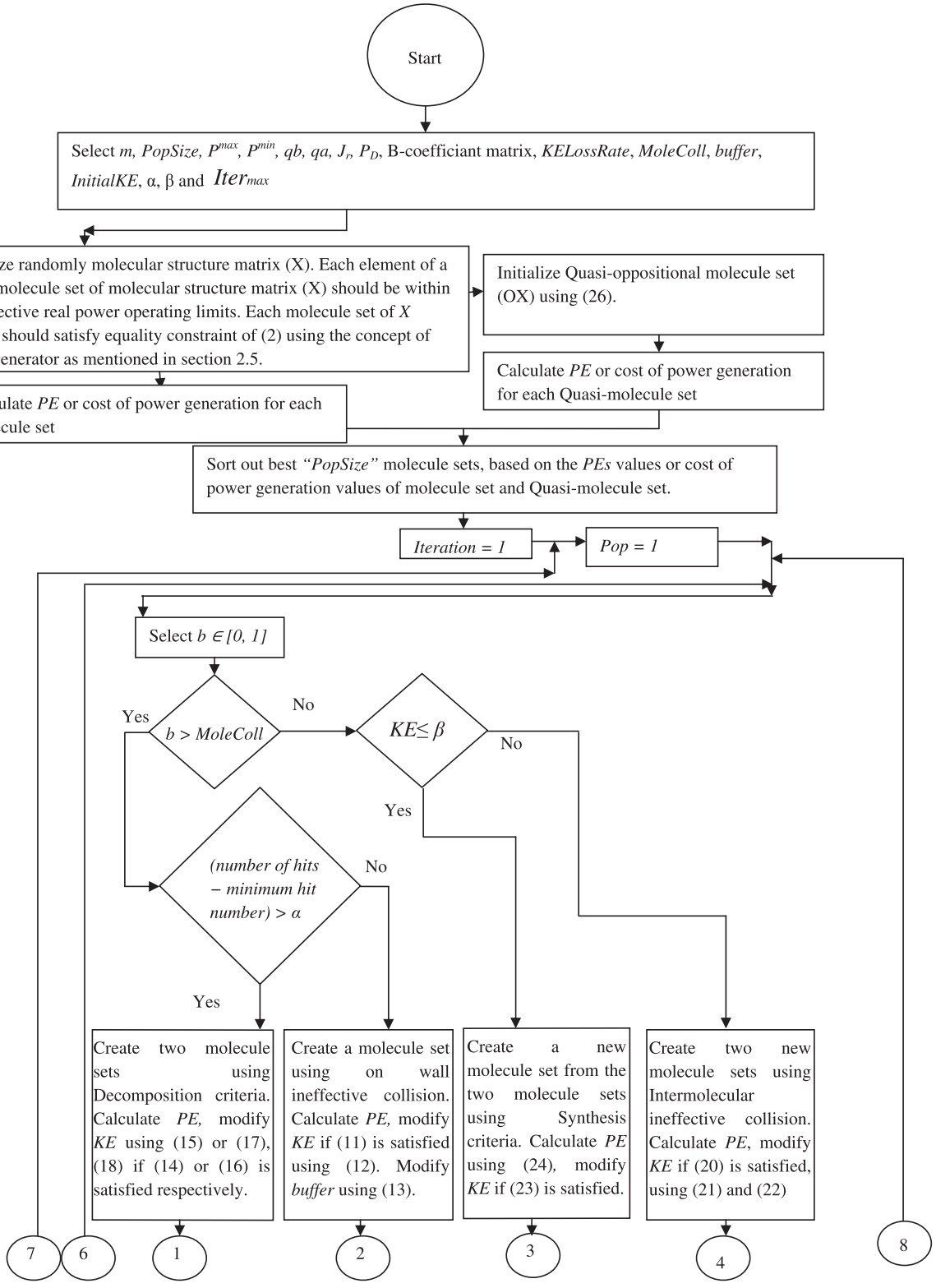

Fig. 1. Flow chart of ORCCRO applied in ELD problems.

Each individual element of the molecular structure represents the real power output of each generator. For initializations choose the number of generator units m and the total number of molecular structure, *PopSize*.

The complete molecular structure is represented in the form of the following matrix:

$$X = X_i = [X_1, X_2, X_3, \ldots X_{PopSize}] \quad \text{where } i = 1, 2, \ldots, PopSize$$

In case of ELD problem, each molecular set is presented as:

$$[X_{i,j}] = [X_{i,1}, X_{i,2}, \ldots, X_{i,m}] = [Pg_{i,j}] = [Pg_{i,1}, Pg_{i,2}, \ldots, Pg_{i,m}];$$

where $i = 1,2,\ldots,PopSize$; $j = 1,2,\ldots,m$. Each molecule set is one of the possible solutions for the ELD problem. The element X_{ij} of X_i is

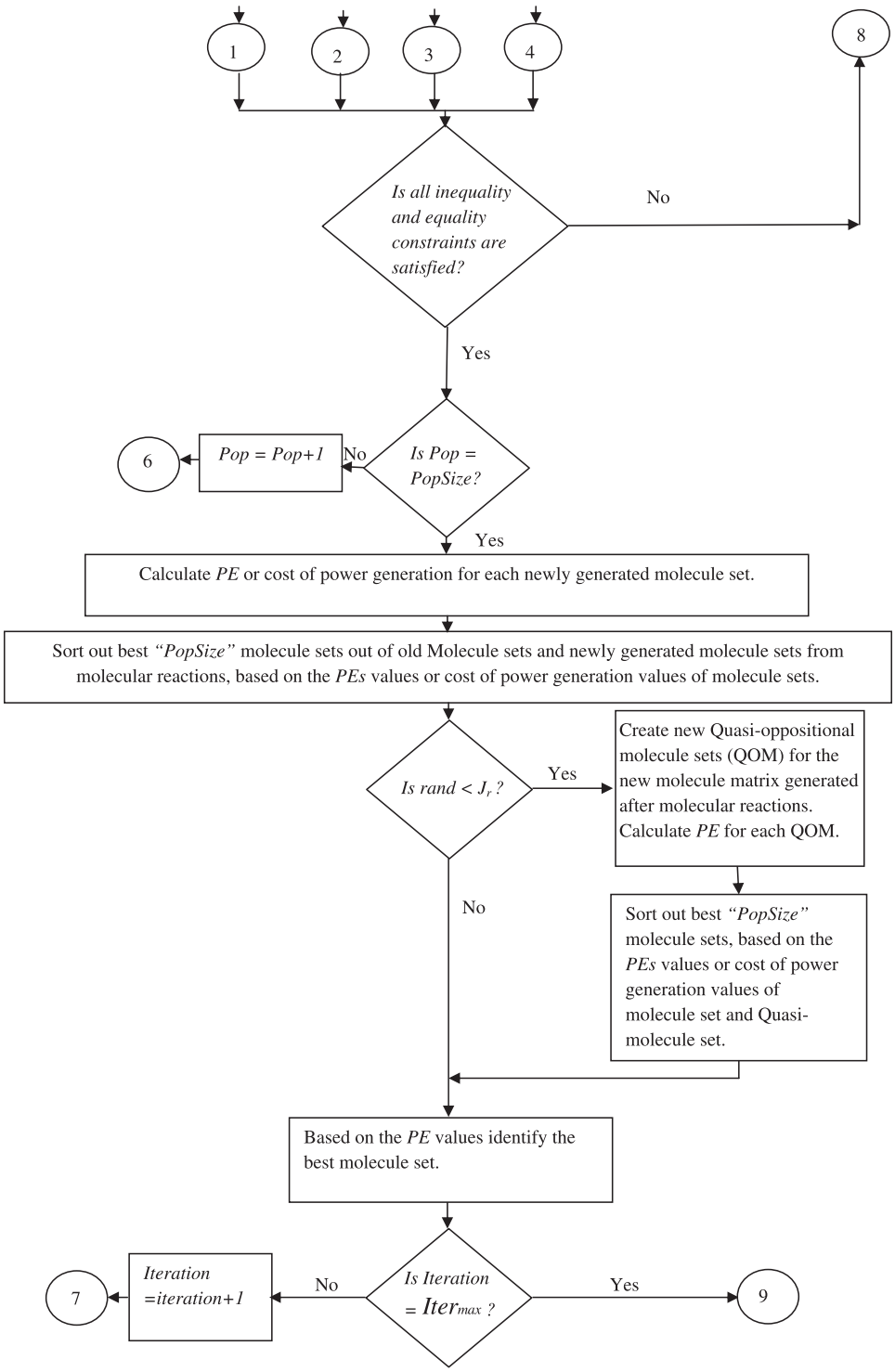

Fig. 1 (*continued*)

the *j*th position component of molecule set *i*. Each element of a given molecule set X_i is initialized randomly within the effective real power operating limits, based on Eq. (4) except last one i.e. *m*th element. The last element of each molecule set X_i is calculated solving Eq. (7) or Eq. (9). In a similar fashion, form Quasi-opposite molecular set using Eq. (26) while satisfying different constraints of Eq. (2) and Eq. (4).

(2) *Initialization of the Molecule set:* Each individual element of the Molecular structure matrix and Quasi-opposite molecular set, i.e. each element of a given molecule set *X* is initialized randomly within the effective real power operating limits, based on Eq. (4) except last one. The last element of each molecule set *X* is calculated solving Eq. (7) or Eq. (9). In a similar way each element of a given Quasi-opposite molecular set *OX* is initialized using Eq. (26) within the effective real power operating limits, based on Eq. (2) and (4).

(3) *Evaluation of PE:* In case of ELD problems, potential energy *PE*, of each molecule set is represented by the total fuel cost of generation for all the generators of that given molecule

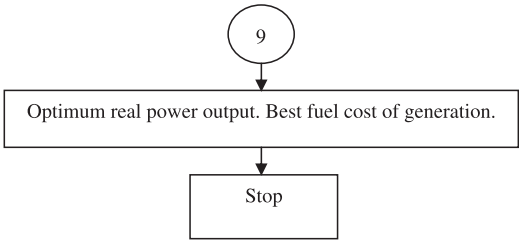

Fig. 1 (*continued*)

set. It is calculated using Eq. (5) for the system having quadratic fuel cost characteristic; using Eq. (1) for the system having valve-point effect; using Eq. (6) for the system having multi-fuel type fuel cost characteristic.

Steps of algorithm to solve ELD problems are given below.

Step (1) For initialization, choose No. of generator units, m: number of molecular structure set, *PopSize*; elitism parameter "*p*". Specify maximum and minimum capacity of each generator, *qb* and *qa* are assign to be the maximum and minimum capacity of each generator, J_r, power demand, *B*-coefficients matrix for calculation of transmission loss. Also initialize the RCCRO parameters like *KELossRate*, *MoleColl*, *buffer*, *InitialKE*, α, and β etc. Set maximum number of iterations, *Itermax*.
Step (2) Initialize each element of a given molecule set of *X* matrix and Quasi-opposite molecular set *OX* using the concept mentioned in "*Initialization of the Molecule set*".
Step (3) Calculate the *PE* value for each molecule set of the molecular matrix and Quasi-opposite molecular matrix for given initial Kinetic Energy (*KE*) i.e. *InitialKE*.
Step (4) Based on their PE_S values, sort out best *PopSize* sets of solution from the *PopSize* sets of molecule and *PopSize* sets of quasi-opposite molecular. Then create the new molecular matrix *X*.
Step (5) Create a random number $b \in [0, 1]$. If *b* is greater than *MoleColl* or there is only one molecule left (at the later stage of iterative procedure, this condition may hold), perform a unimolecular reaction, else perform an intermolecular reaction.
Step (6) If unimolecular reaction is selected, choose one molecule set randomly from the whole *X* matrix and check whether it satisfies the decomposition criterion.

If decomposition condition is satisfied, perform decomposition on that particular molecule set. Create two new molecule sets using the steps mentioned in section IIIB of [26]. Calculate *PE* of the new molecule sets. If the condition mentioned Eq. (14) or Eq. (16) is satisfied, modify *KE* of new molecule sets using Eq. (15) or Eq. (17) and (18). Modify *buffer* using Eq. (19).

If decomposition condition is not satisfied, perform on wall ineffective collision. Create two new molecule sets using the procedure mentioned in sub-Section 3.2.1. Calculate *PE* of the modified molecule set. If the condition mentioned in Eq. (11) is satisfied then modify *KE* of new molecule set using Eq. (12). Modify *buffer* using Eq. (13).

Step (7) From the condition of step 5, if intermolecular reaction is chosen, select two (or more) molecule sets randomly from the molecular matrix *X* and test the synthesis criterion ($KE \leqslant \beta$).

If the condition is satisfied, perform the synthesis steps. Create a new molecule set from the two selected molecule sets following the procedure given in section IIIB of [26]. Calculate *PE* of the new molecule set. After new molecule creation, if the condition of Eq. (23) is satisfied, modify *KE* of new molecule set using Eq. (24).

If synthesis condition ($KE \leqslant \beta$) is not satisfied, perform intermolecular collision. Create two new molecule sets in the neighbourhood of selected molecule sets following the procedure mentioned in sub-Section 3.2.1. Calculate *PE* of the new molecule set. After new molecule sets creation, if condition presented in Eq. (20) is satisfied, modify *KE* of new molecule sets using Eq. (21) and (24).

Step (8) Verify the feasibility of each newly generated molecule set of the modified X matrix, obtained after intermolecular or unimolecular reaction. Individual element of each modified molecule set must satisfy the generator operating limit constraint of Eq. (4). If some elements of a molecule set violate either upper or lower operating limits, then fix the values of those elements of the molecule set at the limit hit by them. Satisfy Real Power Balance constraint of Eq. (2) (in case of lossless system take PL = 0 in (2)) using the concept of slack generator as presented in Section 2.5. If output of slack generator does not meet generator operating limit constraint Eq. (4) discard that new molecule set, and reapply above-mentioned step-5 to step-7 on its old value (before any molecular reaction was performed), until all the constraints are satisfied.
Step (9) Quasi-opposite molecule set (QOM) can be formed from the newly developed molecular set generated in previous steps as per following procedure:

```
if rand < J_r
  for i = 1: PopSize
    for j = 1: m
      QOM(i, j) = rand(qc(j), x_o);
    end
  end
end
```

Newly created quasi-opposite molecule set (QOM) must satisfy the feasible constraints of Eq. (2) and (4). If some elements of a quasi-opposite molecule set violate either upper or lower operating limits, then fix the values of those elements of the quasi-opposite molecule set at the limit hit by them. Satisfy Real Power Balance constraint of Eq. (2) (in case of lossless system take $P_L = 0$ in Eq. (2)) using the concept of slack generator as presented in Section 2.5. If output of slack generator does not meet generator operating limit constraint Eq. (4) discard that new quasi-opposite molecule set, and reapply above-mentioned step-9 on its old value until all the constraints are satisfied.

Step (10) Recalculate the *PE* of each newly generated molecule set i.e. the fuel cost for each power output set of each newly generated molecule set and Quasi-opposite molecular set.
Step (11) Based on their *PE* values, sort out best *PopSize* sets of solution from the *PopSize* sets of molecular set and *PopSize* sets of quasi-opposite molecular set. Then create the new molecular matrix *X*.
Step (12) Based on the *PE* values identify the best molecule set. Here, best term is used to indicate that molecule set of generator power outputs, which give minimum fuel cost. If the best value of present iteration is superior to the best result up to last iteration, then best value of present iteration will be treated as global best solution and that will be stored in a different memory location for comparison in next iteration. Otherwise best result up to last iteration will be treated as global best solution

Table 1
Best power output for 13-generators system (P_D = 2520 MW).

Unit	Power outputs (MW)		Unit	Power outputs (MW)		Unit	Power outputs (MW)	
	ORCCRO	SDE [20]		ORCCRO	SDE[20]		ORCCRO	SDE[20]
1	628.32	628.32	6	159.73	159.73	11	112.14	113.12
2	299.20	299.20	7	159.73	159.73	12	92.40	92.40
3	299.20	299.20	8	159.73	159.73	13	92.40	92.40
4	159.73	159.73	9	159.73	159.73	Total power (MW)	2559.43	2560.43
5	159.73	159.73	10	77.40	77.40	Transmission loss (MW)	39.43	40.43
	Fuel cost ($/h)	24513.91	24514.88					

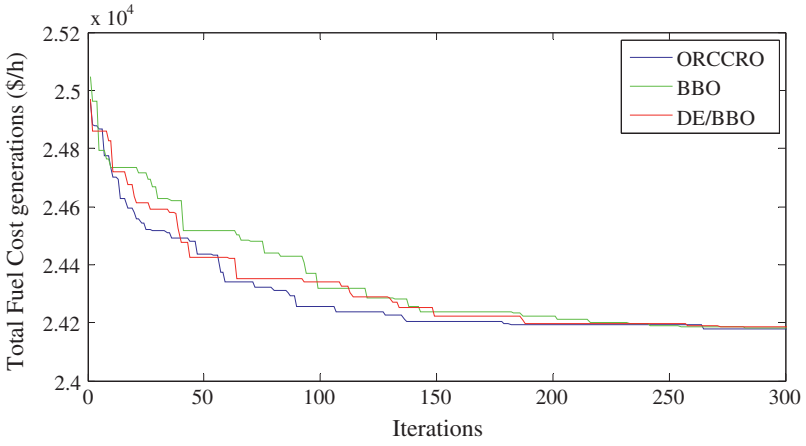

Fig. 2. Convergence characteristic of 13-generators system obtained by ORCCRO, BBO and DE/BBO.

Table 2
Comparison between different methods taken after 50 trials (13-generators system).

Methods	Generation Cost ($/h)			Time/Iteration (S)	No. of hits to minimum solution
	Max.	Min.	Average		
ORCCRO	24513.91	24513.91	24513.91	0.04	50
SDE[20]	NA	24514.88	24516.31	NA	NA
ICA-PSO[19]	24589.45	24540.06	24561.46	0.052	NA*
STHDE[20]	NA	24560.08	NA	NA	NA
BBO	24516.09	24515.21	24515.32	0.15	44
DE/BBO	24515.98	24514.97	24515.05	0.11	46

* NA:- Data Not Available

and that will be stored in that memory location for comparison in next iteration.
Step (13) Go to step 5 for the next iteration. Terminate the process after a predefined number of iterations, $Iter_{max}$.

5. Examples and simulation results

Proposed ORCCRO algorithm has been applied to solve ELD problems in four different test cases and its performance has been compared to several other optimization techniques like DE [20], BBO, DE/BBO, and PSO [7,19] etc. for verifying its feasibility.

5.1. Description of the test system

5.1.1. Test system 1

In this example, 13 generating units with valve-point effect has been considered. Transmission loss has been included in the problem. Power demand is 2520 MW and system data have been taken from [21]. Results obtained from proposed ORCCRO, SDE [20] and different versions of PSO [19], BBO, DE/BBO method have been presented here. Their best solutions are presented in Table 1. The convergence characteristic of the 13-generator systems in case of ORCCRO, BBO, DE/BBO is shown in Fig. 2. Minimum, average and maximum fuel costs obtained by ORCCRO and modified versions of PSO [19], SDE [20], BBO, DE/BBO over 50 trials are presented in Table 2.

5.1.2. Test system 2

A system with 40 generators with valve-point effect has been considered. The input data are available in [15]. The load demand is 10500 MW. Transmission loss has been considered here. The B-loss coefficients for the transmission losses of this system have been taken from the B-loss coefficients of the 6-generator test system [37], by multiplication on rows and columns up to 40 units. The result obtained using proposed ORCCRO method has been compared with BBO, DE/BBO, SDE [20] and GAAPI [20]. Their best solutions are shown in Table 3. Convergence characteristic of the 40-generators system in case of ORCCRO, BBO, DE/BBO is shown in Fig. 3. Minimum, average and maximum fuel costs obtained by RCCRO, BBO, DE/BBO over 50 trials are presented in Table 4.

5.1.3. Test system 3

A 110 generators system having quadratic fuel cost characteristic is used here. The input data of the whole system are taken from

Table 3
Best power output for 40-generators system with transmission loss (P_D = 10500 MW).

Unit	Power outputs (MW)				
	ORCCRO	SDE[20]	GAAPI [20]	BBO	DE/BBO
P_1	111.68	110.06	114	112.54	111.04
P_2	112.16	112.41	114	113.22	113.71
P_3	119.98	120.00	120.00	119.51	118.64
P_4	182.18	188.72	190	188.37	189.49
P_5	87.28	85.91	97	90.41	86.32
P_6	139.85	140.00	140.00	139.05	139.88
P_7	298.15	250.19	300	294.97	299.86
P_8	286.89	290.68	300	299.18	285.42
P_9	293.38	300	300	296.46	296.29
P_{10}	279.34	282.01	205.25	279.89	285.07
P_{11}	162.35	180.82	226.3	160.15	164.69
P_{12}	94.12	168.74	204.72	96.74	94.00
P_{13}	486.44	469.96	346.48	484.04	486.30
P_{14}	487.02	484.17	434.32	483.32	480.70
P_{15}	483.39	487.73	431.34	483.77	480.66
P_{16}	484.51	482.30	440.22	483.30	485.05
P_{17}	494.22	499.64	500	490.83	487.94
P_{18}	489.48	411.32	500	492.19	491.09
P_{19}	512.20	510.47	550	511.28	511.79
P_{20}	513.13	542.04	550	521.55	544.89
P_{21}	543.85	544.81	550	526.42	528.92
P_{22}	548.00	550.00	550	538.30	540.58
P_{23}	521.21	550.00	550	534.74	524.98
P_{24}	525.01	528.16	550	521.20	524.12
P_{25}	529.84	524.16	550	526.14	534.49
P_{26}	540.04	539.10	550	544.43	529.15
P_{27}	12.59	10.00	11.44	11.51	10.51
P_{28}	10.06	10.37	11.56	10.21	10.00
P_{29}	10.79	10.00	11.42	10.71	10.00
P_{30}	89.70	96.10	97	88.28	90.06
P_{31}	189.59	185.33	190	189.84	189.82
P_{32}	189.96	189.54	190	189.94	187.69
P_{33}	187.61	189.96	190	189.13	189.97
P_{34}	198.91	199.90	200	198.07	199.83
P_{35}	199.98	196.25	200	199.92	199.93
P_{36}	165.68	185.85	200	194.35	163.03
P_{37}	109.98	109.72	110	109.43	109.85
P_{38}	109.82	110.00	110	109.56	109.26
P_{39}	109.88	95.71	110	109.62	109.60
P_{40}	548.50	532.47	550	527.82	543.23
Total Power(MW)	11458.75	11474.43	11545.06	11470	11457.83
Loss(MW)	958.75	974.43	1045.06	970.37	957.83
Fuel Cost($/h)	136855.19	138157.46	139864.96	137026.82	136950.77

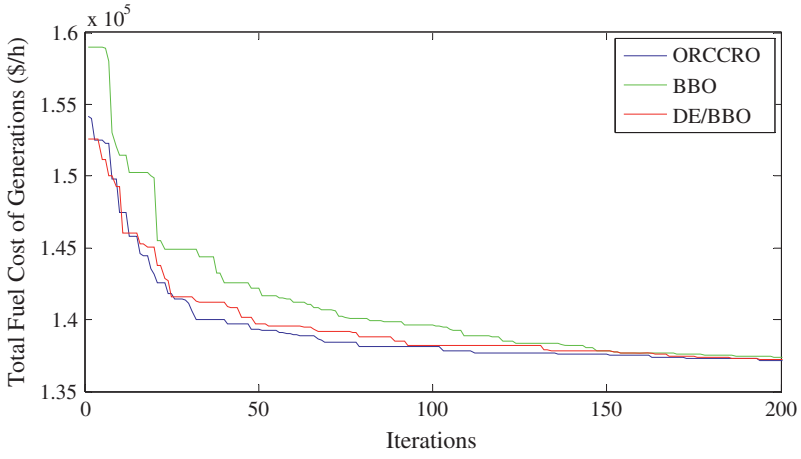

Fig. 3. Convergence characteristic of 40-generators system obtained by ORCCRO, BBO and DE/BBO.

[38]. The load demand is 15000 MW. The best results obtained by proposed ORCCRO is shown in Table 5. Out of 50 trials, minimum, maximum and average fuel cost obtained using ORCCRO, SAB [39], SAF [39], SA [39], BBO, DE/BBO are shown in Table 6. Convergence characteristic obtained using ORCCRO, BBO, DE/BBO is presented in Fig. 4.

Table 4
Comparison maximum, minimum and average value taken after 50 trials (40-generators system).

Methods	Generation Cost ($/h)			Time/iteration (S)	No. of hits to Min. solution
	Max.	Min.	Average		
ORCCRO	136855.19	136855.19	136855.19	0.07	50
BBO	137587.82	137026.82	137116.58	0.20	41
DE/BBO	137150.77	136950.77	136966.77	0.16	45

Table 5
Best power output for 110-generators system (pd = 15000mw).

Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)
P_1	2.40	P_{25}	400.00	P_{49}	8.40	P_{73}	96.85	P_{97}	3.63
P_2	2.42	P_{26}	400.00	P_{50}	8.40	P_{74}	201.25	P_{98}	3.67
P_3	2.42	P_{27}	500.00	P_{51}	8.64	P_{75}	89.99	P_{99}	4.41
P_4	2.44	P_{28}	500.00	P_{52}	12.06	P_{76}	49.99	P_{100}	4.41
P_5	2.44	P_{29}	200.00	P_{53}	12.00	P_{77}	161.01	P_{101}	10.00
P_6	4.01	P_{30}	99.54	P_{54}	12.00	P_{78}	284.23	P_{102}	10.05
P_7	4.02	P_{31}	10.00	P_{55}	12.00	P_{79}	183.91	P_{103}	20.11
P_8	4.01	P_{32}	19.99	P_{56}	25.24	P_{80}	107.53	P_{104}	20.00
P_9	4.00	P_{33}	79.72	P_{57}	25.93	P_{81}	10.07	P_{105}	40.01
P_{10}	66.59	P_{34}	249.79	P_{58}	35.07	P_{82}	12.05	P_{106}	40.01
P_{11}	59.25	P_{35}	359.98	P_{59}	35.08	P_{83}	21.95	P_{107}	50.01
P_{12}	30.23	P_{36}	400.00	P_{60}	45.11	P_{84}	199.22	P_{108}	30.03
P_{13}	52.47	P_{37}	39.41	P_{61}	45.10	P_{85}	324.52	P_{109}	40.06
P_{14}	25.00	P_{38}	69.73	P_{62}	45.18	P_{86}	439.98	P_{110}	20.02
P_{15}	25.14	P_{39}	99.99	P_{63}	184.96	P_{87}	32.33	Fuel Cost ($/h): −198016.29	
P_{16}	25.02	P_{40}	119.70	P_{64}	184.22	P_{88}	22.30		
P_{17}	154.91	P_{41}	157.90	P_{65}	184.64	P_{89}	86.67		
P_{18}	154.99	P_{42}	219.57	P_{66}	185.00	P_{90}	92.52		
P_{19}	154.52	P_{43}	439.98	P_{67}	70.01	P_{91}	59.67		
P_{20}	154.82	P_{44}	559.96	P_{68}	70.01	P_{92}	98.59		
P_{21}	68.93	P_{45}	660.00	P_{69}	70.01	P_{93}	439.98		
P_{22}	68.98	P_{46}	604.13	P_{70}	359.99	P_{94}	499.99		
P_{23}	69.00	P_{47}	5.44	P_{71}	399.96	P_{95}	599.95		
P_{24}	349.872545	P_{48}	5.40	P_{72}	399.87	P_{96}	462.47		

Table 6
Comparison between different methods taken after 50 trials (110-generators system).

Methods	Generation cost ($/h)			Time/iteration (S)	No. of hits to Min. Solution
	Max	Min	Average		
ORCCRO	198016.89	198016.29	198016.32	0.15	48
SAB[39]	NA	206912.9057	207764.73	NA	NA
SAF[39]	NA	207380.5164	207813.37	NA	NA
SA[39]	NA	198352.6413	201595.19	NA	NA
BBO	199102.59	198241.166	198413.45	0.52	41
DE/BBO	198828.57	198231.06	198326.66	0.46	43

5.1.4. Test system 4

A complex system with 160 thermal units with multiple fuel option and valve-point effect is considered here. The input data are available in [13] for 10 units system. For 160 thermal units input data are taken multiplication of each set of values of 10 units system up to 160 units. The system demand is 43200 MW. Transmission loss has not been included. The best result obtained using the proposed ORCCRO algorithm is shown in Table 7. Minimum, average and maximum fuel costs obtained by ORCCRO, ED-DE [40], and different GA [40] methods, BBO, DE/BBO over 50 trials are presented in Table 8. Convergence characteristic of the 160-generator systems obtained by ORCCRO, BBO, DE/BBO is shown in Fig. 5.

5.2. Tuning of parameters for ORCCRO algorithms

To get optimum solution using ORCCRO algorithm, it is necessary to get proper values of different parameter like, kinetic energy loss rate (*KELossRate*), initial kinetic energy (*InitialKE*) and β. Tuning of other ORCCRO parameters like *MoleColl*, α are also very important. For different values of these parameters, minimum fuel costs of generation are evaluated for 160 generators system. For a single value of one parameter, other parameters have been varied for their all possible combinations. Like the parameters of RCCRO jumping rate in OBL is also necessary to tuned to get better efficiency. As for example, when *InitialKE* = 2000; that time β has been varied from 100 to 1000 in suitable steps. At the same time for each value of β, α has been varied from 100 to 2000 in suitable steps. Similarly for each value of α, *MoleColl*, jumping rate(J_r) and *KELossRate* have been varied from 0.1 to 0.9.

However, to present all these results in a table, takes lots of space. Therefore, the detail tuning procedure is not presented here. A brief summarized result is only shown in Table 9.

Too large or small value of molecular structure size may not be capable to get the minimum value of fuel cost. For each molecular structure size (*PopSize*) of 20, 50, 100, 150 and 200, 50 trials have been run. Out of these, molecular structure size of, 50 achieves best fuel cost of generation for this system. For other molecular

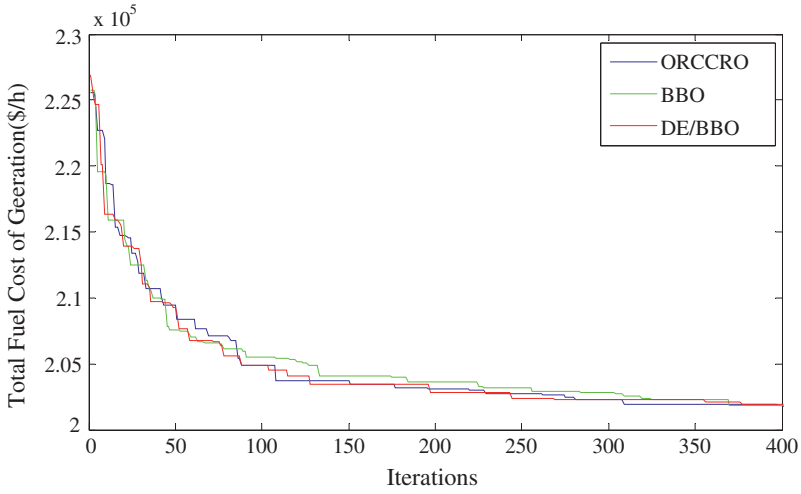

Fig. 4. Convergence characteristic of 110-generators system, obtained by ORCCRO, BBO and DE/BBO.

Table 7
Best power output For 160-generators system (Pd = 43200 mw).

Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)	Unit	Power output (MW)
P_1	213.38	P_{28}	240.09	P_{55}	281.85	P_{82}	210.35	P_{109}	418.50	P_{136}	235.67
P_2	213.62	P_{29}	407.28	P_{56}	235.04	P_{83}	294.06	P_{110}	263.68	P_{137}	301.49
P_3	278.42	P_{30}	255.35	P_{57}	280.11	P_{84}	242.61	P_{111}	214.95	P_{138}	236.29
P_4	238.25	P_{31}	220.97	P_{58}	238.20	P_{85}	286.39	P_{112}	215.10	P_{139}	429.15
P_5	285.43	P_{32}	206.99	P_{59}	436.27	P_{86}	238.89	P_{113}	271.56	P_{140}	266.64
P_6	237.47	P_{33}	285.68	P_{60}	292.89	P_{87}	283.02	P_{114}	235.99	P_{141}	212.69
P_7	309.78	P_{34}	238.10	P_{61}	221.02	P_{88}	244.65	P_{115}	268.60	P_{142}	201.24
P_8	240.27	P_{35}	289.87	P_{62}	209.27	P_{89}	427.17	P_{116}	235.06	P_{143}	277.65
P_9	439.97	P_{36}	241.42	P_{63}	269.63	P_{90}	287.12	P_{117}	277.26	P_{144}	237.23
P_{10}	278.96	P_{37}	293.11	P_{64}	237.11	P_{91}	216.22	P_{118}	240.10	P_{145}	291.73
P_{11}	222.15	P_{38}	241.34	P_{65}	287.60	P_{92}	217.68	P_{119}	439.84	P_{146}	237.83
P_{12}	207.50	P_{39}	420.91	P_{66}	237.54	P_{93}	284.48	P_{120}	282.39	P_{147}	282.03
P_{13}	274.71	P_{40}	277.08	P_{67}	271.21	P_{94}	240.78	P_{121}	217.82	P_{148}	238.07
P_{14}	237.41	P_{41}	222. 50	P_{68}	242.82	P_{95}	274.68	P_{122}	203.02	P_{149}	431.23
P_{15}	283.18	P_{42}	211.20	P_{69}	433.10	P_{96}	234.35	P_{123}	273.98	P_{150}	272.55
P_{16}	248.50	P_{43}	286.51	P_{70}	284.49	P_{97}	294.49	P_{124}	236.98	P_{151}	218.92
P_{17}	296.18	P_{44}	238.77	P_{71}	227.81	P_{98}	240.97	P_{125}	276.88	P_{152}	214.83
P_{18}	245.17	P_{45}	273.00	P_{72}	213.96	P_{99}	420.14	P_{126}	239.41	P_{153}	270.33
P_{19}	437.55	P_{46}	245.42	P_{73}	284.05	P_{100}	259.16	P_{127}	291.85	P_{154}	238.47
P_{20}	283.07	P_{47}	285.78	P_{74}	243.00	P_{101}	223.60	P_{128}	239.46	P_{155}	270.56
P_{21}	223.13	P_{48}	244.94	P_{75}	267.61	P_{102}	208.03	P_{129}	408.90	P_{156}	239.09
P_{22}	202.76	P_{49}	423.01	P_{76}	241.35	P_{103}	283.88	P_{130}	273.39	P_{157}	305.77
P_{23}	266.03	P_{50}	279.38	P_{77}	282.57	P_{104}	242.57	P_{131}	217.84	P_{158}	235.82
P_{24}	244.64	P_{51}	212.08	P_{78}	240.35	P_{105}	281.41	P_{132}	216.23	P_{159}	435.81
P_{25}	264.39	P_{52}	212.06	P_{79}	439.70	P_{106}	241.23	P_{133}	295.57	P_{160}	283.46
P_{26}	244.32	P_{53}	282.31	P_{80}	273.19	P_{107}	293.16	P_{134}	236.97	Fuel cost ($/h):	
P_{27}	284.85	P_{54}	242.06	P_{81}	223.43	P_{108}	240.61	P_{135}	258.96	−10004.20	

Table 8
Comparison between different methods taken after 50 trials (160-generators system).

Methods	Generation cost ($/h)			Time/iteration (S)	No. of hits to Min solution
	Max	Min	Average		
ORCCRO	10004.45	10004.20	10004.21	0.019	48
ED-DE[40]	NA	10012.68	NA	NA	NA
CGA-MU[40]	NA	10143.73	NA	NA	NA
IGA-MU[40]	NA	10042.47	NA	NA	NA
BBO	10010.59	10008.71	10009.16	0.62	40
DE/BBO	10010.26	10007.05	10007.56	0.56	42

structure size, no significant improvement of fuel cost has been observed. Moreover, beyond *PopSize* = 50, simulation time also increases. Best output obtained by ORCCRO algorithm for each molecular structure size is presented in Table 10.

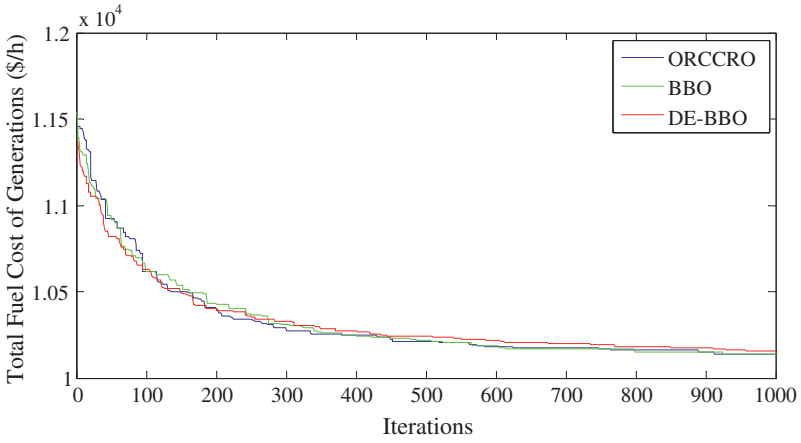

Fig. 5. Convergence characteristic of 160-generator system, obtained by ORCCRO, BBO and DE/BBO.

Table 9
Effect of molecular size on 160-generators system.

Molecular size Size	No. of hitsto Best Solution	Simulation time (S)	Max cost ($/h)	Min cost ($/h)	Average cost ($/h)
20	25	17.15	10006.13	10005.52	10005.83
50	48	19.21	10004.45	10004.20	10004.21
100	22	21.20	10005.76	10005.33	10005.57
150	15	28.20	10005.99	10005.58	10005.87
200	12	34.81	10006.25	10005.90	10006.17

Table 10
Effect of different parameters on performance of orccro (minimum fuel cost obtained for test case-4).

Initial KE	β	α	*MoleColl*	J_r	*KELossRate*					
					0.1	0.2	0.5	0.6	0.8	0.9
2000	1000	2000	0.9	0.9	10005.45	10004.82	10004.39	10004.38	10004.37	10004.41
1800	900	1500	0.8	0.8	10005.10	10004.78	10004.38	10004.37	10004.37	10004.37
1600	800	1300	0.75	0.75	10004.93	10004.66	10004.36	10004.35	10004.35	10004.35
1400	700	1000	0.70	0.7	10004.87	10004.44	10004.33	10004.33	10004.31	10004.34
1200	600	800	0.60	0.6	10004.79	10004.40	10004.32	10004.29	10004.29	10004.35
1000	500	600	0.50	0.5	10004.76	10004.37	10004.32	10004.26	10004.25	10004.33
800	400	400	0.40	0.4	10004.70	10004.34	10004.32	10004.25	10004.23	10004.31
600	300	300	0.30	0.3	10004.66	10004.32	10004.31	10004.25	10004.20	10004.31
400	200	200	0.20	0.2	10004.66	10004.33	10004.31	10004.30	10004.25	10004.31
200	100	100	0.10	0.1	10004.69	10004.33	10004.32	10009.31	10004.31	10004.31

Therefore, optimum values of these tuned parameters are *PopSize* = 50, *InitialKE* = 600, *KELossRate* = 0.8, β=300, *MoleColl* = 0.2, α = 300, J_r = *0.3*. Initial value of *buffer* = 0 is not selected using tuning procedure; rather its value is assumed based on the value presented in sub section IIC of [26].

5.2. Comparative study

5.2.1. Solution Quality

Tables 1, 3, 5 and 7 present the best fuel cost obtained by ORCCRO for 4 different test systems. These costs are better compared to the results obtained by many previously developed techniques specially recently developed techniques like BBO, DE/BBO. These are also shown in Tables 2, 4, 6 and 8. These tables also represent the comparative studies for maximum, minimum and average values, obtained by different algorithms. From the results it is clear that the performance of ORCCRO algorithm is better, in terms of quality of solutions obtained, compared to many already existing techniques.

5.2.2. Computational efficiency

Time taken by ORCCRO to achieve minimum fuel costs, is quite less compared to that obtained by BBO, DE/BBO and many other techniques. These are shown in Tables 2, 4, 6 and 8. These results prove significantly better computational efficiency of ORCCRO. Again for the use of oppositional based learning into RCCRO the systems convergence characteristics are become smooth and get converge within minimum time.

5.2.3. Robustness

Performance of any heuristic algorithms cannot be judged by the results of a single run. Normally their performance is judged after running the programs of those algorithms for certain number of trials. Many numbers of trials should be made to obtain a useful conclusion about the performance of the algorithm. An algorithm is said to be robust, if it gives consistent result during these trial runs. Tables 2, 4, 6 and 8 present that out of 50 numbers of trials for four different test systems. ORCCRO reaches to the minimum costs 50, 50, 48 and 48 times respectively. That means the

efficiency of ORCCRO algorithm to reach minimum solution is 100% and 96% respectively. On the other hand BBO, DE/BBO reach to the minimum costs (44, 41, 41, 40) and (46, 45, 43, 42) times respectively. Therefore performance of ORCCRO is much superior compared to BBO, DE/BBO and many other algorithms, presented in the different literatures.

Therefore, the above results establish the enhanced ability of ORCCRO to achieve superior quality solutions, in a computational efficient and robust way.

6. Conclusion

In this paper, a newly developed ORCCRO algorithm has been successfully implemented to solve different non-convex large scale ELD problems. Analyses of all the simulation results reveal that the performance of ORCCRO in all respect is better in comparison with the previously developed several optimization techniques. The ORCCRO achieves superior quality solutions near global solutions with high convergence speed and robustness compared to other methods. We also explored the effect of parameter setting and various numbers of molecular structures on the ORCCRO. Therefore, ORCCRO can be considered as one of the strongest tools to solve complex ELD problems. In future, ORCCRO can also be tried for solution of complex hydrothermal scheduling, dynamic ELD, optimal power flow problems in search for good characteristics results.

References

[1] El-Keib AA, Ma H, Hart JL. Environmentally constrained economic dispatch using the Lagrangian relaxation method. IEEE Trans Power Syst 1994;9(4):1723–9.
[2] Fanshel S, Lynes ES. Economic power generation using linear programming. IEEE Trans Power Apparatus Syst 1964;PAS-83(4):347–56.
[3] Wood J, Wollenberg BF. Power generation, operation, and control. 2nd ed. John Wiley and Sons; 1984.
[4] Walters DC, Sheble GB. Genetic algorithm solution of economic dispatch with valve point loadings. IEEE Trans Power Syst 1993;8(3):1325–31.
[5] Jayabharathi T, Jayaprakash K, Jeyakumar N, Raghunathan T. Evolutionary programming techniques for different kinds of economic dispatch problems. Elect Power Syst Res 2005;73(2):169–76.
[6] Panigrahi CK, Chattopadhyay PK, Chakrabarti RN, Basu M. Simulated annealing technique for dynamic economic dispatch. Electric Power Components Syst 2006;34(5):577–86.
[7] Gaing Zwe-Lee. Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 2003;18(3):1187–95.
[8] Hou, Y.H., Wu, Y.W., Lu, L.J., Xiong, X.Y., Generalized ant colony optimization for economic dispatch of power systems. In: Proceedings of international conference on power system technology, power-con; 2002, 13–17 October 2002, vol. 1, pp. 225–9.
[9] Nomana N, Iba H. Differential evolution for economic load dispatch problems. Electric Power Syst Res 2008;78(3):1133–322.
[10] Panigrahi BK, Yadav SR, Agrawal S, Tiwari MK. A clonal algorithm to solve economic load dispatch. Elect Power Syst Res 2007;77(10):1381–9.
[11] Panigrahi BK, Pandi VR. Bacterial foraging optimization: Nelder-Mead hybrid algorithm for economic load dispatch. IET Generation, Transm Distrib 2008;2(4):556–65.
[12] Bhattacharya A, Chattopadhyay PK. Biogeography-based optimization for different economic load dispatch problems. IEEE Trans Power Syst 2010;25(2):1064–77.
[13] Chiang CL. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. IEEE Trans Power Syst 2005;20(4):1690–9.
[14] Adhinarayanan, T., Sydulu, M., A directional search genetic algorithm to the economic dispatch problem with prohibited operating zones. In: Proc IEEE/PES Transmiss Distribut Conf Expo. Chicago 2008, IL, 21–24 April; 2008, vol. 1, pp. 1–5.
[15] Sinha N, Chakrabarti R, Chattopadhyay PK. Evolutionary programming techniques for economic load dispatch. IEEE Trans Evol Comput 2003;7(1):83–94.
[16] Selvakumar Immanuel, Thanushkodi K. A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE Trans Power Syst 2007;22(1):42–51.
[17] Panigrahi BK, Pandi VR, Das S. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. Energy Conver Manage 2008;49(6):1407–15.
[18] Chaturvedi KT, Pandit M, Srivastava L. Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. IEEE Trans Power Syst 2008;23(3):1079–87.
[19] Vlachogiannis JK, Lee KY. Economic load dispatch—a comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO. IEEE Trans Power Syst 2009;24(2):991–1001.
[20] Srinivasa Reddy A, Vaisakh K. Shuffled differential evolution for large scale economic dispatch. Electric Power Syst Res 2013;96:237–45.
[21] Coelho LDS, Mariani VC. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve- point effect. IEEE Trans Power Syst 2006;21(2):989–96.
[22] Chiou Ji-Pyng. Variable scaling hybrid differential evolution for large-scale economic dispatch problems. Electric Power Syst Res 2007;77(3–4):212–8.
[23] Panigrahi BK, Pandi VR. Bacterial foraging optimisation: Nelder–Mead hybrid algorithm for economic load dispatch. IET Gener Transm Distrib 2008;2(4):556–65.
[24] Bhattacharya A, Chattopadhyay PK. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. IEEE Trans Power Syst 2010;25(4):1955–64.
[25] Lam AYS, Li VOK. Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans Evol Comput 2010;14(3):381–99.
[26] Lam, Albert, Y.S., Li, Victor, O.K., Yu, James J.Q., 'Real-Coded Chemical Reaction Optimization', IEEE Trans. Evol. Comput., 2012, 16, (3), pp. 339–353.
[27] Tizhoosh H. Opposition-based learning: A new scheme for machine intelligence', In:Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation, Austria, 28th-30th Nov., 2005, pp. 695–701.
[28] Ventresca M, Tizhoosh H. Improving the convergence of back propagation by opposite transfer functions. IEEE International Joint Conference on Neural Networks, Vancouver, BC July 2006;16–21:9527–34.
[29] Rahnamayan S, Tizhoosh H, Salama M. Opposition-based differential evolution. IEEE Trans Evol Comput 2008;12(1):64–79.
[30] Wang H, Liu Y, Zeng S, Li H, Li C. Opposition-based partice swarm algorithm with Cauchy mutation. IEEE Congress Evol Comput, Singapore 2007;25–28:4750–6.
[31] Malisia AR. Investigating the application of opposition-based ideas to ant algorithms. Master's thesis. Ontario, Canada: University of Waterloo, Waterloo; 2007.
[32] Simon D, Ergezer M, Du D. Oppositional biogeography-based optimization. <http://embeddedlab.csuohio.edu/BBO/>.
[33] Happ HH. Optimal power dispatch-A comprehensive survey. IEEE Trans Power Apparatus Syst 1977;PAS-96(3):841–54.
[34] Haiping M, Xieyong R, Baogen J. Oppositional ant colony optimization algorithm and its application to fault monitoring. In: Proc 29th Chin Contr Conf; 2010. p. 3895–8.
[35] Shaw B, Mukherjee V, Ghoshal SP. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. Int J Elect Power Energy Syst 2012;35(1):21–33.
[36] Chatterjee A, Ghoshal SP, Mukherjee V. Solution of combined economic and emission dispatch problems of power systems by an opposition-based harmony search algorithm. Int J Elect Power Energy Syst 2012;39(1):9–20.
[37] Ciornei I, Kyriakides E. Efficient hybrid optimization solution for the economic dispatch with nonsmooth cost function. In: Proc IEEE Power Tech. Bucharest, Romania; June 28–July 2009, vol. 2, p. 1–7.
[38] Orero SO, Irving MR. Large scale unit commitment using a hybrid genetic algorithm. Elect Power Energy Syst 1997;19(1):45–55.
[39] Vishwakarma KK, Dubey HM. Simulated annealing based optimization for solving large scale economic load dispatch problems. Int J Eng Res Technol (IJERT) 2012;1(3):1–8.
[40] Wang Yu, Bin Li, Weise Thomas. Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems'. Inform Sci 2010;180(12):2405–20.