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