Energy Conversion and Management 49 (2008) 3036–3042

Contents lists available at ScienceDirect

Energy Conversion and Management

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

Modified differential evolution algorithm for optimal power flow with non-smooth cost functions

Samir Sayah *, Khaled Zehar

Department of Electrical Engineering, University of Ferhat Abbas, Setif 19000, Algeria

ARTICLE INFO

Article history:
Received 7 November 2007
Accepted 18 June 2008
Available online 3 August 2008

Keywords:
Evolutionary algorithms
Modified differential evolution
Non-smooth cost function
Optimal power flow
Power system optimization

ABSTRACT

Differential evolution (DE) is a simple but powerful evolutionary optimization algorithm with continually outperforming many of the already existing stochastic and direct search global optimization techniques. DE algorithm is a new optimization method that can handle non-differentiable, non-linear, and multimodal objective functions. This paper presents an efficient modified differential evolution (MDE) algorithm for solving optimal power flow (OPF) with non-smooth and non-convex generator fuel cost curves. Modifications in mutation rule are suggested to the original DE algorithm, that enhance its rate of convergence with a better solution quality. A six-bus and the IEEE 30 bus test systems with three different types of generator cost curves are used for testing and validation purposes. Simulation results demonstrate that MDE algorithm provides very remarkable results compared to those reported recently in the literature.

1. Introduction

Optimal power flow (OPF) is one of the main tools for optimal operation and planning of modern power systems. The OPF is, hence, the basic tool that allows electric utilities to determine secure and economic operating conditions for an electric power system. An OPF adjusts the controllable quantities in the system to optimize an objective function, while satisfying a set of physical and operational constraints. This makes the OPF problem a large-scale highly non-linear constrained optimization problem.

The OPF problem has been solved via many traditional optimization methods such as linear programming, non-linear programming, quadratic programming, Newton-based techniques and interior point methods. A comprehensive review of various optimization techniques available in the literature is reported in Refs. [1,2]. Usually, these methods rely on the assumption that the fuel cost characteristic of a generating unit is a smooth, convex function. However, there are situations where it is not possible, or appropriate, to represent the unit's fuel cost characteristic as a convex function. For example, this situation arises when valve-points, unit prohibited operating zones, or multiple fuels are present. Hence, the true global optimum of the problem could not be reached easily. New numerical methods are then needed to cope with these difficulties, specially, those with high speed search to the optimal and not being trapped in local minima.

In recent years, many heuristic algorithms such as genetic algorithms (GA) [3,4], evolutionary programming (EP) [5,6], tabu search (TS) [7], particle swarm optimization (PSO) [8] and simulated annealing (SA) [9], have been proposed to solve the OPF problem, without any restrictions on the shape of the cost curves. The results reported were promising and encouraging for further research in this direction.

Recently, a new evolutionary computation technique, called differential evolution (DE), has been developed and introduced by Storn and Price [10]. DE algorithm is a stochastic population-based search method successfully applied in global optimization problems. DE combines simple arithmetic operators with the classical operators of crossover, mutation and selection to evolve from a randomly generated starting population to a final solution [11,12].

This paper presents an efficient modified differential evolution (MDE) algorithm for solving optimal power flow (OPF) with non-smooth cost functions. Modifications in mutation rule are suggested to the original DE algorithm that explores the solution space with a random localisation, enhancing its rate of convergence for a better solution quality. In order to demonstrate the suitability of the proposed approach, MDE algorithm was applied to the six-bus and IEEE 30 bus test systems with three different types of generator cost curves. Simulation results demonstrate that MDE algorithm is superior to the original DE and appears to be fast providing very remarkable results compared to those reported in the literature recently.

2. Optimal power flow problem formulation

The OPF problem is considered as a general minimization problem with constraints, and can be written in the following form:

* Corresponding author. Tel.: +213 90 74 77 63; fax: +213 36 92 72 10.
E-mail address: samir_pg04@yahoo.fr (S. Sayah).

0196-8904/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enconman.2008.06.014

$$\text{Minimize} \quad f(x,u) \tag{1}$$
$$\text{Subject to}: \quad g(x,u) = 0 \tag{2}$$
$$h(x,u) \leqslant 0 \tag{3}$$

where $f(x,u)$ is the objective function, $g(x,u)$ is the equality constraints and represent typical load flow equations. $h(x,u)$ is the system operating constraints. x is the vector of state variables consisting of slack bus real power P_{G1}, load bus voltages V_L, generator reactive power outputs Q_G, and transmission line loading S_l. Therefore, x can be expressed as:

$$x^T = [P_{G1}, V_{L1} \dots V_{LNL}, Q_{G1} \dots Q_{GNG}, S_{l1} \dots S_{lNB}] \tag{4}$$

where NL, NG and NB are the number of load buses, the number of generators and the number of transmission lines, respectively.

U is the vector of control variables consisting of real power outputs P_G except at the slack bus, generator voltages V_G, transformer tap settings T. Hence, u can be expressed as:

$$U^T = [P_{G2} \dots P_{GNG}, V_{G1} \dots V_{GNG}, T_1 \dots T_{NT}] \tag{5}$$

where NT is the number of regulating transformers.

The objective function for the OPF reflects the cost associated with generating power in the system. The objective function for the entire power system can then be written as the sum of the fuel cost model for each generator:

$$f = \sum_{i=1}^{NG} f_i(\$/h) \tag{6}$$

where f_i is the fuel cost of the ith generator.

The system operating constraints $h(x, u)$ include:

(1) Generation constraints:
For stable operation, generator voltages, real power outputs and reactive power outputs are restricted by the lower and upper limits as follows:

$$V_{Gi}^{\min} \leqslant V_{Gi} \leqslant V_{Gi}^{\max}, \quad i \in NG, \tag{7}$$
$$P_{Gi}^{\min} \leqslant P_{Gi} \leqslant P_{Gi}^{\max}, \quad i \in NG, \tag{8}$$
$$Q_{Gi}^{\min} \leqslant Q_{Gi} \leqslant Q_{Gi}^{\max}, \quad i \in NG. \tag{9}$$

(2) Transformer constraints:
Transformer tap settings are restricted by the minimum and maximum limits as follows:

$$T_i^{\min} \leqslant T_i \leqslant T_i^{\max}, \quad i \in NT. \tag{10}$$

(3) Security constraints:
These incorporate the constraints of voltage magnitudes of load buses as well as transmission line loadings as follows:

$$V_{Li}^{\min} \leqslant V_{Li} \leqslant V_{Li}^{\max}, \quad i \in NL, \tag{11}$$
$$S_{li} \leqslant S_{li}^{\max}, \quad i \in NB. \tag{12}$$

3. Overview of differential evolution algorithm

Differential evolution (DE) is a relatively recent heuristic technique designed to optimize problems over continuous domains [10,11]. In DE, each decision variable is represented in the chromosome (individual) by a real number. As in any other evolutionary algorithm, the initial population of DE is randomly generated, and then evaluated. After that, the selection process takes place. During the selection stage, three parents are chosen and they generate a single offspring which competes with a parent to determine which one passes to the following generation. DE generates a single offspring (instead of two like in the genetic algorithm) by adding the weighted difference vector between two parents to a third parent. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector to which it was compared.

An optimization task consisting of D parameters can be presented by a D-dimensional vector. In DE, a population of N_P solution vectors is randomly created at the start. This population is successfully improved over G generations by applying mutation, crossover and selection operators, to reach an optimal solution [10,11]. The main steps of the DE algorithm are given below:

Initialization
Evaluation
Repeat
  Mutation
  Crossover
  Evaluation
  Selection
Until *(Termination criteria are met)*

3.1. Initialization

Typically, each decision parameter in every vector of the initial population is assigned a randomly chosen value from within its corresponding feasible bounds:

$$X_{j,i}^{(0)} = X_j^{\min} + \mu_j (X_j^{\max} - X_j^{\min}), \quad i = 1, \dots, N_P, \quad j = 1, \dots, D \tag{13}$$

where μ_j denotes a uniformly distributed random number within the range [0,1], generated anew for each value of j. $X_j^{\max}$ and $X_j^{\min}$ are the upper and lower bounds of the jth decision parameter, respectively.

3.2. Mutation

The mutation operator creates mutant vectors X_i' by perturbing a randomly selected vector X_a with the difference of two other randomly selected vectors X_b and X_c, according to the following expression:

$$X_i'^{(G)} = X_a^{(G)} + F(X_b^{(G)} - X_c^{(G)}), \quad i = 1, \dots, N_P \tag{14}$$

where a, b, and c are randomly chosen indices, such that $a, b, c \in \{1,\dots,N_P\}$ and $a \neq b \neq c \neq i$. It should be noted that new (random) values for a, b, and c have to be generated for each value of i. The scaling factor F is an algorithm control parameter in the range [0,2] which is used to adjust the perturbation size in the mutation operator and improve algorithm convergence.

3.3. Crossover

In order to increase the diversity among the mutant parameter vectors, crossover is introduced. To this end, a trial vector X_i'' is created from the components of each mutant vector X_i' and its corresponding target vector X_i, based on a series of D-1 binomial experiments of the following form:

$$X_{j,i}''^{(G)} = \begin{cases} X_{j,i}'^{(G)} & \text{if } \rho_j \leqslant C_R \text{ or } j = q \\ X_{j,i}^{(G)} & \text{otherwise}, \end{cases} \quad i = 1, \dots, N_P, \; j = 1, \dots, D \tag{15}$$

where ρ_j denotes a uniformly distributed random number within the range [0,1), generated anew for each value of j. The crossover constant C_R which is usually chosen from within the range [0,1], is an algorithm parameter that controls the diversity of the population and aids the algorithm to escape from local minima. q is a ran-