Electrical Power and Energy Systems 33 (2011) 236–244

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

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

An investigation about the impact of the optimal reactive power dispatch solved by DE

Juan M. Ramirez [a,*], Juan M. Gonzalez [b], Tapia O. Ruben [c]

[a] Cinvestav-Guadalajara, Av. Científica 1145, Col El Bajio, Zapopan, Jal. 45015, Mexico
[b] Universidad Tecnológica de Manzanillo, Manzanillo, Mexico
[c] Universidad Politécnica de Tulancingo, Hidalgo, Mexico

ARTICLE INFO

Article history:
Received 7 October 2009
Received in revised form 18 June 2010
Accepted 13 August 2010

Keywords:
Optimization
Power system operation
Reactive power dispatch
Voltage stability

ABSTRACT

With the advent of new technology based on power electronics, power systems may attain better voltage profile. This implies the proposition of careful strategies to dispatch reactive power in order to take advantage of all reactive sources, depending on size, location, and availability. This paper proposes an optimal reactive power dispatch strategy taking care of the steady state voltage stability implications. Two power systems of the open publications are studied. Power flow analysis has been carried out, which are the initial conditions for Transient Stability (TS), Small Disturbance (SD), and Continuation Power Flow (CPF) studies. Steady state voltage stability analysis is used to verify the impact of the optimization strategy. To demonstrate the proposal, PV curves, eigenvalue analyses, and time domain simulations, are utilized.

1. Introduction

The problem of reactive power dispatch is generally bundled with the problem of maintaining load voltages within pre-specified limits. The generator voltage set-point values V_{Gi}^{ref} are optimized with respect to certain performance criteria subject to the reactive-power-balance constraints, the load voltage acceptable limits, the available limits on the generated reactive power, and the limits on voltage generators. The generation-based reactive power dispatch falls under the category of the optimal power flow (OPF).

Since transformer tap ratios and outputs of shunt capacitor/reactors have a discrete nature, while reactive power output generators, bus voltage magnitudes and angles are, on the other hand, continuous variables, the reactive power optimization problem is formulated as mixed-integer, nonlinear problem [1,2].

Algorithms based on the principles of natural evolution have been applied successfully to a set of numerical optimization problems. With a good degree of parallelism and stochastic characteristics, they are adequate for solving intricate optimization problems, such as those found in reactive optimization, distribution systems planning, expansion of transmission systems, and economic dispatch [3–9]. Publications present an extensive list of works concerning the application of evolutionary techniques to power systems issues [10,21,22]. In general, these applications concentrate primarily on power system planning, followed by distribution systems.

Lai and Ma [3] have presented a modified evolutionary strategy to solve the reactive power dispatch, obtaining good results. Other authors [5,6] have applied the same algorithm for other power system problems, reporting results using the IEEE30 system. A simplified evolution strategy has been used in [6] and compared with genetic algorithms, and the Lai and Ma algorithm. More recently, a proposal quite similar to [3] has been presented in [7]. In spite of these efforts, evolutionary techniques have not yet explored completely power system applications [11].

Differential Evolution (DE) algorithm has been considered a novel evolutionary computation technique used for optimization problems. The DE and some other evolutionary techniques exhibit attractive characteristics such as its simplicity, easy implementation, and quick convergence. Generally speaking, all population-based optimization algorithms, no exception for DE, suffer from long computational times because of their evolutionary/stochastic nature. This crucial drawback sometimes limits their application to off-line problems with little or no real-time constraints.

In these kind of algorithms, within an *n*-dimensional search space, a fixed number of vectors are randomly initialized, and then new populations are evolved over time to explore the search space and locate the optima. Differential Evolutionary strategy (DE) uses a greedy and less stochastic approach in problem solving. DE combines simple arithmetical operators with the classical operators of recombination, mutation and selection to evolve from a randomly generated starting population to a final solution. The fundamental

* Corresponding author. Tel.: +52 33 3777 3600.
E-mail addresses: jramirez@gdl.cinvestav.mx (J.M. Ramirez), mgonzale@gdl.cinvestav.mx (J.M. Gonzalez), rtapia@gdl.cinvestav.mx (T.O. Ruben).

0142-0615/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijepes.2010.08.019

idea behind DE is a scheme whereby it generates the trial parameter vectors. In each step, the DE mutates vectors by adding weighted random vector differentials to them. If the fitness of the trial vector is better than that of the target, then the target vector is replaced by the trial vector in the next generation.

In this paper an optimal reactive power dispatch is proposed, where the objective function is the minimization of active power losses while maintaining system voltage security. In coordination with the problem formulation, Continuation Power Flow (CPF) is applied to evaluate and maintain the voltage security margin of the system. The control variables are the generation voltages, taps position, and capacitor/reactor banks. The problem is solved by the Differential Evolution algorithm. Usually, the problem is proposed as an objective function subject to constraints related to physical limits, especially on active and reactive power generation. Optimization techniques are applied to determine the steady-state optimal operating conditions, where voltage magnitudes and angles at all buses are evaluated for specific levels of load and generation. Evidently the results of any optimization technique will impact on the power system stability. In this paper the implication on steady state voltage stability is taken into account.

2. Optimal reactive power dispatch formulation

The optimal reactive power dispatch goal is to minimize active power losses and improve the voltage profile by setting generator bus voltages, VAR compensators, and transformer taps. This problem may be expressed as follows

min $f = f_{\text{losses}}$

such that

$$\begin{aligned} &P_i^d - P_i(V,\theta) = 0 \quad i \in NB - 1 \\ &Q_i^d - Q_i(V,\theta) = 0 \quad i \in NPQ \end{aligned} \tag{1a}$$

$$\begin{aligned} &V_{k\min} \leqslant V_k \leqslant V_{k\max} \quad k = 1,2,\ldots,NB \\ &\text{Tap}_{l\min} \leqslant \text{Tap}_l \leqslant \text{Tap}_{l\max} \\ &B_{\text{shunt\_}j\min} \leqslant B_{\text{shunt\_}j} \leqslant B_{\text{shunt\_}j\max} \end{aligned} \tag{1b}$$

where f_{losses} represents the system losses; NB represent the system buses set; NPQ represents the PQ-buses set. Eq. (1a) represents the load flow equations. V is the bus voltage magnitude, Tap_l represents the l-th transformer's tap position, and B_{shunt_j} is the shunt susceptance located at bus j. The generator bus voltages, the transformer tap-settings, and capacitor/reactor banks are the control variables. $P_i^d - Q_i^d$ are the active and reactive power demand at bus i, respectively. In this paper, constraints (1a), (1b) are handled through the objective function's penalization, where the corresponding penalty parameters are chosen empirically based on experience and the particular application.

3. Summary on the Differential Evolution algorithm

Differential Evolution (DE) is a floating-point encoding evolutionary algorithm for global optimization over continuous spaces [12–14], which can work with discrete variables. DE creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness value.

DE has three control parameters: amplification factor of the difference vector-F, crossover control parameter-CR, and population size-NP. The original DE algorithm keeps fixed all three control parameters during the optimization process. However, there exists a lack on knowledge of how to find reasonably good values for the DE's control parameters for a given function [15].

Although the DE algorithm has been shown to be a simple, yet powerful, evolutionary algorithm for optimizing continuous functions, users are still faced with the problem of preliminary testing and hand-tuning of the evolutionary parameters prior to start up the actual optimization process. As a solution, self-adaptation has proved to be highly beneficial for automatically and dynamically adjusting evolutionary parameters, such as crossover and mutation rates. Self-adaptation is usually used in Evolution Strategies [16]. Self-adaptation enables an evolutionary strategy to adapt itself to any general class of problem, by reconfiguring itself accordingly, and does this without any user interaction.

3.1. DE's description

This subsection provides the basic background on the DE algorithm [14,17].

An optimization algorithm is concerned with finding a vector $\mathbf{x}$ so as to minimize $f(\mathbf{x})$; $\mathbf{x} = (x_1, x_2, \ldots, x_D)$. D is the dimensionality of vector $\mathbf{x}$. The variables' domains are defined by their lower and upper bounds: $x_{j,\text{low}}$, $x_{j,\text{upp}}$; $j \in \{1, \ldots, D\}$. The initial population is selected uniform randomly between the lower ($x_{j,\text{low}}$) and upper ($x_{j,\text{upp}}$) bounds defined for each variable x_j. These bounds are specified by the user according to the problem's nature.

DE is a population-based algorithm and vector $\mathbf{x}_{i,G}$; $i = 1, 2, \ldots$ NP is an individual in this population. NP denotes population size and G the generation. During one generation for each vector, DE employs mutation, crossover, and selection operations to produce a trial vector (offspring) and to select one of these vectors with the best fitness value.

Once initialized DE mutates randomly chosen vectors to produce an intermediary population of NP mutant vectors, $\mathbf{v}$. Each vector in the current population is then recombined with a mutant to produce a trial population of NP trial vectors. During recombination, trial vectors overwrite the mutant population, so that a single array can hold both populations.

3.2. Algorithm

The DE algorithm main steps are:

(a) Set the parameters: F, CR, NP, and the generation counter $G = 1$.
(b) Generate the initial population randomly: $\mathbf{x}_{i,0}$ $i = 1, 2, \ldots, \text{NP}$ from solution space.
(c) While (stopping criterion is not met).
(d) For each vector $\mathbf{x}_{i,G} = \{x_{1i,G}, x_{2i,G}, \ldots, x_{Di,G}\}$, $i = 1, 2, \ldots, \text{NP}$.

(1) Choose three indexes r_1, r_2, r_3 within the range [1, NP] randomly. They should be mutually different and also different from index i.

(2) (Mutation). Generate the mutant vector $\mathbf{v}_{i,G+1}$ according to

$$\boldsymbol{v}_{i,G+1} = \boldsymbol{x}_{r1,G} + F(\boldsymbol{x}_{r2,G} - \boldsymbol{x}_{r3,G}) \tag{2}$$

The scale factor, $F \in (0, 1+)$, is a positive real number that controls the rate at which the population evolves. While there is no upper limit on F, effective values are seldom greater than 1.0.

(3) (Crossover). DE crosses each vector with a mutant vector. Generate a new vector $\mathbf{u}_{i,G+1}$, where

$$\boldsymbol{u}_{ji,G+1} = \begin{cases} \boldsymbol{v}_{ji,G+1} \text{if } r \text{ and } (j) \leqslant CR \text{ or } j = rn(i), \\ \boldsymbol{x}_{ji,G} \text{ if } r \text{ and } (j) > CR \text{ and } j \neq rn(i), \end{cases} \tag{3}$$

$r(j) \in [0, 1]$ is the jth evaluation of the uniform random generator number. $rn(i) \in (1, 2, \ldots, D)$ is a randomly chosen index. The crossover probability, $CR \in [0, 1]$, is a user defined value that controls the fraction of parameter values that are copied from the mutant.