Electrical Power and Energy Systems 62 (2014) 461–468

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

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

Application of binary group search optimization to distribution network reconfiguration

Saeed Teimourzadeh, Kazem Zare *

University of Tabriz, P.O. Box 51666-15813, Tabriz, Iran

ARTICLE INFO

Article history:
Received 6 July 2013
Received in revised form 13 April 2014
Accepted 30 April 2014
Available online 4 June 2014

Keywords:
Binary group search optimization
Reconfiguration
Forward–backward power flow

ABSTRACT

This paper presents the binary group search optimization algorithm (BGSO) for solving the optimal distribution network reconfiguration problem. The objective of the optimization is loss reduction in distribution system. The proposed GSO algorithm is introduced with fundamental modifications, such as introducing binary form of the GSO in order to be fit for reconfiguration and all binary form problems. All formulation of conventional GSO has been modified for accessing a novel powerful binary searching algorithm. Additionally, the forward–backward sweep, load flow is used in this paper, due to its accuracy. The Comparative studies are conducted on standard networks to verify the validity of the proposed algorithm. The simulations on different test cases confirm validity of the BGSO in solving optimal distribution network reconfiguration problem.

Introduction

Distribution network reconfiguration (DNR) is an efficient measure of distribution networks optimization, which optimizes through determining the switches condition. DNR is an effective way to reduce losses in the distribution networks. DNR also helps load balancing, improve voltage quality and enhance system security [1]. Distribution networks are generally designed in loop configuration however; they operate in radial configuration. These systems also include large quantity of sectionalizing switches and some of tie switches [2].

Network reconfiguration is the process of changing the topology of distribution systems by changing the open/closed status of tie and sectionalizing switches. For the large scale distribution systems, choosing the branches in order to find the optimal configuration is a challenging problem, due to the many candidate-switching combinations in the distribution systems. Therefore, network reconfiguration is a complicated combinatorial and non-differentiable constrained optimization problem.

Generally there are three categories for solving network reconfiguration problem. The first category includes classical mathematical and analytical analysis methods such as, Linear programming [3], integer programming [4] and dynamic programming [5]. These algorithms are difficult to be practically implementation because the burden of calculation would exponentially increase with the scale increase of the distribution networks and they maybe get stuck in local minima.

The second category includes heuristic methods. The branch exchange method in [1,6], fuzzy multi-objective approach in [7], optimal flow pattern in [8] and efficient network reconfiguration using minimum cost maximum flow-based branch exchanges and random walks-based in [9], are listed in this category. These algorithms are correlated with the operation sequence of switches or with the network initial state. Therefore, it is difficult to obtain the global optimal solution.

Artificial intelligence (AI) based algorithms, such as simulated annealing (SA) in [10,11], genetic algorithm (GA) in [12], an ant colony search in [13] and particle swarm optimization in [2,14,15] could be listed in third category. In order to solve DNR problem, a mix-integer hybrid differential evolution is reported in [16] and variable scaling hybrid differential evolution in [17]. The tabu search (TS) methodology and its modifications are reported in [18–20] as an efficient tool in order to solve the DNR problem.

Recently, power system applications have benefited from a novel and powerful optimization algorithm. This algorithm is based on animal searching behavior and called group search optimization algorithm (GSO). This algorithm firstly presented at 2009 by He et al. [21]. Based on pervious works of the authors [22,23], GSO and its modifications have better performance in comparison with recently related published papers.

* Corresponding author. Tel./fax: +98 411 3300829.
E-mail addresses: teimourzadeh91@ms.tabrizu.ac.ir (S. Teimourzadeh), kazem.zare@tabrizu.ac.ir (K. Zare).

http://dx.doi.org/10.1016/j.ijepes.2014.04.064
0142-0615/

In this paper, firstly a brief overview of the conventional GSO is presented. The conventional GSO algorithm is in continues searching space mode. However, for the reconfiguration problem, a binary searching tool is required. The main contribution of paper is to propose basic modifications to the conventional GSO, in order to change it to a binary searching tool. Afterwards, the proposed BGSO is implemented for distribution system reconfiguration problem. In order to verify the efficiency of the proposed method, comprehensive simulation studies are carried out. The simulation studies incorporate 33-node, 69-node and 119-node distribution test systems. The obtained results are compared with the recently published techniques. The simulation and comparison results validate accuracy of the proposed method.

The remainder of the paper is organized as follows: 'Group search optimization algorithm (GSO)' provides a brief overview of the basics of GSO and the proposed BGSO. 'Problem formulation' presents the distribution network reconfiguration formulation. In 'Reconfiguration problem solution mechanism', presents the implementation of BGSO for reconfiguration problem is presented. 'Test results' reports the simulation studies and 'Conclusion' provides the conclusion of the findings and contributions of the paper.

Group search optimization algorithm (GSO)

This section presents a brief overview of group search optimization algorithm. Afterwards, the modification procedure of the GSO for providing a binary searching tool is presented.

Basics of GSO

Group search optimization (GSO) is a novel optimization algorithm which is based on animal searching behavior and their group-living theory [22]. This theory mainly based on the producer–scrounger (PS) model and the animals scanning mechanisms are employed metaphorically for designing an optimum searching strategy for solving the optimization problems [21].

The population of GSO algorithm is called a group and each individual in the population is called a member. In n-dimensional search space, the ith member in kth searching iteration has a current position $X_i^k \in \mathbb{R}$ and a head angle $\varphi_i^k = (\varphi_{i1}^k, \ldots, \varphi_{i(n-1)}^k) \in \mathbb{R}^{n-1}$. Where $\mathbb{R}$ is the set of real numbers and φ_{ik} is polar angle of ith member relative to the kth dimension. The search direction of ith member, $D_i^k(\varphi_i^k)$ could be calculated from φ_i^k via a polar to Cartesian coordinate transformation [24]:

$$d_{i1}^k = \prod_{q=1}^{n-1} \cos(\varphi_{iq}^k) \tag{1}$$

$$d_{ij}^k = \sin\left(\varphi_{i(j-1)}^k\right) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k) \qquad j = 2, \ldots, n-1 \tag{2}$$

$$d_{in}^k = \sin\left(\varphi_{i(n-1)}^k\right) \tag{3}$$

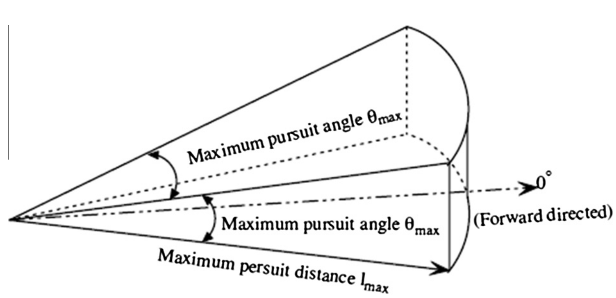

Fig. 1. Scanning field in 3-D space.

$$D_i^k(\varphi_i^k) = (d_{i1}^k, d_{i2}^k, \ldots, d_{in}^k) \tag{4}$$

The GSO group includes three types of members: producers, scroungers and rangers. At each iteration, a group member, which has a best fitness value, is chosen as the producer. The producer scans the search space for better states. In this methodology, the producer uses its vision ability to search the space. Vision ability is the ability of testing some points around its position. Fig. 1 depicts the scanning field in 3-D searching space.

At the kth iteration the producer X_p behaves as follows:

(1) The producer will scan at zero degree and then scan laterally by randomly sampling three points in the scanning field as follow:

$$X_z = X_p^k + r_1 l_{\max} D_p^k(\varphi^k) \tag{5}$$

$$X_r = X_p^k + r_1 l_{\max} D_p^k(\varphi^k + r_2\theta_{\max}/2) \tag{6}$$

$$X_l = X_p^k + r_1 l_{\max} D_p^k(\varphi^k - r_2\theta_{\max}/2) \tag{7}$$

where $r_1 \in \mathbb{R}^1$ is a normally distributed random number with zero mean and one standard deviation and $r_2 \in \mathbb{R}^{n-1}$ is a uniformly distributed random number sequence in the range of $(0,1)$.

(2) The producer will then find the new point .If the new point has a better value in comparison with its current position, producer flies to the new point. If not, it will stay in its current position and turn its head using Eq. (8):

$$\varphi^{k+1} = \varphi^k + r_2 a_{\max} \tag{8}$$

where $a_{\max} \in \mathbb{R}^1$ is the maximum turning angle.

(3) If the producer cannot find a better area after a iterations, it will turn its head back to zero degree as follows:

$$\varphi^{k+a} = \varphi^k \tag{9}$$

where a is a constant.

During each searching bout, a number of group members are selected as scroungers. The scroungers will keep searching for opportunities to join the resources found by the producer. At the kth iteration, the area copying behavior of the ith scrounger can be modeled as a random walk toward the producer.

$$X_i^{k+1} = X_i^k + r_3 \circ (X_p^k - X_i^k) \tag{10}$$

where $r_3 \in \mathbb{R}^n$ is a uniform random sequence in the range $(0,1)$. Operator "$\circ$" is the Hadamard product or the Schur product, which calculates the entry-wise product of two vectors. During scrounging, the ith scrounger will keep searching for other opportunities to join the producer. The rest of the group members will be dispersed from their current positions. Random walks, which are thought to be the most efficient searching method for randomly distributed resources are employed by the rangers. At the kth iteration, a ranger generates a random head angle φ_i using (8), and then it chooses a random distance from (11) and moves to the new point using (12).

$$l_i = a r_1 l_{\max} \tag{11}$$

$$X_i^{k+1} = X_i^k + l_i D_i^k(\varphi^{k+1}) \tag{12}$$

More details on GSO could be found in [21].

Binary group search optimization (BGSO)

Producer

In this section, all the members of GSO group are equal to 0 or 1 and X_p is the producer. As previously mentioned, the producer member has vision ability. In order to simulated the producer