

A Biased Random Key Genetic Algorithm Applied to the Electric Distribution Network Reconfiguration Problem

H. de Faria Jr.^{a,b}, M. G. C. Resende^c, D. Ernst^b

^aUFABC – Universidade Federal do ABC, CECS, Rua Santa Adélia, N^o 166, Santo André, SP, 09.210-170 Brazil

^bMontefiore Institute, Quartier Polytech 1,10, Allée de la découverte 4000 ,B28, Université de Liège, Belgium

^cMathematical Optimization and Planning, Amazon.com, 333 Boren Ave. N, Seattle, WA 98109 USA

Corresponding author: e-mail: haroldo.faria@ufabc.edu.br, Tel.: (+352)691996841

Abstract

This work presents a biased random-key genetic algorithm (BRKGA) to solve the electric distribution network reconfiguration problem (DNR). The DNR is one of the most studied combinatorial optimization problems in power system analysis. Given a set of switches of an electric network that can be opened or closed, the objective is to select the best configuration of the switches to optimize a given network objective while at the same time satisfying a set of operational constraints. The good performance of BRKGAs on many combinatorial optimization problems and the fact that it has never been applied to solve DNR problems are the main motivation for this research. A BRKGA is a variant of random-key genetic algorithms, where one of the parents used for mating is biased to be of higher fitness than the other parent. Solutions are encoded by using random keys, which are represented as vectors of real numbers in the interval (0,1), thus enabling an indirect search of the solution inside a proprietary search space. The genetic operators do not need to be modified to generate only feasible solutions, which is an exclusive task of the decoder of the problem. Tests were performed on standard distribution systems used in DNR studies found in the technical literature and the performance and robustness of the BRKGA were compared with other GA implementations.

Keywords: Distribution network reconfiguration, Biased random-key genetic algorithms, Optimization, Power losses

1. Introduction

Combinatorial optimization problems that involve a large finite number of alternatives arise in many scientific research fields, industry and government. In this type of problem, the number of combinations to be tested to find an optimal solution often grows exponentially with the size of the problem, which makes complete enumeration of solutions infeasible in real-world problems. The electric power systems research field has many problems which belong to the combinatorial optimization field, such as unit commitment (Roque et al. 2014), transmission network design (Faria Jr. et al. 2005), and distribution network reconfiguration (DNR) (Civanlar et al. 1988). DNR belongs to the class of network design problems which include transportation problems, computer network restoration and telecommunication network design (Delbem et al. 2004). These problems are generally *NP-Hard*.

Electric power distribution systems are located after the transmission and sub-transmission systems and are generally structured in mesh but operated in radial configuration to reduce short-circuit fault levels and to facilitate the implementation of effective coordination and protection schemes (Swarnkar et al.

2011). The problem consists in altering the open or closed status of normally closed (NC) sectionalizing switches and normally open (NO) tie-line switches to optimize a given objective of the network while satisfying various operational constraints and maintaining the radiality of the distribution system. The radiality constraint should be satisfied without islanding of any node(s). This fact makes the problem extremely complicated and computationally hard. Fig. 1 shows a distribution system composed of 14 nodes or buses where node 1 represents the substation node and the other nodes represent electrical loads. All the power to attend the loads come from node 1. Dark lines represent the NC switches and dotted lines represent the NO switches. It can be seen that the system is radial and that all loads are connected to bus 1 and are, thus, energized.

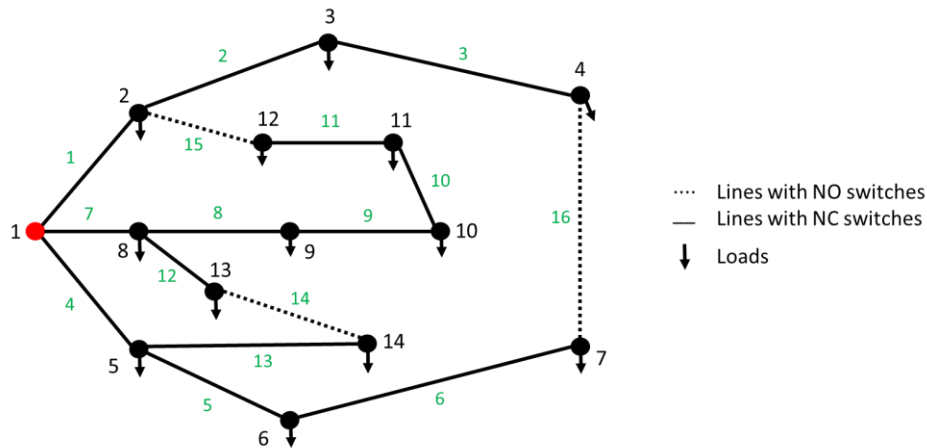


Fig. 1 14-node distribution network

Usually, DNR studies are used for minimization of electric power losses in the network, service restoration after power outages, load balancing and planning studies. Network reconfiguration for loss reduction is the most explored version of DNR. It is a highly complex combinatorial, nondifferentiable and constrained nonlinear mixed integer optimization problem, due to the high number of switching elements in a distribution network, and to the nonlinear characteristics of the constraints used to model the electrical behavior of the system (Carreño et al. 2008).

The main methods used for solving the DNR for loss reduction can be separated into two main groups: exact mathematical methods and approximate methods. Very large real-world problems pose serious computational challenges for exact methods despite recent advances in finding provably optimal solutions to combinatorial optimization problems using methods such as cutting planes, dynamic programming and branch and bound techniques (Gonçalves and Resende 2010). Regarding DNR, this type of algorithm is applied only to simplified models of the electrical network with an approximated loss function as well. For the ac network model, there is no exact representation of the losses in the exact methods. Approximate methods, on the other hand, include the class of heuristic and metaheuristic methods which cannot prove the optimality of the solution found, but can tackle big and complex real-world problems efficiently. These methods can produce many high-quality solutions in reduced computational time and find the optimal solution most of the time (Yang 2010). The majority of methods applied to the DNR for loss reduction belong to the metaheuristic class of methods. A metaheuristic applies and coordinates more than one heuristic, such as local search, using the strengths of each one to efficiently explore the search space. They

include genetic algorithms (GA), simulated annealing, tabu search, scatter search, ant colonies, variable neighborhood search, GRASP, and path-relinking. There are many ways to classify metaheuristics. These include, trajectory-based versus population-based, nature-inspired versus non-nature inspired, memoryless versus memory-based, etc. Genetic algorithms, for example, are nature-inspired, population-based, with memory. Tabu search are trajectory-based with memory. GRASP is trajectory-based (Festa et al. 2006).

Various heuristic and metaheuristic techniques have been proposed to solve DNR problems. The first application of a GA to solve the DNR for loss reduction was proposed in (Nara et al. 1992). In (Carreno et al. 2008), a modified genetic algorithm presented by Chu-Beasley is used to solve the DNR for loss reduction. This work proposes a codification where, instead of representing the switching devices, the entire network configuration resulting from the switching is used as the individual and considered as a tree graph, represented as a vector with the arcs in the tree (branches) sequentially organized, from top to bottom of the network, being the root node (generally the substation) the top. A combination of the binary and discrete Particle Swarm Optimization is proposed in (Li et al. 2008) to solve the loss reduction problem. The method identifies groups of branches to represent the network and each group has unidimensional encoding. In (Santos et al. 2010), a node-depth encoding based on graph theory is proposed to solve very large scale DNR problems. A multi-objective evolutionary algorithm is used in conjunction with the node-depth encoding and two crossover operators: preserve ancestor operator (PAO) and change ancestor operator (CAO). These operators generate only feasible configurations, that is, radial DNs that supply power to the entire network. An explicit representation of the radiality constraints in DNR problems was proposed in (Lavorato et al. 2012). This contribution enables the solution of the problem using an integer programming technique. The analytical formulation proposed was solved using a nonlinear branch-and-bound algorithm with modest solution times. The work presented in (Braz and Souza 2011) uses graph theory to represent the network and a GA to solve the reconfiguration problem. The objective function comprises electrical losses and switching mitigation. Two novel network representations that generate only radial topologies were proposed: Subtractive sequential encoding and Additive sequential encoding. The drawback is the decodification process applied to the chromosomes to evaluate the fitness of the solution. In all metaheuristic techniques, the encoding of the solution is fundamental for the efficiency of the method. The encoding ideally should be able to generate only feasible solutions, reducing the size of the search space and running times of the algorithm.

The rest of the paper is organized as follows: Section II presents the formulation for the problem of distribution network reconfiguration for loss reduction. Section III introduces biased random key genetic algorithms. Section IV describes the codification for DNR problems used in this paper. Section V presents the results of the study and Section VI draws some conclusions and points out future developments.

2. Problem Formulation

The DNR is an important tool for many power system operation and planning problems. It can be used for power loss reduction, service restoration, optimization of network voltage profiles, maximum accommodation of distributed generation and other objectives. This work deals with the power loss reduction problem, whose formulation is presented next.

$$\text{Min } f = \sum_{ij=1}^{|\Omega_l|} k_{ij} r_{ij} \frac{P_{ij}^2 + Q_{ij}^2}{V_i^2} \quad (1)$$

subject to

$$P_{Si} - P_{Di} - \sum_{j \in \Omega_{bi}} k_{ij} P_{ij} = 0 \quad \forall i \in \Omega_b \quad (2)$$

$$Q_{Si} - Q_{Di} - \sum_{j \in \Omega_{bi}} k_{ij} Q_{ij} = 0 \quad \forall i \in \Omega_b \quad (3)$$

$$S_{ij} \leq \bar{S}_{ij} \quad \forall (ij) \in \Omega_l \quad (4)$$

$$\underline{V}_i \leq V_i \leq \bar{V}_i \quad \forall i \in \Omega_b \quad (5)$$

$$k_{ij} \in \{0,1\} \quad \forall ij \in \Omega_l \quad (6)$$

$$g \in G \quad (7)$$

In this formulation, i and j represent generic nodes of the electrical distribution system, where Ω_l is the set of all branches ij of the network connecting nodes i and j , and Ω_b is the set of all nodes i . The symbol r_{ij} stands for the electrical resistance of branch ij . The symbols P_{ij} and Q_{ij} are the active and reactive power flows of branch ij . The elements of P_{ij} and Q_{ij} are given by (8) and (9), respectively, where g_{ij} is the conductance of branch ij and b_{ij} is the susceptance of branch ij .

$$P_{ij} = V_i^2 g_{ij} - V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \quad (8)$$

$$Q_{ij} = -V_i^2 b_{ij} - V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) \quad (9)$$

P_{Di} is the active power demand at node i and P_{Si} is the active power supply at the same node. Q_{Di} is the reactive power demand at node i and Q_{Si} is the reactive power supply at the same node. The objective function (1) represents the power losses of the distribution system operation. Equations (2) and (3) represent the power flow balance equations, derived from Kirchhoff's current law. Equation (4) represents operational limits on branch capacity where S_{ij} is the apparent power flowing in branch ij and \bar{S}_{ij} is the apparent power capacity of the branch. Equations (5) are the operational limits on the value of voltages at each node i of the network, where \underline{V}_i and \bar{V}_i are the minimum and maximum acceptable voltage magnitudes at node i , respectively. Equation (6) represents the binary nature of k_{ij} . The circuit between buses ij is connected if the corresponding value is equal to one and is not connected if it is equal to zero. Equation (7) represents the radiality constraint of the DNR problem. It states that the graph g of the solution must belong to a set G composed of all allowed network structures, i.e. the set excluding meshed and islanded networks. Many heuristic techniques used for solving the DNR problem consider constraint (7) implicitly, applying equation (10).

$$M = n_b - 1 \quad (10)$$

Where M is the number of branches of the solution and n_b is the number of nodes, where $n_b = |\Omega_b|$. However, this condition is necessary but not sufficient to guarantee the radiality constraint (Schmidt et al. 2005). Metaheuristic techniques normally ensure the radiality constraint in their solutions using graph theory or inside evolutionary operators. A feasible solution to the DNR for loss reduction is a distribution network that satisfies constraints (2) - (7) and whose electrical power losses can be calculated to obtain the value of the objective function f . The BRKGA for DNR proposed in this work uses a set of rules derived from graph theory that ensure the feasibility of solutions generated by the metaheuristic as will be explained in Section IV.

3. Biased Random Key Genetic Algorithms

Biased random key genetic algorithms (BRKGA) are nature inspired metaheuristics derived from the genetic algorithms with random keys (RKGA) introduced by Bean (1994) for solving combinatorial optimization problems involving sequencing. In both methods, a chromosome is represented by a vector of randomly generated real numbers in the interval (0,1) called *keys*. Both methods search for solutions to the problem in a continuous n -dimensional unit hypercube and not in the problem's search space directly. This way, a de-codification must be done to map the solutions in the hypercube to the problem's search space. The component responsible for this mapping is called the *Decoder*. Decoders can vary in complexity depending on the problem being solved. They can be very simple, requiring a simple ordering of the generated keys, or can be complex algorithms composed of heuristics and local search. The BRKGA is a general framework for optimization composed of two structures: The genetic algorithm (GA) and the decoder. The decoder is problem dependent and must be devised for each problem tackled. The GA is problem independent, simple to implement and can be used without modification to solve any problem. The algorithm starts with an initial population of p vectors of n random keys. This initial population is evolved by the BRKGA over a number of iterations called *generations*. The initial population is then decoded so that the *fitness* of each individual can be computed. The population is then divided into a small set p_e of elite individuals with the best fitness, and another set of $p - p_e$ individuals. To form the population of the next generation, the set p_e is copied, unchanged, to the next generation. The BRKGA uses the same strategy of classical GAs to avoid entrapment in local optima by introducing mutants into the population. The algorithm inserts p_m mutants into the population and completes the number of individuals by generating $p - p_e - p_m$ vectors of random keys using parametrized uniform crossover (Spears and DeJong 1991). Let a and b be the vectors chosen for mating and let c be the offspring produced. In the crossover, $c[i]$, the i -th component of the offspring vector, receives the i -th key of one of its parents. It receives the key $a[i]$ with probability ρ_a and $b[i]$ with probability $\rho_b = 1 - \rho_a$.

A BRKGA differs from a RKGA in the way parents are selected for mating during crossover. In Bean's algorithm, both parents are chosen randomly from the entire population. In a BRKGA, a parent is always chosen from the elite set, which introduces the elitism principle in the reproduction process. This modification is sufficient to make the biased version of the GA to outperform the unbiased version (Gonçalves et al. 2014). The populations of individuals are evolved in sequence until a stopping criterion is reached. This criterion can be number of generations, running time or quality of solution. The best solution found over all generations is returned as the final result of the algorithm. Fig. 2 shows a BRKGA framework for solving optimization problems where the decoder is the problem dependent part of the algorithm.

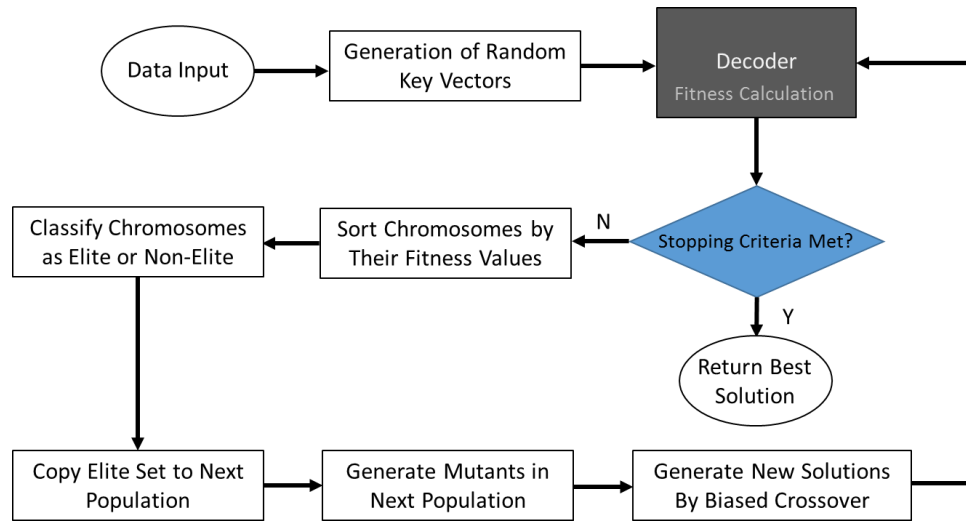


Fig. 2 BRKGA Framework (Gonçalves and Resende 2010)

4. Codification for DNR Problems

Codification is an adequate way to represent a single element which belongs to the search space of a problem, thus, it represents the information of a solution proposal (Carreño et al. 2008). After being decoded, the fitness of the solution should be easily calculated so that a given metaheuristic algorithm can use this information in the search process. The BRKGA framework allows the decodification of encoded solutions to result in exclusive feasible solutions, without reliance on GA operators due to the independence of GA and decoder.

Graph theory can be advantageously used to aid in the codification of solutions of DNR problems. As illustrated by Figure 1, a distribution network can be seen as a graph G composed of a set of nodes N and a set of edges $E - G(N, E)$. The codification used in this work is derived from graph theory and uses a set of rules to correct infeasible individuals and, thus, generate only feasible vectors during decodification. This set of rules was proposed in (Swarnkar et al. 2011) to be used in conjunction with any metaheuristic technique. In a problem of DNR for loss reduction, feasibility of a solution means it is radial, without isolated nodes from the network. The radial configurations of a distribution network are called trees of its associated graph. Consider that every edge of the distribution network graph (DNG) contain a switch. A tree with N nodes contains $N - 1$ graph edges or twigs. The edges that were removed to form the tree are called links. These links form a cotree, which is the complement of the tree. The number of links of a DNG is given by $l = E - (N - 1)$, which is usually much less than the twigs. Thus, the links can be used in the codification of solutions of metaheuristic techniques, reducing the size of the solution vector. In the following, some terms are defined.

Principal node: the junction of three or more elements of the DNG.

Exterior node: the node located at the perimeter of the DNG.

Interior node: the node located inside the perimeter of the DNG.

For the 14-node distribution network given in Fig. 1, the principal nodes of the system are nodes 1,2,8, and 5. The exterior nodes are 1,2,3,4,5,6, and 7. The interior nodes are 8,9,10,11,12,13, and 14.

Loop vector: the set of elements constituting a closed path in a DNG. This closed path cannot contain in its interior another closed path.

Common branch vector: the set of elements which are common between any two loop vectors of a DNG.

Prohibited group vector: the set of the common branch vectors. From each of them, if one element is opened, then one or more interior nodes of the DNG will be islanded. The size of a prohibited group vector cannot be greater than l .

As an example, the three groups of vectors for the system of Fig. 1 are given in Table 1.

Table 1 – Loop vectors, Common branch vectors and Prohibited group vectors of 14-node system

| Loop vectors | Common branch vectors | Prohibited group vectors |
|------------------------------------|-----------------------|----------------------------------|
| $L_1 = [1,7-11,15]$ | $C_{12} = [7]$ | $P_1 = [C_{12}, C_{13}, C_{23}]$ |
| $L_2 = [4,12-14,7]$ | $C_{13} = [8-11,15]$ | |
| $L_3 = [8-11,15,2-3,16,5-6,12-14]$ | $C_{23} = [12-14]$ | |

With these definitions in mind, the switches that are actually links of a cotree, will be used to represent a solution of the DNR problem. The solution vector represents a cotree which must have a corresponding tree that is feasible. This is accomplished by forming the solutions vectors in accordance to the following set of rules:

Rule 1: each candidate switch must belong to its corresponding loop vector.

Rule 2: only one candidate switch can be selected from one common branch vector.

Rule 3: all the common branch vectors of a prohibited group vector cannot participate simultaneously to form an individual.

These set of rules guarantee the production of feasible individuals, meaning that only radial configurations without islanded nodes are built, avoiding the necessity of mesh checks on solutions. Rule 1 prevents the islanding of exterior node(s), Rule 2 prevents the islanding of interior node(s) and Rule 3 prevents the islanding of principle interior node(s) of the distribution network, respectively. Using the set of three rules aforementioned, one can build a feasible solution to the 14-node system. Since the links of the cotree of a distribution network can be used to represent a solution, the links or NO switches of the 14-node system form a solution to the DNR problem. This solution is viable if it complies with rules 1,2, and 3. Fig. 3 illustrates a viable solution or chromosome of the 14-node system.

| | | |
|---|----|---|
| 7 | 13 | 2 |
|---|----|---|

Fig. 3 Viable chromosome of the 14-node system

The representation of solutions using loop vectors greatly reduces the size of the problem's search space (Wang et al. 2009). If the binary coding strategy is adopted for the 14-node system, the length of each chromosome would be 16 (number of branches) and the search space is $2^{16} = 65,536$ individuals. If the representation using loop vectors is used, the length of each chromosome is in accordance with the total number of system fundamental loop vectors. One gene locus corresponds to one fundamental loop, and

its allele can be any branch that forms the fundamental loop. The value of the allele is the number of the branch whose switch is opened in this fundamental loop. In this representation, the size of the search space for the 14-node system would be $7 \times 5 \times 13 = 455$ individuals.

In a BRKGA framework, these set of rules can be used to build the decoder, which maps a vector of random keys into a solution of the optimization problem and computes the cost of this solution. The advantage of the BRKGA is that it is totally independent of the decoder. The BRKGA genetic operators do not need to be modified to generate only feasible solutions as all the other GA applications to DNR problems encountered in the literature. This is an exclusive task of the decoder. The BRKGA carries out the evolutionary process on a population of random keys inside a proprietary search space (hypercube). An important noteworthy fact is that decoders should be deterministic, meaning that a vector of random keys is always mapped to the same point in the search space of the problem being solved. Fig. 4 shows a pseudo-code adapted from Gonçalves and Resende (2010) for the minimization of $f(x)$, where $x \in X$ and X is a discrete set of solutions and $f: X \rightarrow \mathbb{R}$.

```

1 BRKGA( $|P|, |P_e|, |P_m|, n, \rho_a$ )
2 Initialize value of the best solution found:  $f^* \leftarrow \infty$ ;
3 Generate a population  $P$  with  $n$  vectors of random keys;
4 while stopping criteria not satisfied do
5     Decode and evaluate the cost of each new solution in  $P$ ;
6     Partition  $P$  into two sets:  $P_e$  and  $P_{\bar{e}}$ 
7     Find best solution  $x^+$  in  $P$ :
8      $x^+ \leftarrow \operatorname{argmin}\{f(x) | x \in P\}$ ;
9     if  $f(x^+) < f^*$  then
10          $x^* \leftarrow x^+$ ;
11          $f^* \leftarrow f(x^*)$ 
12     end
13     Initialize population of next generation:  $P^+ \leftarrow P_e$ ;
14     Generate set  $P_m$  of mutants, each mutant with  $n$  random keys;
15     Add  $P_m$  to population of next generation:  $P^+ \leftarrow P^+ \cup P_m$ ;
16     foreach  $i \leftarrow 1$  to  $|P| - |P_e| - |P_m|$  do
17         Select parent at random from  $P_e$ ;
18         Select parent b at random from  $P_{\bar{e}}$ ;
19         foreach  $j \leftarrow 1$  to  $n$  do
20             Throw a biased coin with probability  $\rho_a > 0.5$  of resulting heads;
21             if heads then  $c[j] \leftarrow a[j]$ ;
22             else  $c[j] \leftarrow b[j]$ ;
23         end
24         Add offspring  $c$  to population of next generation  $P^+ \leftarrow P^+ \cup \{c\}$ ;
25     end
26     Update population:  $P \leftarrow P^+$ ;
27 end
28 return  $x^*$ ;

```

Fig. 4 – Pseudo-code for a BRKGA

Fig. 5 shows how the set of rules are used to form the decoder inside the algorithm. The decoder is called in line 5 of the pseudo-code together with the evaluation of the fitness of solutions using a power flow program.

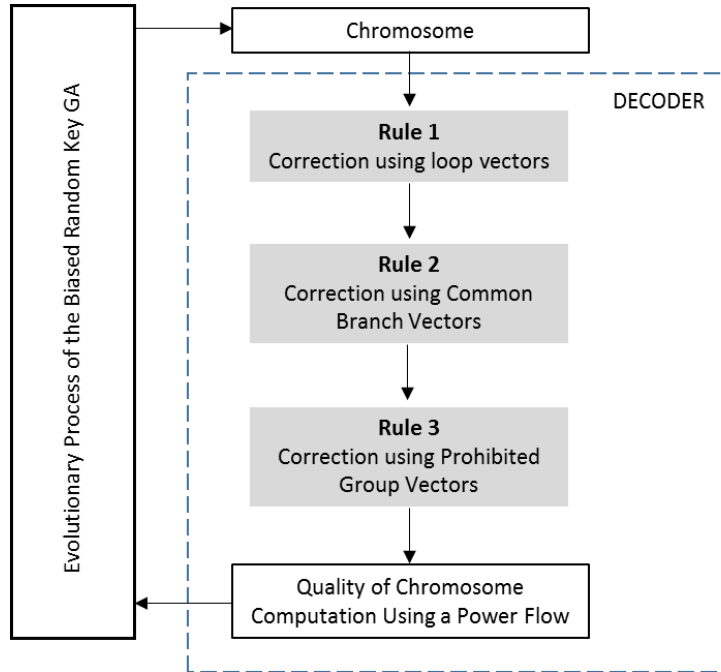


Fig. 5 Decoder of a BRKGA for DNR problems

Some basic details about the computational implementation of the correction rules are given next. Rule 1 is a procedure that receives a random key vector as input and decodes it such that the first allele belongs to the first loop vector, the second allele belongs to the second loop vector and so forth. Fig. 6 shows how this is done for the first allele of a given random key vector and the first loop vector of a graph.

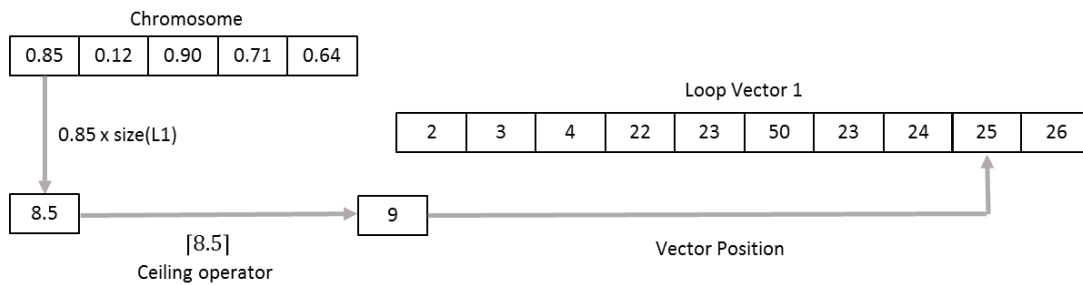


Fig. 6 Decodification of a random key vector to comply with rule 1

Rules 2 and 3 are procedures that analyze the decoded vector produced by rule 1 to check for violations. If a violation is detected, a substitute for the value of the first allele that is not obeying the corresponding

rule is searched in the corresponding loop vector. If the violation is not eliminated, the second allele and its corresponding loop vector are used in the correction process. The application of the rules is sequential.

5. Results

In this section, the computational results obtained from the application of the BRKGA to solve two standard test systems found in DNR literature are presented. The algorithm performance is also compared with other GA implementations.

Two standard IEEE distribution systems were used to test the efficiency of the BRKGA algorithm. The first system is a 33-bus distribution system (Savier and Das 2007) with 37 branches and 5 normally open switches. The second system is 69-bus distribution system (Baran and Wu 1989) with 73 branches and 5 normally open switches. Both network topologies have five loop vectors, seven common branch vectors and six prohibited group vectors. Table 2 gives the initial configuration of the tested systems. The systems were considered balanced and loads were modeled as constant power to enable comparisons with other GAs.

Table 2 - Initial configurations of tested systems

| <i>Data</i> | <i>Systems</i> | |
|-------------------------|----------------|----------------|
| | <i>33-node</i> | <i>69-node</i> |
| Number of Buses | 33 | 69 |
| Number of Branches | 37 | 73 |
| Number of Open Switches | 5 | 5 |
| Active Load (MW) | 3.7 | 3.8 |
| Reactive Load (Mvar) | 2.3 | 2.7 |
| Nominal Voltage (kV) | 12.66 | 12.66 |
| Active Losses (MW) | 0.208 | 0.239 |
| Minimum Voltage (pu) | 0.911 | 0.903 |

Table 3 shows the optimal solutions of the test systems used in the experiments.

Table 3 – Optimal solutions of the test systems

| Test system | Optimal configuration | Real power loss (MW) | Minimum node voltage (pu) |
|----------------|-----------------------|----------------------|---------------------------|
| 33-node | 7,9,14,37,32 | 0.1389 | 0.9423 |
| 69-node | 14,56,61,69,70 | 0.0997 | 0.9423 |

The proposed BRKGA was implemented with the parameters given in Table 4 in some of the experiments. BRKGAs parameters are adjusted based on guidelines provided in Gonçalves and Resende (2010).

Table 4 – BRKGA parameters

| Topology | Population size | Maximum generation | Size of elite set | Size of mutant set |
|----------------|-----------------|--------------------|-------------------|--------------------|
| 33-node | 50 | 30 | 7 | 5 |
| 69-node | 80 | 30 | 10 | 8 |

The probability ρ_a of choosing the elite allele of the elite parent during crossover was fixed at the value of 0.7 for all simulations. The algorithm was developed in Matlab[®] using the Matpower toolbox (Zimmerman et al. 2011), and the simulations were done on a personal computer with an Intel Core i7-6700HQ @2.6 GHz with 16GB of RAM.

5.1. 33-node Test System

The BRKGA was independently run 50 times – each time with a different seed for the pseudo-random number generator – and CPU times to optimal solution were recorded. Fig. 7 shows a time to target plot for the BRKGA using a population of 50, 80, and 90 individuals. As with most stochastic search methods, the continuous random variable time to target solution of a BRKGA has an empirical distribution that approximates a shifted exponential distribution. These graphs are used to characterize the running times of stochastic algorithms for combinatorial optimization. Time-to-target (TTT) plots display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis (Aeix et al. 2007). The Fig. 7 shows that the population size does not affect the time to target plots for this particular instance of the DNR problem. In 50% of the runs, the algorithm was able to find the optimal solution in less than 19 seconds for the three population sizes. It would be very useful for comparison purposes between stochastic algorithms if TTT plots were constructed in other technical references, but only CPU times and number of generations until convergence are usually presented. This makes the comparison a bit difficult. The reduced number of generations required for convergence of the BRKGA attests its efficiency and the implementation of a parallel variant can be very promising in terms of running times.

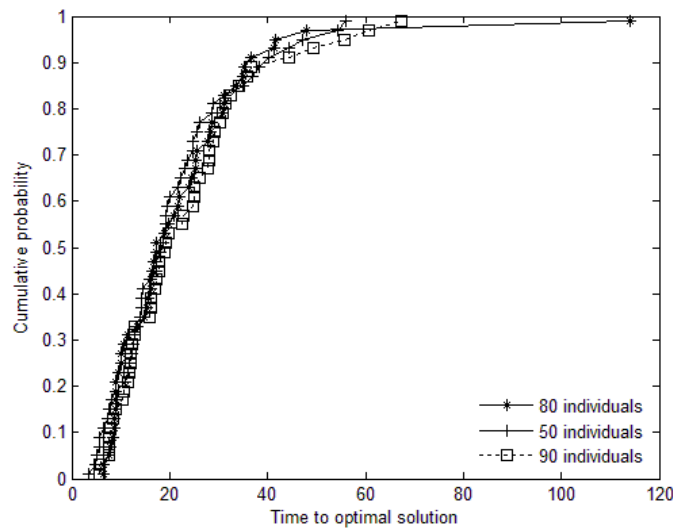
**Fig. 7 – Time to optimal solution plots for the 33-node system**

Fig. 8 shows a convergence characteristic of the algorithm to find the optimal solution for an independent run of the BRKGA on the 33-node system using a population of 100 individuals. Only 6 generations were needed to reach the optimal solution. The 100-individual population proved to be efficient in terms of running times and number of iterations to reach the optimum. Around 8 seconds were needed to reach the optimum. The discrete random variable iterations to target solution, on the other hand, has an empirical shifted geometric distribution. The Fig. 9 illustrates the empirical distribution of number of iterations of the BRKGA to find the optimal solution.

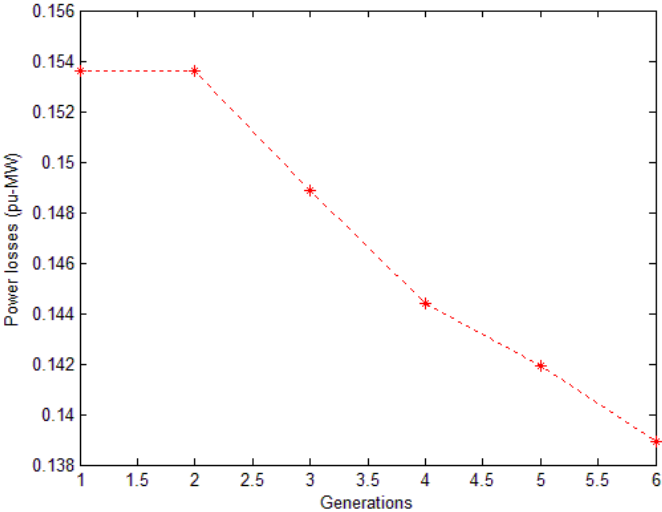


Fig. 8 Convergence characteristic for a particular run of the algorithm for the 33-node system

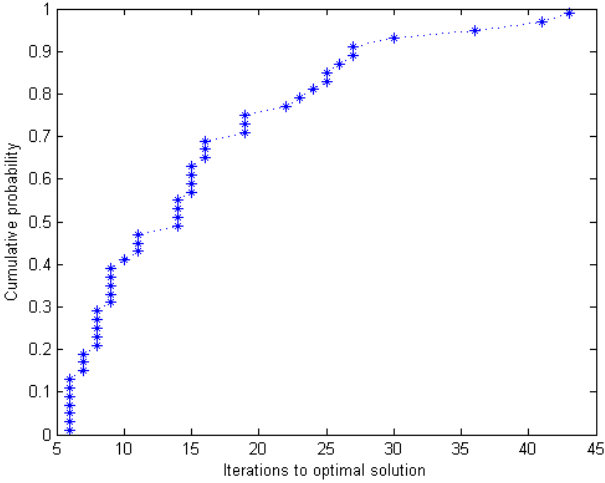


Fig. 9 - Iteration count distribution to optimal solution of the BRKGA

According to Fig. 9, in 41% of the runs, the BRKGA could find the optimal solution in less than 10 iterations. Despite the absence of TTT plots in other technical references, the BRKGA was compared with other GA implementations using the methodology presented in (Braz and Souza 2011). An effective metric for

practical applications is proposed, where the idea is to compare the results of ten algorithm runs. The mean time to convergence T_C is used as an approximation of complexity and is given by:

$$T_C = \frac{g'}{g_{MAX}} T_M \quad (10)$$

where g' is the average of generations required to convergence, g_{MAX} is the generations upper limit used as stop criteria, and T_M is the running time average. Table 5 shows computational results comparing the performance of four GAs with the BRKGA. The GAs used in the comparisons differ from each other in the type of solution encoding. They are called *conventional*, *improved*, *SSE*, and *ASE*. The number of generations used as stop criteria is set to 50 for all the GA implementations. The population size of the BRKGA was set to 100 individuals.

Table 5 – 33-Node system comparative results between GA variants for ten independent runs

| GA | Results | | | | T_C |
|--------------|---------|--------|---------|-------|-------|
| | Best | | Average | | |
| | g' | $t(s)$ | g' | T_M | |
| Conventional | 30 | 30.0 | 13.7 | 27.0 | 7.34 |
| Improved | 40 | 76.0 | 31.7 | 62.7 | 39.75 |
| SSE | 29 | 78.0 | 27.2 | 61.0 | 33.18 |
| ASE | 0 | 89.0 | 8.4 | 107.5 | 18.06 |
| BRKGA | 18 | 24.9 | 25.6 | 37.1 | 19.04 |

Only the BRKGA and ASE were robust enough to reach the optimal solution in 100% of the executions. Thus, T_C is an effective comparative metric only for ASE and BRKGA, since the other methods did not achieve the same results in all runs. These results attest the efficiency and robustness of the BRKGA encoding. The BRKGA also outperforms the other GAs in terms of algorithm running times due to the reduced size of solution encoding. Other technical works also compare solution quality, but the best known solution (BKS) for the 33-node system is the one found by the BRKGA, as confirmed by (Zhu 2002), (Ramos et al. 2005) and (Enacheanu et al. 2008).

5.2. 69-node Test System

For five independent runs of the algorithm, the BRKGA was able to find the optimal solution in an average CPU time of 26.7 seconds. The BRKGA did not use any form of heuristic spark, which is a means of igniting the search engine of GAs by inserting an individual of better fitness into the population at the beginning of the search process.

Fig. 10 shows a convergence characteristic of the algorithm to find the optimal solution for an independent run of the BRKGA on the 69-node system using a population of 100 individuals. Only 5 generations and 10 seconds were needed to reach the optimal solution.

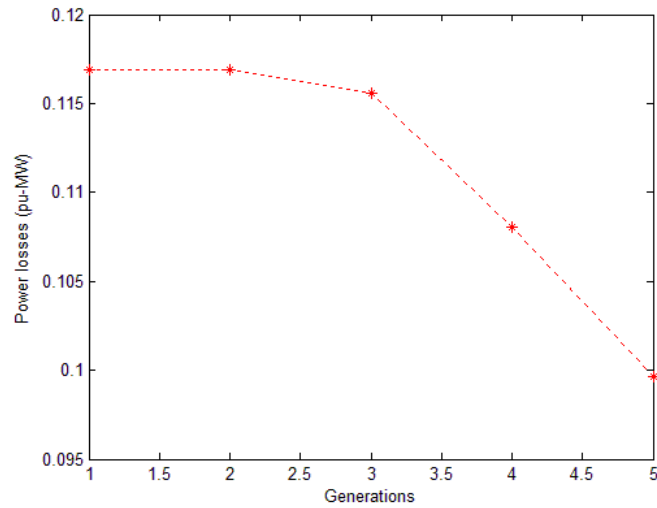


Fig. 10 Convergence characteristic for a particular run of the algorithm for the 69-node system

Fig. 11 shows a time to target plot for the BRKGA using a population of 80 individuals and 100 individuals. It can be seen that there is a 49% probability that the algorithm will find the optimal solution under 29.13 seconds using a smaller population. The bigger population accounts for longer running times according to Fig. 11 but the number of generations to reach the optimum is reduced in comparison to the smaller sized population.

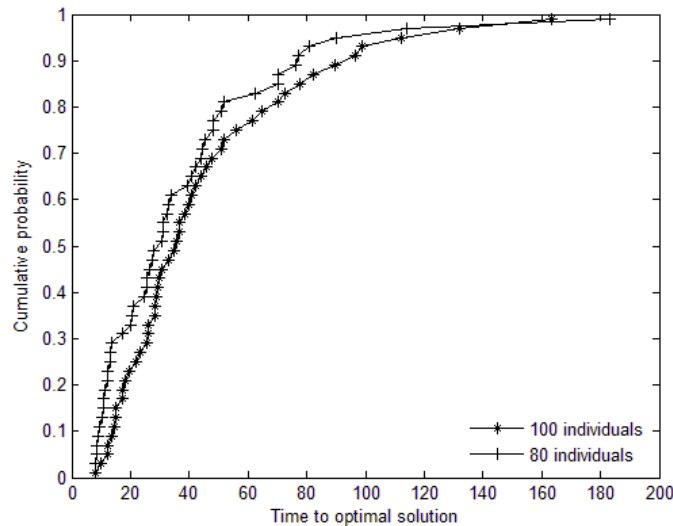


Fig. 11 Time to target plot to find the optimal solution for the 69-node system

For the 69-node system, comparisons were carried out with respect to a conventional GA and an improved immune genetic algorithm (Wang et al. 2009). The algorithm was executed 100 times to enable

comparisons with results presented in the article. The metric T_C was computed accordingly. Table 6 shows computational results comparing the performance of two GAs with the BRKGA.

Table 6 – 69-Node system comparative results between GA variants for one hundred independent runs

| GA | Results | | | T_C |
|--------------|----------------|----------------|-------|-------------------------|
| | <i>Best</i> | <i>Average</i> | | |
| | g' | g' | T_M | |
| Conventional | 10 | 57.30 | 20.08 | 8.52 |
| IIGA | 3 | 7.61 | 3.32 | 0.19 |
| BRKGA | 5 | 21.28 | 39.29 | 19.04 |

The performance of the BRKGA for the 69-Node system can also be compared with respect to average convergence generation with an artificial immune algorithm (Wenchuan and Jiaju 2006), a hybrid genetic particle swarm optimization algorithm (Zhang et al. 2007), and a hybrid intelligent algorithm (Zifa et al. 2005). These algorithms converge, on average, in 34, 43 and 30.20 generations, respectively, whereas the BRKGA converges in 21.28 generations. These results show that the BRKGA is among the best metaheuristics for DNR problems and its performance can be substantially improved with the application of heuristic mechanisms such as restart procedures to prevent convergence to local optima (Gonçalves and Resende 2010) and heuristic spark (Swarnkar et al. 2011).

6. Conclusions and Future Developments

This paper proposes a biased random key genetic algorithm to solve a classic power system combinatorial optimization problem called distribution network reconfiguration. This is the first application of a BRKGA to a DNR problem in the technical literature. The BRKGA produces solutions that are called random keys inside the a real-valued interval (0,1) and a decoder is used to map these solutions into the problem's search space. The decoder is built applying a set of rules derived from graph theory that guarantees the generation of only feasible solutions to the problem. This procedure reduces the size of the search space and avoids time consuming feasibility checks on solutions. Elitism is used efficiently by always using an elite parent on the crossover phase and by copying the entire elite set of one generation onto the next generation. Time to target plots for the BRKGA using different population sizes were drawn to assess the performance of the algorithm. The bigger sized populations consumed more CPU times but were able to converge to the optimum in fewer generations in comparison to the smaller populations.

The algorithm performed extremely well on two benchmark test systems widely used for DNR studies, being able to find the optimal solution in a reduced number of generations most of the time. The BRKGA was compared with other GA implementations found in the technical literature using an efficient metric. One can conclude, based on the comparisons, that the BRKGA is among the best metaheuristics for DNR problems. The two test systems used are considered medium sized problems and future investigations will be done on larger systems to assess the performance of the method using multiple populations, heuristic improvement mechanisms and GPU parallel computing. The use of multiple populations permits the exchange of information regarding good individuals found in each of these populations and parallel processing can reduce the computation time to find the optimal solution on hard problem instances.

References

- Roque, L.A.C., Fontes, D.B.M.M., Fontes F.A.C.C.: A hybrid biased random key genetic algorithm approach for the unit commitment problem. *J. of Combinatorial Optimization* 28, 140–166 (2014)
- Faria Jr., H., Binato, S., Resende, M., Falcão, D.M.: Transmission network design by a greedy randomized adaptive path relinking approach. *IEEE Trans. Power Syst.* 20, 43–49 (2005)
- Civanlar, S., Grainger, J., Lee, S.: Distribution feeder reconfiguration for loss Reduction. *IEEE Trans. Power Delivery* 3(3), 1217–1223 (1988)
- Delbem, A.C.B., Carvalho, A.C.P.L.F.D., Policastro, C.A., Pinto, A.K.O., Honda, K., Garcia, A.C.: Node-Depth Encoding for Evolutionary Algorithms Applied to Network Design. *Proc. GECCO* 1, 678-687 (2004)
- Swarnkar A., Gupta, N., Niazi, K.R.: A novel codification for meta-heuristic techniques used in distribution network reconfiguration. *Electric Power Syst. Research* 7(81), 1619–1626 (2011)
- Carreño, E.M., Romero, R., Padilha-Feltrin, A.: An efficient codification to solve distribution network reconfiguration for loss reduction problem. *IEEE Trans. Power Syst.* 4(23), 1542–1551 (2008)
- Gonçalves, J., Resende, M.: Biased random-key genetic algorithms for combinatorial optimization. *J. of Heuristics* 17(5), 487–525 (2010)
- Yang, Xin-She: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley (2010)
- Festa, P., Pardalos, P., Pitsoulis, L., Resende, M.: GRASP with path-relinking for the weighted MAXSAT problem. *ACM J. of Experimental Algorithmics* 11, 1–16 (2006)
- Nara, K., Shiose, A., Kitagawa, M., Ishihara, T.: Implementation of genetic algorithm for distribution systems loss minimum re-configuration. *IEEE Trans. Power Syst.* 3(7), 1044–1051 (1992)
- Carreno, E. M., Romero, R., Padilha-Feltrin, A.: An efficient codification to solve distribution network reconfiguration for loss reduction problem. *IEEE Trans. Power Syst.* 4(23), 1542–1551 (2008)
- Li, Z., Chen, X., Yu, K., Sun, Y., Liu, H.: A hybrid particle swarm optimization approach for distribution network reconfiguration problem. *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 20-24 (2008)
- Santos, A.C., Delbem, A.C.B., London, J.B.A.Jr., Bretas, N.G.: Node-Depth encoding and multiobjective evolutionary algorithm applied to large-scale distribution system reconfiguration. *IEEE Trans. Power Syst.* 3(25), 1254–1265 (2010)
- Lavorato, M., Franco, J.F., Rider, M.J., Romero, R.: Imposing radiality constraints in distribution system optimization problems. *IEEE Trans. Power Syst.* 1(27), 172–180 (2012)
- Braz, H.D.M., Souza, B.A.: Distribution network reconfiguration using genetic algorithms with sequential encoding: subtractive and additive approaches. *IEEE Trans. Power Syst.* 2(26), 582–593 (2011)
- Schmidt, H.P., Ida, N., Kagan, N., Guaraldo, J.C.: Fast reconfiguration of distribution systems considering loss minimization. *IEEE Trans. Power Syst.* 3(20), 1311–1319 (2005)
- Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. *ORSA J. on Computing* 6, 154–160 (1994)

Spears, W.M., DeJong K.A.: On the virtues of parameterized uniform crossover. Proceedings of the Fourth International Conference on Genetic Algorithms, 230–236 (1991)

Gonçalves, J.F., Resende, M., Toso, R.F.: An experimental comparison of biased and unbiased random-key genetic algorithms. *Pesquisa Operacional* 34, 143–164 (2014)

Wang, C., Zhao, A., Dong, H., Li, Z.: An improved immune genetic algorithm for distribution network reconfiguration. *IEEE International Conference on Information Management*, (2009)

Savier, J.S., Das, D.: Impact of network reconfiguration on loss allocation of radial distribution systems. *IEEE Trans. Power Del.* 4(22), 2473–2480 (2007)

Baran, M. E., Wu, F. F.: Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Trans. Power Del.* 2(4), 1401–1407 (1989)

Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J.: Matpower: Steady-state operations, planning and analysis tools for power systems research and education. *IEEE Trans. Power Syst.* 1(26), 12–19 (2011)

Aiex, R.M., Resende, M., Ribeiro, C.C.: TTTPLOTS: A perl program to create time-to-target plots. *Optimization Letters* 1, 355–366 (2007)

Zhu, J. Z.: Optimal reconfiguration of distribution network using the refined genetic algorithm. *Elsevier Elect. Power Syst. Res.* 62, 37–42 (2002)

Ramos, E. R., Expósito, A. G., Santos, J. R., Iborra, F. L.: Path-based distribution network modeling: Application to reconfiguration for loss reduction. *IEEE Trans. Power Syst.* 2(20), 556–564 (2005)

Enacheanu, B., Raison, B., Caire, R., Devaux, O., Bienia, W., Hadjsaid, N.: Radial network reconfiguration using genetic algorithm based on the matroid theory. *IEEE Trans Power Syst.* 1(23), 186–195 (2008)

Rama Rao, P. V. V., Sivanagaraju, S.: Radial distribution network reconfiguration for loss reduction and load balancing using plant growth simulation algorithm. *Int. J. on Elect. Eng. and Inform.* 2(4), 266–277 (2010)

Wenchuan, M., Jiaju, Q.: An artificial immune algorithm to distribution network reconfiguration. *Proceedings of the CSEE* 26(17), 25–29 (2006)

Zhang, C. Q., Zhang, J. J., Gu X. H.: The application of hybrid genetic particle swarm optimization algorithm in the distribution network reconfigurations multi-objective optimization. *Third Int. Conf. on Natural Computation proceeding.* (2), 455–459 (2007)

Zifa, L., Shaoyun, G., Yixin, Y.: A hybrid intelligent algorithm for loss minimum reconfiguration in distribution networks. *Proceedings of the CSEE* 5(15), 73–78 (2005)