



Artificial immune algorithm applied to distribution system reconfiguration with variable demand



Simone S.F. Souza^{a,*}, Ruben Romero^{a,1}, Jorge Pereira^{b,c,2,3}, João T. Saraiva^{b,d,2,4}

^a Electrical Engineering Department, Campus of Ilha Solteira, Unesp, Univ Estadual Paulista, Av. Brazil 56, 15385-000 Ilha Solteira, SP, Brazil

^b INESC TEC, Av. Dr. Roberto Frias, 378, 4200-465 Porto, Portugal

^c Faculty of Economics, University of Porto, Av. Dr. Roberto Frias, s/n, 4200-464 Porto, Portugal

^d Faculty of Engineering, University of Porto, Av. Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:

Received 13 August 2015

Received in revised form 3 April 2016

Accepted 11 April 2016

Keywords:

Distribution system reconfiguration

Variable demands

Artificial immune systems

Copt-aiNet

ABSTRACT

This paper presents a new methodology to solve the reconfiguration problem of electrical distribution systems (EDSs) with variable demand, using the artificial immune algorithm Copt-aiNet (Artificial Immune Network for Combinatorial Optimization). This algorithm is an optimization technique inspired by immune network theory (aiNet). The reconfiguration problem with variable demand is a complex problem of a combinatorial nature. The goal is to identify the best radial topology for an EDS in order to minimize the cost of energy losses in a given operation period. A specialized sweep load flow for radial systems was used to evaluate the feasibility of the topology with respect to the operational constraints of the EDS and to calculate the active power losses for each demand level. The algorithm was implemented in C++ and was evaluated using test systems with 33, 84, and 136 nodes, as well as a real system with 417 nodes. The obtained results were compared with those in the literature in order to validate and prove the efficiency of the proposed algorithm.

© 2016 Elsevier Ltd. All rights reserved.

Introduction

In recent years, many investments have been directed toward electrical distribution systems (EDSs) in order to modernize and automate their operations, and increase profitability. These modernization efforts have aimed at meeting several requirements, such as improving the reliability, efficiency, and security of the system, and satisfying strict regulatory rules. In light of these aims, much research is currently being conducted to solve the distribution system reconfiguration (DSR) problem using new approaches and techniques.

The DSR problem consists of identifying the best radial topology for the EDS through the opening and closing of switches in order to optimize an objective function, which is typically the minimization of active power losses. This is subject to the technical operational constraints of the EDS, such as the condition of radiality, nodal

voltage limits, branch current capacity limits, and the first and second Kirchhoff's laws (active and reactive power balance). Apart from this typical formulation, the DSR can also be performed primarily to improve the voltage levels, to maintain or enhance the reliability of the network, to help network operators to isolate faults more quickly, and to help prepare plans for preventive maintenance actions [1].

The DSR problem is a complex and combinatorial problem that can be modeled as a mixed-integer nonlinear programming (MINLP) problem [2]. Thus, as the size of the EDS increases, it becomes more difficult to solve this problem using exact methods. As a result, intelligent optimization techniques, such as heuristic and meta-heuristic algorithms, artificial neural networks, and artificial immune systems, among others, are increasingly being applied to solve this problem. These techniques include effective strategies that reduce the search space, enabling the best solution or at least good-quality solutions to be found.

The DSR problem has been widely addressed in the literature, yet most of the approaches have only considered one demand level at each consumption node, corresponding to the peak level. However, since demand varies over time, some authors have addressed the problem considering the variation in demand, for example, separating the demand into the 24 h of the day and following the daily variation of the demand curve. Accordingly, the DSR problem

* Corresponding author. Tel.: +55 18 3743 1236.

E-mail addresses: simonefrutuoso.mat@gmail.com (S.S.F. Souza), ruben@dee.feis.unesp.br (R. Romero), jpereira@inesctec.pt (J. Pereira), jsaraiva@fe.up.pt (J.T. Saraiva).

¹ Tel.: +55 18 3743 1236.

² Tel.: +351 222094000.

³ Tel.: +351 225571100.

⁴ Tel.: +351 225081400.

with variable demand aims at identifying an unique radial topology that operates at different demand levels, while minimizing the cost of energy losses in a given operation period.

The literature includes references that have used several approaches to address the DSR problem considering fixed demand, such as heuristic algorithms [2,3] and meta-heuristics including genetic algorithms [4], simulated annealing [5], tabu search [6], ant colony [7], artificial neural networks [8], and artificial immunological algorithms [9]. On the other hand, few references have considered the DSR problem with variable demand, as in [10–13].

In this paper, the Copt-aiNet (Artificial Immune Network for Optimization) [14] is proposed to solve the DSR problem with variable demand. This technique is inspired by biological immune systems, providing a computational emulation of the main properties and functionalities of the organism with respect to immune network theory (aiNet). To evaluate the affinity of each antibody, a specialized sweep load flow for radial EDSs is used [15] for each demand level in order to estimate the cost of energy losses in each operation period.

Notably, in this paper, the DSR problem is analyzed with variable demand, representing a natural extension of paper [9] in which the authors analyzed the DSR problem considering only fixed demand.

This paper presents results using test systems with 33, 84, and 136 nodes, and a real system with 417 nodes. These results are compared with the results available in the specialized literature in order to assess the efficiency of the proposed approaches.

Mathematical model for the DSR problem with variable demand

Considering a symmetrical and balanced system with a unique radial topology, the DSR problem with variable demand can be modeled as an MINLP problem, as described in (1)–(9) [16]:

$$\text{Min } v = \sum_{d \in \Omega_d} \sum_{ij \in \Omega_l} c_d \Delta_d [g_{ij} x_{ij} (V_{i,d}^2 + V_{j,d}^2 - 2V_{i,d} V_{j,d} \cos \theta_{ij,d})] \quad (1)$$

s. a.

$$Ps_{i,d} - Pd_{i,d} - \sum_{j \in \Omega_{bi}} (x_{ij} P_{ij,d}) = 0 \quad \forall i \in \Omega_b, \forall d \in \Omega_d \quad (2)$$

$$Qs_{i,d} - Qd_{i,d} - \sum_{j \in \Omega_{bi}} (x_{ij} Q_{ij,d}) = 0 \quad \forall i \in \Omega_b, \forall d \in \Omega_d \quad (3)$$

$$\underline{V} \leq V_{i,d} \leq \bar{V} \quad \forall i \in \Omega_b, \forall d \in \Omega_d \quad (4)$$

$$(P_{ij,d}^2 + Q_{ij,d}^2) \leq \overline{S_{ij,d}^2} * x_{ij} \quad \forall ij \in \Omega_l, \forall d \in \Omega_d \quad (5)$$

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

$$\sum_{(ij) \in \Omega_l} x_{ij} = n_b - 1 \quad (7)$$

In this formulation:

Ω_l is the set of circuits;

Ω_b is the set of nodes;

Ω_{bi} is the set of nodes connected at node i ;

Ω_d is the set of demands;

c_d^l is the cost of energy losses at demand level d ;

Δ_d is the duration of demand level d ;

g_{ij} is the conductance of circuit ij ;

$V_{i,d}$ is the voltage magnitude at node i at demand level d ;

$\theta_{ij,d}$ is the phase angle difference between nodes i and j at demand level d ;

b_{ij} is the susceptance of circuit ij ;

$P_{ij,d}$ is the active power flow that goes from node i to node j at demand level d ;

$Q_{ij,d}$ is the reactive power flow that goes from node i to node j at demand level d ;

$Ps_{i,d}$ is the active power supplied by the substation at node i at demand level d ;

$Qs_{i,d}$ is the reactive power supplied by the substation at node i at demand level d ;

$Pd_{i,d}$ is the active power demanded at node i at demand level d ;

$Qd_{i,d}$ is the reactive power demanded at node i at demand level d ;

\underline{V} is the minimum voltage magnitude;

\bar{V} is the maximum voltage magnitude;

$\overline{S_{ij,d}}$ is the maximum apparent power of circuit ij at demand level d ;

n_b is the number of nodes in the system;

x_{ij} is the binary decision variable that represents the state (connected or disconnected) of circuit ij .

In this formulation, (1) represents the objective function of the DSR problem with variable demand. It corresponds to the cost of energy losses to be minimized in the EDS. The mathematical model also considers physical constraints, the specifications of system components, and operational conditions.

Constraints (2) and (3) represent the active and reactive nodal balance equations in which the active and reactive power flows $P_{ij,d}$ and $Q_{ij,d}$ are calculated using (8) and (9), respectively.

$$P_{ij,d} = V_{i,d}^2 g_{ij} - V_{i,d} V_{j,d} (g_{ij} \cos \theta_{ij,d} + b_{ij} \sin \theta_{ij,d}) \quad (8)$$

$$Q_{ij,d} = -V_{i,d}^2 b_{ij} - V_{i,d} V_{j,d} (g_{ij} \sin \theta_{ij,d} - b_{ij} \cos \theta_{ij,d}) \quad (9)$$

Constraint (4) represents the voltage magnitude limits for each node of the EDS, as defined by regulatory standards. The power flow in each circuit is limited by (5). Constraint (6) corresponds to the binary nature of the decision variables, according to which x_{ij} can take two values as follows: when it is 0 (zero), circuit ij is open (or disconnected), and when it is 1 (one), circuit ij is closed (or connected).

Constraint (7) presents one of the necessary conditions to guarantee the radial operation of the EDS, namely, that a solution to the problem must have $(n_b - 1)$ active circuits. The other necessary condition is that the system must be connected (i.e., all nodes connected). This condition is guaranteed by (2) and (3). Thus, satisfying (2), (3), and (7) ensures that any feasible solution, as well as the optimal solution, will be radial [17].

Artificial immune systems

Artificial immune systems (AISs) are compounds of a set of intelligent algorithms inspired by the functioning of biological immune systems. Similar to other bio-inspired methods and meta-heuristics, the main objective of AISs is to solve complex problems that cannot be addressed in a timely manner by classical optimization methods [9,18].

In the literature, AISs have been widely used to solve optimization problems. In this context, the immune algorithms most frequently mentioned in the literature are the CLONALG algorithm (Clonal Selection Algorithm) [19], the aiNet (Artificial Immune Network) [20], the Opt-aiNet algorithm [18], the B-cell (BCA) algorithm [21], and the Copt-aiNet algorithm [14].

In this paper, the Copt-aiNet is used to solve the DSR problem with variable demand.

Copt-aiNet algorithm

The Copt-aiNet was originally proposed in [14]. This technique is an extension of the aiNet to solve combinatorial optimization problems, and can be described in the following steps:

- 1 Generate an initial population (P) with N antibodies;
- 2 **While** (stopping criterion is not satisfied) **do**
- 3 **While** (stabilizing criterion is not satisfied) **do** (clonal expansion)
- 4 Evaluate the affinity (objective function) for each antibody in population P ;
- 5 Select the n best antibodies in population P , obtaining the set $P_{\{n\}}$;
- 6 Reproduce the n best antibodies selected $P_{\{n\}}$, generating a population (C) with Nc clones;
- 7 Submit the population of clones (C) to a hypermutation process, obtaining a subpopulation (C^*) of mutated antibodies;
- 8 Evaluate the affinity of each antibody that belongs to the subpopulation (C^*), re-select the n best antibodies ($C_{\{n\}}^*$), and add them to population P ;
- 9 Replace d low-affinity antibodies with new antibodies ($P_{\{d\}}$) (metadynamic process);
- End while**
- 10 Eliminate all memory antibodies that have a resemblance larger than threshold S (similarity rate) in the clonal suppression;
- 11 If the size of population P is lower than N , generate antibodies to complete the population (population control);
- 12 If there are no improvements during k iterations, execute weak mutation for all antibodies in population P ;
- 13 Apply strong mutation to the n best antibodies in the population (memory antibodies $P_{\{M\}}$);
- End while**

The number of clones Nc used in the cloning process (step 6) for each antibody is given by (10) [18], in which β is a cloning factor between 0 and 1, N is the total number of antibodies in population P , and $round(\cdot)$ is the rounding operator to the nearest integer.

$$N_c^i = round\left(\frac{\beta N}{i}\right) \quad (10)$$

The mutation rate (α) of each clone is defined by (11) [18], in which ρ is a damping control parameter for the exponential function and f^* is the normalized value of affinity f , which is calculated according to (12) for minimization problems and according to (13) for maximization problems [9]. In these expressions, f_{max} is the greatest affinity and f_{min} is the lowest affinity.

$$\alpha = \exp(-\rho f^*) \quad (11)$$

$$f^* = \frac{f}{f_{max}} \quad (12)$$

$$f^* = \frac{f_{min}}{f} \quad (13)$$

Each clone undergoes a mutation process, as proposed in [22], in which m is the number of mutations that each antibody will suffer, α is the mutation rate, and $N(0, 1)$ is a Gaussian random variable with a mean of 0 and standard deviation of 1.

$$m = round(\alpha * N(0, 1)) \quad (14)$$

The stabilization criterion for population P depends on each problem, but commonly, when the memory set $P_{\{M\}}$ does not change over a specified number of iterations, the population is then considered to be stable.

Codification of the DSR problem with variable demand

In this work, the coding of the DSR problem proposed in [23] is used, wherein the EDS is represented as a tree (graph theory) composed of an array of arcs (branches). The encoding vector has a dimension of n_l (number of branches) and represents the whole electrical system storing the branches of the system. In this coding scheme, the first $(n_b - 1)$ elements of the vector indicate the branches of the radial topology (set $N1$), and the branches between positions n_b and n_l (set $N2$) indicate the connection elements (branches of the radial configuration), as illustrated in Fig. 1.

For example, a topology of a test system with 14 nodes is shown in Fig. 2. This topology can be represented as (15), where all elements between 1 and $(n_b - 1)$ belong to the network topology (continuous line), and all others, from n_b to the end of the vector, are connection elements that are disconnected in this configuration (dotted line). This representation is not fixed, given that the same topology can be represented as (16), which can help to diversify the search depending on the kinds of operators being used.

$$[C_1 \ C_5 \ C_{10} \ C_2 \ C_7 \ C_{11} \ C_3 \ C_6 \ C_{12} \ C_4 \ C_8 \ C_{13} \ C_9 \ (C_{14} \ C_{15} \ C_{16})] \quad (15)$$

$$[C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6 \ C_7 \ C_8 \ C_9 \ C_{10} \ C_{11} \ C_{12} \ C_{13} \ (C_{14} \ C_{15} \ C_{16})] \quad (16)$$

This codification ensures that the initial population only includes radial topologies. Therefore, it is possible to develop search operators that preserve the radial topology, avoiding this type of infeasibility.

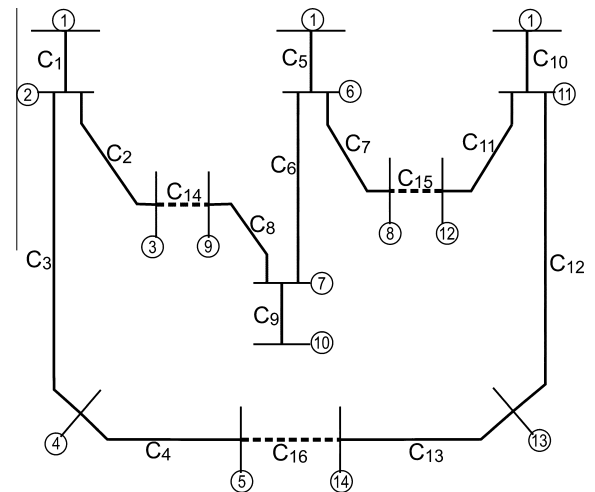


Fig. 2. Example of codification of the 14-node system.

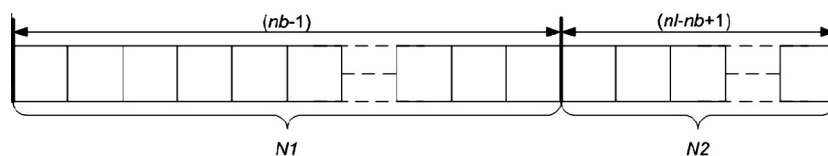


Fig. 1. Codification vector for the 14-node system.

Methodologies to solve the DSR problem with variable demand

In this section, the Copt-aiNet algorithm is applied to solve the DSR problem with variable demand.

Application of the Copt-aiNet algorithm

The following sections describe the application of the Copt-aiNet to solve the DSR problem with variable demand.

Strategy to generate the initial population

To generate the initial population (P) for the Copt-aiNet, the Prim heuristic presented in [23] is used. This heuristic considers the codification of the problem presented in the previous section and is described in the following steps.

-
- 1 D is the set of nodes present in the current configuration;
 - 2 $D = \phi, N1 = \phi, N2 = \phi$;
 - 3 Add the node to substation (root node) to D ;
 - 4 Find all branches with a vertex in D ;
 - 5 Select a branch, identified in the previous step, to enter into the topology (randomly);
 - 6 If the selected branch generates a loop in the topology, include it in $N2$, otherwise include the opposite vertex in D and the branch in $N1$;
 - 7 Return to step 3 until all branches are selected.
-

At the end of the process, the vector $[N1, N2]$ represents a topology of the EDS. This strategy always generates topologically feasible solutions (radial), contributing to the efficiency and diversity of the algorithms proposed in this paper.

Affinity evaluation operator

In this step, the affinities of each antibody (candidate solution) in population P are calculated. This value represents the cost of the energy losses associated with a topology in the operation period for all demand levels under analysis. In order to obtain the cost of losses, a specialized sweep load flow for radial EDSs [15] is run for each demand level. To analyze several demand levels, the power flow is run for each demand, and then expression (17) is used to obtain the affinity value of the associated antibody. In this expression, Nd is the number of demand levels to be analyzed, K_i is the cost of energy losses for demand level i , and T_i and P_i are the duration and value, respectively, of the active power losses associated with demand level i .

$$f = \sum_{i=1}^{Nd} [K_i * T_i * P_i] \tag{17}$$

Selection operator

In the selection operator, the n best antibodies, named memory antibodies, are selected generating a subpopulation of antibodies called $P_{(n)}$. To select the n best antibodies, the algorithm takes into account the affinity values, ordering them from the lowest affinity value (the one of highest quality) to the highest affinity value (the one of poorest quality). The subpopulation $P_{(n)}$ is submitted to the cloning and hypermutation processes, as well as to the re-selection process aimed at inserting the n best matured antibodies into the population (P).

Cloning operator

The cloning operator generates a subpopulation of clones (C). The subpopulation C is composed of Nc clones obtained through

the antibodies in subpopulation $P_{(n)}$. The number of clones that each antibody can generate is calculated using (10).

Hypermutation operator

After obtaining a subpopulation of clones (C), the hypermutation operator is carried out to generate a new subpopulation of matured clones (C^*). Initially, it is necessary to calculate the number of mutations of an antibody i in the iterative process of the Copt-aiNet using (14). Eq. (11) defines the mutation rate (α), and (13) establishes the normalized affinity (f^*). Following these calculations, a random mutation is performed as described in the following steps.

-
- 1 Read C and obtain Nc (number of clones in population C);
 - 2 **for** $i = 1$ until Nc **do**
 - 3 Calculate the number of mutations (m) for antibody i ;
 - 4 **for** $j = 1$ until m **do**
 - 5 Select randomly a circuit $l \in N2$ of antibody i . This position l represents a disconnected circuit;
 - 6 Connect/close the selected circuit and identify the loop formed. Choose a circuit in the loop formed and replace it with circuit l ;
 - 7 **end for**
 - 8 Store the matured antibody i in C^* ;
 - 9 **end for**
 - 10 **return** C^* ;
-

Fig. 3 presents an illustrative example of the hypermutation process. In this example, the antibody shown in (16) is used. In this process, a circuit belonging to set $N2$, i.e., a disconnected circuit, is chosen at random. Then, this circuit is connected and a loop is formed in the system. This loop should be identified and stored. In the sequence, another circuit belonging to the loop formed is chosen at random to replace the initial circuit. Finally, the circuits are exchanged (circuit 15 is replaced by circuit 11), generating a matured antibody. This hypermutation strategy always generates topologically feasible antibodies.

Metadynamic operator

In the metadynamic operator, the worst d antibodies of the population (P) are replaced by new d antibodies generated randomly using the strategy presented in section ‘Strategy to generate the initial population’.

Stopping criterion for the clonal expansion process

The clonal expansion process is executed until the population is stabilized and no more memory antibodies are modified over k generations.

Clonal suppression operator

In the clonal suppression operator, the memory set $P_{(M)}$ is evaluated by comparing the similarity between its antibodies. If the

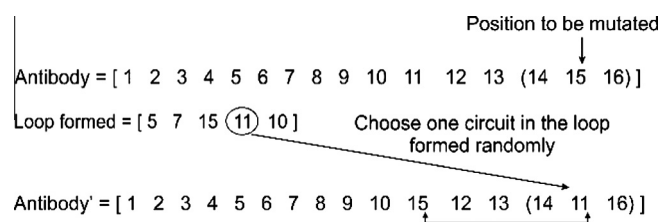


Fig. 3. Hypermutation example for the 14-node system.

similarity between a pair of antibodies is larger than a threshold S (similarity rate), then one of the antibodies is discarded through the clonal suppression process. This means that similar antibodies are eliminated in order to provide greater population diversity.

Control of the population's size

After the clonal suppression operator, the population size is checked to see whether it is lower than N . If so, new antibodies are generated randomly using the strategy presented in section 'Strategy to generate the initial population' in order to complete the population.

Weak mutation operator

The weak mutation operator is executed if there are no improvements in the affinity of the antibodies belonging to the memory set ($P_{\{M\}}$) over k iterations of the Copt-aiNet. In this case, the weak mutation occurs in all elements of population P through the hypermutation operator presented in Section 'Hypermutation operator'.

Strong mutation operator

The strong mutation operator is performed on each generation of the Copt-aiNet in order to conduct an intensified search (local improvement) of the antibodies in the memory set ($P_{\{M\}}$). This process uses the local search heuristic proposed in [23] and detailed in [9].

Stopping criterion

The Copt-aiNet stops if the affinity value of the best antibody in the population does not change for a specified number of iterations, and the average affinity value of the antibodies in the population does not change by more than a specified percentage over a given number of iterations. If the above two conditions hold simultaneously, then the algorithm ends, indicating that a final solution has been obtained. If not, the iteration counter is increased by 1. If the maximum number of iterations is reached, then stop. If not, return to Step 2 of the algorithm.

Results

This section presents the results obtained from applying the Copt-aiNet to the test systems with 33 [3], 84 [24], and 136 [6] nodes, and to a real system with 417 [25] nodes. The algorithm was implemented in C++ 6.0[®] [26]. All tests were carried out using a PC with an Intel Core i7 3.1 GHz processor.

Configuration of the used demand levels

The tests considered an operation period of 24 h, and for each hourly period, a demand level was specified for each node in the test system. The demand levels were organized into three sets: (a) residential, (b) commercial, and (c) industrial. For each demand level, a typical load diagram was specified, as illustrated in Fig. 4 (a–c).

In order to obtain a specific demand level, the load factor of the final consumer at each node was multiplied by the active and reactive power associated with the typical load curves. The selection of the type of consumer at each node of the EDS was done at random, assuming that 60% of the consumers were residential, 25% were commercial, and 15% were industrial. The type of consumer defined for each node of each EDS can be found in [13].

Table 1 presents the load factors associated with each type of consumer over the 24 h of the operation period and the cost of energy losses. Fig. 5 shows how the cost of energy losses varied across the 24-h period. In each hourly period, these costs were

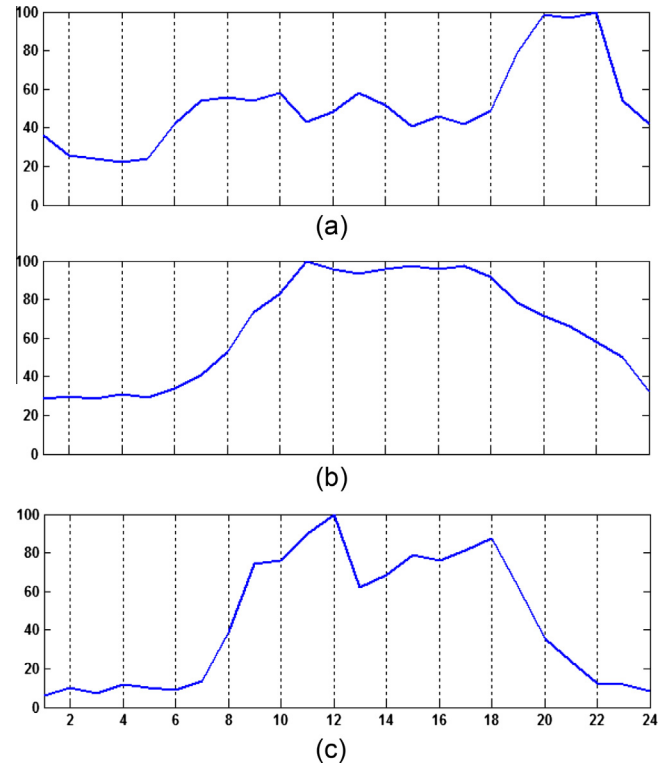


Fig. 4. Typical active power demand load curves.

Table 1

Load factors of the consumers and cost of energy losses.

Demand level	Consumer kind			Cost (US\$/kW h)
	Residential	Commercial	Industrial	
1	0.3600	0.2838	0.0625	0.0650
2	0.2600	0.2973	0.1000	0.0650
3	0.2400	0.2838	0.0750	0.0650
4	0.2200	0.3108	0.1188	0.0650
5	0.2400	0.2938	0.1000	0.0650
6	0.4200	0.3378	0.0875	0.0650
7	0.5400	0.4054	0.1375	0.1100
8	0.5600	0.5270	0.3875	0.1100
9	0.5400	0.7297	0.7438	0.1100
10	0.5800	0.8311	0.7625	0.1100
11	0.4300	1.0000	0.9000	0.1100
12	0.4800	0.9595	1.0000	0.1100
13	0.5800	0.9324	0.6188	0.1100
14	0.5200	0.9595	0.6875	0.1100
15	0.4100	0.9730	0.7875	0.1300
16	0.4600	0.9595	0.7625	0.1300
17	0.4200	0.9730	0.8125	0.1300
18	0.4900	0.9189	0.8750	0.1300
19	0.7900	0.7838	0.6188	0.1500
20	0.9840	0.7162	0.3563	0.1500
21	0.9700	0.6622	0.2375	0.1500
22	1.0000	0.5811	0.1250	0.0650
23	0.5400	0.5000	0.1188	0.0650
24	0.4200	0.3229	0.0832	0.0650

multiplied by the active losses calculated by the power flow routine in order to obtain the value of the objective function associated with each candidate topology.

Parameters of the Copt-aiNet

The results for all of the systems were obtained using the parameters shown in Table 2. These parameters were established after running the algorithm in a number of trial tests.

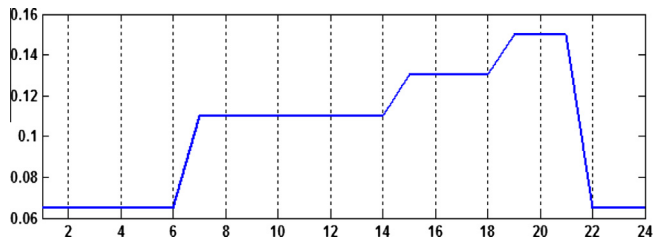


Fig. 5. Cost of energy losses (US\$/kWh).

Table 2
Parameters.

Parameters	33-, 84-, 136-node systems	417-node system
N	50	60
β	0.3	0.3
ger	30	70
n	10	30
d	1	1
ρ	4	5
S	80%	80%
est	2 ger	2 ger
k	2 ger	2 ger

In Table 2, N is the number of antibodies in the population, β is the cloning factor, ger is the number of generations, n is the number of memory antibodies, d is the number of antibodies to be replaced in the metadynamic process, ρ is the damping control parameter for the exponential function, S is the similarity rate, est is the criterion for the stability of the population, and k is the weak maturing criterion.

It is noteworthy that the parameters used in the Copt-aiNet to solve the test systems differ only in the number of ger , the ρ value, and the number of memory antibodies (n), which demonstrates the robustness of the proposed methodology.

33-, 84-, 136-, and 417-node systems

The well-known 33-, 84-, and 136-node test systems and the 417-node real system had nominal voltages of 12.66 kV, 11.40 kV, 13.80 kV, and 11.00 kV, respectively.

Table 3 shows the initial and final topologies (i.e., before and after reconfiguration) found using the Copt-aiNet, along with the cost of energy losses for each tested EDS. For all of the tests, the Copt-aiNet led to a reduction between 10% and 31% in the cost of energy losses.

In order to evaluate the performance of the Copt-aiNet, it was run 30 times. In all runs, the algorithm found the best solutions for all systems, but with a different number of iterations and computational time.

Tables 4 and 5 present the number of iterations and the computational time required for the 30 executions to find the solution for each test system. Table 4 includes data regarding $iTmax$, $iTmin$, $iTmean$, and dev , representing the larger, the lower, the mean, and the standard deviation of the execution time for the number of iterations needed to find the best solution.

Table 5 presents four types of data. $Tmax$ and $Tmin$ are the maximum and the minimum time, respectively, required to find the solution, while $Tmean$ is the mean time taken by the 30 executions, and dev is the standard deviation of that number.

According to the information presented in Tables 4 and 5, the Copt-aiNet found the solutions for all tested EDSs in all executions of the cross-reference test with low computational time, proving the efficiency and performance of the proposed algorithm.

Table 3
Results for the 33-, 84-, 136-, and 417-node systems.

System	Before reconfiguration		After reconfiguration		Reduction (%)
	Topology	Cost of Energy Losses (US\$)	Topology	Cost of Energy Losses (US\$)	
33	33-34-35-36-37	187.86	7-9-14-28-32	128.81	31.43
84	84-85-86-87-88-89-90-91-92-93-94-95-96	456.41	7-34-39-63-72-83-84-86-88-89-90-92-95	410.53	10.05
136	136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156	288.50	7-38-51-54-84-90-96-106-118-126-135-137-138-141-144-145-147-148-150-151-155	256.89	10.95
417	1-5-15-16-26-31-53-54-55-75-82-94-96-97-106-107-119-136-138-154-155-156-168-169-177-179-194-195-201-207-211-214-219-241-256-258-282-297-302-314-321-354-359-362-364-385-388-395-396-404-407-423-424-426-431-436-445-446-449	637.88	1-2-13-15-16-26-31-40-41-50-59-73-82-94-96-97-111-115-136-146-150-155-156-158-163-168-169-178-179-190-191-194-195-209-230-254-256-267-270-294-310-321-354-362-385-389-392-395-403-404-423-424-426-436-437-439-446-449-466	529.66	16.96

Table 4
Number of iterations needed to find the solutions.

System	$iTmin$	$iTmax$	$iTmean$	dev
33	1	2	1.43	0.50
84	2	5	3.37	1.11
136	8	24	16.83	4.82
417	26	68	48.83	12.51

Table 5
Computational time needed to find the solutions.

System	$Tmin$ (s)	$Tmax$ (s)	$Tmean$ (s)	dev
33	0.158	0.523	0.365	0.108
84	3.236	6.458	5.142	0.691
136	15.589	29.258	21.891	4.481
417	158.245	214.152	183.687	17.613

Comparison with the literature

Table 6 presents the best solutions available in the literature for each test system. Comparisons were performed with the results considering fixed and variable demands. To obtain the value of the cost of energy for the fixed demands, the topologies presented in these papers were used along with the affinity function to obtain the value of the cost.

The results for the variable demands were compared with the results for the fixed demands in order to analyze the results from an operational point of view.

As can be seen in Table 6, the results for the variable demands had lower values for the cost of power losses than the results con-

Table 6
Comparison with the literature.

System	Topology	Cost of energy losses (US\$)	Ref.	Kind of analysis
33	7-9-14-28-32	128.81	[13]	Variable demand
	7-9-14-32-37	134.30	[9]	Fixed demand
84	7-34-39-63-72-83-84-86-88-89-90-92-95	410.53	[13]	Variable demand
	7-13-34-39-42-55-62-72-83-86-89-90-92	417.29	[9,17]	Fixed demand
136	7-38-51-54-84-90-96-106-118-126-135-137-138-141-144-145-147-148-150-151-155	256.89	[13]	Variable demand
	7-35-51-90-96-106-118-126-135-137-138-141-142-144-145-146-147-148-150-151-155	272.96	[7,17]	Fixed demand
417	1-2-13-15-16-26-31-40-41-50-59-73-82-94-96-97-111-115-136-146-150-155-156-158-163-168-169-178-179-190-191-194-195-209-230-254-256-267-270-294-310-321-354-362-385-389-392-395-403-404-423-424-426-436-437-439-446-449-466	529.66	[13]	Variable demand
	5-13-15-16-21-26-31-54-57-59-60-73-86-87-94-96-97-111-115-136-142-149-150-155-156-158-163-168-169-178-179-191-195-199-214-221-254-256-266-282-317-322-325-358-362-369-392-395-403-404-416-423-426-431-436-437-446-449-466	530.29	[9]	Fixed demand

sidering fixed demands. This is because considering variable demand offers a more realistic representation of the problem. As a result, a unique topology is obtained that can operate in all 24 demands, making it unnecessary to perform various reconfiguration operations. When considering fixed demand, the topology is best suited to a specific demand and would not be suitable for variable demand, because it would require many reconfiguration operations over an operating period. So, from an operational point of view, considering variable demand is more viable, and in general produces a lower cost of power losses than when considering fixed demand.

To compare the results with the literature, references [7,9,17] for fixed demands and the reference [13] for variable demands were used. References [7,9,17] present the best results available in the literature for the reconfiguration problem with fixed demand; for this reason, these works were used in the comparison. Ref. [13] was used because it presents results using the same electric power systems, and the same configurations and parameters of variable demands as used in this paper; however, the methodology used to solve the problem is different, as it uses four versions of the VNS meta-heuristic.

For the 33-, 84-, 136-, and 417-node systems, the Copt-aiNet algorithm found the best solutions reported in the literature, as presented in Table 6. From these comparisons, it can be concluded that the Copt-aiNet is an efficient method because its final solutions are consistent with the best solutions found in the specialized literature for the configurations presented in this paper.

Conclusions

In this paper, a new methodology was presented to solve the DSR problem with variable demand, using the artificial immune algorithm Copt-aiNet with the objective of minimizing the cost of the energy losses in the EDS.

According to the results, the Copt-aiNet algorithm was able to find the best solutions available in the literature for all tested systems due to the weak and strong mutation strategies that enhance the ability to search the solution space of the problem. These strategies enabled the algorithm to find the best solution in a smaller number of iterations and, consequently, with reduced computational time.

The Copt-aiNet algorithm showed very good stability and reliability in solving the DSR problem with variable demand; in all runs

of the algorithm in the cross-reference tests, the best solutions were found for all of the tested systems. The obtained results for all test systems were compared with those reported in the literature in order to evaluate the efficiency of the proposed method.

Finally, we conclude that the Copt-aiNet algorithm proposed to solve the DSR problem demonstrates good performance, efficiency, low computational time, and robustness.

Acknowledgments

The authors gratefully acknowledge INESC TEC, Porto, Portugal and CNPq/Brazil and CAPES-PDSE/Brazil (Proc. No. BEX 3660/14-1) for supporting this research.

References

- [1] Guimarães MAN, Lorenzetti JFC, Castro CA. Reconfiguration of distribution system for voltage stability margin enhancement using tabu search. In: Proc Powercon. p. 1556–61.
- [2] Merlin A, Back H. Search for a minimal-loss operating spinning tree configuration in an urban power distribution system. Power Syst Comput Conf 1975;1:1–18.
- [3] Baran ME, Wu FF. Network reconfiguration in distribution systems for loss reduction and load balancing. IEEE Trans Power Deliv New York 1989;4(2):1401–7.
- [4] Mendoza J, Lopez R, Morales D, Lopez E, Dessante P, Moraga R. Minimal loss reconfiguration using genetic algorithms with restricted population and addressed operators: real applications. IEEE Trans Power Syst 2006;21(2):948–54.
- [5] Chang HC, Kuo CC. Network reconfiguration in distribution systems using simulated annealing. Electr Power Syst Res Lausanne 1994;29(3):227–38.
- [6] Zhang D, Fu Z, Zhang L. An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems. Electr Power Syst Res Lausanne 2007;77(5–6):685–94.
- [7] Carpaneto E, Chicco G. Distribution system minimum loss reconfiguration in the hyper-cube ant colony optimization framework. Electr Power Syst Res 2008;78:2037–45.
- [8] Salazar H, Gallego R, Romero R. Artificial neural networks and clustering techniques applied in the reconfiguration of distribution systems. IEEE Trans Power Deliv New York 2006;21(3):1735–42.
- [9] Souza SSF, Romero R, Franco JF. Artificial immune networks Copt-aiNet and Opt-aiNet applied to the reconfiguration problem of radial electrical distribution systems. Electr Power Syst Res 2015;119:304–12.
- [10] Bueno EA, Lyra C, Cavellucci C. Distribution network reconfiguration for loss reduction with variable demands. In: Proc 2004 IEEE Latin America transmission and distribution conference. p. 384–9.
- [11] Queiroz LMO, Lyra C. Adaptive hybrid genetic algorithm for technical loss reduction in distribution networks under variable demands. IEEE Trans Power Syst 2009;24(1):445–53.
- [12] Inoue T, Takano K, Watanabe T, Kawahara J. Distribution loss minimization with Guaranteed error bound. IEEE Trans Smart Grid 2014;5(1):102–11.

- [13] Possagnolo LHF. Distribution systems reconfiguration operating in several demand levels through of the variable neighborhood search (Master's Thesis). Campus of Ilha Solteira, Unesp, Univ Estadual Paulista; 2015 [in Portuguese].
- [14] de Souza JS, Zuben FJV, de Castro LN, Sousa JS. An immune evolutionary algorithm for multiple rearrangements of gene expression data. *Genet Program Evolvable Mach* 2004;5:157–79.
- [15] Shirmohammadi D, Hong HW, Semlyen A, Luo GX. A compensation based power flow method for weakly meshed distribution and transmission networks. *IEEE Trans Power Syst* 1988;3(2):753–62.
- [16] Franco JF, Lavorato M, Rider MJ, Romero R. An efficient implementation of tabu search in feeder reconfiguration of distribution systems. In: *Proc IEEE PES general meeting*. p. 1–8.
- [17] Lavorato M, Franco JF, Rider MJ, Romero R. Imposing radiality constraints in distribution system optimization problems. *IEEE Trans Power Syst New York* 2012;27(1):172–80.
- [18] de Castro LN, Timmis J. An artificial immune network for multimodal function optimization. In: *Proc. of IEEE world congress on evolutionary computation*. p. 699–704.
- [19] de Castro LN, Von Zuben FJ. The clonal selection algorithm with engineering applications. In: *Proc of workshop on artificial immune systems and their applications*. p. 36–7.
- [20] de Castro LN, Zuben FJV. aiNet: an artificial immune network for data analysis. In: Abbas H, Sarker R, Newton C, editors. *Data mining: a heuristic approach*. Idea Group Publishing; 2001.
- [21] Kelsey J, Timmis J. Immune inspired somatic contiguous hypermutation for function optimisation. In: *Proc GECCO*. p. 207–18.
- [22] de Franca FO, Von Zuben FJ, de Castro LN. An artificial immune network for multimodal function optimization on dynamic environments. In: *Proc GECCO*, Washington, DC, USA. p. 289–96.
- [23] Carreño EM, Romero R, Feltrin AP. An efficient codification to solve distribution network reconfiguration for loss reduction problem. *IEEE Trans Power Syst New York* 2008;23(4):1542–51.
- [24] Chiou JP, Chang CF, Su CT. Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems. *IEEE Trans Power Syst New York* 2005;20(2):668–74.
- [25] Bernal-Agustin JL. Application of genetic algorithms to the optimal design of power distribution systems 346 f. Tesis. Zaragoza: University of Zaragoza; 1998. 1998.
- [26] Borland 6.0 Version. C++ Builder.