Distribution System Reconfiguration with Variable Demands Using the Clonal Selection Algorithm

Simone S. F. Souza, Ruben Romero, *Senior Member IEEE*
Electrical Engineering Department
UNESP – Univ. Estadual Paulista
Ilha Solteira, SP, Brazil
simonefrutuoso.mat@gmail.com, ruben@dee.feis.unesp.br

Jorge Pereira
INESC TEC and Faculty of Economics, University of Porto
Porto, Portugal
jpereira@inesctec.pt

J. T. Saraiva
INESC TEC and Faculty of Engineering, University of Porto
Porto, Portugal
jsaraiva@fe.up.pt

***Abstract*—This paper describes the application of the clonal selection algorithm to the reconfiguration problem of distribution networks considering non-uniform demand levels. The Clonal Algorithm, CLONALG, is a combinatorial optimization technique inspired in the immunologic bio system and it aims at reproducing the main properties and functions of this system. The reconfiguration problem of distribution networks with non-uniform demand levels is a complex problem that aims at identifying the most adequate radial topology of the network that complies with all technical constraints in every demand level while minimizing the cost of active losses along an extended operation period. This work includes results of the application of the Clonal algorithm to distribution systems with 33, 84 and 136 buses. These results demonstrate the robustness and efficiency of the proposed approach.**

***Index Terms*—Distribution System Reconfiguration, Variable Demands, Clonal Selection Algorithm, Artificial Immune Systems, Mixed-Integer Nonlinear Programming Problem.**

I. Introduction

In recent year distribution networks were the object of large investments in order to modernize them and to improve their automation level. These investments are designed to improve the reliability, the efficiency and the security of these networks. In this scope a lot of work has also been done at the scientific level namely to address the reconfiguration problem of distribution networks usually termed as DSR, Distribution System Reconfiguration.

The DRS problem aims at identifying the most adequate radial topology of the distribution system taking advantage of the operation of sectionalizers and breakers so that an objective is optimized. Typically, this search for a solution of this problem is driven by the minimization of active losses, provided that a number of operation conditions are satisfied, namely related with nodal voltage ranges, branch flow limits, radial nature of the system in operation and first and second Kirchhoff Laws. Apart from this typical general formulation, the DSR problem can also address the improvement of the voltage profile, keep or enhance the reliability level of the system, to isolate faults and to perform preventive maintenance actions [1].

The DSR problem has combinatorial nature and it can be modeled as a mixed integer non-linear optimization problem (MINLOP) [2]. Accordingly, as the size of the system under analysis increases, the solution of this problem using exact optimization approaches becomes increasingly difficult. This is the main motivation for the adoption of alternative methods as heuristics, artificial networks, metaheuristics including genetic algorithms, simulated annealing and Tabu Search. In recent years the use of immunologic artificial systems has also been reported to solve the DSR problem because it includes strategies that enable reducing the search space so that a good solution can be identified in an efficient way.

As mentioned before, the DSR problem has been widely addressed in the literature considering fixed demand levels. However, some authors address this problem considering variable demand levels. In this case, the DSR problem aims at identifying the most adequate radial topology of the distribution system (one unique topology) that complies with all operation constraints in all considered demand levels and that minimizes the overall cost of energy losses along an extended operation period.

Among the most relevant research works in the literature on the DSR problem assuming fixed demand levels it is possible to mention approaches using heuristic algorithms [2-3], metaheuristics as Genetic Algorithms [4], Simulated Annealing [5], Tabu Search [6], Ant Colonies [7], GRASP [8], Artificial Neural Networks [9] and Immunologic Systems [10]. On the other hand, references [11-13] address the DSR problem assuming variable demand levels.

In this work we solve the DSR problem considering variable demand levels using the Clonal Selection Algorithm (CLONALG) [14]. In this algorithm a population of antibodies is submitted to a selection, a cloning and a hypermutation process that envisages the improvement of the affinity of the anti-bodies, in this case represented by the cost of energy losses associated with each radial topology under analysis. The algorithm also includes a metadynamic procedure that is designed to maintain the diversity level of the population, substituting in each iteration the worse anti bodies by new ones generated in a random way. In order to evaluate the quality of each candidate radial topology it is run a power flow study specially designed to address radial networks [15] for each demand level to be analysed. As a result, it is possible to obtain the cost of energy losses along the specified operation period. In order to illustrate the application of the CLONALG, this paper also includes results obtained for distribution networks with 33, 84 and 136 buses. The results confirm the robustness and the efficiency of the proposed approach.

Apart from this Introduction section, this paper is structured as follows. Section II details the CLONALG algorithm and Section III describes its application to the DSR problem. Then, Section IV presents the illustrative results considering three test systems and finally Section V outlines the main conclusions of this research.

II. CLONAL SELECTION ALGORITMH

The CLONALG algorithm was originally proposed in [14], and its development was inspired on the biological principle of the clonal selection of B lymphocytes that takes place in the immunologic system. It can be applied to pattern recognition problems, machine learning and also to combinatorial optimization problems.

The general flowchart of the CLONALG applied to optimization problems is Figure 1 and its main steps as detailed below [14].

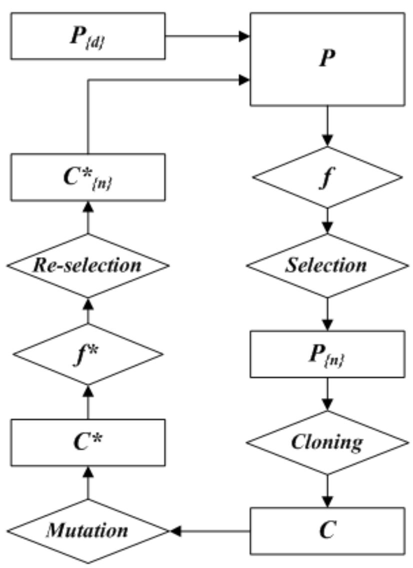

Figure 1. Flowchart of the CLONALG algorithm.

Step 1: Generate a population P with N antibodies interpreted as candidate solutions to the problem under analysis.

Step 2: Evaluate the affinity (objective function) of each antibody and select (selection procedure) the n better antibodies in the population P, obtaining the set $P_{\{n\}}$;

Step 3: Reproduce (cloning procedure) the n selected antibodies, creating a population C with N_c clones. The number of clones of each antibody is proportional to its affinity;

Step 4: The population of clones C is submitted to an hypermutation process, in which the mutation rate is inversely proportional to the affinity of the antibody. At the end of this process it is obtained a population C^* of mutated antibodies;

Step 5: The affinity of each antibody in C^* is now evaluated. Then the n better antibodies in C^* are selected creating the set $C^*_{\{n\}}$. These n antibodies in $C^*_{\{n\}}$ will then be used to replace antibodies in the original population P.

Step 6: d antibodies having low affinity are substituted by new antibodies in $P_{\{d\}}$ (diversity/metadynamic process). In this process, the antibodies having lower affinity are assigned a larger probability of being substituted.

Step 7: The steps 2 to 6 are repeated until the stopping criterium is valid.

The antibodies correspond to candidate solutions and can be coded in real or binary format depending on the problem. Each antibody generates a total number of N_c. The clones can go under mutation using a rate that is inversely proportional to its affinity (objective function). During the execution of the algorithm, the antibodies having lower affinity are substituted by new antibodies generated in a random way.

Using [16], the number N_c of clones generated in step 3for each antibody i is given by (1). In this expression β is a multiplicative factor in [0,1], N is the total number of antibodies in the population P, and *round*(.) is the operator that outputs the nearest integer of its argument.

$$N_c^i = round(\frac{\beta N}{i}) \tag{1}$$

Following [16], the mutation rate α of its clone is given by (2). In this expression ρ is a parameter that controls the output of the exponential function and D^* is the normalized value of the affinity function.

$$\alpha = exp(-\rho D^*) \tag{2}$$

The D^* parameter is calculated using (3) for maximization problems and (4) for minimization [16]. In these expressions, D_{max} and D_{min} represent the maximum and the minimum values of the affinity function.

$$D^* = \frac{D}{D_{max}} \tag{3}$$

$$D^* = \frac{D_{min}}{D} \tag{4}$$

The number of mutations affecting each clone of an antibody [17] is given by (5). In this expression m is the number of mutations that will affect a clone of an antibody, *round*(.) is the operator that outputs the nearest integer of its argument and $N(0,1)$ is value taken from a Gaussian probability function with o mean and standard deviation 1.

$$m = round(\alpha * N(0,1)) \tag{5}$$

III. PROPOSED METHODOLOGY

This section details the application of the CLONALG algorithm to the solution of the DSR problem. This includes the codification of the candidate solutions, the strategy to generate the initial population and the operators (evaluation of the affinity, cloning, hypermutation and metadynamic used in the CLONALG algorithm.

A. *Coding of the Candidate Topologies*

In this work each candidate solution to the DSR problem was coded using the approach detailed in [18] using integer numbers to indicate the branches that are disconnected/ opened. This coding strategy has the advantage of reducing the search space as well as only working with topologic feasible solutions thus allowing the developed CLONALG algorithm to be efficient, fat and robust.

In order to implement this coding strategy, we use (6) to calculate the number of fundamental loops in the graph of the grid under analysis e it is obtained the size of the vector that is used to code each candidate solution. In this expression, LF o is the number of fundamental loops in the graph, n_l is the number of branches and n_b is the number of buses. After obtaining the number of fundamental loops in the graph, it is necessary to identify and store them. It should also be noticed that branches in purely radial areas are set as connected along all the CLONALG algorithm because this is a necessary condition to have all the demand supplied.

$$LF = n_l - n_b + 1 \tag{6}$$

Figure 2 presents a test system with 14 buses and 16 branches. As indicated above, branch 9 in red in Figure 2 is a terminal one and so it is set as connected all through the CLONALG algorithm in order to ensure that the demand in bus 10 is supplied. Therefore, when using (6), n_l and n_b are taken as 15 and 13 and so $LF = 15- 13 +1 =3$. Figure 2 presents the 3 fundamental loops in the graph. These loops are coded in (7) considering the branches in each one.

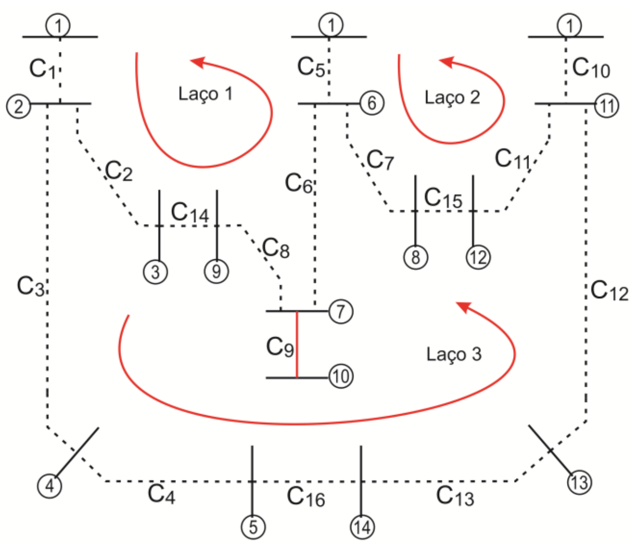

Figure 2. Fundamental loops of the 14 bus system.

$$\begin{aligned} L_1 &= [C_1, C_2, C_{14}, C_8, C_6, C_5] \\ L_2 &= [C_5, C_7, C_{15}, C_{11}, C_{10}] \\ L_3 &= [C_1, C_3, C_4, C_{16}, C_{13}, C_{12}, C_{10}] \end{aligned} \tag{7}$$

This 14 bus test system has 3 fundamental loops. Therefore, in order to ensure that every candidate solution is radial, three branches have to be disconnected in each solution. This means that the coding vector of each candidate solution has three positions, each one related with one branch that is opened in the topology associated with that solution. Departing from the fundamental loops in (7) we choose in a random way a branch to be disconnected, so that a radial topology is obtained. For instance, choosing the branches C_8, C_{11} and C_4 we obtain the coding vector for a candidate solution as given by (8).

$$[8 \quad 11 \quad 4] \tag{8}$$

When choosing the three branches to be opened in a specific candidate solution, we must ensure that each of them belongs to one fundamental loop and that they are not repeated. In fact, it is usual that different fundamental loops share some branches. This means that the same branch should be selected only once, although being shared by more than one fundamental loop.

B. *Strategy to Generate the Initial Population*

In order to generate the initial population of the CLONALG algorithm we used concepts detailed in the previous section and the coding of the candidate solutions and the fundamental loops of the graph of the network.

The initial population is constituted by a number of antibodies (each of them associated to a candidate solution) that are generated in a random way. Its solution is coded by a vector having as many positions as the number of fundamental loops and in each position it is an integer associated to the branch that is disconnected in the fundamental loop. Following [18], the strategy to generate the initial population P is described by the Pseudocode 1 given below. Given that each candidate solution is associated to an opened branch in each fundamental loop, we are ensuring that all candidate solutions are radial, that is they are feasible from a topologic point of view.

Pseudocode 1: Heuristic to generate the initial population

```
Read the fundamental loops (L) and obtain the number of loops (LF);
Define the size of the population (N);
do i=1 until N do
    for j=1 until LF do
        At random, select a circuit that belongs to the
        fundamental loop j (Lj), so that the circuit does not form
        part of the solution topology i in another fundamental
        loop;
    end for
end for
```

C. *Operator to Evaluate the Affinity*

The operator to evaluate the affinity is responsible for the calculation of the value of the affinity of each candidate solution (each antibody) in the population P. This value is associated with the cost of the energy losses associated with that topology when supplying a specified demand level in each bus. In order to obtain the amount of losses it is run a power flow study specially designed to address radial networks [15]. It is important to mention that if a particular topology associated with an antibody is infeasible from a technical point of view (namely because nodal voltage limits or branch current limits are violated) it is then penalized in an inherent way when running the power flow. If several demand scenarios are under analysis, then a power flow exercise is run for each of them and at the end (9) is used to obtain the affinity value of the associated topology. In this expression Nd is the number of demand levels to be analysed, K_i is the cost of energy losses for the demand level i, and T_i and P_i are the duration and amount of energy losses associated with the demand level i.

$$f = \sum_{i=1}^{Nd} [K_i * T_i * P_i] \quad (9)$$

D. Selection Operator

Along the iterative process of the CLONALG algorithm, the selection operator is in charge of selecting the antibodies to use in the cloning and hypermutation processes as well as selecting the best mutated antibodies from the population P.

The selection is performed using the value of the affinity function of each antibody in the population P. After computing the value of the affinity function for each antibody according to the concepts detailed in section III.C, it is possible to conduct the selection process identifying the best n antibodies (the ones associated with the smaller cost of energy losses along the planning period) in the population P in order to create a subpopulation of antibodies denoted as $P_{\{n\}}$.

E. Cloning Operator

After completing the selection step of the CLONALG algorithm and obtaining a subpopulation $P_{\{n\}}$ it is run the cloning operator in order to create a subpopulation of clones C. This population of clones C includes Nc clones of each antibody in the subpopulation $P_{\{n\}}$. The number of clones to obtain from each antibody in $P_{\{n\}}$ is obtained using (1).

F. Hypermutation Operator

The hypermutation operator is used to create mutated antibodies in the neighborhood of the antibodies included in the population of clones C described in Section III.E. This leads to a new population of mutated clone antibodies designated C^*. In order to implement the hypermutation process it is necessary to calculate the mutation rate α using (2) and then identify the number of mutations that each antibody will undergo using (5). After doing this, a number of random mutations are performed according to the Pseudocode 2 below.

Pseudocode 2: Hypermutation Operator

```
Read C and obtain Nc (number of clones of population C);
for i=1 until Nc do
        Calculate the number of mutations (m) for the antibody i;
        for j=1 until m do
                Select randomly a position l of the antibody i. This
                position l represents a fundamental loop (l ∈ LF);
                Choose a circuit of the fundamental loop selected that
                is not disconnected in another position of the antibody i
                and use it to replace the circuit of the position l;
        end for
        Store in C* the matured antibody i;
end for
return C*;
```

Figure 3 illustrates the hypermutation process just described. In this example we used the fundamental loops of the 14 bus test system indicated in (7). The antibody in (8) was selected to undergo the hypermutation process.

Considering this example, let us admit that position 2 of the vector associated with this antibody was randomly selected to undergo mutation. This means that the mutation process will affect a branch in the fundamental loop 2. It should then be randomly selected a branch in loop 2 so that the change of its position (connected/disconnected) originates a different code regarding the original antibody. Given that branch 11 was already in the vector associated to this antibody it can no longer be selected. Let us then assume that branch 10 was chosen which means that branch 10 will substitute branch 11 in the original antibody creating a new antibody in the neighborhood of the initial one.

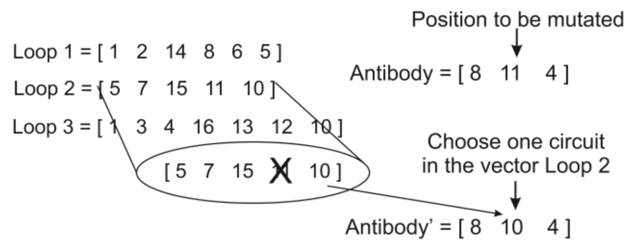

Figure 3. Hypermutation process.

After mutating all the antibodies using the procedure that was just illustrated, we finally get a new population of mutated cloned antibodies designated as C^*.

G. Metadynamic Operator

The metadynamic operator is used to maintain or increase the diversity of the population of antibodies used along the CLONALG algorithm. In each iteration, this operator is used to generate new antibodies and to substitute worse antibodies in the population P (the ones having the largest values of the affinity parameter) by the new ones. In particular, the d worse antibodies in the population P are substituted by d new antibodies generated in a similar way as what was used to create the initial population, as described in Section III.B.

H. Stoping Criteria

The CLONALG algorithm stops if the affinity value of the best antibody in the population did not change for a specified number of iterations and the average value of the affinity value of the antibodies in the population doesn't change more than a specified percentage along a specified number of iterations. If the two above conditions hold, then the algorithm ends indicating that a final solution was obtained. If not, the iteration counter is increased by 1. If the maximum number of iterations was reached then Stop indicating that a final solution was not obtained. If not, return to Step 2 of the algorithm.

IV. Applications and Results

This section presents the results that were obtained using the CLONALG algorithm in the solution of the DSR problem. The CLONALG algorithm was implemented in the Borland C++ 6.0® [19] platform. In order to obtain the active losses for each demand level and for each candidate solution (each antibody in the population) it was used the power flow model detailed in [15]. The tests were done using distribution systems with 33, 84 and 136 buses and the respective data is available in [20], [21] and [6].

A. Used Demand Scenarios

The tests were developed considering a period of 24 hours, and for each of these hourly periods it was specified a demand level for each node in the test systems. The demand was organized in three classes as follows (a) residential, (b) commercial and (c) industrial and for each of them a typical load diagram was specified as illustrated in Figure 4.a), b) and c). In order to obtain the demand in a specific bus the load factor of the end consumer in that bus is multiplied by the active and reactive powers associated with the typical load curves below. The selection of the type of end consumer connected to each bus was done in a random way assuming that 60% of the consumers are residential, 25% are commercial and 15% are industrial. Table I presents the load factors associated to each type of consumer along the 24 hours of the simulation period.

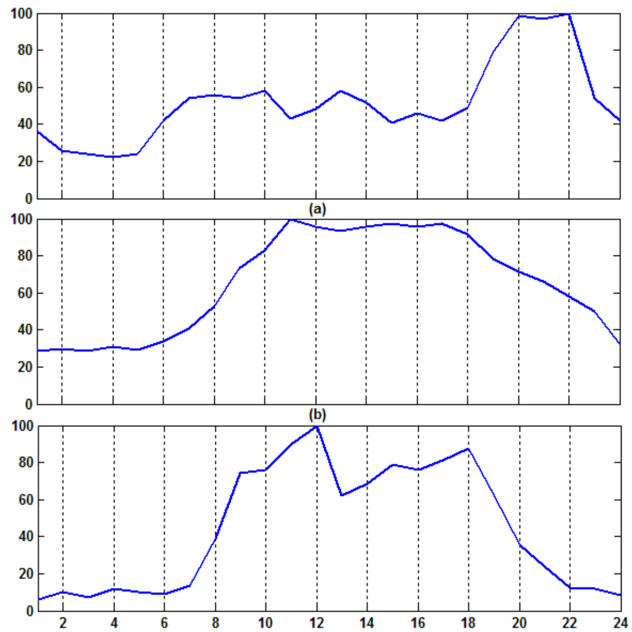

Figure 4. Typical active power demand load curves.

Figure 5 shows how the cost of active losses varies along the 24 hours. In each hourly period, these values are multiplied by the active losses calculated by the power flow study in order to obtain the value of the affinity function associated with a particular antibody, that is with a candidate topology solution.

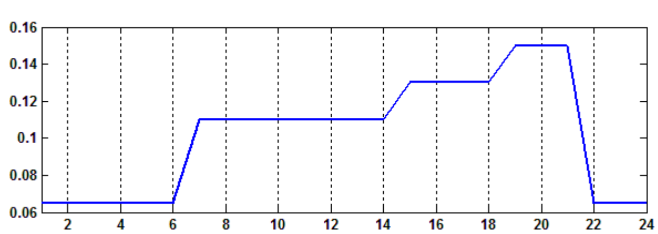

Figure 5. Cost of the energy losses (US$/kWh).

TABLE I. Load factors of the consumers and cost of active losses.

Demand level	Consumer type			Cost (US$/kWh)
	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

B. Parameters of the CLONALG Algorithm

The tests using the three mentioned distribution systems were conducted using the parameters in Table II where ε is the tolerance of the power flow algorithm detailed in [15].

TABLE II. Parameters used in the CLONALG.

Paramters	Distribution system	
	33 and 84 buses	136 buses
N	30	50
β	0.3	0.3
ger	30	50
n	10	10
d	5	5
ρ	4	4
ε	10^{-6}	10^{-6}

C. Distribution System with 33 Buses

This test system has 33 buses, 32 of which are demand buses. It is connected to the upward voltage level by one substation and it has 37 branches. It is established at 12.66 kV, and the total active and reactive powers are 3715 kW and 2315 kVAr [20].

TABLE III. Results obtained for the 33 bus system.

Topology	Open branches	Daily cost of losses (US$)
initial	33-34-35-36-37	187.86
1 demand level	7-9-14-32-37	134.30
final	7-9-14-28-32	128.81

Table III details the results obtained by the CLONALG algorithm for this system. The solution identified by the CLONALG has a total daily cost of 128.81 US$, that represents an improvement of 31.43% regarding the cost of the initial solution. On the other hand, the topology presented in [10] just for one demand level has a larger cost of energy losses when compared with the cost associated with the topology selected by the CLONALG. Finally, the computation time was 0.294 seconds.

D. Distribution System with 84 Buses

This test system is established at 11.40 kV, it has 84 buses, from which 83 are demand buses. It is connected to the upstream voltage level by one substation and the total active and reactive powers are 28350 kW and 20700 kVAr [21]. Table IV present the results for this test system, regarding the initial topology, the topology for just one demand level and the topology output by the CLONALG algorithm.

The solution identified by the CLONALG algorithm has a daily cost of losses of 410.53 US$, that corresponds to an improvement of 10.05% regarding the cost associated with the initial topology. On the other hand, the topology mentioned in [10] just for one load level has a larger cost of energy losses regarding the topology obtained by the CLONALG. Finally, the computation time required by the CLONALG was 3.978 seconds.

TABLE IV. Results obtained for the 84 bus system.

Topology	Open branches	Daily cost of losses (US$)
initial	84-85-86-87-88-89-90-91-92-93-94-95-96	456.41
1 demand level	7-13-34-39-42-55-62-72-83-86-89-90-92	417.29
final	7-34-39-63-72-83-84-86-88-89-90-92-95	410.53

E. Distribution System with 136 Buses

This test system has 136 buses, one substation, 135 demand buses and 155 branches. Its nominal voltage is 13.80 kV, and the total active and reactive loads are 18313.80 kW and 9384.82 kVAr [6]. Table V details the results that were obtained, namely the initially used topology, the topology given in [10] just for one demand level and the topology that was identified by the CLONALG algorithm.

TABLE V. Results obtained for the 136 bus system.

Topology	Open branches	Daily cost of losses (US$)
initial	136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156	288.50
1 demand level	7-35-51-90-96-106-118-126-135-137-138-141-142-144-145-146-147-148-150-151-155	272.97
final	7-38-51-54-84-90-96-106-118-126-135-137-138-141-144-145-147-148-150-151-155	256.89

The daily cost of the energy losses associated with the topology obtained by the CLONALG is 256.89 US$, which represents a reduction of 10.95% regarding the cost associated with the initial topology. On the other hand, the topology in [10] for one demand level has a cost of energy losses larger than the one obtained by the CLONALG. Finally, the computation time of the CLONALG algorithm was in this case of 19.752 seconds.

V. CONCLUSION

This work describes the application of the CLONALG algorithm to the Distribution System Reconfiguration problem considering nodal demand levels that change along the planning period, 24 hours in this case. The search for the most adequate topology is driven by the minimization of the cost of active losses, assuming a unit cost that also varies along the day. The CLONALG algorithm was tested using 3 distribution test systems, with 33, 84 and 136 buses and in all analysed cases the CLONALG was able to identify topologies that are feasible from a technical point of view and also better regarding solutions provided in the literature. Finally, the CLONALG algorithm had a very satisfactory behavior, namely considering its robustness, efficiency and computation time.

ACKNOWLEDGMENTS

The first author wishes to thank INESC TEC, Porto, Portugal and CNPq/Brazil and CAPES-PDSE/Brazil (proc. Nº BEX 3660/14-1) for supporting this research.

The research leading to this work is being carried out as a part of the evolvDSO project (Development of methodologies and tools for new and evolving DSO roles for efficient DRES integration in distribution networks) - www.evolvdso.eu. This project is funded by the European Commission under the seventh framework program (FP7) under grant agreement number 608 732.

REFERENCES

[1] M. A. N. Guimarães, J. F. C. Lorenzetti, C. A. Castro, “Reconfiguration of distribution system for voltage stability margin enhancement using tabu search,” *in Proc. Powercon*, 2004, pp. 1556-1561.

[2] A. Merlin, H. Back, “Search for a minimal-loss operating spinning tree configuration in an urban power distribution system,” *in Proc. 1975 Power System Computation Conference*, pp. 1-18.

[3] S. Civanlar, J. J. Grainger, S. S. H. Lee, “Distribution feeder reconfiguration for loss reduction”, *IEEE Transactions on Power Delivery*, vol. 3, n. 3, pp. 1217-1223, 1988.

[4] K. Nara, A. Shiose, M. Kitagawa, T. Ishihara, “Implementation of genetic algorithm for distribution systems loss minimum reconfiguration”, *IEEE Transactions on Power Systems*, vol. 7, n. 3, pp. 1044-1051, 1992.

[5] H. C. Chang, C. C. Kuo, “Network reconfiguration in distribution systems using simulated annealing”, *Electric Power Systems Research*, vol. 29, n. 3, pp. 227-238, 1994.

[6] D. Zhang, Z. Fu, L. Zhang, “An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems,” *Electric Power Systems Research*, vol. 77, n. 5-6, pp. 685-694, 2007.

[7] E. Carpaneto, G. Chicco, “Distribution system minimum loss reconfiguration in the hyper-cube ant colony optimization framework”, *Electric Power Systems Research*, vol. 78, pp. 2037-2045, 2008.

[8] S. S. F. Souza, M. Lavorato, R. Romero, “Specialized GRASP applied to the problema of distribution reconfiguration systems”, *In Proc. IEEE PES Transmission and Distribution Latin America (T&D-LA)*, 2012, pp. 1-6. (in Portuguese).

[9] H. Salazar, R. Gallego, R. Romero, “Artificial neural networks and clustering techniques applied in the reconfiguration of distribution systems,” *IEEE Transactions on Power Delivery*, vol. 21, n. 3, pp. 1735-1742, 2006.

[10] S. S. F. Souza, R. Romero, J. F. Franco, “Artificial immune networks Copt-aiNet and Opt-aiNet applied to the reconfiguration problem of radial electrical distribution systems,” *Electric Power Systems Research*, vol. 119, pp. 304-312, 2015.

[11] E. A. Bueno, C. Lyra, C. Cavellucci, “Distribution network reconfiguration for loss reduction with variable demands,” *in Proc.*

2004 IEEE Latin America Transmission and Distribution Conference, pp. 384-389.
[12] L. M. O. Queiroz, C. Lyra, "Adaptive hybrid genetic algorithm for techinical loss reduction in distribution networks under variable demands," *IEEE Transactions on Power Systems*, vol. 24, n. 1, pp. 445-453, 2009.
[13] T. Inoue, K Takano, T. Watanabe, J. Kawahara, "Distribution loss minimization with Guaranteed error bound," *IEEE Transactions on Smart Grid*, vol. 5, n. 1, pp. 102-111, 2014.
[14] L. N. de Castro, F. J. Von Zuben, "The clonal selection algorithm with engineering applications", *In: Proc. GECCO*, 2000, pp. 36-39.
[15] D. Shirmohammadi, H. W. Hong, A. Semlyen, G. X. Luo, "A compensation based power flow method for weakly meshed distribution and transmission networks", *IEEE Transactions on Power Systems*, vol. 3, n. 2, pp. 753-762, 1988.
[16] L. N. de Castro, "Immune engineering : development and application of computational tools inspired by artificial immune systems", 2001, PhD. Thesis, UNICAMP, Campinas/Brazil. (in Portuguese).
[17] F. O. de Franca, F. J. Von Zuben, L. N. de Castro, "An Artificial Immune Network for Multimodal Function Optimization on Dynamic Environments", *In: Proc. GECCO*, 2005, pp. 289–296.
[18] J. Mendoza, R. Lopez, D. Morales, E. Lopez, P. Dessante, R. Moraga, "Minimal loss reconfiguration using genetic algorithms with restricted population and addressed operators: Real applications," *IEEE Transactions on Power Systems*, vol. 21, n. 2, pp. 948-954, 2006.
[19] Borland 6.0 Version, C++ Builder.
[20] M. E. Baran, F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, New York, vol. 4, n. 2, pp. 1401-1407, 1989.
[21] J. P. Chiou, C. F. Chang, C. T. Su, "Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems", *IEEE Trans. on Power Systems*, vol. 20, no. 2, pp. 668-674, 2005.