Research Article

Evolutionary Optimization of Electric Power Distribution Using the Dandelion Code

Jorge Sabattin,¹ Carlos Contreras Bolton,² Miguel Arias,¹ and Víctor Parada²

¹Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile, Avenida Ecuador 3659, Santiago, Chile

²Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Avenida Ecuador 3659, Santiago, Chile

Correspondence should be addressed to Jorge Sabattin, jorge.sabattin@usach.cl

Received 16 August 2011; Revised 12 November 2011; Accepted 12 November 2011

Academic Editor: Edgar Carreno

Copyright © 2012 Jorge Sabattin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Planning primary electric power distribution involves solving an optimization problem using nonlinear components, which makes it difficult to obtain the optimum solution when the problem has dimensions that are found in reality, in terms of both the installation cost and the power loss cost. To tackle this problem, heuristic methods have been used, but even when sacrificing quality, finding the optimum solution still represents a computational challenge. In this paper, we study this problem using genetic algorithms. With the help of a coding scheme based on the dandelion code, these genetic algorithms allow larger instances of the problem to be solved. With the stated approach, we have solved instances of up to 40,000 consumer nodes when considering 20 substations; the total cost deviates 3.1% with respect to a lower bound that considers only the construction costs of the network.

1. Introduction

Electric power distribution must satisfy user needs within a given geographic area, and for this purpose, an electric network that can identify a primary and a secondary distribution network is used. In the primary network, electric power is distributed in a tree topology, an activity in which electric companies invest approximately 50% of the transport network's capital [1] and which gives rise to the requirement of efficient planning of the distribution network, taking into account the minimization of power loss. One way of approaching the planning is through mathematical programming models, which, although useful to solve various real situations [2, 3], have limitations when dealing with problems of large size.

The feeders allow power to be transferred from the substations to places close to the final consumers, using a voltage level that can satisfy the requirements of each consumer. The power flows radially along conductors, which, because of their nature, offer resistance to the flow, causing part of this power to be dissipated as heat. These conductors have different costs depending on their cross sections and types of materials, and, therefore, the flow must be adjusted to minimize the loss. This paper considers the design problem

that also involves minimizing the amount of materials and installation costs.

Energy distribution from a substation to the consumption points can be represented by a tree in which the substation corresponds to the root and the tree's nodes represent the consumers. This tree can be coded as an array of integer numbers that represent the labels of the nodes. If we consider the bijection between a Cayley tree [4] and a string of n characters, the number of possible trees given by the Prüfer number [5] is n^{n-2} . Furthermore, because each connection can be established with different types of conductors, the combinatorial degree of the problem can reach dimensions that make it computationally unmanageable.

The optimum generation of electric power distribution trees is a problem that has been studied in the literature and is known as the distribution tree problem (DTP) [6]. One way of facing this challenge is through heuristic and metaheuristic methods. Under this approach, a comparative analysis of computational performance between simulated annealing and tabu search has shown that it is possible to obtain good quality solutions for a group of instances that, although larger than previously reported in the literature, are still far from the sizes that are actually handled [7].

Various authors have studied the DTP by considering genetic algorithms (GAs), which are stochastic search methods based on principles of evolution and natural selection [8]. Typically, they use coding based on integer numbers (binary in some cases) to represent a tree, but this type of representation involves the problem of having to define adequate genetic operators to visit only feasible solutions. In this respect, Najafi et al. [9] use a binary coding scheme that allows solving a large distribution network that, in practice, is reduced to 90 nodes. Considering a similar representation, Li and Chang [10] propose a new GA supported by a minimal spanning tree algorithm and illustrate the method with a small network. The computational performance of a GA appears to become worse when the networks are larger, requiring even longer computing time to solve problem instances of approximately 500 nodes [11]. Real networks can have hundreds of thousands of nodes [12], and thus, the algorithms described above may require extremely long computing times, making it difficult to construct a computational system to support decision making in this field. The greatest computational expense in the GA is associated with the calculation of the power flow at each new solution visited and with the handling of the operators that control the feasibility of the solution. One way of reducing this expense involves using a coding based on trees, which has been explored in the literature by considering other optimization problems in graphs [13].

There are several ways of representing trees within a GA, but there is little consensus as to which of these representations is “the best” [14]. Several researchers have proposed that the representation of a tree must have certain properties for the implemented algorithm to operate efficiently [13, 15, 16]. One way of generating this representation involves using the dandelion code [17], which belongs to the Cayley family of codes [5]. The results show that the dandelion code satisfies the properties identified by Palmer and Kershenbaum [16] and therefore constitutes an interesting representation for an evolutionary search. When using the dandelion code for the DTP, each distribution tree can be evaluated without concern for the feasibility when applying the cross-over and mutation operators. In this way, the computational effort is reduced exclusively to the evaluation of the power flow. This paper tackles the problem of the optimal generation of large electric power distribution trees through a GA, using the dandelion code to represent trees.

The second section of the paper describes the main components of the model, and the third section presents the results. The last section presents the conclusion of the study.

2. Problem Modeling and Representation

Let $\Gamma = \{\tau_1, \tau_2, \dots, \tau_m\}$ be defined as the set of all possible trees with n nodes, and let S be the set of all the integer strings of length $n - 2$. The dandelion code establishes a relationship between a code $D \in S$ and a tree $\tau \in \Gamma$ such

that to decode a tree $\tau \in \Gamma$ associated with D , a function $\varphi_D: [2, n - 1] \rightarrow [1, n]$ is defined such that $\varphi_D(i) = d_i$ for each $i \in [2, n - 1]$ [14].

2.1. Representation of the Problem. To represent a distribution tree, all of the substations are labeled with natural numbers from 1 to N , and the consumer nodes are labeled with natural numbers from $N + 1$ to M ; an artificial node “0” is created to represent the source of supply for all of the substations. In turn, each substation feeds the consumer points following a tree structure. Thus, a dandelion code associated with a set of distribution trees can be identified with an array of $n - 2$ elements. Moreover, to ensure that each substation be the root of each tree, the positions corresponding to the substations have the value “0,” which represents the artificial node.

As an example, consider an electric distribution system with $N = 3$ substations and 9 consumer points. Every set of distribution trees can be represented by a string of 10 natural numbers in $\{1, 2, 3, \dots, 10\}$. To decode a given string, such as $C = (4, 6, 2, 2, 4, 8, 7, 6, 11, 12)$, we first define a string A_c with consecutive numbers from 2 to $M - 1$ (in this case, $A_c = (2, 3, 4, 5, 6, 7, 8, 9, 10, 11)$) such that the first $N - 1$ positions correspond to the labels of the substations. The first two positions of C have the value 0, so matrix R is constructed with the elements of A_c in the first row and the elements of C in the second row:

$$R = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 0 & 0 & 2 & 2 & 4 & 8 & 7 & 6 & 11 & 12 \end{bmatrix}. \quad (1)$$

Function $\varphi(\cdot)$, corresponding to the dandelion code function, is applied where, for each element of the first row, there is a corresponding element directly below it in the second row. In this example, $\varphi(2) \rightarrow 0$; $\varphi(3) \rightarrow 0$; $\varphi(4) \rightarrow 2$; $\varphi(5) \rightarrow 2$; $\varphi(6) \rightarrow 4$; $\varphi(7) \rightarrow 8$; $\varphi(8) \rightarrow 7$. A cycle is detected and separated from $\varphi(\cdot)$, storing it in an ordered list from smallest to largest, such as in the regular dandelion decoding algorithm. Then, we continue with the following element of the first row— $\varphi(9) \rightarrow 6$, $\varphi(10) \rightarrow 11$, $\varphi(11) \rightarrow 12$ —until the elements of A_c are finished. The tree is built as follows: all the nodes that represent the substations are connected with the artificial substation. In this example, nodes 1, 2, and 3 are connected with node 0. The cycles and the last node are added to one of these substations; in this case, 7, 8, and 12 are added to substation 1. A substation node is chosen on which to hang the cycles. In this case, the substation is labeled 1. To select substation 1, the criterion of choosing the substation with the lowest label that appears fewer times in C is applied. In the case of a tie, the one with the smallest label is chosen. The nodes are then added to the tree, as indicated by function $\varphi(\cdot)$. In this way, the code's bijection is retained, and a good locality is insured. Figure 1 shows the electric distribution tree corresponding to code C .

Any set of electric distribution trees that solves the DTP can be represented by the proposed model, with M substations and N consumer points. We call this representation D-DTP. For the GA code C representing a chromosome, each chromosome represents a set of electric distribution trees, and each set of trees represents a solution to the DTP.

FIGURE 1: Electric distribution tree corresponding to code $C = (4, 6, 2, 2, 4, 8, 7, 6, 11, 12)$.

2.2. Initial Population and Objective Function. The initial population is generated using a modified Prim algorithm [18]. Each of the solutions of the initial population is represented by the D-DTP, which, in the GA, is equivalent to an individual. To evaluate the fitness of each individual, (2) is used, which allows the minimum construction cost and the minimum power loss cost to be determined:

$$\text{Min } f(y, x) = f_1(y) + f_2(y, x), \quad (2)$$

where $f_1(y)$ is the investment cost in equipment and $f_2(y, x)$ is the power loss cost.

The construction cost between the f_1 nodes is proportional to the Euclidean distance of the arc between the nodes that form the tree, multiplied by the cost of the conductor used in each section. Furthermore, the cost of the losses f_2 is proportional to the square of the current that flows along each section and is determined using a power flow algorithm [19].

2.3. Experimental Design. To perform the experiment, the test instances are generated first, which have the following input data: active power P , reactive power Q , and (X_1, X_2) , which contains the position of the nodes in geographic space. The instances that must be created are 35,000, 40,000, and 45,000 points, which correspond to the consumption nodes. Additionally, for each instance, 20 points equivalent to the substations are considered. P is generated randomly for each node, with values between 0 and 1. In contrast, Q is generated such that it fulfills the electric power relations that satisfy the condition $Q = P \tan(\theta)$, where θ is generated by satisfying $\cos(\theta) > 0.8$. Similarly, the active power and the reactive power of the substations are generated. Additionally, the total power of the substations must be greater than the sum of the powers at the consumption nodes, including the losses. The geographic locations of the nodes are generated randomly with nominal values between 0 and 1. A conductor table (Table 1) must also be considered that contains the resistance, the impedance, the current capacity, and the price of each conductor.

2.4. Parameter Calibration. The main parameters of a GA are: population size, the cross-over probability, and the mutation probability. In this paper, we will use the parameters proposed by Grefenstette [20], which are 30 to 50

TABLE 1: Conductor data.

Conductor	Impedance (Ohm/Km)	Resistance (Ohm/Km)	Current Capacity (A)	Cost (dollars)
1	0.00010	0.0016	0.08429	8000
2	0.00010	0.0008	0.12644	9000
3	0.00009	0.0005	0.14232	10000
4	0.00008	0.0004	0.18071	12000
5	0.00008	0.0008	0.21073	13000
6	0.00007	0.0004	0.24624	17000

TABLE 2: Network data.

Base power (kVA)	1000
Available power (kVA)	21000
Required power (kVA)	17000
Operating time (years)	10
Base voltage (kV)	12
Number of consumers	500
Number of substations	20

individuals for the population size, 0.90 to 0.95 for the cross-over probability, and 0.01 to 0.05 for the mutation probability. For the termination criterion parameter, we used the number of generations, which is 1000 in this case.

Although it is recommended to use a low mutation probability [21], experiments prior to our work have shown that when the probability increases, the results improve. For this reason, a calibration of this parameter is made. The set of values studied for the mutation probability is the following: 0.005, 0.01, 0.02, 0.04, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18, 0.20, and 0.22.

The calibration process was executed 5 times with an instance of 500 nodes and 20 substations. The network's data are shown in Table 2. As an example, for a given node $P = 1$, considering a base power of 1000 kVA, the P value for this node becomes 1000 kW. The results of the calibration show that the probability of 0.16 achieves the best results, as shown in bold numbers in Table 3.

2.5. Hardware and Software. The equipment used had a 2-Quad-Core CPU 2.00 GHz Intel Xeon with 16 Gbytes of RAM, and the operating system was Ubuntu 10.04.2 LTS Kernel 2.6.32-28-generic, compiler GNU C (4.4.3).

3. Results

For each instance, the following input data were considered: 20 substations, base power 1000 kVA, base voltage 12 kV, and an operating time of 10 years.

Figure 2 shows that the algorithm starts with good initial solutions, then moves away from the initial solution, and finally converges to a good-quality solution. The good initial solutions are explained because the initial population is generated with Prim's algorithm.

TABLE 3: Study of the mutation.

Mutation (%)	0.01	0.01	0.02	0.04	0.08	0.10
Av. installation cost (US\$)	167895	169552	187395	170235	171649	161885
Av. loss cost (US\$)	39757425	18641845	6069982	189316	519619	1462243
Av. total cost (US\$)	39925320	18811398	6257377	359551	691268	1624,128
Mutation (%)	0.12	0.14	0.16	0.18	0.20	0.22
Av. installation cost (US\$)	160996	166901	157098	166722	157471	156557
Av. loss cost (US\$)	1570343	157148	121282	2839201	215340	1684931
Av. total cost (US\$)	1731339	324050	278381	3005924	372810	1841488

FIGURE 2: Cost versus number of generations.

TABLE 4: Results.

Number of nodes	35000	40000	45000
Total cost (average) (US\$)	1021351	1069387	1132179
Minimum cost (US\$)	1013664	1068609	1123047
Minimum bound (US\$)	970082	1036867	1102479
Average time (HH:MM:SS)	87:17:00	110:54:46	127:00:18
Minimum time (HH:MM:SS)	84:58:47	105:33:30	118:59:16
Available power (kVA)	20000	21000	22000
Required power (kVA)	16000	17000	18000

To validate the results obtained with the GA, the results are compared with a lower bound. To find such bound, use is made of a Prim algorithm with which the minimum construction cost of the network is sought. In this way, comparisons of the results are established that validate the results obtained with the GA. This can be observed in all of the test instances (Table 4).

4. Conclusions

This paper proposes a solution for the DTP that has a real application in the optimization of electric distribution networks. The problem is approached using GAs, proposing a model that considers construction and power loss costs. To perform the search in the distribution tree space, the dandelion code is used. The proposed approach shows that

the code used is efficient in the representation of trees, and its use allows real problems to be solved using GAs.

The approach used also allows one to find solutions to problems with extremely large instances, e.g., for the instance of 45,000 nodes, which requires a computing time of 118:59:16.

References

- [1] G. Kjolle, L. Rolfseng, and E. Dahl, "Economic aspect of reliability in distribution system planning," *IEEE Transactions on Power Delivery*, vol. 5, no. 2, pp. 1153–1157, 1990.
- [2] U. G. Knight, "The logical design of electrical networks using linear programming methods," *Proceedings of the IEE A*, vol. 107, no. 33, pp. 306–314, 1960.
- [3] M. A. El-Kady, "Computer-aided planning of distribution substation and primary feeders," *IEEE transactions on power apparatus and systems*, vol. 103, no. 6, pp. 1183–1189, 1984.
- [4] A. Cayley, "A theorem on trees," *Quarterly Journal of Mathematics*, vol. 23, pp. 376–378, 1984.
- [5] H. Prüfer, "Neuer beweis eines satzes über permutationen," *Archiv für Mathematik und Physik*, vol. 27, pp. 742–744, 1918.
- [6] V. Parada, J. A. Ferland, M. Arias, P. Schwarzenberg, and L. S. Vargas, "Heuristic determination of distribution trees," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 861–869, 2010.
- [7] V. Parada, J. A. Ferland, M. Arias, and K. Daniels, "Optimization of electrical distribution feeders using simulated annealing," *IEEE Transactions on Power Delivery*, vol. 19, no. 3, pp. 1135–1141, 2004.
- [8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1st edition, 1989.
- [9] S. Najafi, S. H. Hosseini, M. Abedi, A. Vahidnia, and S. Abachezadeh, "A framework for optimal planning in large distribution networks," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 1019–1028, 2009.
- [10] Y. Li and X. Chang, "A MST-based and new GA supported distribution network planning," in *Proceedings of the International Conference on Mechatronic Science, Electric Engineering and Computer (MEC '11)*, pp. 2534–2538, 2011.
- [11] I. J. Ramirez-Rosado and J. L. Bernai-Agustin, "Genetic algorithms applied to the design of large power distribution systems," *IEEE Transactions on Power Systems*, vol. 13, no. 2, pp. 696–703, 1998.
- [12] A. S. Pabla, *Electric Power Distribution*, Tata McGraw-Hill, New York, NY, USA, 2004.
- [13] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*, Springer, 2nd edition, 2006.

- [14] T. Paulden and D. K. Smith, "From the Dandelion Code to the rainbow code: a class of bijective spanning tree representations with linear complexity and bounded locality," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 2, pp. 108–123, 2006.
- [15] F. Rothlauf, "Towards a theory of representations for genetic and evolutionary algorithms," University of Bayreuth, Alemania, 2001.
- [16] C. C. Palmer and A. Kershenbaum, "Representing trees in genetic algorithms," in *Proceedings of IEEE Conference on Evolutionary Computation*, vol. 1, pp. 379–384, 1994.
- [17] S. Picciotto, "How to encode a tree," Univ. California, California, 1999.
- [18] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.
- [19] M. Arias, H. Sanhueza, and V. H. Quintana, "Nuevo algoritmo para cálculo de flujo de potencia en alimentadores de distribución," in *the 6th Congreso Chileno de Ingeniería Eléctrica*, Punta Arenas, Chile, 1995.
- [20] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.
- [21] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," University of Michigan, USA, 1975.