

# Heuristic Determination of Distribution Trees

Víctor Parada, Jacques A. Ferland, Miguel Arias, *Senior Member, IEEE*, Pablo Schwarzenberg, and Luis Vargas, *Senior Member, IEEE*

**Abstract**—Electrical distribution networks develop configurations that deviate from the original long-term plan. The Distribution Trees Problem (DTP) is one means of measuring this development, which finds the deviation between long-term planning and the optimal topology for the actual conditions of the network. Each feasible solution corresponds to a set of directed out-trees rooted at the substations. DTP takes into account characteristics of the substations and consumer demand. It also determines the optimal topology of the network to distribute electrical energy at minimum cost. In this paper, we use two search techniques to solve this problem: 1) simulated annealing and 2) tabu search. Nine different problems within 500 to 30 000 consumer points and 20 substations were used to calibrate the parameters of both methods and to compare their efficiency. The numerical results indicate that the efficiency of simulated annealing decreases as the problem size increases, and that tabu search is more efficient than simulated annealing.

**Index Terms**—Distribution trees problem (DTP), electrical energy distribution, simulated annealing, tabu search.

## I. INTRODUCTION

**D**ISTRIBUTION networks planning has been of academic interest since 1960 [1] when the first mathematical models were proposed. In the long term, the expansion of networks is studied by defining the location and selecting the proper transformers at the substations, and by selecting the routes of new feeders and the type of conductor to be used. The objective is to satisfy the current and future demands at minimal total cost. This includes investment in the installation of transformers and feeders to decrease the non-linear power loss cost due to the resistance of power flow in the conductors of the feeder.

Distribution planning is frequently classified in three different ways [2], [3]. In the first approach, expansion planning consists of the optimal design for expansion of the distribution network

Manuscript received May 15, 2008; revised February 05, 2009, June 04, 2009, August 01, 2009. This work was supported in part by the Millennium Institute Complex Engineering Systems Grant P01-34, in part by NSERC Grant OGP 0008312, and in part by DICYT-USACH. Paper no. TPWRD-00375-2008.

V. Parada is with the Informatics Engineering Department of the University of Santiago, Santiago, Chile. he is also with the Millennium Institute Complex Engineering Systems, Santiago, Chile (e-mail: vparada@diinf.usach.cl).

J. A. Ferland is with the Department of Computer Science and Operations Research, University of Montreal, Montreal, QC H3C 3J7, Canada (e-mail: ferland@iro.umontreal.ca).

M. Arias is with the Electrical Engineering Department of the University of Santiago, Santiago, Chile (e-mail: Miguel.arias@usach.cl).

P. Schwarzenberg is with the Informatic Engineering Department of the University of Santiago, Santiago, Chile.

L. S. Vargas is with the Electrical Engineering Department, University of Chile. He is also a member with the Millennium Institute Complex Engineering Systems, Santiago, Chile (e-mail: lvargas@cec.uchile.cl).

Digital Object Identifier 10.1109/TPWRD.2009.2033929

for both the future growth of current customers and new forecasted loads. The second approach is the Greenfield planning structure, which is based on the optimal design of the whole distribution network for real consumption by assuming there is no real existing network. From this ideal network we can determine differences with the existing network. Known as operation planning, the third approach consists of optimizing the operation of the existing network by analyzing different operation combinations in order to decrease the system losses in a given time horizon. This paper deals with a combination of the last two approaches. That is, from the existing substations, and considering the location and the magnitude of the actual demand points, we look to identify the topology of minimal losses and to minimize the installation cost, while conditions of radiality, ampacity of the conductors and demand requirements are satisfied. We refer to this problem as a Distribution Tree Problem (DTP).

In the normal operation the primary distribution network has a radial topology (i.e., each consumer point is supplied with energy by a unique pathway from a power substation). In this way, the distribution network is a set of directed out-trees rooted at the power substations. The DTP considered in this paper determines the primary distribution network using radial topology for a specific period, assuming that:

- the installation costs and the physical locations of substations and consumer points are known;
- additional transformers may have to be added to some power substations;
- each consumer point has a fixed power demand;
- each consumer point is connected to a substation along a path that may include other consumer points;
- there is a set of conductor types available with known parameters (ampacity, resistance, reactance, and cost per unit length);
- the selected conductors for each path are capable of transmitting the necessary power flow;
- the sum of the capacities of all substations is sufficient to supply power to all customers; several consumer points may belong to the same client.

The DTP can be seen as a special case of the planning problem of a primary distribution network for a single period using a similar objective function, where the demand is known, where the locations of the substations are already selected, and where we account for the current components of the system. Also, the DTP is different from the networks reconfiguration problem, where the distribution network in normal operating conditions is adjusted to react to emergencies or to reduce the operating cost. This is done by using the feeders currently available in the physical network where no new components can be installed in the system. Indeed, the DTP solution may include new transformers in the substations, new branches, and

new feeder conductors on existing branches, which results in a modified physical network.

Exact and heuristic techniques have been used to deal with both design and reconfiguration problems arising in electric distribution. These problems are large scale, non-linear, and combinatorial in nature. Chang and Wu [4] have studied a problem similar to DTP. They solve the optimal reliability design problem by determining a solution of the equations system that results from the Karush-Kuhn-Tucker conditions for a non-linear optimization problem. The algorithm was tested with a problem including only 3 substations and 4 consumer points. In a previous work [5] based on a non-linear network flow model, we studied the computational performance of Simulated Annealing to solve problems in up to 20 substations and 295 consumer points. Nevertheless, since the Prim's algorithm was used at each iteration, it was not possible to solve large-sized problems. On the other hand, the reconfiguration problem has been formulated as a binary quadratic programming problem as in [6], [7] and as an integer programming problem that can be seen as a Steiner tree problem in Avella, *et al.* [8]. The largest problem studied includes one substation and 69 consumer points. In order to tackle the computational difficulties that typically are found in reconfiguration problems, heuristic methods such as Tabu Search [9], Simulated Annealing [10], and genetic algorithms [11] have been used. Genetic algorithms also have been used for representing and solving design problems [12], for encoding (representing) the solution, and for simultaneously specifying the location of the substations and required feeders.

The remainder of this paper is organized as follows: In Section II we introduce a specification of the DTP problem, and we summarize a procedure to evaluate the cost of a solution. Two local search techniques are presented in Section III. First we characterize the structure of the neighborhood, and then we introduce a procedure to determine an initial solution. Then the two procedures that have been implemented (Simulated Annealing and Tabu Search) to solve the DTP are summarized. In order to compare the efficiency of the techniques, the numerical results obtained for 9 different problems including 20 substations and 500 to 30 000 consumer points, are analyzed in Section IV. Section V concludes the paper.

## II. PROBLEM SPECIFICATION

Let  $G = (U, A)$  be a directed graph associated with a DTP, where the set of nodes is given by  $U = \{r\} \cup M \cup N$ , and  $r$  is the artificial node,  $M$  is the set of substations, and  $N$  is the set of consumer points. Three types of arcs exist in  $A$ : arcs connecting pairs of consumer points, arcs connecting a substation to consumer points, and fictitious arcs connecting an artificial node to substation nodes. A primary distribution network having a radial topology corresponds to a directed spanning out-tree  $\Gamma = (U, A_\Gamma)$  rooted at the artificial node in the directed graph  $G = (U, A)$ . Note that such a directed spanning out-tree includes all arcs connecting the artificial node to the substations, unless some substations are not used (Fig. 1).

The problem can be formulated as a nonlinear integer program as follows:

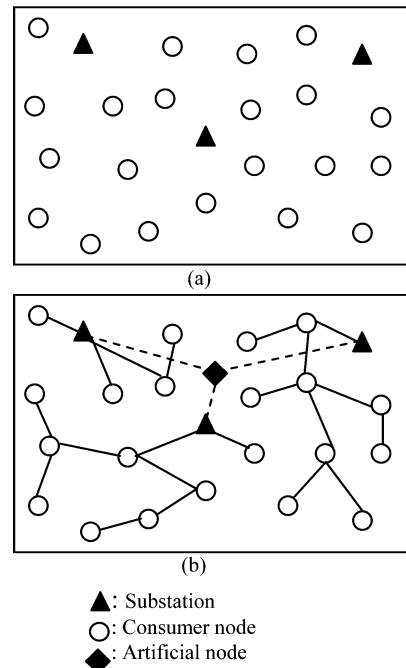


Fig. 1. DTP Representation.

### Input data

$$G = (U, A); U = \{r\} \cup M \cup N;$$

$M$  set of substations;

$N$  set of consumer nodes;

$W$  set of conductor types;

$A_j^-$  set of incidents arcs to node  $j$ ;

$A_j^+$  set of emergent arcs from node  $j$ ;

$B \subseteq A; B = \{a; a = (r, m_j), j = 1, 2, \dots, |M|\}$ ;

$R_w$ : Electrical resistance of conductor type  $w$ ;

$d_j$ : power required by the consumer node  $j$ .

### Variables

$x_a$ : power flow on arc  $a$ ;

$y_a = 1$  if arc  $a$  is part of a tree and  $y_a = 0$  otherwise;

$z_{aw} = 1$  if a conductor of type  $w$  is assigned to arc  $a$  and

$z_{aw} = 0$  otherwise;

$V_a^s$ : voltage at the beginning of the arc  $a$ .

### Parameters

$\alpha, \beta$ : adjustment parameters;

$$\begin{aligned} \text{Min } f(x) = & \alpha \sum_{a \in A} R_w z_{aw} \left( \frac{x_a}{V_a^s} \right)^2 y_a \\ & + \beta \sum_{a \in A} c_w z_{aw} y_a. \end{aligned} \quad (1)$$

Subject to

$$\sum_{a \in A_j^-} x_a y_a - \sum_{a \in A_j^+} x_a y_a = d_j \quad \forall j \in U \quad (2)$$

$$\sum_{a \in A} R_w z_{aw} \left( \frac{x_a}{V_a^s} \right)^2 y_a = \sum_{a \in A} x_a - \sum_{j=1}^{|M|+|N|} d_j \quad (3)$$

$$\sum_{a \in A} y_a = |M| + |N| \quad (4)$$

$$\sum_{w \in W} z_{aw} = 1 \quad \forall a \in A \quad (5)$$

$$x_a \leq k_w z_{aw} y_a \quad \forall a \in A \quad (6)$$

$$x_a \geq 0 \quad \forall a \in A \quad (7)$$

$$V_a^s \geq 0 \quad \forall a \in A \quad (8)$$

$$y_a = 0 \text{ or } 1 \quad \forall a \in A \quad (9)$$

$$z_{aw} = 0 \text{ or } 1 \quad \forall a \in A, w \in W. \quad (10)$$

The model for DTP has an objective function that minimizes both the power loss and the cost involved in the investment and the installation of conductors. Constraints (2) specify the satisfaction of the demand at each node, while (3) ensures the total energy balance on the network. The equality (4) implies that the topology must be a tree. By means of (5) only one type of conductor is assigned to each arc. Constraint (6) guarantees the satisfaction of the capacity of each arc. Constraints (7), (8), (9), and (10) state the valid range for the variables. As to solve this problem by a mathematical programming algorithm could become a computationally difficult task even for small instances, in this study we propose two approaches based on metaheuristics. In order to explain such approaches, first let us define a procedure to evaluate the objective function.

We associate an investment and installation cost with each arc  $a = (r, j)$ ,  $j \in M$ , corresponding to the cost of adding capacity to substation  $j$ . Denoted by  $w \in W_a$ , a selection of conductors allowing a power supply capacity equal to  $P_j(w)$ , the cost of modifying the current components to reach this selection  $w$  is represented as  $c_w$ . Then for each fictitious arc  $a = (r, j)$ ,  $j \in M$  connecting the artificial node  $r$  and a substation  $j$ ,  $w_a$  is selected as the cheapest among those  $w \in W_a$  supplying sufficient power, i.e.:

$$w_a = \text{Argmin}_{w \in W_a} \{c_w; P_j(w) \geq x_a\}.$$

Note that for the fictitious arc  $a = (r, j)$ ,  $j \in M$ , we indicate the current power supply capacity of substation  $j$  by  $w' \in W_a$ , hence  $c_{w'} = 0$ . Furthermore, we assume the existence of a dummy selection  $w''$  providing a very large capacity  $M_3$  at a very large cost  $M_4$ .

The second and the third components of the cost function depend on the type of feeder conductors used on the arcs of the directed spanning out-tree  $\Gamma$  connecting a substation and a consumer point, or connecting two consumer points (i.e.,  $a \in A_\Gamma - \{(r, j); j \in M\}$ ). Several conductor types with different capacities and characteristics can be used. In general, investment and installation costs increase with the capacity of the conductor, whereas power loss decreases. Hence, a trade-off is required in the feeder size selection in order to minimize the total cost. The procedure to evaluate these costs includes two steps:

Step 1) For each arc  $a$  of the directed spanning out-tree  $\Gamma$  connecting a substation and consumer point, or connecting two consumer points (i.e.,  $a \in A_\Gamma - \{(r, j); j \in M\}$ ), the conductor type  $w_a$  is selected as the cheapest among those in set  $W_a$  of conductor types that can be used on segment  $a$ , having a power flow  $u_w$  large enough then:

$$w_a = \text{Argmin}_{w \in W_a} \{c_w; u_w \geq x_a\},$$

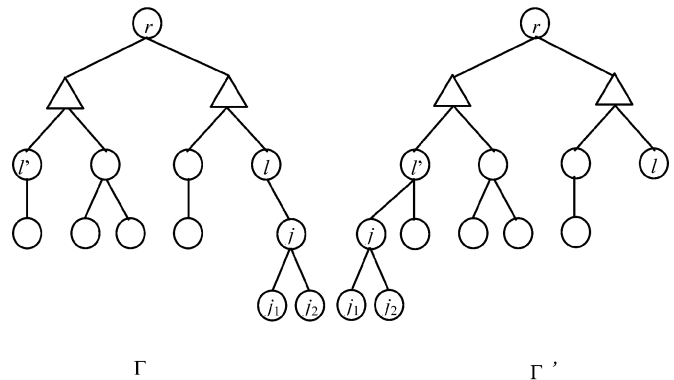


Fig. 2. Neighborhood.

where  $c_w$  is the investment and installation cost per kilometer for conductor type  $w$ , and  $u_w$  is its power flow capacity. Note that for all  $a \in A_\Gamma - \{(r, j); j \in M\}$ , we assume the existence of a dummy conductor type  $w'' \in W_a$  having a very large flow capacity  $u_{w''} = M_1$  and a very large unit cost  $c_{w''} = M_2$ .

Step 2) Modify the conductor type in order to reduce power-loss by increasing its size (hence increasing the capacity and the investment cost) to reduce total cost.

To properly modify the selection of the conductor type on some arcs and additional transformers in the substations, an iterative load-flow algorithm from Arias, *et al.* [13] is used. This algorithm generates the magnitude and the phase angle of the voltage at the consumer points, and the real and reactive power flowing in each arc of the directed spanning out-tree. Analysis of load-flow algorithms for distribution feeders is beyond the scope of this paper and different approaches can be found in [14] and [15].

### III. DTP STRATEGY

In this paper, we are using two different neighborhood search techniques to deal with the DTP: Simulated Annealing and Tabu Search. A neighborhood search technique is an iterative procedure allowing movement from a current solution  $S$  to a new one  $S'$  in the neighborhood  $N(S)$  of the current solution  $S$ , until an admissible solution  $S^*$  which satisfies the stopping criterion, is found.

#### A. General Description

In our implementation of the two local search techniques, the structure of the neighborhood of any solution  $S$  is specified as follows. Let  $\Gamma = (U, A_\Gamma)$  (where  $A_\Gamma \subset A$ ) is the directed spanning out-tree rooted at the artificial node associated with the current solution  $S$  as illustrated in Fig. 2. For each consumer point  $j \in N$ , the (unique) edge  $(l, j)$  on the path between  $r$  and  $j$  allows the identification of the predecessor node of  $j$  in  $\Gamma$ . The set of consumer nodes having  $j$  on their path from  $r$  are the set of successor nodes of  $j$ . In Fig. 2,  $l$  is the predecessor of  $j$ , and  $\{j, j1, j2\}$  is the set of successors of  $l$  in the directed spanning out-tree  $\Gamma$ .

The following procedure (illustrated in Fig. 2) is used to generate a directed spanning out-tree  $\Gamma' = (U, A_{\Gamma'})$  associated with a neighborhood solution  $S'$ :

- Randomly select a consumer point  $j \in N$ ;
- Disconnect node  $j$  from its predecessor  $l$  by eliminating edge  $(l, j)$ ;
- Reconnect node  $j$  to a new node  $l'$  selected randomly in the set  $M \cup N - \{\text{successor of } j\} - \{l\}$  in order to generate a new directed spanning out-tree  $\Gamma'$ . Then, it is possible to reach any feasible solution, in particular the optimum solution, from any given solution belonging to the feasible domain.

It follows that the number of neighborhood solutions of any solution  $S$  is smaller than  $|N|(|M| + |N| - 1)$ .

To generate an initial solution  $S_0$  for the local search techniques, consider the underline graph (non-directed)  $G' = (U, A')$  obtained from  $G = (U, A)$  by eliminating the orientation on the arcs. Denote by  $l_{ij}$  the distance between the locations of nodes  $i$  and  $j$  associated with edge  $(i, j) \in A'$ . The fictitious edges  $(r, j)$  link the super-substation  $r$  and substations  $j \in M, l_{rj} = 0$ . An initial solution is generated by randomly constructing a spanning tree with a random version of the well-known Prim's algorithm [16]. Initially the set of nodes included in the current sub-tree is  $L_e = \{r\} \cup M$ , and the set of nodes left to be included is  $L_d = N$ ; the corresponding sub-tree includes only the fictitious edges. At each iteration, a new edge is added to the current sub-tree by randomly selecting a node  $j \in L_d$  and linking it to the closest node  $l$  already in the sub-tree, such that the substation  $ss$  providing the power to  $l$  has the capacity  $P_{ss}$  to also provide the power  $d_j$  required by  $j$ . The associated directed spanning out-tree is obtained by orienting the edge to specify paths from the artificial node to the substations and the consumer points.

### B. Simulated Annealing Procedure

Simulated Annealing allows non-descent modification to avoid being trapped in a local minimum. Originally the approach was used to simulate the evolution of an unstable physical system toward a thermodynamically stable equilibrium-point at a fixed temperature. Kirkpatrick *et al.* [17] were the first to use it to solve optimization problems.

At each iteration of this probabilistic technique, a solution  $S'$  is selected randomly in neighborhood  $N(S)$  of the current solution  $S$ . Solution  $S'$  replaces  $S$  as the current solution if  $\delta = f(S') - f(S) < 0$ . But  $S'$  can replace  $S$  even if  $\delta \geq 0$  (i.e., no improvement of  $f$  is induced by moving from  $S$  to  $S'$ ) according to probability decreasing as the value of  $\delta$  increases. More specifically,  $S'$  replaces  $S$  with probability  $e^{-\delta/T}$  where  $T$  is a parameter (referred to as the temperature factor) decreasing with the number of iterations. In our representation  $f(S)$  is evaluated with the two-steps procedure described in Section II.

Any implementation of the Simulated Annealing technique is characterized by the values selected for the two parameters associated with the technique: the initial temperature  $T_0$ , and the number of iterations  $L(T)$  completed using a specific temperature  $T$ . An additional parameter  $\alpha \in (0, 1)$  must be specified to modify temperature  $T$ . Even if the geometric cooling schedule  $T = \alpha T$  cools more rapidly than a logarithmic schedule, it is nevertheless widely used. The values of these parameters in our

implementation are analyzed in Section IV. Finally, the stopping criterion is specified in terms of a minimal value  $T_{\min}$  that the temperature can reach.

### C. Tabu Search Procedure

The Tabu Search procedure also avoids being trapped in a local minimum [18]. At each iteration, the best solution  $S'$  in a subset  $N'(S)$  of the neighborhood  $N(S)$  in the current solution  $S$  is selected to be the current solution of the next iteration. As long as  $f(S') < f(S)$ , the procedure behaves like a descent method. But if  $f(S') \geq f(S)$ , then movement from  $S$  to  $S'$  does not improve but may even deteriorate the objective function. Nevertheless, this movement allows escape from a local minimum.

Now, since the value of the objective function  $f$  is not necessarily monotonic in its behavior, decreasing during the resolution, a safeguard against cycling is required. This is provided through a memory mechanism to forbid returning to an earlier solution by keeping the most recent modifications used to generate the sequence of current solutions in a short-term Tabu List TL. More specifically, suppose that  $S' \in N'(S)$  selected to be the current solution for the next iteration is generated by eliminating the arc  $(l, j)$ . To avoid cycling, we forbid the arc  $(l, j)$  to be in the directed spanning out-tree associated with any solution during the next  $|TL|$  iterations. The mechanism is implemented by using a cyclic Tabu List TL that is updated at each iteration by including the arc  $(l, j)$ . Since the length  $|TL|$  of the list is fixed, if the list is full when the arc  $(l, j)$  is included, then the oldest element of the list is eliminated. Accordingly, a solution is said to be tabu (and cannot be selected as the current solution for the next iteration) if it is generated by reconnecting a node  $j$  to a node  $l'$ , and if arc  $(l', j)$  is on the Tabu List TL. Hence the subset  $N'(S)$  is obtained by eliminating the solutions of  $N(S)$  that are tabu.

The preceding mechanism has the drawback of giving a tabu status to some solutions that were not generated before. Since some of these solutions might be very good, we introduce an aspiration criterion to bypass the tabu status of these solutions. Hence, even if a neighbor solution is tabu, it is nevertheless included in  $N'(S)$  whenever its value is smaller than the value  $f(BS)$  of the best solution generated so far.

Note that in our implementation, we are not generating the whole subset  $N'(S)$  at each iteration, but rather only a fixed number of its elements. This number of elements is a parameter of the procedure analyzed in Section IV.

The stopping criterion is specified in terms of a maximum number (ITTMAX) of consecutive iterations where the objective function  $f$  does not improve.

Periodically during the resolution, when the rate of improvement slows down, it is recommended that the search be intensified in promising regions. This mechanism is known as intensification. Diversification of the search at various stages of the resolution is also recommended. The criteria to detect the proper moment to diversify or to intensify are discussed in Section IV.

In the diversification stage, in order to search the feasible domain more extensively, we use a larger neighborhood  $ND(S)$  where the elements are generated by disconnecting and reconnecting one or two consumer points elsewhere. The procedure

TABLE I  
TEST SYSTEMS

Instance N°	Number of Consumer nodes	Lower Bound \$	Power Available [MW]	Total Consumption [MW]	Initial Solution \$
1	500	151374	553	305	281505
2	1000	230084	551	310	340531
3	2000	327730	519	297	412696
4	3000	409824	526	301	536279
5	4000	485947	512	292	718051
6	5000	537586	528	305	748346
7	10000	770717	549	316	920423
8	20000	1106693	530	298	1294277
9	30000	1363016	543	313	1648472

to update the Tabu List is modified to include both arcs (that were earlier eliminated) to generate the neighbor solution. Accordingly, a neighbor solution is tabu if one of the arcs used to reconnect one node is on the Tabu List.

In the intensification stage, the search is reinitialized at the best solution BS generated so far, and the neighborhood  $NI(S)$  is also different. The element of  $NI(S)$  are obtained by disconnecting a consumer point  $j$  from its current predecessor  $l$  and reconnecting it to the node  $l'$  in the set  $M \cup N - \{\text{successor of } j\} - \{l\}$  inducing the distribution network with the smallest investment and installation cost.

#### IV. NUMERICAL RESULTS

We use nine different randomly-generated problems to complete the numerical experimentation of the procedures. One of these problems is used to calibrate the parameters of both procedures. Later all nine problems are used to compare the efficiency of the two procedures. The tests are run on an Intel(R) CPU T2300 1.66 GHz Dual Core having 1 GB of RAM.

To achieve a comparison without introducing relative advantages or bias in favor of one of the procedures, an object-oriented design to manage the functions common to both procedures is defined. Functions to generate the initial solution and to generate neighbor solutions randomly, are defined in an abstract class, and both the **Simulated Annealing Procedure** and the **Tabu Search Procedure** are derived from this common class. The functions to generate the neighbor solutions in the intensification and the diversification stages are incorporated in the **Tabu Search Procedure**.

##### A. Problems Generation

Each of the nine problems is generated randomly, and has 20 substations and a number of consumer points as indicated in Table I. The coordinates of these nodes are randomly generated in a rectangle and the substations are located where the largest concentrations of consumer points are found. The power available at each substation lies in the interval from 20 MVA to 40 MVA. The consumer points requirements are randomly generated in order that the total power required at the consumer points does not exceed the total power available at the substations. Finally, the conductor types correspond to those typically used in the industry.

For each problem, a lower bound on total cost is obtained as follows: generate a minimum spanning tree (in terms of dis-

TABLE II  
RESULTS FOR DIFFERENT VALUES OF  $T_0$ 

$T_0$	Number of Iterations (average)	Average Cost Solution \$	Average Cost Improvement \$
7000	14860	334635	5896
10000	15292	333603	6928
12000	15512	338522	2009
15000	15734	339445	1086
20000	16184	340531	0

tances) of the underlying graph for the problem; then for each edge of the tree, select the cheapest conductor to determine the investment and installation cost.

In Table I, the first and second columns refer to the problem number and the number of consumer nodes respectively. The fourth and the fifth columns include the total power available at substations and the total power required at the consumer nodes respectively. Finally, the third and the sixth columns indicate the cost of the lower bound and of the initial solution respectively.

##### B. Calibration of Parameters

In our approach, the parameters are considered sequentially in order to determine their best values, keeping others fixed to some preliminary values recommended in the literature. The criterion to select the best value for a parameter is specified in terms of the average cost improvement with respect to the initial solution.

Problem 2 in Table I, which has 20 substations, 10 000 consumer points, and 6 feeder types, is used to calibrate the parameters. Three successive resolutions of the problem are run.

Typically, at the early stage, the SA procedure allows to move widely in the feasible domain, and at the end, the last third of the iterations is used to reach convergence. In preliminary testing, it was observed that a stopping criterion based on a specified number of successive iterations without improving the objective function was not efficient. Hence we use a stopping criterion specified in terms of a smallest value  $T_{\min}$  that temperature  $T$  can reach. Preliminary testing indicates that value  $T_{\min} = 10^{-40}$  is appropriate.

The initial value  $T_0$  allows, in a sense, the control of how widely the feasible domain is explored during the earlier iterations. Indeed, the rate of acceptance of a neighborhood solution increases with the value of  $T_0$  to allow more extensive exploration of the feasible domain.

Table II summarizes the results obtained with five different values for  $T_0$  : 7 000, 10 000, 12 000, 15 000, and 20 000. In these tests, the value of  $\alpha$  is fixed at 0.9. The best value for the average cost improvement is obtained when  $T_0 = 10 000$ . Furthermore, the efficiency of the procedure seems to decrease as the value of  $T_0$  increases from 10 000. For  $T_0 = 20 000$ , no improvement of the objective function is observed during more than 16 000 iterations.

Seventeen different values are used to analyze the effect of  $\alpha$  : 0.80, 0.81, ..., 0.95, 0.96. The value of  $T_0$  is fixed at 10 000. The results in Table III indicate that the best value for the average cost improvement is obtained when  $\alpha = 0.85$ . Note that there are two cases in the calibration process of  $T_0$  and  $\alpha$  that do

TABLE III  
RESULTS FOR DIFFERENT VALUES OF  $\alpha$

$\alpha$	Number of Iterations (average)	Average Cost Solution \$	Average Cost Improvement \$
0,80	10785	336105	4426
0,81	11380	337117	3414
0,82	12274	335654	4877
0,83	13246	336039	4492
0,84	14044	340266	265
0,85	15292	333603	6928
0,86	16507	333939	6592
0,87	18079	336627	3904
0,88	19918	337906	2625
0,89	21931	337511	3020
0,90	24409	338715	1816
0,91	27585	336935	3596
0,92	31598	337098	3433
0,93	36852	338639	1892
0,94	43654	332983	7548
0,95	53657	338231	2300
0,96	68172	340531	0

not improve the quality of the initial solution ( $T_0 = 20\,000$   $\alpha = 0.96$ ). We know that when  $T_0$  is very low, SA transforms into a local search algorithm; on the other hand, when  $T_0$  is high and the value of  $\alpha$  is high and constant, the convergence is slow. Sometimes, this behavior can occur in other instances.

To determine the number of iterations  $LT(T)$  at temperature  $T$ , three different strategies are analyzed:

- 1)  $LT(T) = \text{constant}$ ;
- 2)  $LT(T) = \lceil 1 + \log(T_0/T) \rceil$ ;
- 3)  $LT(T) = \lceil 1 + \log(T_0/\Delta T)^2 \rceil$ .

where  $\Delta T = T_0$ , if  $T = T_0$ , and  $\Delta T = \text{Preceding temperature} - T$ , in another case. Furthermore,  $\lceil a \rceil$  denotes the smallest integer greater than or equal to  $a$ .

Preliminary experiments indicate that strategy a) induces improvement for small problems, but for larger problems 7, 8, and 9, the initial solution cannot be improved. Strategy b) allows the improvement of the initial solution only for problems with less than 10 000 consumer points. On the other hand, using strategy c) the procedure can improve the initial solution for all problems except for the largest containing 20 000 consumer points. Therefore, to compare the **Simulated Annealing Procedure** and the **Tabu Search Procedure**, strategy c) is used.

The criterion to detect the proper moment to diversify or to intensify in the **Tabu Search Procedure** is specified as follows: diversification is applied during the first  $ndiv$  iterations; hence  $DIV = \text{True}$  for the  $ndiv$  iterations; then,  $DIV = \text{False}$  for the rest of the iterations.

The intensification criterion is satisfied whenever the current best solution is not improved during  $noimp = 5$  consecutive iterations of the regular stage. Then the  $INT = \text{True}$  and the current best solution BS become the current solution CS. The Intensification criterion remains satisfied until a solution better than BS is generated. Then we return to the regular stage. Note that the value  $noimp = 5$  was determined using preliminary tests.

TABLE IV  
NEIGHBORHOOD SIZE TS

Neighborhood Size NEIG	Average Cost Solution \$	Average Cost Improvement \$
10	316851	23680
15	306778	33753
20	307575	32956
25	303783	36748
30	306339	34192
35	295574	44957
40	307314	33217

TABLE V  
TABU LIST SIZE FOR TS

Size of Tabu List	Average Cost Solution \$	Average Cost Improvement \$
3	314881	25650
4	312609	27922
5	319344	21187
6	321853	18678
7	319772	20759
8	318965	21566

TABLE VI  
DIVERSIFICATION EFFECT

Diversification N° of Iterations	Average Cost Solution \$	Average Cost Improvement \$
0	318179	22352
5	324640	15891
10	319617	20914
15	319666	20865
20	315612	24919
25	318737	21794

The value of  $ITMAX$  for the stopping criterion was fixed to 20 using preliminary tests. Stopping criterion is specified in terms of the maximal number of consecutive iterations where BS is not improved. It follows that the procedure stops if during 5 consecutive iterations in the usual stage are followed by 15 consecutive iterations in the intensification stage, the value of BS remains unchanged.

The calibration of the **Tabu Search Procedure** was carried out considering the influence of the parameters in the quality of the solution from the greater to lesser values. In the first place we calibrated the size of the neighborhood; then the size of the tabu list, and finally the effect of the diversification, measured as the number of diversified search iterations (Tables IV, V and VI). Note that this calibration procedure allows for the partial exploration of the possible values of each parameter. It only illustrates the impact of the parameters in the solution process, but does not assure the delivery of optimum values of NEIG, TL, and  $ndiv$ .

Table IV includes the results obtained for seven different neighborhood sizes  $NEIG$ : 10, 15, 20, 25, 30, 35, and 40. In these tests, the size of the Tabu List is fixed at 4, and the diversification strategy is applied during the first  $ndiv = 5$  iterations. The best average cost improvement is obtained with  $NEIG = 35$ .

TABLE VII  
RESULTS WITH SA

Instance N°	Average Solution Cost \$ SA	Cost Improvement \$	Percentage of Improvement [%]	Running Time [s]
1	223959	57546	20.44	0.53
2	298129	42402	12.45	0.95
3	390460	22236	5.39	1.97
4	511695	24584	4.58	3.55
5	685091	32960	4.59	7.15
6	719397	28949	3.87	7.64
7	919789	634	0.07	18.49
8	1292195	2082	0.16	57.77
9	1648472	0	0	152.20

TABLE VIII  
RESULTS WITH TS

Instance N°	Average Solution Cost \$ TS	Cost decreasing \$	Percentage of Improvement [%]	Number of Neighbor Solutions	Running Time [s]
1	215247	66258	23.54	22225	0.29
2	300412	40119	11.78	14700	1.34
3	379184	33512	8.12	32970	1.82
4	458861	77418	14.44	46340	5.31
5	590606	127445	17.75	60550	6.21
6	619350	128996	17.24	109200	10.48
7	820869	99554	10.82	166215	18.74
8	1202549	91728	7.09	126630	49.31
9	1451307	197165	11.96	294630	183.43

Six different sizes for the Tabu List  $|TL|$  are compared: 3, 4, 5, 6, 7, and 8. The results in Table V are obtained with  $|NEIG| = 35$  and  $ndiv = 5$ . The best average cost improvement is obtained with a Tabu List of size  $|TL| = 4$ . It is worth noting that the solutions are always better than those found with the **Simulated Annealing Procedure** (see Tables II and III).

Diversification was applied during the first  $ndiv$  iterations for 6 different values of  $ndiv$  (0, 5, 10, 15, 20, and 25) using  $|NEIG| = 35$  and  $|TL| = 4$ . The results in Table VI indicate that the best average cost improvement is obtained when the search is diversified during the first 20 iterations.

### C. Comparison of Proposed Procedures

By using the nine problems described in Section IV-A, we compare the two procedures where the parameters are fixed according to the calibration analyses of Section IV-B. The results are summarized in Tables VII and VIII.

For the **Simulated Annealing Procedure**, if the number of iterations at temperature  $T$  is  $NT = \lceil 1 + \log(T_0/\Delta T)^2 \rceil$ , then the stopping criterion using  $T_{min} = 10^{-40}$  induces a maximal number of iterations equal to 423 831 when  $\alpha = 0.85$ . Hence, this is the number of iterations or the number of solutions generated in the resolution of each problem in Table VII.

In order to have a fair comparison of the two procedures, we have to take into account that at each iteration of the **Tabu Search Procedure**, 35 solutions are generated while only one is generated at each iteration of the **Simulated Annealing Procedure**. For the same problem,

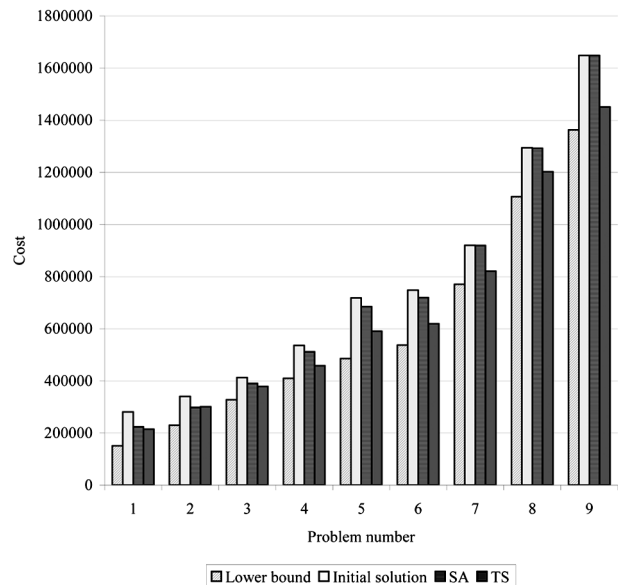


Fig. 3. SA and TS cost functions.

the number of solutions with TS is on the fifth column of Table VIII. In all cases, we can see that TS visits fewer neighborhood solutions than SA.

The results in Table VII indicate that the efficiency (specified in terms of the percentage of the cost improvement with respect to the initial solution) of the **Simulated Annealing Procedure** decreases as the problem size increases. Furthermore the percentage of improvement is very small for larger problems, and the procedure is unable to improve the initial solutions of the largest problem.

The results obtained using the **Tabu Search Procedure** are summarized in Table VIII. The **Tabu Search Procedure** generates an improvement for the problems which is much more stable than the **Simulated Annealing Procedure**. As expected, the number of iterations (and hence the number of solutions generated) increases with the problem size.

If we compare the results in Tables VII and VIII, we observe that the **Tabu Search Procedure** generates solutions with a better percentage of improvement than the **Simulated Annealing Procedure** for all problems, even if the number of neighborhood solutions generated is smaller. Hence, the **Tabu Search Procedure** is clearly more efficient than the **Simulated Annealing Procedure**. Fig. 3 also shows a graphic comparison of the efficiency of both methods.

Fig. 4 illustrates the behavior of the value of the current best solution during a typical execution of the **Simulated Annealing Procedure** by solving problem 8. The straight line corresponds to the value of the initial solution. In the first 150 000 iterations, the **Simulated Annealing Procedure** examines solutions having a worse value than the initial solution. Afterward, the value of the objective function improves. This behavior can be explained by the fact that several solutions involving changes in the type of conductors are examined in the early stage of the procedure which causes a cost increase. Nevertheless, as the process

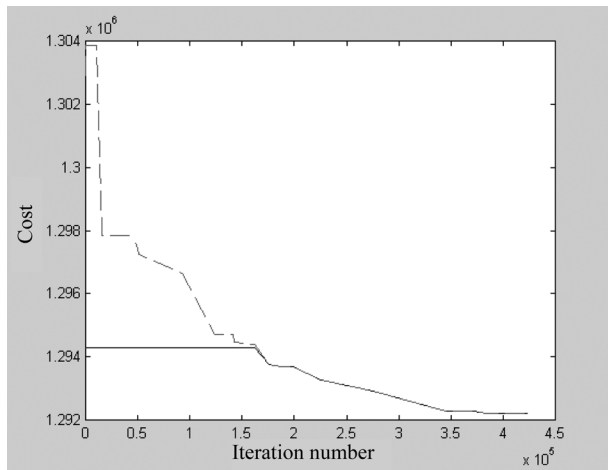


Fig. 4. Iteration progress of SA for instance 8.

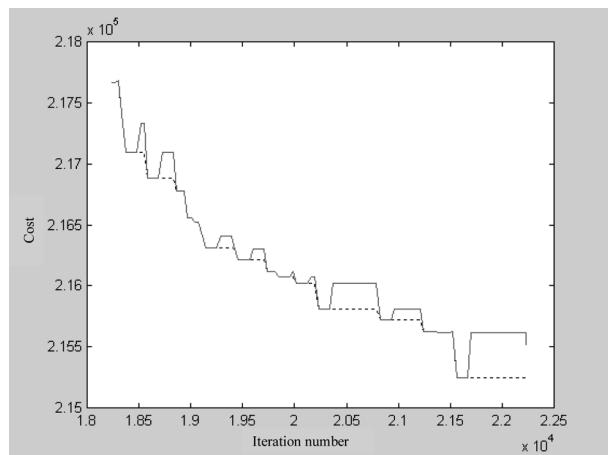


Fig. 5. Iteration progress of TS for instance 1.

advances, the search reaches a stable solution after 400 000 iterations.

Fig. 5 illustrates the last iterations of the **Tabu Search Procedure** applied to problem 1. The first curve corresponds to the cost of the current solution, and its lower envelope illustrates the cost of the current best solution.

## V. CONCLUSION

In this paper we introduce and compare two neighborhood search techniques (the **Simulated Annealing Procedure** and the **Tabu Search Procedure**) to solve the Distribution Tree Problem. The solution of this problem is the set of ideal distribution out-trees in order to supply the current demand. This solution (the ideal network) can compare with the real network to evaluate the deviation between both, and then plan the actions to reduce this deviation. Both methods can solve problems with 20 substations and up to 20 000 consumer points. The largest problem including 30 000 consumer points can be solved efficiently only with the **Tabu Search Procedure**. The results indicate that the **Tabu Search Procedure** is more efficient in general, and it requires less neighborhood solutions

than the **Simulated Annealing Procedure** to reach a similar level of efficiency. In the analysis of a distribution network corresponding to a given region, each node can not only identify one consumer point but a set of them representing for example, a sub-region. Then, a network with 30 000 consumer points seems to be adequate to reflect the situation of a medium size city. Based on this consideration, the computer program developed can be used for engineers to analyze distribution networks at different planning levels: the expansion, the structural and the operational level. To run each situation it is only required to adequately input the data.

## REFERENCES

- [1] U. G. Knight, "The logical design of electrical networks using linear programming methods," *Proc. Inst. Elect. Eng.*, vol. 33, pp. 306–314, 1960.
- [2] H. L. Willis, H. Tram, M. V. Engel, and L. Finley, "Optimization applications to power distribution," *IEEE Comput. Appl. Power*, vol. 8, no. 4, pp. 12–17, Oct. 1995.
- [3] S. K. Khator and L. C. Leung, "Power distribution planning: A review of models and issues," *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1151–1158, Aug. 1997.
- [4] W. F. Chang and Y. C. Wu, "Optimal reliability design in an electrical distribution system via a polynomial-time algorithm," *Int. J. Elect. Power Energy Syst.*, vol. 25, pp. 659–666, Oct. 2003.
- [5] V. Parada, J. A. Ferland, M. Arias, and K. Daniels, "Optimization of electrical distribution feeders using simulated annealing," *IEEE Trans. Power Del.*, vol. 19, no. 3, pp. 1135–1141, Jul. 2004.
- [6] C. T. Huddleston, R. P. Broadwater, and A. Chandrasekaran, "Reconfiguration algorithm for minimizing losses in radial electric distribution-systems," *Elect. Power Syst. Res.*, vol. 18, pp. 57–66, Jan. 1990.
- [7] N. D. R. Sarma and K. S. P. Rao, "A new 0-1-integer programming method of feeder reconfiguration for loss minimization in distribution-systems," *Elect. Power Syst. Res.*, vol. 33, pp. 125–131, May 1995.
- [8] P. Avella, D. Villacci, and A. Sforza, "A Steiner arborescence model for the feeder reconfiguration in electric distribution networks," *Eur. J. Oper. Res.*, vol. 164, pp. 505–509, Jul. 16, 2005.
- [9] K. K. Li, T. S. Chung, G. J. Chen, and G. Q. Tang, "A tabu search approach to distribution network reconfiguration for loss reduction," *Elect. Power Compon. Syst.*, vol. 32, pp. 571–585, Jun. 2004.
- [10] Y. J. Jeon, J. C. Kim, J. O. Kim, J. R. Shin, and K. Y. Lee, "An efficient simulated annealing algorithm for network reconfiguration in large scale distribution systems," *IEEE Trans. Power Del.*, vol. 17, no. 4, pp. 1070–1078, Oct. 2002.
- [11] J. Z. Zhu, "Optimal reconfiguration of electrical distribution network using the refined genetic algorithm," *Elect. Power Syst. Res.*, vol. 62, pp. 37–42, May 28, 2002.
- [12] B. Turkay and T. Artac, "Optimal distribution network design using genetic algorithms," *Elect. Power Compon. Syst.*, vol. 33, pp. 513–524, May 2005.
- [13] M. Arias, H. Sanhueza, and V. H. Quintana, "Nuevo algoritmo para cálculo de flujo de potencia en alimentadores de distribución," in *XI Congreso Chileno de Ingeniería Eléctrica*, Punta Arenas, Chile, 1995.
- [14] S. K. Goswami and S. K. Basu, "Direct solution of distribution-systems," *Proc. Inst. Elect. Eng. C Gen. Transm. Distrib.*, vol. 138, pp. 78–88, Jan. 1991.
- [15] G. X. Luo and A. Semlyen, "Efficient load flow for large weakly meshed networks," *IEEE Trans. Power Syst.*, vol. 5, no. 4, pp. 1309–1316, Nov. 1996.
- [16] R. Prim, "Shortest connection networks and some generalization," *Bell Syst. Tech. J.*, vol. 36, pp. 1389–1401, 1957.
- [17] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [18] F. Glover and M. Laguna, *Tabu Search*. Norwell, MA: Kluwer, 1997.

**Víctor Parada** received the M.S. degree in industrial engineering from the Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil, and the D.Sc. degree in computer science and systems engineering from the Universidade Federal do Rio de Janeiro.

Currently, he is Professor at the University of Santiago of Chile. His current research interests are on heuristic methods to solve practical optimization problems.



**Jacques A. Ferland** received the Ph.D. degree in operations research from the Stanford University, Stanford, CA, in 1971.

Currently, he is full Professor in the Department of Computer Science and Operations Research of the Université de Montréal, Montréal, QC, Canada. His research interests include fundamental and applied research projects.

**Miguel Arias** (SM'01) received the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 1987 and 1990, respectively.

Since 1980, he has been with the Electrical Engineering Department of the University of Santiago, Santiago, Chile, University of Santiago, where he is Titular Professor. His main research interests are in the areas of planning, analysis, and operation and control of electric powers and distribution systems.

**Pablo Schwarzenberg** is currently pursuing the M.Eng. degree in informatics at the University of Santiago, Santiago, Chile.

He is an Informatics Engineer at the University of Santiago.

**Luis S. Vargas** (SM'08) received the M.Sc. degree from the Universidad de Chile, Santiago, in 1987 and the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada.

Since 1994, he has been with the Universidad de Chile where he is currently an Associate Professor. His main research interests are in the areas of supply and demand for the forecasting and expansion planning of energy systems.