Contributions Concerning the Determination of Radial Structures from Meshed Networks

HORIA ANDREI
Faculty of Electrical Engineering
University Valahia Targoviste
18-20 Blv. Unirii, Targoviste, Dambovita
ROMANIA
GIANFRANCO CHICCO
Dipartimento di Ingegneria Elettrica
Politecnico di Torino
Corso Duca degli Abruzzi, 24-10129, Torino
ITALY
COSTIN CEPISCA
Faculty of Electrical Engineering
University Politehnica Bucharest
313 Splaiul Independentei, sector 6, Bucharest
ROMANIA
ION CACIULA
Faculty of Electrical Engineering
University Valahia Targoviste
18-20 Blv. Unirii, Targoviste, Dambovita
ROMANIA

Abstract – **One of the most important aspects of the electro-energetic systems analysis, with practical applications mainly to distribution system reconfiguration and optimization, is the identification of the radial structures that can be extracted from meshed network. Even the enumeration of the possible radial configurations extracted out of a weakly meshed network with given supply points can be a challenging task, due to the specific physical links that do not allow for defining a simple and general method for the radial structure determination. We propose, in this paper, a fast and efficient backtracking algorithm, to enumerate and identify all the possible radial configurations that can be obtained from a given meshed structure of a network. The procedure is illustrated and applied to a 33-node reconfigurable distribution systems widely used in the literature.**

Keywords: – radial structures, reconfiguration, topology, distribution systems.

1 Introduction

In the topological analysis of the electric circuits, the generation of the trees of the associated graphs to obtain radial configurations is a particularly challenging issue. For a connected graph with the total number of nodes *N,* the generation of the trees presumes the examination of all the possible combinations of the branches, in order to verify which of them does not contain loops and passes through all of circuit nodes. If *L* is the total number of graph branches, using this method the number of branch combinations to be examined is C_L . This method leads to a great number of operations, which increases exponentially with the number of nodes. For example, for a complete graph the number of solutions is *N*

Common methods for tree generation have been introduced in the literature. For instance, the iterative methods presented in [1], [2], [3] and [4] are based on elementary transformations through

which a tree is generated from another tree. The iterative methods described in [5], [6] and [7] are based on the modification of the branch-to-node incidence matrix through the elementary operations with rows and columns.

In this paper, the generation of the branch tree is addressed by using an efficient backtracking-based programming technique.

2 The calculation of the number of radial configurations

Let's consider a complete network composed of N nodes, B branches and S supply points. Any radial configuration obtained from this network has $Q = B\text{-}N\text{+}S$ open branches. The proposed procedure for extracting the radial configurations out of a meshed network structure starts from the observation that, given the complete network, all the branches in series along a path between two nodes can be conveniently replaced by a single equivalent branch r associated to a multiplicity factor m_r equal to the number of series branches. In fact, opening the equivalent branch r would correspond to opening one of the branches in series, so that there would be m_r possible combinations of open branches.

After the identification of the nodes in the complete network, the first step is then formation of the reduced network, by substituting all the series connections of the branches with their equivalent connection. The reduced network has N_R nodes and B_R branches. The number of open branches in each radial configuration is $Q_R = B_R\text{-}N_R\text{+}S$, with $Q_R=Q$.

A specific algorithm for identifying the radial configurations, as the one proposed in Section 3, is then applied to the reduced network, to obtain the resulting K radial configurations. Each of these radial configurations $k=1,\ldots,K$ corresponds to a set $\boldsymbol{R}^{(k)}$ of open branches. In order to calculate the total number of radial configurations that can be extracted out of the complete network, it is possible to associate to each configuration $k= 1,\ldots, K$ of the reduced network, characterized by the equivalent open branches $r \in \boldsymbol{R}^{(k)}$, the number n_k of different sets of open branches in the complete network. Considering the equivalent branch $r \in \boldsymbol{R}^{(k)}$ with its multiplicity m_r, its contribution to the number of open branches is equal to its multiplicity. Then,

$$n_k = \prod_{r \in R^{(k)}} m_r \tag{1}$$

The total number of radial configurations for the complete network is then found as

$$n_{total} = \sum_{k=1}^{K} n_k \tag{2}$$

3 The backtracking algorithm

In the search of some fundamental principles of the algorithms elaboration, backtracking is one of the most general techniques. Many problems which require the search of a solution set or an optimal solution which must satisfy certain restrictions can be resolved using this method [8].

In many applications of this method, the solution can be expressed under the shape of a vector $(x_1, x_2, \ldots x_n)$ where $x_1 \in S_i$, $\forall\ i \in \{1,2\ldots,n\}$, the sets S_i being finite and ordinate. Many times, for being resolved, the problem requires the founding of a vector which maximizes (minimizes or satisfies) a certain condition $P(x_1, x_2, \ldots x_n)$. Sometimes, we search all the vectors that satisfy P.

We presume that m_i is the cardinal of the set S_i. Be m $(m=m_1 * m_2 * \ldots * m_n)$ vectors with n components that can satisfy condition P. A direct method for founding the solutions, known as the brut force method, would be that to generate all these vectors and to choose only those that verify the conditions of the problem.

The backtracking method has as advantage the ability to get to the same answer, but making less tries. The base idea is to build the solution vector component by component and to use modified conditions $P_k(x_1,x_2,\ldots x_k)$, sometimes named functions, utile to verify if the formed vector has a success chance. The main advantage of this method is the next: if we find that the partial vector $(x_1, x_2,,.. x_k)$ can't lead to a "possible" solution, then $m_{k+1},\ldots m_n$ remained untested vectors can be totally ignored. Many of the problems that are resolved using the backtracking method require that the solutions satisfy a complex set of restrictions.

For any problem, these restrictions can be divided in two categories: implicit and explicit.

Definition 1.The explicit restrictions are rules that impose every x_k to take values from a given set.

The explicit restrictions depend of the specific condition i of each problem that must be resolved. In other words, all the vectors that satisfy the explicit restrictions define the space of possible solutions $S_1xS_2\ldots xS_n$.

Definition 2.The implicit restrictions are rules that determine which vectors from the domain of

the x_k solutions satisfy function criterion. Thus, the implicit restrictions describe the way in which the component elements x_k must be tied among them.

The input information, e.g. the explicit restrictions, includes the number of nodes, the number of branches and the topological structure of circuit. For the graph corresponding to the circuit, the node-to-node connection matrix $\boldsymbol{A}_{(N-1)x(N-1)}$ contains binary elements 1 and 0, respectively corresponding to the existence or not of a branch connecting the pair of nodes identified by the matrix row and column, so that

$$A_{ij} = \begin{cases} 1, \text{if a branch exists between nodes } i \text{ and } j \\ 0, \text{if a branch does not exists between nodes } i \text{ and } j \end{cases}$$

The matrix A is symmetrical, that is, $A_{ij} = A_{ji}$ for $i, j = 1,...,N$. The number of non-null elements in the off-diagonal portion of the matrix A is equal to twice the number of the graph branches.

The algorithm has been implemented in the C++ language. The solution is generated by using a set of vectors, each of which will contain the nodes of a tree.

If we consider that the generation of the solutions is made in pushdown storage, then at the base of pushdown storage the number of the first node (supply node) will be loaded. The filling of the pushdown storage with the following elements, from 2 to N, will take note of two conditions (e.g. the implicit restrictions):

1. a branch exists between the new node and the previous one;
2. the new node does not make a loop with the previous ones.

The output information contains the trees (radial configurations) and the co-trees (the open branches or the branches which are not contained in the tree).

In order to explain the details of the procedure, a simple illustrative example is provided here for a meshed system with $N = 5$ nodes, $B = 6$ branches and $S = 1$ supply points (Figure 1). The number of open branches for each radial configuration is $Q = B-N+S = 2$.

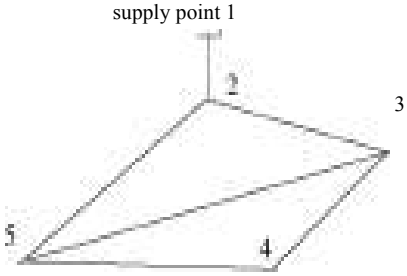

Figure 1. Layout of the system for the illustrative example

The list of integer values $v = [vi\ v_2\ ...]$ with variable size is introduced for assisting the description of the example. The basic rule concerning the selection of the components of v is that for $j > 0$ the components v_{j+1} of v may contain the number of any node connected to the node v_{2j}. The connections are considered to be orientated, so that it is possible to use a connection previously accepted, but only in the opposite direction.

The procedure starts from the root node and continues until all the possible radial configurations are reached. The first component is set to $v_1 = 1$, so that $v = [1]$. Then, the second component starts from $v_2 = 1$ and is increased by one until a node with a new connection to v_2 is found. Hence, $v_2 = 2$, and the list is updated to $v = [1\ 2]$. The third component starts from $v_3 = 1$ and is incremented until any connection to v_2 is found, thus stopping at $v_3 = 1$, so that $v = [1\ 2\ 1]$. The fourth component then starts from $v_4 = 1$ and is increased to search a new connection to node $v_3 = 1$. Since no new connection is found, the component v_4 is eliminated and the procedure operates backtracking to the component v_3, which is updated by increasing it by one unit with respect to its previous value, thus searching for any new connection to node $v_2 = 2$ starting from $v_3 = 2$. The connection is found for $v_3 = 3$, so that $v = [1\ 2\ 3]$. The procedure continues by considering the fourth component starting from $v_4 = 1$ and increasing it to search for a new connection to node $v_3 = 3$. The connection is found for $v_4 = 2$, and $v = [1\ 2\ 3\ 2]$. For the fifth component, the starting value $v_5 = 1$ is satisfactorily connected to node $v_4 = 2$, thus updating the list $v = [1\ 2\ 3\ 2\ 1]$. The sixth component starts from $v_6 = 1$ and increases from 1 to 5 without finding any new connection to node $v_5 = 1$, so that v_6 is eliminated and the procedure operates another backtracking to v_5, to perform a new search starting from the increased value $v_5 = 2$ and searching for any new connection to node $v_4 = 2$. Since the orientated connection from node 2 to node 3 has already been found, the solution is $v_5 = 5$, and $v = [1\ 2\ 3\ 2\ 5]$. The procedure continues by setting $v_6=1$ and searching for a new connection to $v_5 = 5$, obtaining $v_6 = 2$ and updating $v = [1\ 2\ 3\ 2\ 5\ 2]$. The next component starts from $v_7 = 1$, and the search for a new connection to node $v_6 = 2$ directly provides $v_7 = 1$, thus leading to $v = [1\ 2\ 3\ 2\ 5\ 2\ 1]$. Successively, starting from $v_8 = 1$ the search for new connections to node $v_7 = 1$ provides no solution,

so that the component v_8 is eliminated. The search continues by using the initial (increased) component $v_7 = 2$ and searching for new connections to $v_6 = 2$, again with no solution. The component v_7 is then eliminated as well, performing another backtracking to the upgraded component $v_6 = 3$ and searching again for new connections to node $v_5 = 5$. The connection from node 5 to node 3 does exist, but it cannot be selected, since is would close a loop in the system on nodes 2, 3 and 5. Then, the search proceeds with the increased value $v_6 = 4$, finding the connection from node 5 to node 4. Hence, $v = [1\ 2\ 3\ 2\ 5\ 4]$. At this point, all the nodes have been reached and the first radial structure has been formed. The procedure identifies the corresponding two open branches (3-4) and (4-5).

The search continues by upgrading $v_6 = 5$ and searching for new connections to $v_5 = 5$. There is no solution, so that the procedure operates backtracking on v_5. Again, v_5 cannot be upgraded, so that there is a new backtracking on v_4, setting $v_4 = 3$ and searching for new connections to node $v_3 = 3$. The solution is found for $v_4 = 4$, so that the list is updated to $v = [1\ 2\ 3\ 4]$. The search continues with the same scheme, finding successively the eight solutions reported in Table 1.

The procedure stops when a complete backtracking to the root node has been performed. This procedure is able to find out all the solutions in succession by using the same scheme. If the backtracking is continued by upgrading the first component v_1, this corresponds to change the root node, so that it is possible to find out all the radial configurations that can be obtained by assuming any node of the network as root node. Clearly, this option is not directly useful to analyze a distribution system with specified location of the root node.

Table 1. Radial solution for the illustrative example

solution	V	*open branches*
1	[1 2 3 2 5 4]	(3,4) - (3,5)
2	[1 2 3 4 3 2 5]	(3,5) - (4,5)
3	[1 2 3 4 3 5]	(2,5) - (4,5)
4	[1 2 3 4 5]	(2,5) - (3,5)
5	[1 2 3 5 4]	(2,5) - (3,4)
6	[1 2 5 3 4]	(2,3) - (4,5)
7	[1 2 5 3 5 4]	(2,3) - (3,4)
8	[1 2 5 4 3]	(2,3) - (3,5)

4 Example

An example of calculation of the total number of radial configurations that can be extracted out of a weakly meshed distribution system is provided here by considering a distribution system structure widely referenced in the literature [9]. The complete network (Figure 2) has $N = 33$ nodes, $B = 37$ branches and $S = 1$ supply points. Any radial configuration obtained from this network is composed of $Q = B\text{-}N+S = 5$ open branches.

The *reduced network* is formed by substituting all the series connections of the branches with an equivalent connection, as shown in Figure 3, where the equivalent branches are identified by using the letters from A to N. The number of series connections corresponding to each equivalent connection $r \in R$ is taken into account by means of the multiplicity factor m_r, as shown in Table 2. For instance, in the reduced network the path from node 1 to node 20, containing three series branches in the complete network, is associated to the equivalent connection labelled as C with multiplicity factor $m_c = 3$

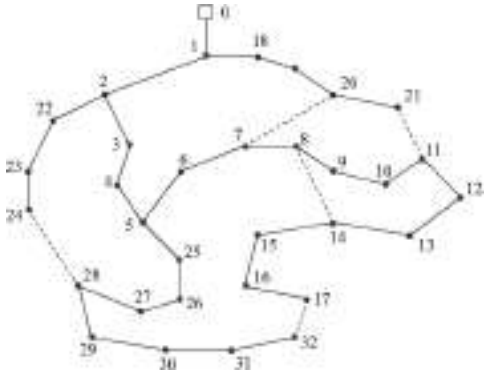

Figure 2. Complete network

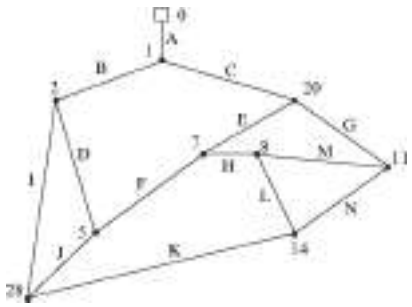

Figure 3. Reduced network

The proposed algorithm for identifying the radial configurations is then applied to the reduced network, obtaining the results shown in Table 3, with $K = 463$ resulting radial configurations. Each radial configuration $k = 1,\ldots, K$ corresponds to a set of open equivalent branches $r \in \boldsymbol{R}^{(k)}$, each of which has multiplicity m_r, and its contribution n_k to the number of open branches is calculated from (1). For instance, the case $k = 1$ corresponds to the open branches D, I,

F, H and L in the reduced network, with the corresponding multiplicities indicated in Table 2, so that $n_1 = m_D\, m_1\, m_F\, m_H\, m_L = 24$.

The total number of radial configurations for the complete network under test, calculated from (2), is $n_{,o,} = 50751$. The computation time is less than half a second on a 1.8 GHz Intel Pentium® IV personal computer.

Table 2. Equivalent branch multiplicity

r	m_r
A	**1**
B	**1**
C	**3**
D	**3**
E	**1**

r	m_r
F	2
G	2
H	1
I	4
J	4

r	m_r
K	**8**
L	**1**
M	**3**
N	**3**

5 Conclusions

An efficient procedure for computing the number of radial configurations that can be extracted from a meshed network has been presented.

The results of the proposed method are of particular interest for setting up reference values for the optimal reconfiguration of distribution systems [10-22]. The optimal reconfiguration is a global optimization problem with different local minima, so that specific deterministic and meta-heuristic procedures have been used to search for the global optimum. These procedures typically analyze a very high number of radial configurations, but none of them is able by definition to ensure that the global optimum has been found. The knowledge of the number of possible radial configuration allows for setting up an indicator of the relative number of configurations analyzed, defined as the ratio between the number of configurations analyzed and the total number n_{tot} for the same network. Furthermore, for small systems in which the relative number of configurations is significantly high, the use of global optimization methods could be ineffective, since the number of configurations analyzed could be high (sometimes even analyzing the same configuration more than once) without ensuring that the global optimum is reached.

Resorting to exhaustive search or setting up suitable control mechanisms applied to the evolution of the numerical procedure (e.g., of branch and bound type) would be better alternatives in these cases

References

[1] Mayeda W and Seshu S, *Generation of Tree without Duplications*, IEEE Trans. Circuit Theory, CT-12, 181-185, 1965.

[2] Paul A T Jr., *Generation of Direct Trees and 2 - Trees without Duplications*, IEEE Trans. Circuit Theory, **CT - 14,** 354 - 356, 1967.

[3] Mayeda W, *Graph Theory,* John Wiley and Sons, New York, 1972.

[4] Lin PM, *Symbolic Network Analysis,* Studies in Electric and Electronic Engineering, Elsevier, Science Publishing Company Inc., Amsterdam-Oxford-New-York-Tokyo, 1991.

[5] Mayeda W, Hakimi SL, Chen WK and Deo N, *Generation of Complete Trees,* IEEE Trans. Circuit Theory, **CT-15,** pp.101-105, 1968.

[6] Hassoun M M and Lin P M, *A new Network Approach to Symbolic Simulation of Large - Scale Networks,* Proc. IEEE - ISCAS, 806-809, 1989.

[7] Iordache M and Dumitriu L, *Tree or s-Forest Generation Diakoptic Method Used in Large - Scale Electric and Electronic Circuit Symbolic Analysis,* Proc. IEEE-SMACD, 103-113, 1994.

[8] Andrei H, Cepisca C, Chicco G, Dragusin V and Spinei F, *An efficient topological algorithm for determination of the equivalent network and network functions for reciprocal multipole,* WSEAS Transactions on Systems, Vol.5, No.l, Jan. 2006, 71-77.

[9] Baran M E and Wu F F, *Network reconfiguration in distribution systems for loss reduction and load balancing*, IEEE Trans. on Power Delivery 4 (1989) 1401- 1407.

[10] Augugliaro A, Dusonchet L and Riva Sanseverino E, *Genetic, Simulated Annealing and Tabu Search Algorithms: Three Heuristic Methods for Optimal Reconfiguration and Compensation of Distribution Networks,* European Trans, on Electric Power 9, 1 (1999) 35-41.

[11] Carpaneto E, Chicco G and Roggero E, *Comparing deterministic and simulated annealing- based algorithms for minimum losses reconfiguration of large distribution systems,* Proc. IEEE Porto Power Tech 2001, Porto, Portugal, Sept. 10-13, 2001, paper PST3-237.

[12] Chang H - C and Kuo C-C, *Network reconfiguration in distribution systems*

using simulated annealing, Electric Power Systems Research 29 (1994) 227-238.

[13] Cherkaoui R, Bart A and Germond AJ, *Optimal configuration of electrical Distribution networks using heuristic methods,* Proc. 11th PSCC, Avignon, France (Aug. 1993) 147-154.

[14] Jeon Y-J, Kim J-C, Kim J-O, Shin J-R and Lee KY, *An efficient Simulated Annealing Algorithm for Network Reconfiguration in Large – Scale Distribution Systems,* IEEE Trans, on Power Delivery 17, 4 (October 2002) 1070-1078.

[15] Mori H and Ogita Y, *A Parallel Tabu Search Based Method for Reconfigurations of Distribution Systems*, Proc. IEEE/PES Summer Meeting, Seattle, WA (July 2000), 1, 73-78.

[16] Nara K, Shiose A, Kitagawa M and Ishihara T, *Implementation of genetic algorithm for distribution systems loss minimum reconfiguration,* IEEE Trans, on Power Systems 7, 3 (August 1992) 1044-1051.

[17] Parada V, Ferland JA, Arias M and Daniels K, *Optimization of electrical distribution feeders using simulated annealing,* IEEE Transactions on Power Delivery 19, 3 (July 2004) 1135-1141.

[18] Peponis G and Papadopoulos M, *Reconfiguration of radial distribution networks: application of heuristic methods on large-scale networks,* IEE Proc. Gen. Trans. Distr. 142**,** 6 (Nov. 1995) 631-638.

[19] Reeves CR (ed.), *Modern heuristic techniques for combinatorial problems* (ISBN 0-470-22079-1) Blackwell Scientific Publ, Oxford, UK (1993).

[20] Sarfi RJ, Salama MMA and Chikhani AY, *A survey of the state of the art in distribution system reconfiguration for system loss reduction,* Electric Power Systems Research 31 (1994) 61-70.

[21] Civanlar S, Grainger J, Yin H and Lee S, *Distribution feeder reconfiguration for loss reduction*, IEEE Trans, on Power Delivery 3 (1988) 1217 - 1223.

[22] Andrei H, Caciula I , and Chicco G, *An efficient algorithm for forming radial structures from meshed networks,* Proc. the 6th WESC, Torino, July (2006), 175-180.