An Improved Mechanism of Leaf Node Identification for Radial Distribution Networks

D. P. Sharma[1], A. Chaturvedi[1], G.Purohit[2] and G.Prasad[1]
[1]National Institute of Technology Karnataka, Surathkal- 575025, India
[2]Banasthali University, Rajasthan-304022, India.

Abstract-*Optimal operational & control aspects of distribution networks have been a thrust research area in academics as well as in industries since last two-three decades. In day to day practice, every one of us uses services offered by public utility distribution networks namely, water distribution network, electrical power distribution network etc. Operational topology of power distribution networks are radial in nature and hence are termed as radial distribution networks (RDN$_s$) .Network reconfiguration has been exercised as one of the prime and widely adopted approach for operational, maintenance and control activities of an RDN. Since last two decades, researchers have been using evolutionary computation based techniques (Genetic Algorithm [1], Simulating Annealing etc) for optimal network reconfiguration. The choices of network topology for the specific purpose/application requires a careful analysis of its merits and demerits. In RDN$_s$, the ultimate performance of a specific network topology is usually assessed by an iterative algorithm known as Load flow analysis (LFA) and its execution results in estimation of voltages, currents and losses profiles which in turn decides, whether the obtained network topology is good or bad. Thus, an obvious need is in favor of developing a conceptual frame work for faster load flow algorithm especially to meet near real-time operation requirements. In this paper, a simple and computationally efficient method for terminal (leaf) nodes identification is presented and thus, by integrating this subroutine in LFA definitely leads to an efficient and faster LFA algorithm.*

I. INTRODUCTION

In the last two decades, many efficient and reliable load flow analysis techniques are widely used and reported for power system operation, control and planning. At the same time as the developments of automated distribution systems solutions have increased considerably, for these solutions to be efficient, there arises a need for inherent mechanism that leads to a faster LFA algorithm. According to various reported studies in the literature, methods dealt with the load flow analysis are divided into two categories. The first group of methods is based on the forward-backward sweep mechanism, while solving ladder networks. On the other hand, the second groups of methods are obtained by modifying the existing methods such as Newton-Raphson, Gauss-Seidel etc.

Nanda et. al reported load flow study [2] , here the procedure for finding leaf node is of order $O(n^2)$ and used repeatedly. Where n denotes number of nodes. So it is obvious that this approach does not lead to an efficient load flow analysis.

Mok et. al suggested an load flow analysis [3], that was based upon reverse sweep ,in that researchers used inefficient method for leaf identification. Arvindhababu et. al have used matrix with reverse sweep for load flow analysis [4],which also used inefficient procedure of leaf finding.

Prasad et al suggested a simple algorithm for load flow Analysis [5]. In this work, author exploits the properties of tree with efficient use of data structure. However, it is based upon an inefficient procedure for finding leaf node, which requires step execution of an order $O(n^2)$, and same procedure when used repeatedly in convergence loop ,further made it more inefficient.

The paper outline is as follows; the proposed algorithm is reported in section 2, whereas the pseudo code for leaf identification is given in section 3.The step by step procedure for leaf node identification is illustrated on a simple example network in section 4. Complexity analysis of proposed algorithm is discussed and

978-1-4244-8052-4/11/$26.00 ©2011 IEEE

reported in section 5. Implementations of proposed algorithm for two practical RDN are presented in section 6, while section 7 concludes the paper.

II. PROPOSED ALGORITHM

The proposed algorithm is quite different from existing algorithm [5]. In existing algorithm the sending node list scanned repeatedly for finding leaf nodes. While, the proposed procedure scans the sending node list only for once and finally it counts the frequency of occurrences of each node present in the list. The nodes having zero frequency of occurrence will be treated as the leaf nodes and the nodes for which frequency of occurrence is greater than one represent the junction nodes of the network. Execution of this procedure leads to creation of a new array (termed as Leaf array) and it contains all the leaf nodes of the network. The time and space complexity of the proposed algorithm are of order O (n), that is a remarkable improvement as compared to [5], where the time and space complexity is of order $O(n^2)$ and O(n) respectively.

III. PSEUDO CODE FOR LEAF NODE IDENTIFICATION

The proposed algorithm can be executed using the following steps.

Where N : No of node in network.
Leaf : Array of Leaf
SEND : Array used to store the all sending nodes of an RDN
Temp : Array used to store the frequency of occurrences
i : denote the index of Array

1. Copy the list of sending node into an array SEND.

2. Initialize an array Temp with 0
/ Array that store frequency of all types of node of a network*/*

3. Initialize Leaf = φ ;
/ Initially Leaf array will have no node.*/*

4. for i =1:N
/ Loop used for counting the frequency.*/*
Temp(SEND(I))=Temp(SEND(I))+1;
end

5. for i=1:N
*If(Temp(i)==0) /*Condition of Leaf */*
Leaf=Leaf U (i)
end
end

6. Return Leaf.

IV. EXAMPLE NETWORK

Consider sample 11-nodes RDN as shown in Fig.1.

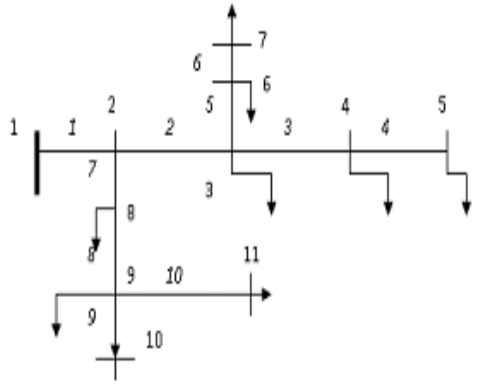


Fig1. Single line Sample RDN

Step 1: Prepare Array **SEND**

SEND	1	2	3	4	3	6	2	8	9	9	-
Index of Array →	1	2	3	4	5	6	7	8	9	10	11

The numbers stored in the above array are in the range of 1 to 11, thus the numbers which are not present in the **SEND** array indicate leaf nodes, and is the requisite outcome of the algorithm.

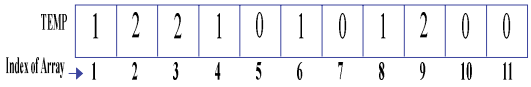
Step 2:

Since the specified range is 1 to 11, so initialize an array **Temp** of size 11(same as number of network nodes) with all of its elements zero as shown below:

TEMP	0	0	0	0	0	0	0	0	0	0	0
Index of Array →	1	2	3	4	5	6	7	8	9	10	11

Step 3:
Subsequently, the array **Temp** is updated with the frequency (occurrences) of nodes [6]. The

calculation of frequency is based upon the values stored in the **SEND** array.



Step 4:

Now for any arbitrary value of i (with in the range of 1 to 11) if **Temp (i) = 0** in the updates array **Temp** (result of step 3) then its corresponding index of array Temp will be a leaf node .In Temp array; the value of **Temp (i)** is zero for i=5, 7, 10, 11, thus node indexed as 5,7,10 and 11 are leaf nodes in 11-bus RDN.

V. COMPLEXITY ANALYSIS OF ALGORITHM

A simplified analysis of the time and space complexity [6, 7] of the proposed algorithm is briefed as:

A. *Time Complexity*:

The proposed algorithm does not include any nested loop. Hence, the time complexity of this algorithm is of order n, i.e. O(n).

B. ***Space Complexity***:

The proposed algorithm uses two arrays. **SEND** array is used to store sending node present in the network and **TEMP** array is used to store the frequency (occurrence) of each node of the network. The size of both the array, i.e. of SEND and **TEMP** array is equal to the number of nodes present in the network. Thus, the space complexity of this algorithm is also remains of order n i.e. O(n).

Thus, concluding the proposed algorithm is linear with reference to time and space both.

VI. TEST RESULTS

The proposed algorithm of leaf node identification is tested during LFA simulation run for two different RDN, i.e., 34-bus and 69-bus RDN. The number of steps execution required for getting converge LFA solutions using proposed algorithm is stated and compared with Prasad et. al.[5].

A. For 69-bus RDN, integration of the proposed algorithm of the leaf node identification as a subroutine in LFA simulation run, requires approximately 24275 steps execution, whereas , this figure approaches close to 42500 (approximately) when tested on using algorithm reported by prasad et. al.[5]. Thus, these statistical values of step execution shows a saving of 42.88% on using proposed algorithm compared to one reported in [5].

B. For 34-bus RDN, on using proposed approach, the number of steps execution are 5500(approximately) , while on using algorithm reported in [5], this figure attains a value 9600(approximately). Hence, also obtained saving in steps execution is 42.7%.

VII. CONCLUSIONS

Observations indicated in above paragraph clearly shows a remarkable saving in steps execution to get converge load flow solutions. It is also worth mentioning that this saving further improved as the network's size increases. An obvious outcome of this saving is a faster load flow algorithm, which further shortens the time required for network reconfigurations and thus supports near real-time requirements.

REFERENCES

[1] Goldberg, D. E. (1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading.

[2] J. Nanda, M.S. Srinivas, M. Sharma, S.S. Dey, L.L. Lai, "New findings on Radial Distribution system load flow algorithms", *Proc. IEEE conference*, 2000, pp. 1157-1161.

[3] S. Mok, S. Elangovan, C. Longjian, M. M. A. Salama, "A new approach for power flow analysis of balanced distribution systems" *Electric Machines and Power Systems*, Vol. 28, 2000, pp. 325-340.

[4] P. Aravindhababu, S. Ganapathy, K.R.Nayar, " A novel technique for the analysis of radial distribution systems", *International Journal of Electrical Power and Energy Systems*, Vol. 23, 2001, pp. 167-171.

[5] K. Prasad, N. C. Sahoo, A. Chaturvedi and R. Ranjan "A simple approach in branchcurrent computation in load flow analysis of radial distribution systems", International Journal of Electrical Engineering Education, in press (paper no. 3871).

[6] Cormen T. H., C.E. Leiserson, and R.L Rivest, Introduction to Algorithms, Prentice-Hall of India Pvt. Ltd., New Delhi, 2001.