Visibility Representation of Distribution System One-line Diagrams

P.S. Nagendra Rao and Ravishankar Deekshit
Department Of Electrical Engineering;
Indian Institute Of Science
Bangalore, INDIA
080-293-2365
nagendra@ee.iisc.ernet.in

Abstract—An novel method of generating one-line diagrams of radial distribution systems in the form of a Visibility graph is proposed in this paper. An algorithm that automatically generates an aesthetically 'pleasing' and 'readable' Visibility diagram is presented. In addition to the basic property of a Visibility graph that nodes and edges be represented by axis-parallel horizontal and vertical lines respectively, a new set of layout specifications (Aesthetic Criteria) suitable for easy visualization of the distribution system have been proposed. This work is based on the premise that in general, the network data of distribution systems does not contain any geographical information of node locations. Therefore the algorithm assumes that only the identity of the terminal nodes of all the edges are known.The proposed algorithm automatically determines node positions such that the specified aesthetic criteria are satisfied.

Keywords—Radial distribution systems,Visibility representation, Graphic User Interface (GUI), One-line diagram, Automatic graph drawing.

1. INTRODUCTION

Efficient planning and operation of distribution systems require many tools. One such useful tool is a program that can generate automatically the system single-line diagram *which is easy to read. The existing GUIs in use in* modern distribution systems provide generally a manual scheme for generating the one-line diagram. Such schemes are time-consuming,inflexible and prone to error. The problem of generating one-line diagrams would be relatively easy if some position information of nodes (geographical location,relative position etc) is available. But unfortunately many distribution systems do not have any information of geographical locations of various components in their data-base. Therefore, drawing algorithms that can generate the co-ordinates of the buses automatically are needed.

Most of the distribution networks are radial and we restrict our discussions here to such networks only. The topology of a radial distribution system can be represented by a graph with a tree structure. Various busses such as load points and branching points correspond to the nodes of the tree and the transmission line-segments correspond to the branches/edges of the graph. The substation transformer bus could be treated as the root node. Hence, the problem of generating a one-line diagram of a radial distribution system could be viewed as essentially that of generating the drawing of a rooted tree.

There have been a few (but very limited) attempts in the recent past wherein application of graph drawing algorithms for the generation of distribution system diagrams has been investigated. Ong et al., [1] have studied the application of three tree drawing algorithms (i.e the spring embedder algorithm,the controlled spring embedder algorithm and the rooted tree algorithm) for automatic generation of single line diagrams of distribution systems; The authors in [1], themselves state that the performance of the spring embedder algorithm, originally proposed by Fruchterman et al., [2] is not very good for networks with unsymmetrical structures such as the distribution network. The controlled spring embedder algorithm is a modification of the spring embedder algorithm. Here, in order to obtain good diagrams, the user has to provide the locations of some components using his judgement. The algorithm then places the other components relative to these fixed components. However, the high level of human intervention required is a distinct limitation. The rooted tree algorithm [3] is a popular tree drawing algorithm. This algorithm is useful for drawing graphs wherein, the nodes at the same distance (in terms of number of intermediate nodes) from the root node are to lie along a line.We therefore see that none of these algorithms are really attractive for generating one-line diagrams of distribution networks. In [4], the authors describe a new method of generation of one-line diagrams in which all branches are constrained to lie along only two mutually perpendicular directions. The authors in [4] use a Depth First Search (DFS) based traversal to identify all laterals of a feeder and then assign directions and edge distances such that a specified set of aesthetic criteria are satisfied.

In this work, a new method of automatically generating one-line diagrams of radial distribution systems in the form of a Visibility graph is presented. The branches and busses of *the distribution network are represented by axis-parallel* vertical lines and horizontal lines respectively. In addition a set of aesthetic criteria that enhance the 'readability' of the diagram are specified. The algorithm identifies and places all the components of the distribution feeder such that the resulting diagram satisfies the proposed specifications.

2. TERMINOLOGY

Graph related terminology used in the paper is introduced below . We use the terms bus-node, graph-network and edge-branch as synonymns. The directional terms 'upstream' and 'downstream' with reference to a node, represent the directions towards and away from the substation (root) respectively, from that node.

- Child node or off-spring:A Node that is adjacent to and downstream of a node 'i' is called the child node of node 'i'.
- Parent node:The node that is adjacent to and upstream of a given node 'i' is called the parent node of node 'i'.
- Root node: The node having no parent node.
- Base node: Node having more than one off-spring node.
- Leaf node: Node having no off-spring or child nodes .
- Sub-trees at a node:Trees rooted at the node consisting of only one of its child nodes and all the descendent nodes and edges of the particular child node.
- Lateral: A sub-graph that is framed by a leaf node and a base node.
- Level of a Node : The distance of a node from the root node in terms of intermediate edges.

3. The Graph Drawing problem

The objective of any graph drawing algorithm is to generate a diagram that has a pleasing appearance and is 'appropriate' for the particular application. These qualifying terms are rather subjective and hence, there is a need to translate them to a set of objective criteria.

Therefore, development of a graph drawing algorithm can be viewed to consist of two stages.The first is evolving a set of (non-subjective) specifications for the drawing,which ensure that the drawing complying with these specifications would be 'readable'. The second step would be the development of an algorithm, which can generate the co-ordinates of the nodes, such that the specifications evolved above are met when edges are drawn between these nodes.

The proposed specifications

A Visibility diagram of a graph is a drawing of the graph using axis parallel, disjoint horizontal lines for nodes and vertical lines for edges [5]. Visibility representations have been traditionally used for visualization of planar graphs in the areas of VLSI design and Computer science. A radial distribution system can be drawn as a planar graph with a tree structure and can therefore be drawn as a Visibility diagram. Thick horizontal lines correspond to the nodes of the distribution network and the vertical lines denote the feeder segments. Apart from above two requirements that nodes and edges be both represented by mutually perpendicular lines, we propose a set of layout specifications. The proposed specifications are:

1. All nodes at the same distance from the root node i.e at the same node level lie along the same horizontal line.
2. All line segments are of equal length.
3. All branches of a lateral are in a straight line.
4. The orientation of the longest lateral is fixed first and subtrees originating from this lateral are placed on either side of it based on the relative sizes of sub-trees.
5. At every stage, larger laterals are always laid out closer to the main lateral than the smaller ones.
6. No branches intersect.

It is seen, from specifications 4 and 5 that the layout of the graph depends on the size of each sub-tree. This neccesitates that we specify a measure to quantify the relative sizes of sub-trees. In our approach, we choose the largest path length possible from the root of the sub-tree to any other node in the sub-tree as an index of its size (W).

4. Solution approach

A close examination of the layout specifications reveals that node positions are determined based on choice of laterals. The main feeder is to be laid out vertically upwards and laterals belonging to sub-trees originating from the main feeder are placed to its left or right (alternately) depending on the relative sizes of the sub-trees. In addition, larger laterals are placed closer to the main lateral. Thus the procedure for computation of the co-ordinates of all nodes is subdivided into the following steps.

- Selection of main feeder
- Identification of laterals
- Assignment of directions.
- Computation of co-ordinates

Selection of the main feeder

In our approach, the longest lateral is chosen as the main feeder or lateral as follows. Firstly, we assign a level to each node (based on the number of edges between the node and the root node) and hence determine the node (N_{mx}) with the maximum level. Then we identify the main lateral by backtracking from node N_{mx} to the root node and picking up all the visited nodes.

Identification of laterals

Once the main feeder is chosen, other laterals are identified by a Depth First Search (DFS) procedure [6] starting from each base node of the main lateral. To start with, we assign a weight W to each node and sort all child nodes in decreasing order of their assigned weights. Starting from the base node nearest to the leaf node of the main lateral, a DFS traversal is performed on each sub-tree incident at each base node. A lateral is formed by picking up all the nodes visited starting from a base node and ending at a leaf node. In the DFS traversal, since all child nodes are sorted in decreasing order of their weights (W), at every stage, the longer laterals are always visited before smaller laterals. This step is illustrated with the help of Fig 1. Consider the sub-tree T originating from the base node 2 of a lateral. T1 and T2 are subtrees rooted at base node 10. The DFS traversal starts from base node 2 and proceeds in the direction of the dotted line. The traversal is decided by the weights W assigned to each node. Since $W(11) > W(15)$, the traversal always picks

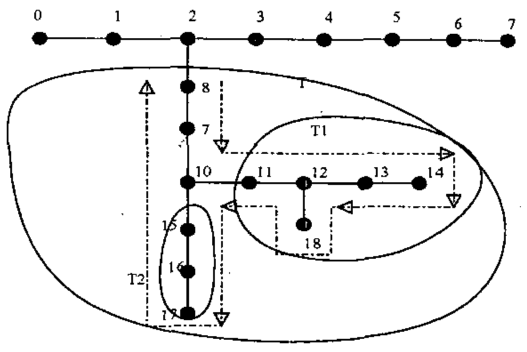

Figure 1. Formation of laterals

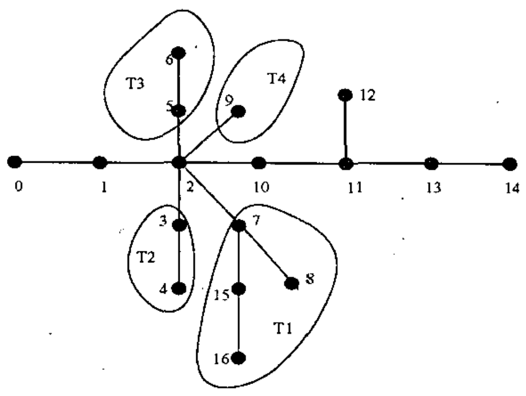

Figure 2. Orientation of laterals

up nodes belonging to sub-tree T1 ahead of those belonging to sub-tree T2. Similarly at base node 12, the sub-tree with nodes 13 and 14 are picked ahead of node 18. Therefore, the order in which the leaf nodes are visited during the BFS traversal are 14,18 and 17 starting respectively from base nodes 2,12 and 10. Therefore the laterals in this sub-tree are identified as L1={2,8,7,10,11,12,13,14}, L2={12,18} and L3={10,15,16,17}.

Assigning directions to laterals

The layout specifications require that the longest lateral or main feeder is first laid out vertically and the other sub-trees that originate from this lateral are placed on either side of it. Therefore nodes belonging to sub-trees rooted on the main feeder are assigned a direction depending on whether they should appear to its left or right. In our procedure, sub-trees originating from the same node on the main feeder are assigned alternate directions based on their relative weights. Therefore, the direction assigned to a sub-tree decides the directions of laterals contained in the sub-tree.As an example,consider Fig 2 in which T1,T2,T3 and T4 are sub-trees that are rooted at base node 2 on the main lateral.

The longest lateral consists of the following set of nodes: {0,1,2,10,11,13,14}. Therefore, this set is identified as the main lateral. It is seen that the size of the sub-trees rooted at base node 2 are such that $W(T1) \geq W(T2) \geq W(T3) \geq W(T4)$. Let T1 be assigned a direction to the right of the main lateral.The direction assigned to sub-tree T2 is then to the left of T. Sub-trees T3 and T4 are respectively assigned rightward and leftward directions. Therefore, the laterals L1={2,7,15,16},L2={7,8} (belonging to sub-tree T1) and L2={2,5,6} (belonging to sub-tree T3) are assigned directions to the right of the main lateral. Similarly laterals L3={2,3,4} and L4={2,9} are assigned a leftward direction.

Computation of co-ordinates

From the first two specifications, it follows that the 'y' co-ordinate assigned to a node depends on the specified position of the root node (XRoot,YRoot),level of the i^{th} node (level($Node_i$)) and the constant length of each branch (HEIGHT). For the i^{th} node,its 'y' co-ordinate is computed as: $Y(Node_i) = YRoot + level(Node_i) * HEIGHT$. According to aesthetic-3, all the branches belonging to a lateral should lie along a straight line and hence all nodes belonging to a lateral are assigned the same 'x' co-ordinate. Therefore, once the main lateral is identified, all its nodes are assigned an 'x' co-ordinate equal to that of the root node (XRoot). Subsequently the DFS traversal identifies the remaining laterals as explained in the previous sections. As and when a lateral is formed, it is placed depending on the direction assigned to it and the nearest available 'x' co-ordinate in that direction. In order to keep track of the available 'x' co-ordinates at each node level, two one dimensional arrays $Level\_array1$ and $Level\_array2$, each of size equal to the maximum node level, have been defined. The arrays $Level\_array1$ and $Level\_array2$ are used to indicate the available 'x' co-ordinate positions to the right and left respectively of the main feeder. After assignment of an 'x' co-ordinate to each node in the main lateral, all the elements of the array $Level\_array1$ are initialized to a value equal to XRoot plus a predefined sibling distance (SIB_DIST). Similarly all elements of $Level\_array2$ are assigned to a value XRoot-SIB_DIST. As and when a lateral is identified, all its nodes are assigned the same 'x' co-ordinate. This 'x' co-ordinate value is chosen as follows:
Let $L = \{N_1, N_2, \ldots, N_n\}$ be the nodes belonging to a lateral that has been identified currently by the DFS traversal. Let $NL = \{NL_1, NL_2, \ldots, NL_n\}$ be their respective node levels.Since, the arrays $Level\_array1$ and $Level\_array2$ are indexed by the node levels, the 'x' co-ordinate of all the nodes belonging to L are assigned a value equal to max(Level_array1(NL)),if its orientation is to the right,or to a value min(Level_array2(NL)), if the lateral has been previously assigned a leftward direction. All the corresponding

elements of the arrays $Level\_array1$ or $Level\_array2$ are updated to the next available 'x' co-ordinate values. The identification of laterals, assignment of directions to each lateral and computation of co-ordinates are all carried out during the same DFS traversal.

5. MAJOR STEPS IN THE PROPOSED PROCEDURE

The major steps in the proposed procedure are as follows.

1. From the given data determine the degree of all nodes and identify the leaf nodes (nodes of degree 1) and also identify the parent and child nodes for each node
2. Determine the weight for each node.This is done by traversing the graph from each of the leaf nodes to the root node and updating the weights of visited nodes (which are all initialized to zero to start with).The scheme of updating is chosen such that at the end of the traversals the weights assigned to each node corresponds to the largest possible path length on the downstream side of the node.
3. Sort the child nodes of each node in the decreasing order of their weights
4. Determine the node levels of all the busses and hence the Maximum Node Level (MNL)
5. Identify the leaf node with the maximum level. Backtrack from this leaf node to the root node and pick up all the nodes visited and form the main lateral or feeder.
6. For all the nodes N_i belonging to the main feeder, assign co-ordinates as follows:
$X(N_i)$=XRoot,$Y(N_i)$=YRoot+HEIGHT*level(N_i)
7. Initialize the arrays $Level\_array1$ *and* $Level\_array2$
8. For each base node B_i on the main feeder,
(a) Assign directions to all sub-trees (except the sub-tree which contains a part of the main lateral) rooted at B_i and performing a DFS traversal over each sub-tree, identify the laterals that are contained in the respective sub-trees. As and when a lateral is formed determine co-ordinates of all its nodes as described previously and update the elements of the array $Level\_array1$ or $Level\_array2$ to the next available 'x' co-ordinate value
9. Draw the graph by starting from the root node. Represent all nodes by thick horizontal lines hence completing the visibility representation

6. ILLUSTRATIVE EXAMPLE

The application of the algorithm is first illustrated with a 37 node example whose data is given in Table 1. The value assigned to some of the constant parameters are:(XRoot,YRoot)=(0,0),HEIGHT=1,SIB_DIST=2. The set of leaf nodes are first identified. For each node, its parent node,Level, Weight, the number of offspring nodes (nchld) and set of child nodes (CN) are determined. For the sake of illustration these values (for the first fifteen nodes) are given in Table 2. In the next step, the main feeder is identified by backtracking from leaf node 11 (the leaf node with the Maximum Node Level(MNL=11)) till the root node is reached and picking up all the visited nodes. The co-ordinates of the nodes belonging to the main lateral assigned.

Table 1. Input data - Edge list

0 - 1	1 - 2	2 - 3	3 - 4	4 - 5	5 - 6
6 - 7	7-8	8 - 9	9 - 10	10 - 11	1 - 12
12 - 13	13 - 14	14 - 15	4 - 16	16 - 17	17 - 18
17 - 19	2 - 20	20 - 21	21 - 22	6 - 23	23 - 24
24 - 25	13 - 26	26 - 27	14 - 28	28 - 29	16 - 30
30 - 31	20 - 32	20 - 33	18 - 34	34 - 35	23 - 36
24 - 37	-	-	-	-	-
Bus No 0:Sub- station bus(Root)					

Table 2. Parent,child and weight information

Node	Parent	Level	Weight	nchld	(CN,W(CN))
1	0	1	10	2	(2,9)(12,4)
2	1	2	9	2	(3,8)(20,2)
3	2	3	8	1	(4,7)
4	3	4	7	2	(5,6)(16,4)
5	4	5	6	1	(6,5)
6	5	6	5	2	(7,4)(23,2)
7	6	7	4	1	(8,3)
8	7	8	3	1	(9,2)
9	8	9	2	1	(10,1)
10	9	10	1	1	(11,0)
11	10	11	0	0	-
12	1	2	4	1	(13,3)
13	12	3	3	2	(14,2)(26,1)
14	13	4	2	2	(28,1)(15,0)
15	14	5	0	0	-

The two arrays $Level\_array1$ and $Level\_array2$, each of size MNL , are initialized to the values +2 and -2 respectively. These values indicate the currently available 'x' co-ordinates on either side of the main feeder.

The other laterals are formed by performing a DFS traversal along each sub-tree rooted at a base node. It is seen that nodes {6,4,2,1} are the base nodes ($nchld$ > 2) on the main lateral. Starting from node 6, which is the base node nearest to the leaf node of the main lateral, it is seen that there is one sub-tree that has to be traversed. This sub-tree is assigned a direction to the right of the main lateral. Starting a DFS traversal at node 6, the first leaf node visited is node 37. Therefore, the first lateral to be identified consists of the set of nodes: {6,23,24,37}. The 'x' co-ordinate of the base node 6 has been already determined and the co-ordinates of the remaining nodes of the lateral are computed as follows:

The levels of nodes 6,23,24 and 37 are respectively 6,7,8 and 9. Therefore the 'x' co-ordinates of these nodes are found as : X(23)=X(24)=X(37)=max($Level\_array1(\{6,7,8,9\})$)=2. The elements of the array $Level\_array1$ are updated to the next available position as shown : $Level\_array1(6) = Level\_array1(7) = Level\_array1(8) = Level\_array1(9) = 2+2 = 4$. The next leaf node picked up in the DFS traversal is node 25 and hence the next lateral consists of nodes {24,25}. The level of nodes 24 and 25 are 8 and 9 respectively. Therefore: X(25)= $max(Level\_array1\{8,9\}) = 4$ and the array $Level\_array1$ is updated as before. Laterals belonging to all sub-trees are similarly laid out and the final diagram is shown in Fig 3.

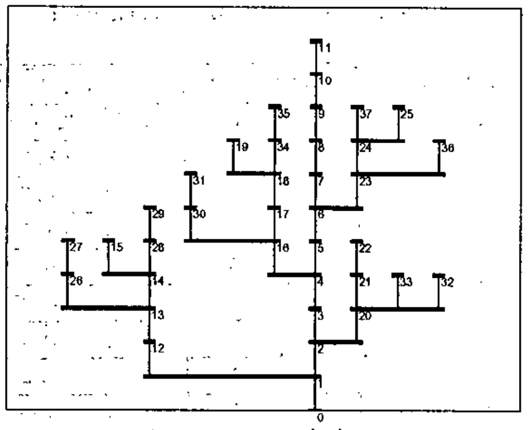

Figure 3. Visibility diagram of the 37 node system

The algorithm has been used to generate, consistently very good diagrams, of a large number of practical distribution networks. Here, we give some examples as thumb-nail diagrams in Fig 4. The first three examples correspond to three 11 KV feeders -Bukkasagara, Saradagi and Devagiri, in rural South India with 138,160 and 172 nodes respectively. The fourth example is that of a 150 node system whose connectivity information has been generated with an aim, to create a complex test case for the algorithm.

7. Conclusions

Automatic generation of visibility diagrams of radial distribution systems has been presented. A new set of aesthetic criteria suitable for easy visualization of a radial distribution system are proposed. The algorithm is illustrated with a detailed example. One-line diagrams of some practical feeders as generated by the algorithm have been presented. The proposed method can be a valuable tool for providing an effective GUI for system engineers involved in planning and operation of distribution networks.

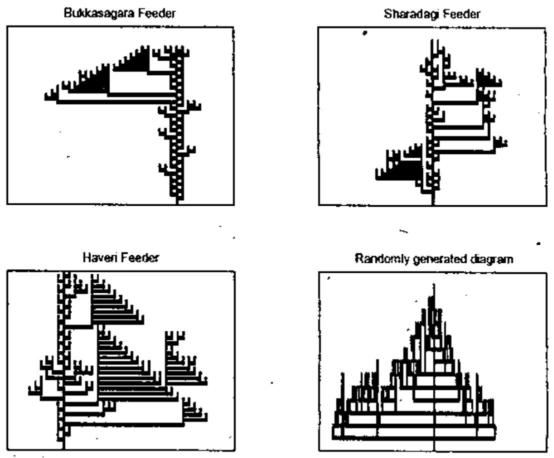

Figure 4. Sample Diagrams Generated by the Algorithm

References

[1] Y.S Ong, H.B Gooi and C.K Chan, "Algorithms for automatic generation of one-line diagrams",*IEE Proc-Gener,Transm,Distrib.*,Vol 147, No. 5, Sept 2000,pp 292-298

[2] Fruchterman T, Thomas M.J and Reingold E, "Graph Drawing by Force directed placement" ,*Software-Practice and Experience*, Vol 21, 1991,pp 1129-1164.

[3] John Q. Walker II, "A Node-positioning Algorithm for General Trees",*Software-Practice and Experience*, Vol 20(7), July 1990,pp 685-705

[4] P.S Nagendra Rao and Ravishankar Deekshit,"Automatic generation of distribution system one-line diagrams",*Proceedings of National Power Systems Conference ,IIT Kharagpur, India*,Vol 1,pp 166-170.

[5] Mark Petrick, "Applications of the canonical ordering",*CS762* , Lecture 24,March 13,2002,pp 1-6.

[6] Narsingh Deo, "Graph theory with applications to engineering and computer science", *Prentice Hall*,June 1974.