International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Restoration of Power Supply in a Multiple Feeder Distribution Network using Dijkstra's Algorithm

Shridevi Narasannavar[A*], Rashmi Kannur[A] and S B Karajgi[A]

[A]PG Student, Department of Electrical and Electronics Engineering, SDM College of Engineering and Technology, Dharwad, Karnataka, India

Accepted 27 May 2014, Available online 01 June2014, **Vol.4, No.3 (June 2014)**

Abstract

Faults in a distribution network cause partial or complete blackout of feeder. It is necessary to supply the power to the remaining healthy part of a feeder after isolation of faulty section. This can be achieved by proper opening and closing of tie-line switches (normally open switches) and sectional switches (normally closed switches). With that the topology of a network will change and the losses will be redistributed. In this paper Dijkstra's algorithm is used as a high speed switching algorithm, which gives information about the status of the switches to restore the supply to all the load points in the event of outage of any line or a transformer/feeder. It also gives the minimum impedance path from source to all the load points. In this paper IEEE standard 16 node distribution test system is considered for a case study. Load flow analysis is carried out using a MATLAB tool.

Keywords: *Reconfiguration, Distribution Network, Dijkstra's Algorithm, Restoration of Power Supply*

1. Introduction

It is known that distribution networks are built as interconnected meshed networks, while in operation they are arranged in the form of a radial line structures. This means that distribution systems are divided into subsystems of radial feeders, which contain a number of normally-closed (sectional) switches and a number of normally open (tie-line) switches. From graph theory, a distribution network can be represented with a graph of G (n, l) that contains a set of nodes 'n' and a set of branches 'l'. Every node represents either a source node (supply transformer) or a sink node (customer load point), while a branch represents a feeder section that can either be connected (switch closed) or open (switch open). The network is radial, so that feeder sections form a set of trees where each sink node is supplied from exactly one source node. Therefore, the distribution network reconfiguration problem is to find a radial operating structure that minimizes the system power loss. In fact, this problem can be viewed as a problem of determining an optimal tree of the given graph. However, real distribution systems contain many nodes and branches therefore total number of trees is extremely large. Dijkstra's algorithm finds the Minimum Spanning Tree (MST) in a graph from the source node to all the load points.

Whenever the power supply interruption occurs in distribution system due to a fault, it is imperative to bring back the system to an optimal target network. The problem of obtaining target network by changing the topology of the network is called as power supply restoration. The objectives are to restore the supply to maximum number of customers. The constraints are to maintain the radial nature of the distribution network and number of switching operation should be within the limit specified by the operators. Dijkstra's algorithm isolates only faulty section and restores the power supply to the maximum number of customers and reduces the distribution power loss while maintaining the radial nature of distribution network.

A.Merlin, H.Back, proposed a branch and bound method for the problem of finding the network configuration with minimum line losses. Reconfiguration problem is considered as a mixed-integer, non-linear optimization problem (A.Merlin *et al*, 1975). It begins with all branches closed (a complete graph) and performs a procedure of opening branches which carry the least current. This method is a greedy algorithm that does not necessarily guarantee feasibility of the final solution.

Nahman et al. presented another heuristic approach (J. Nahman *et al*, 1994; V. Glamocanin, 1990). The algorithm starts from a completely empty network, with all switches open and all loads disconnected. Load points are connected one by one by switching branches onto the current sub-tree. The search technique also does not necessarily guarantee global optima.

Two elements comprise knowledge based (KB) expert system (ES): knowledge base and inference engine. Each component of the knowledge base describes a piece of knowledge serving to determine the restoration procedure. In other words, the knowledge base is an alternative expression of a computer program. An inference engine

*Corresponding author **Shridevi Narasannava** and **Rashmi Kannur** are PG students; **S B Karajgi** is working as Professor

interprets the knowledge base, reasoning values of the specified parameters. However, this method has no guarantee to find an optimal target configuration; moreover the maintenance of a large-scale ES has turned out to require heavy cost, thus greatly offsetting the advantage of introducing expert systems (F.F.WU *et al*, 1988)

Optimization algorithms include genetic algorithm (GA), ant colony search algorithm (ACSA), artificial bee colony algorithm (ABC), simulated annealing (SA), Tabu search, particle swarm optimization (PSO), heuristic algorithms, and so forth (H. D. De Macedo Braz *et al*, 2011; A. Saffar *et al*, 2011; J. Olamaei *et al*, 2011; V. Parada *et al*, 2004; A. Y. Abdelaziz *et al*, 2010; H.T.Yang *et al*, 2010) can be used for restoration problem. These optimization techniques have a major drawback i.e., the time consumption increases exponentially with size of the de-energized areas. So far large area it may not converge within the prescribed time (Sudhakar *et al*, 2004)

To find best path for the power to flow, graph theory based minimum spanning tree (MST) approach can be used. There are number of MST algorithms which could be used to solve the restoration problems (T.D Sudhakar *et* al, 2011; S.Mohanram *et al*, 2011). One such a methodology is a Dijkstra's algorithm. Graph Theory can be applied to distribution networks considering buses or transformers as nodes, distribution line as a path between the nodes, and impedance of line as weight on path. While applying Graph Theory Technique the search time increases linearly with respect to the de-energized areas.

2. Dijkstra's Algorithm

It is a minimum spanning tree algorithm for finding the shortest paths from a source node to all other nodes in a network. Dijkstra's algorithm (Narsingh Deo, (2005) finds the shortest path from source node to destination node in order of increasing weight on path. The algorithm is iterative. At the first iteration the algorithm finds the closest node from the source node, which must be the neighbor of source node. At the second iteration the algorithm finds the second closest node from the source node. This node must be the neighbor of the either the source node or the closest node to the source node. At the third iteration the third closest node must be the neighbor of the source node or the first two closest nodes, and so on. Thus at the K^{th} iteration, the algorithm finds the minimum spanning tree for the source node, which includes all the nodes thus giving an optimal path.

The flow chart for finding the shortest path from a specified vertex s to another specified vertex t, is given in fig 1. A simple weighted diagraph G of n vertices is described by n by n matrix D = [d_{ij}], where,

d_{ij}= length of the directed edge from vertex i to vertex j,

$d_{ij} \geq 0$,

d_{ij}=0,

$d_{ij}=\infty$, if there is no edge from i to j

Dijkstra's algorithm labels the vertices of the given digraph. At each stage in the algorithm some vertices have permanent labels and other temporary labels. The algorithm begins by assigning a permanent label 0 to the starting vertex s, and a temporary label ∞ to the remaining n-1 vertices. In the next step algorithm finds the closest vertex from vertex s and assigns the permanent label to this vertex. Remaining n-2 vertices have the temporary label ∞. Likewise in each iteration one vertex gets permanent label, according to the following rules:

1. Every vertex j that is not yet permanently labeled gets a new temporary label whose value is given by

Min [old label of j, (old label of i + d_{ij})],

Where, i is the latest vertex permanently labeled, in the previous iteration, and d_{ij} is the direct distance between vertices i and j. If i and j are not joined by an edge, then $d_{ij} = \infty$.

2. The smallest value among all the temporary labels is found, and this becomes the permanent label of the corresponding vertex. In case of a tie, select any one of the candidates for the permanent labeling.

Steps 1 and 2 are repeated alternatively until the destination vertex t gets a permanent label.

At the end all the vertices will be selected forming a minimum spanning tree from the source vertex.

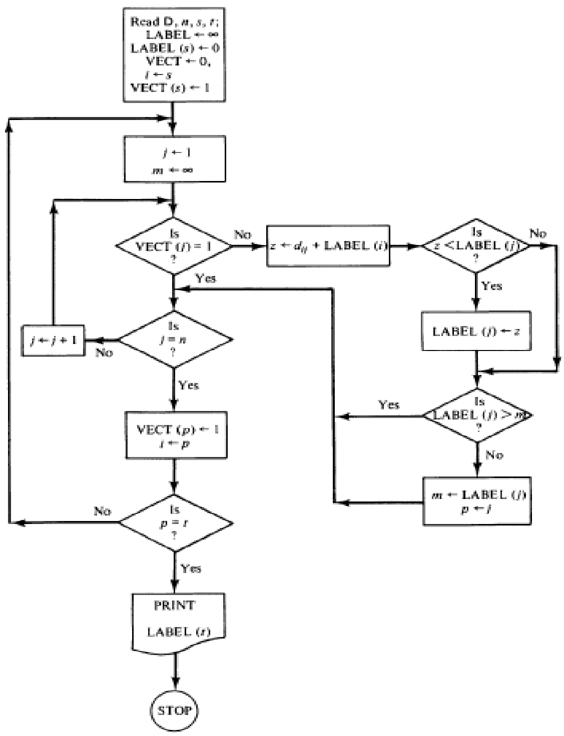

Fig.1 Flowchart for Dijkstra's Algorithm

3. Implementation of Dijkstra's Algorithm

The objective of this work is to find optimal reconfiguration and restoration of power supply to affected loads by changing the configuration of distribution network through appropriate switching of tie-lines and sectional lines switches.

In applying graph theory to the distribution networks buses are considered as node, feeders/ distribution transformers are considered as source node, distribution lines are considered as a paths, and impedance of the line is considered as weight on path.

Objective function

- Restore supply to maximum number of customers

Subjected to the constraints

- Retaining the radial nature of the network
- Minimum distribution line losses

3.1. Description of Test System

IEEE standard 16 node test system is shown in figure1, line and load data are given in Table1. System consists of three feeders with the base power of 100MVA and Base voltage of 23kV. The total System load is 28.7 MW, 17.3 MVAr. Total injected reactive power at different load is 11.4 MVAr.

3.2 Simulation

The test system is simulated on MATLAB frame and tested for Dijkstra's algorithm. Following assumptions are made in the test system

- The system is represented on a per phase basis
- The load along a feeder section is represented as constant P, Q loads placed at the end of the lines.
- It is assumed that each line in a system is associated with the switch.

The Dijkstra's algorithm is applicable only to networks consisting of single source. For the network consisting of multiple sources (transformers/ feeders) following procedure can be adopted.

- Solve the Dijkstra's algorithm considering source 1 (feeder1) alone as active i.e. feeder 1 is supplying power to the loads which comes under feeder 2 and 3 including its own load.
- Note down the impedance value from feeder 1 to all the load points
- Repeat the above steps considering each source active at a time
- Compare the impedances
- Select source for each load point which gives the least impedance path

The above procedure is time consuming if the number of sources (feeders /transformers) considered increases. To account for this problem virtual source node (which acts like a source to all the transformers/ feeders) is created. Then check for Dijkstra's algorithm to find the shortest impedance path from this virtual source node to all the load points. This will also suggest from which source (transformer/ feeder) the particular load point has a shortest impedance path. It will also suggest status of all switches (open/close) in the network. Accordingly the switches are made open/closed. Then load flow analysis is carried out to get the total losses in MATLAB.

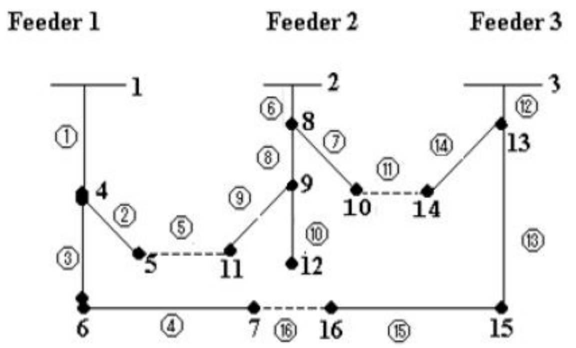

Fig.2 IEEE standard 16 nodes Test System

Table 1 Line and load Data for IEEE standard 16 nodes Test System

Line	Resistance R in Ω	Reactance X in Ω	Receiving node		
			P in MW	Q MVAr	Injected MVAr
1-4	0.075	0.1	2.0	1.6	0
4-5	0.08	0.11	3.0	1.5	0
4-6	0.09	0.18	2.0	0.8	0
6 -7	0.04	0.04	1.5	1.2	0
2-8	0.11	0.11	4.0	2.7	1.1
8-9	0.08	0.11	5.0	3.0	1.2
8-10	0.11	0.11	1.0	0.9	0
9-11	0.11	0.11	0.6	0.1	0
9-12	0.08	0.11	4.5	2.0	1.2
3-13	0.11	0.11	1.0	0.9	0
13-14	0.09	0.12	1.0	0.7	0.6
13-15	0.08	0.11	1.0	0.9	3.7
15-16	0.04	0.04	2.1	1.0	0
5-11	0.04	0.04	--	--	1.8
10-14	0.04	0.04	--	--	0
7-16	0.09	0.12	--	--	1.8

Initially three switches i.e. 5, 11 and 16 were open. On running the load flow on the test system, the initial system power loss is found to be 511.44 kW and 590.37 kVAr. The system is first reconfigured, using Dijkstra's Algorithm to get optimal power flow path.

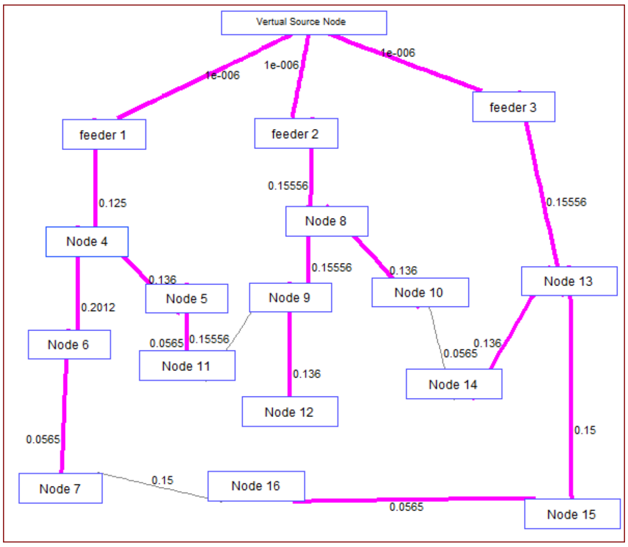

Fig.3 Network topology after the reconfiguration

The resultant reconfigured network of test system is shown in fig 2. The load flow is once again carried out on the reconfigured system and the results are compared in table 2. It is observed that the losses have reduced to 493.15 kW and 571.51 kVAr.

Table 2 Results for initial and reconfigured network for standard IEEE 16 nodes test system.

Status	Switches opened	Real power loss	Reactive power loss
Before reconfiguration	5-11 10-14 7-16	511.44 kW	590.37 kVAr
After reconfiguration	9-11 10-14 7-16	493.15 kW	571.51 kVAr

Table 3 Switching status and load flow analysis result for each line outage

Line under outage	Switches to be open			Real power loss in kW	Reactive power loss in kVAr
1-4	1-4	4-6	10-14	960.37	1065.51
4-5	4-5	10-14	7-16	676.63	764.57
4-6	4-6	9-11	10-14	547.68	615.04
6-7	6-7	9-11	10-14	500.14	568.05
2-8	2-8	8-9	7-16	849.39	1036.67
8-9	8-9	10-14	7-16	707.75	882.209
8-10	8-10	9-11	7-16	466.13	544.90
9-11	9-11	10-14	7-16	493.15	571.51
9-12	9-11	10-14	7-16	Supply restoration to node 12 is not possible	
3-13	3-13	13-15	9-11	637.65	763.68
13-14	13-14	9-11	7-16	511.94	589.62
13-15	13-15	9-11	10-14	559.34	685.31
15-16	15-16	9-11	10-14	523.14	628.30

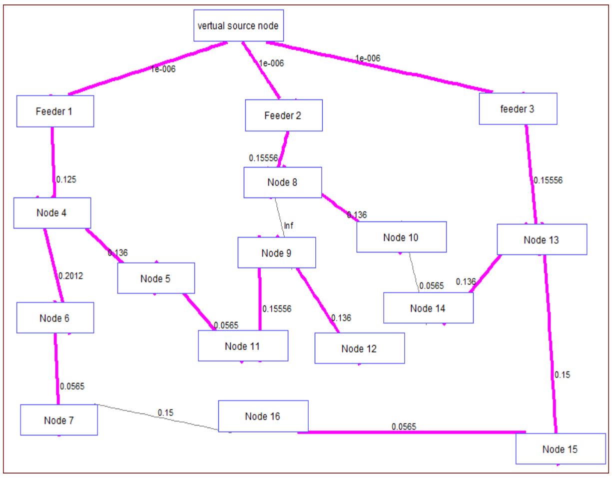

Fig.4 Network Topology when the fault is in section '8'

To simulate the fault at section '8', switch associated with line '8' is made open to isolate the fault. Dijkstra's algorithm is run by making this line impedance as infinite. Infinite impedance indicate that line is open. From the simulated results it is observed that sections '11', '16' are to be opened to get optimal power flow path. This is shown in figure 3. Similarly Figure 4 indicates that sections '6', '8', '16' are to be opened for the fault on section '6'. Similar results were obtained for the fault simulated on each branch separately. The load flow analysis is carried out for each of these cases, and the results are tabulated in the table 3.

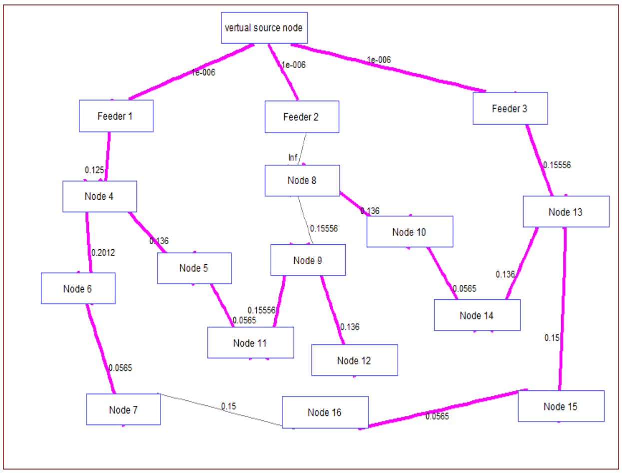

Fig.5 Network Topology when the fault is in section '6'

4. Conclusion

In this paper Dijkstra's algorithm for multiple feeder distribution networks (IEEE 16 node test system) is carried out by creating a virtual source node. It is observed that the algorithm finds out shortest impedance path from this virtual source to all the load points. This will also suggest from which source (feeder/ transformer) the particular load point has the shortest impedance path. This reduces the computation burden on the operator by providing single optimal solution instead of all the possible switching options. It is also observed that final network topology is independent of initial status of the network switches. This algorithm can be implemented for any complex distribution network

References

A. Merlin, H. Back, (1975), Search for a minimal loss operating spanning tree configuration for an urban power distribution system. *Proceedings of the 5th Power System Computation Conference, (PSOC), Cambridge*. pp1–18

J. Nahman, G. Strbac, (1994), A new algorithm for service restoration in large-scale urban distribution systems, *Elect. Power Syst. Res. 29*, pp181_ 192.

V. Glamocanin, (1990), Optimal loss reduction of distributed networks, *IEEE Trans. Power Syst*. 5 (3) pp774_ 782.

F.F. WU, and A. Monticelli, (1988), Analytical tools for power system restoration - *Conceptual design, IEEE Trans. On Power Systems*, Vol. 3, No. 1, pp.10-16.

H. D. De Macedo Braz and B. A. de Souza, (2011), Distribution network reconfiguration using genetic algorithms with sequential encoding: subtractive and additive approaches, *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp.582–593.

A. Saffar, R. Hooshmand, and A. Khodabakhshian, (2011), A new fuzzy optimal reconfiguration of distribution systems for loss reduction and load balancing using ant colony search-based algorithm, *Applied Soft Computing Journal*, vol. 11,no. 5, pp. 4021–4028.

J. Olamaei, T. Niknam, and S. B. Arefi, (2011), Distribution feeder reconfiguration for loss minimization based on modified honey bee mating optimization algorithm, *Proceedings of the 2nd International Conference on Advances in Energy Engineering*, pp. 304–311.

V. Parada, J. A. Ferland, M. Arias, and K. Daniels, (2004), Optimization of electrical distribution feeders using simulated annealing, *IEEE Transactions on Power Delivery*, vol. 19, no. 3, pp. 1135–1141.

A. Y. Abdelaziz, F. M. Mohamed, S. F. Mekhamer, and M. A. L. Badr, (2010) Distribution system reconfiguration using a modified Tabu Search algorithm, *Electric Power Systems Research*, vol. 80, no. 8, pp. 943–953.

H.-T.Yang, Y.-T.Tzeng, and M.-S. Tsai, (2010), Loss-minimized distribution system reconfiguration by using improved multi-agent based particle swarm optimization, *Proceedings of the Asia-Pacific Power and Energy Engineering Conference (APPEEC '10)*.

\

Sudhakar, Vadivoo, Slochanal, and Ravichandran, (2004), Supply Restoration in Distribution Network Using Dijkstra's Algorithm, *International Conference on Power Systems Technology, POWERCON 2004, Singapore*, pp. 640-645

T.D. Sudhakar, and K.N. Srinivas, (2011), Power system reconfiguration based on Prim's algorithm, *1st International Conference on Electrical Energy Systems*, pp. 12-20.

S. Mohanram, T.D. Sudhakar, (2011), Power system restoration using reverse delete algorithm implemented in FPGA, *International conference on Sustainable Energy and Intelligent systems (SEISCON)*, pp.373-378.

Narsingh Deo,(2005), *Graph theory with application to engineering and computer science Prentice Hall of India*, pp. 292- 96.