GRAPH THEORY-BASED OPTIMISATION AND CONTROL THEORY FOR SMART GRIDS

Shobha Mahadevappa
Assistant Professor Department of Mathematics
Government First Grade College,Shorapur-585224 (K.S)

ABSTRACT : The technology of the twenty-first century is utilised in a smart grid to distribute power in a dependable and effective manner. They serve as the power grids for the two-way exchange of information and electricity. This study focusses on determining the lightest tie-set path between two nodes in a network and optimising smart grid technology utilising tie-sets. Tie-set management devices can be used to address local issues with energy distribution, facilitating simple communication and effectively lowering power outages [7]. The shortest path connecting the source and destination nodes is known as the lightest tie set path.

Keywords: Smart Grid, Tie-set Graph Theory, Dijkstra's algorithm.

I. INTRODUCTION:

A central producing station and millions of transmission lines are the two main components of the traditional power grids that are used to distribute electricity to houses. To satisfy today's electrical demands, a smart grid gives this electricity system a digital makeover. Smart meters and other specialised equipment are used to automate the management and control of power distribution [8]. Distributed control of energy should take the place of centralised control as there is decentralised power generation and a demander can also be a supply. Additionally, this will aid with scalability.
Divide the network into ring structures such as tie-sets or loops to provide distributed control [7]. The notion of tie-set graphs can be applied to local management. Another way to describe a basic tie-set is as "fundamental circuit' in a graph that is created by joining a spanning tree (communication paths) and a chord (non-communication path) in relation to a graph (network) that forms a loop.

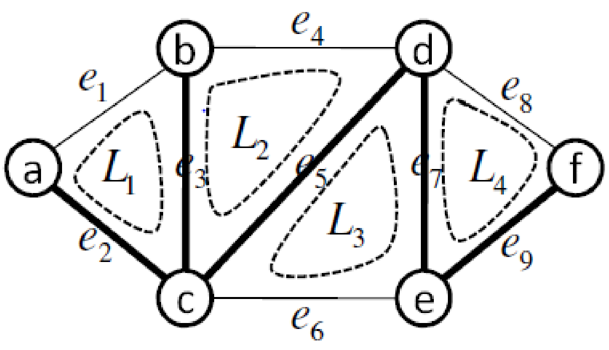


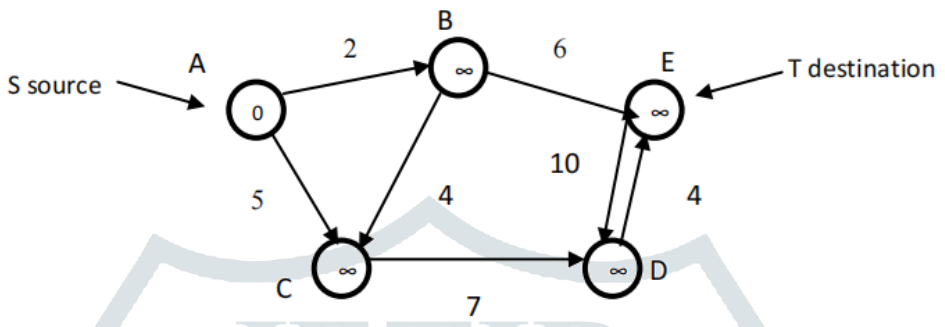
Figure-1: Example of a fundamental system of tie-sets

In every tie-set, a leader node is chosen to decide on the best course of action and resolve distributed control issues. It also resolves resource allocation quandaries and determines how to employ the distributed resources allotted to each node. In order to solve distribution challenges, leader nodes must communicate with the tie-set. If the leader nodes are in the same tie-set, then communication can be facilitated by passing tie-set communication messages; if not, then the quickest method of communication is to employ a routing algorithm between the leader nodes. [7]

II. METHOD OF MATHEMATICAL FORMULATION :

Using a routing table, Dijkstra's algorithm determines the shortest path between the two leader nodes in the tie-set. We build an undirected graph G= (V, E) that is bi-connected, with vertices V= {v_1, v_2, v_3..v_n} and edges E= {e_1, e_2,...e_m}. Let Li= {e_1^i, e_2^i, e_3^i,..} represent the set of edges that make up a loop in G. The tie-set is the collection of edges Li. We choose a leader node in each tie set that makes critical choices, figures out the best course of action for dispersed problems, and solves them. There is often a need for tie-set communication while handling distributed challenges [7].

To facilitate efficient communication between the tie-set leaders, we designate the "lightest tie." The "lightest tie set path" is defined by us. Out of all the routes that might connect two leaders, this one has the least amount of weight. We utilise Dijkstra's technique to determine the minimal distance between the two leaders. Dijkstra's algorithm is employed.



In the tie-set graph, each leader node has a table connected to it. All of the other leader nodes in graph G are included in the table. The "next" node that an entry must travel to in order to reach the destination node is the first of two fields in each item in the table. The weight or the separation between the 'next' node and the s would be the second entry [5]. And the bare minimum would be this. The shortest path between source node "S" and destination "T" can be found by rotating the table among nearby leader nodes and using Dijkstra's algorithm on it.

III. ALGORITHM:

Every leader node in this approach possesses a copy of the entire topology, or all of the network's nodes. A table with the fields node Y, weight (the distance from node X to Y), and next node (the node that connects X to Y) is present in each node X. The nodes are split into two sets by the algorithm: tentative andpermanent. Let's be the starting point for the Dijkstra's algorithm, located at [3].

1. Assign a rough list value for each column to each node in the source node table. For the source of the current node, the weight must be '0,' and for all other (unvisited) nodes, it must be infinity.

2. At first, the permanent list is empty and the source node is in the tentative list.

3. Move the source node to the permanent list (as visited node) and add all the neighbours along with their distance to the tentative list.

4. Among all the nodes in the tentative list, choose the node with the shortest cumulative distance. Move it to the permanent list.

5. Add each unprocessed neighbour of last moved node (to permanent list) to tentative list if not already there.

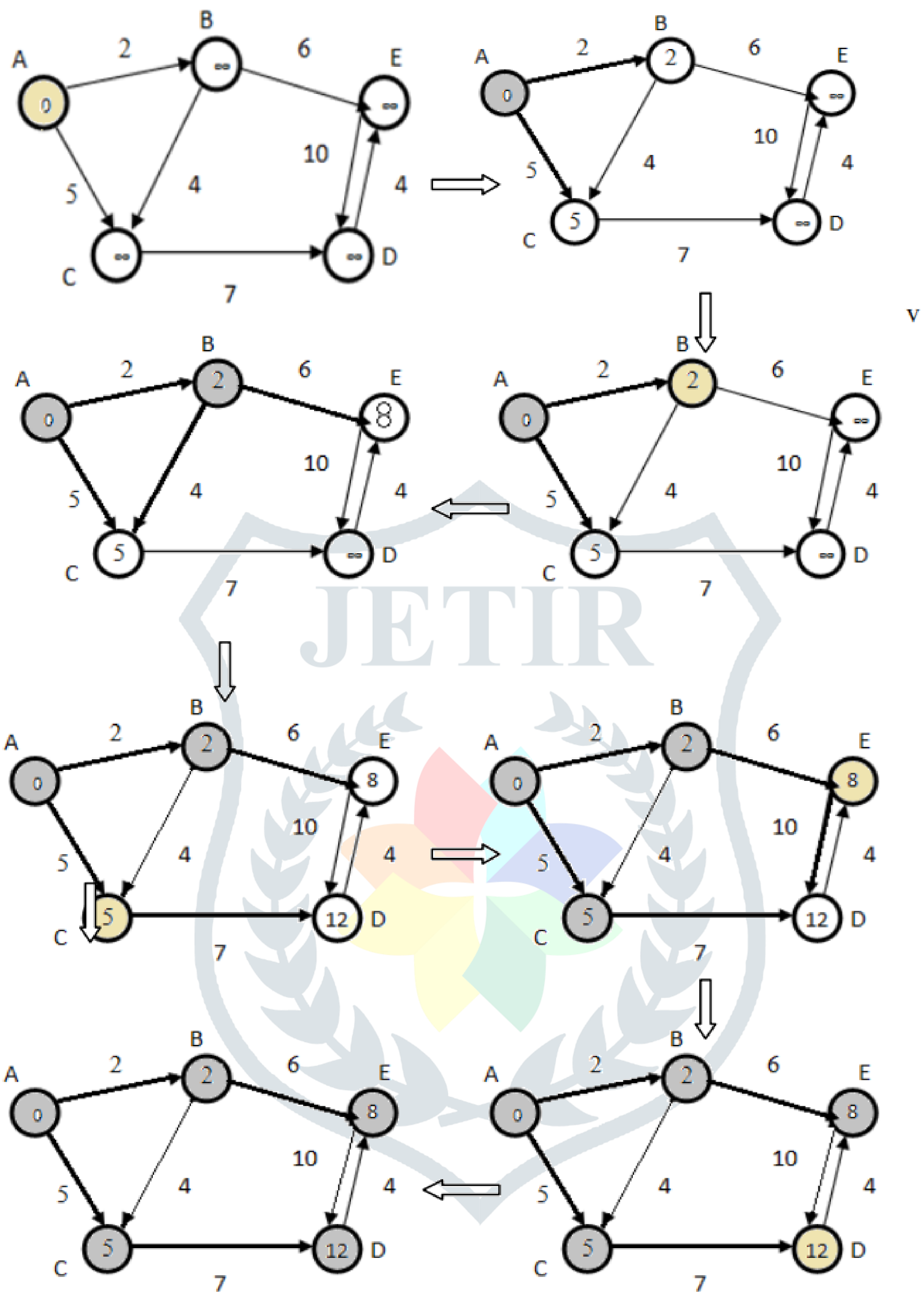
6. If neighbour is already present in the tentative list with larger cumulative cost, replace it with new one For example, If the current node X is marked with a tentative distance of 9, and the edge connecting it with a neighbour Y has length 6, then the distance to Y (through X) will be 9+6=15. If this distance is less than the previously recorded tentative distance of Y, then overwrite that distance.

6. Repeat steps 4, 5 and 6 until the tentative list is empty.

7. Once the node has been moved to the permanent list, its distance is minimal and final.

- ◆ Unvisited node
- ◆ Current node
- ◆ Visited node

Every node X has a table associated with it which contains the following fields- node, weight, next. Node field gives the name of other nodes in the network, weight gives the distance of those nodes from the given node X and next node gives the name of the node through which X is connected to the other node [5]. Initial and final tables for the source node A are given for the following example. There are five nodes in the network with the distance between every two nodes mentioned on the edge joining the two nodes. We will find the shortest distance between the source node A and every other node using Dijkstra's algorithm [3]



Initial table for node A:

NODE	WEIGHT	NEXT
A	0	-
B	∞	-
C	∞	-
D	∞	-
E	∞	-

After applying Dijkstra's algorithm, to the network, we get the final table for A:

NODE	WEIGHT	NEXT
A	2	-
B	3	-
C	6	-
D	11	C
E	8	B

This way the source node has found the shortest distance to all the nodes in the network and communicates through these shortest routes. Therefore, our solution provides an efficient way of communication between points in a smart grid network.

RESULTS AND DISCUSSION :

The shortest path between the tie-sets can be found by applying Dijkstra's algorithm. The method suggested by Dijkstra's algorithm determines the shortest path between two leader nodes in the smart grid to provide the best possible distribution of electricity across the system. The two nodes' communication efficiency is increased when they use the shortest distance algorithm. Additionally, network management employing loops as the fundamental management unit is made possible by the theory of tie-sets.

CONCLUSION :

The next energy future for more intelligent supply of electricity is the smart grid system. Distribution and transmission are both modernised by using tie-set on smart grid. Global optimisation has been achieved by the application of tie-set graph theory to local electricity distribution optimisation. This suggests that a network can benefit from distributed control utilising tie-sets. Unlike two-node disjoint pathways, it makes the network fault resilient because connections can still be maintained even in the event of a failure in one of the links. Finding the shortest path between leader nodes of tie-sets using Dijkstra's algorithm facilitates simple communication and lowers electricity loss as the distance of electricity transmission decreases. As a result, Dijkstra's method can be applied to tie-sets to facilitate efficient communication and quick resource transfer across nodes.

REFERENCES :

1. Touzene, V. Ustimenko, Graph Based Private Key Crypto System, International Journal on Computer Research, Nova Science Publisher, volume 13 (2006), issue 4, 12p.
2. BehrouzA.Forouzan, Sophia Chung Fegan. Data Communications and Networking (4th edition). New Delhi: Tata McGraw Hill.
3. Dijkstra's algorithm, from http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm.
4. Ellis Horowitz, SartajSahni, Dinesh Mehta. Fundamentals of Data Structures in C++ (2nd edition). University Press.
5. J. A. Thas, Generalised polygons, in F. Buekenhout (ed), Handbook in Incidence Geometry, Ch. 9, North Holland, Amsterdam, 1995.
6. J. Tits, Sur la trialite et certains groupes qui s'en deduisent, Publ. Math. I.H.E.S, 2 (1959), 15-20. [29]
7. Kiyoshi Nakayama, Norihiko Shinomiya. Distributed Control Based on Tie-Set Graph Theory for Smart Grid Networks. Paper presented at 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT).
8. Smart Grid, from http://energy.gov/oe/technology-development/smart-g.
9. Thomas H.Cormen, Charles E. Leiserson, Ronlad L. Rivest, Clifford Stein. Introduction to Algorithms. London: The MIT Press.
10. V. A. Ustimenko, Coordinatisation of regular tree and its quotients, in "Voronoi's impact on modern science, eds P. Engel and H. Syta, book 2, National Acad. of Sci, Institute of Matematics, 1998, 228p.