International Journal of Mathematical Archive-6(6), 2015, 172-175
Available online through www.ijma.info ISSN 2229 – 5046

OPTIMIZATION AND CONTROL THEORY FOR SMART GRIDS USING GRAPH THEORY

PALLAVI DANI*[1], NIKILA PRABHAKARAN[2]

[1,2]School of Computer Science and Engineering, VIT University, India.

(Received On: 08-05-12; Revised & Accepted On: 29-06-15)

ABSTRACT

A smart grid is 21st century technology that is used for reliable and efficient delivery of electricity. They are the power grids for bidirectional flow of electricity and information. This paper focuses on the optimization of smart grid technology using tie-sets and finding the lightest tie-set path between two nodes in a network. Using tie-set as management units can solve electricity distribution problems locally thus helping in easy communication and effective reduction of loss of electricity [3]. The lightest tie set path is the shortest path between the source and the destination nodes.

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

1. INTRODUCTION

The traditional power grids used for distribution of electricity consists of a central generating station and millions of transmission lines for delivering the electricity to our homes. A smart grid provides a digital makeover to this electricity system to meet today's electricity demands. It automates management and control of electricity distribution using special devices like smart-meter [6]. Since there is decentralized power generation as a demander can be a supplier too, the centralized control should be replaced by distributed control of electricity. This will also help in scalability.

The distributed control can be done by partitioning the network into loops or ring structure like tie-sets [3]. The tie-set graph theory can be used for local management. A fundamental tie-set can also be said as 'fundamental circuit' in a graph which is formed by adding a chord (non-communication path) to a spanning tree (communication paths) with respect to a graph (network) forming a loop.

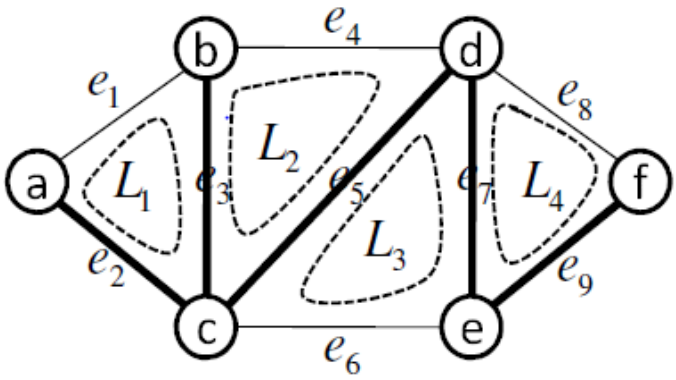


Figure-1: Example of a fundamental system of tie-sets

Each node in a tie-set has information about the tie-set to which it belongs. That information is captured in a table using distributed algorithm. If a problem which cannot be solved in a tie-set emerges, then the tie-set will contact its neighbouring tie-sets to help solve that problem. For example, if a tie-set is in need of electricity, and its neighbouring tie-set has excess of electricity then it can obtain it from the adjacent nodes to resolve the problem.

Corresponding Author: Pallavi Dani*[1]
[1,2]School of Computer Science and Engineering, VIT University, India.

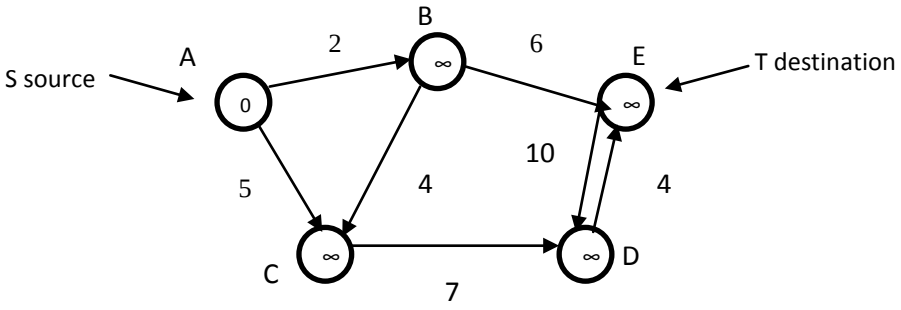
A leader node is elected in each tie-set which determines optimal control for distributed problems and solves them. It also decides how to use the distributed resources allocated to each node and resolves dilemma regarding allocation of resources. When solving distribution problems, communication among tie-set is necessary, this is done by leader nodes. If the leader nodes are there in the same tie-set, then communication can take place easily by passing tie-set communication message otherwise if they are in different tie-sets then a routing algorithm between leader nodes need to be used for fastest communication [3].

2. METHOD

Dijkstra's algorithm finds the shortest distance between the two leader nodes of the tie-set, using a routing table.

We define a bi-connected graph and undirected graph G= (V, E) with a set of vertices V= {v_1, v_2, v_3..v_n} and a set of edges E= {e_1,e_2,...e_m}. Let Li= {e_1^i, e_2^i, e_3^i..} be a set of edges which constitute a loop in G. The set of edges Li is called the tie-set. In each tie set we select a leader node that takes important decisions and determines the optimal control for distributed problems and solves them. When solving for distributed problems, communication among tie-sets are frequently required [3].

For effective communication among the leaders of the tie-set we define the 'lightest tie set path'. This is the minimum weight path among all the possible paths between two leaders. For finding the minimum distance between the two leaders we use Dijkstra's algorithm. We use Dijkstra's algorithm.



Every leader node in the tie-set graph has a table associated with it. The table consists of all the other leader nodes in graph G. Each entry in the table has two fields; first one is the 'next' node to which it has to go to, to reach the destination node. The second entry would be the weight or the distance between s and 'next' node [5]. And this would be the minimum distance. By circulating the table among neighbouring leader nodes and applying Dijkstra's algorithm to it, we get the shortest path between the source node 'S' and destination 'T'.

3. RESULTS AND DISCUSSION

The use of Dijkstra's algorithm provides us with the shortest way of communication between the tie-sets. For optimum distribution of electricity in the smart grid, the solution proposed using Dijkstra's algorithm finds the shortest path between two leader nodes in the smart grid. The shortest distance algorithm between the two nodes increases the efficiency of communication between them. Also, the theory of tie-sets helps to realize network management using loops as basic management unit.

4. ALGORITHM:

In this algorithm, each leader node has a copy of the whole topology, i.e., all the nodes present in the network. Each node X has a table having fields- node Y, weight (distance from node X to Y), next node (node connecting X to Y). The algorithm divides the nodes into 2 sets, tentative and permanent. Let's' be the source node at which we will start the Dijkstra's algorithm [1].

1. Assign to every node in the source node table, a tentative list value for the fields. The weight has to be '0' for the (current node) source and infinity for all other (unvisited) nodes.
2. Initially, the source node is in the tentative list and the permanent list is empty.
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.

7. Repeat steps 4, 5 and 6 until the tentative list is empty.
8. Once the node has been moved to the permanent list, its distance is minimal and final.

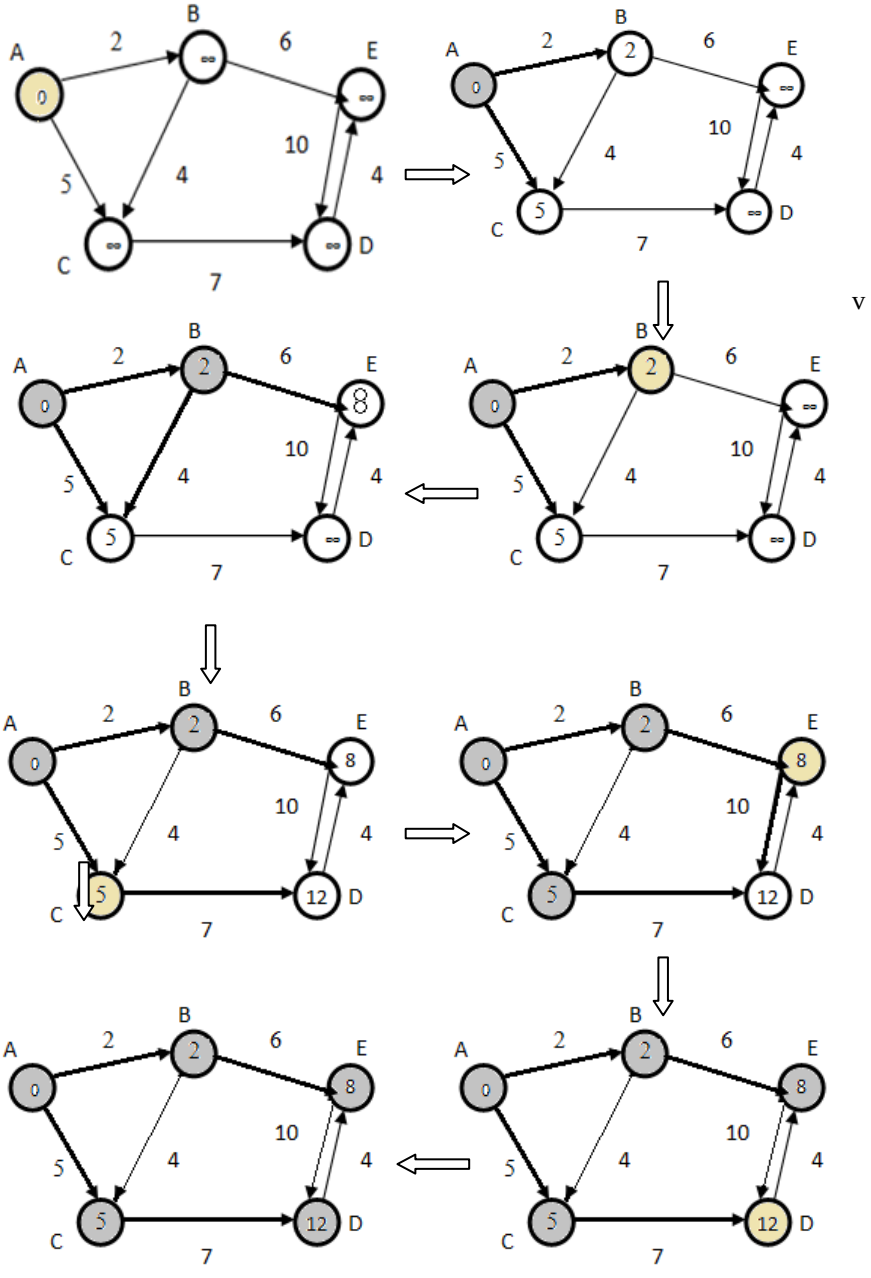
Unvisited node

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 [4].



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	0	-
B	2	-
C	5	-
D	12	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.

5. CONCLUSION

Smart grid system is the new energy future for smarter electricity delivery. Using tie-set on smart grid modernizes both transmission and distribution. The tie-set graph theory used for local optimization of distributing electricity has resulted in global optimization. This indicates that distributed control using tie-sets is beneficial to a network. It makes the network fault tolerant as compared to two-node disjoint paths because even if one link fails, connections can still be maintained. Using Dijkstra's algorithm to find the shortest path between leader nodes of tie-sets helps in easy communication and reduces the loss of electricity as the distance of transfer of electricity is reduced. Thus Dijkstra's algorithm can be used for effective communication between the tie-sets resulting in fast transfer of resources between nodes for network management.

REFERENCES

1. Dijkstra's algorithm, from http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm.
2. Ellis Horowitz, SartajSahni, Dinesh Mehta. Fundamentals of Data Structures in C++ (2nd edition). University Press.
3. 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).
4. Thomas H.Cormen, Charles E. Leiserson, Ronlad L. Rivest, Clifford Stein. Introduction to Algorithms. London: The MIT Press.
5. BehrouzA.Forouzan, Sophia Chung Fegan. Data Communications and Networking (4th edition). New Delhi: Tata McGraw Hill.
6. Smart Grid, from http://energy.gov/oe/technology-development/smart-grid.

Source of support: Nil, Conflict of interest: None Declared