A Comparative Study Of Minimal Spanning Tree Algorithms

Conference Paper · March 2020

DOI: 10.1109/ICMCECS47690.2020.240900

4 authors:


Peace Ayegba
Landmark University


Joyce Ayoola
Landmark University


Emmanuel Oluwatobi Asani
Landmark University

Aderemi Elisha Okeyinka
Ibrahim Badamasi Babangida University, Lapai, Nigeria

A COMPARATIVE STUDY OF MINIMAL SPANNING TREE ALGORITHMS

Peace Ayegba
Department of Computer Science
Landmark University
Kwara, Nigeria
ayegba.peace@lmu.edu.ng

Joyce Ayoola
Department of Computer Science
Landmark University
Kwara, Nigeria
ayoola.joyce@lmu.edu.ng

Emmanuel Asani
Department of Computer Science
Landmark University
Kwara, Nigeria
asani.emmanuel@lmu.edu.ng

Aderemi Okeyinka
Department of Computer Science
Landmark University
Kwara, Nigeria
okeyinka.aderemi@lmu.edu.ng

2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS) 978-1-7281-3126-9/20/$31.00 ©2020 IEEE 10.1109/ICMCECS47690.2020.240900

***Abstract*—The minimal spanning tree is a structure used in solving certain types of combinatorial optimization problems. Popular application areas include network design such as roads, telephone, electrical and cable-laying. There are two popular algorithms for acquiring the minimum spanning tree of a graph; These are Kruskal's algorithm and Prim's algorithm. The goal of this study is to implement these algorithms and find out which of them is more efficient. To achieve this, we implement and evaluate the complexity of the algorithms. In this paper both methods have been implemented on the road network of the 36 states in Nigeria and results of the implementation are presented. However, since this work is taken from a research in progress, the computational complexity of the two algorithms will be carried out and reported in another paper in the future.**

Keywords—Kruskal, Prim's, Graph, Minimal Spanning Trees, Complexity

I. INTRODUCTION

Several network design problems such as telephone cable networks, computer networks, electrical and road networks are usually represented with graphs. The nodes in the graphs can represent the phone stations, computers or the cities while the edges represent the cable lines, the links or the roads. Spanning trees are generally used to find paths that connect all nodes in a network. A spanning tree is a subgraph that includes all vertices in the associated graph with a minimum number of paths, that is, it contains no cycles. The minimal spanning tree (MST) is a popular combinatorial optimization problem that is utilized in developing a spanning tree of an edge-weighted undirected graph whose sum of weights is minimal, whereas in the maximum spanning tree, the sum of weights is equivalent or higher than the sum of weights of every other spanning tree. The MST has several direct and indirect applications in the design of these networks. Arogundade et al. [1] applied Prim's algorithm in the transportation system of the Odeda local government in Nigeria. The graph used consisted of 88 nodes and 96 paths where the nodes represent the villages and the connecting roads represent the edges between the nodes. It was concluded that Prim's algorithm proved effective in providing the shortest routes and in reducing the fuel cost for transportation in that LGA. Effanga and Uwe [2] performed a similar implementation but made use of Prim's algorithm in producing the minimum spanning tree of the 36 state capitals in Nigeria with Yenagoa as the starting node. They propose the construction of the optimal road network discovered, in the design of telecommunication networks, transportation and even petroleum pipelines.

A. Minimal spanning trees (MST)

Several algorithms for finding minimal spanning trees have been developed. The algorithms are categorized as line-based MST and node-based MST algorithms. Kruskal's algorithm and Reverse Delete algorithm are line-based algorithms while the node-based algorithm includes Prim's algorithm and Dijkstra's algorithm. Other modified algorithms for finding MST's have been proposed. Some include the Bee algorithm-simulated annealing weighted minimal spanning tree algorithm proposed by Saravanan and Madheswaran [3], A cut-set based algorithm which gives all possible MST's in a graph proposed by Yamada et al. [4], and a quick MST algorithm which utilizes a divide and conquer approach with K-means by Akomolafe et al. [5].

Given a connected undirected graph U= (M, N), the spanning tree is formally defined as a tree U= (M, N') and N' ⊆ N. For a connected, undirected weighted graph U=M, N, w the minimum spanning tree (MST) has the least weight, and the weight or cost of the tree T is defined as:

$$c(t) = \sum_{n \in (T)} c(n) \qquad (1)$$

Some properties of spanning trees are:

- The fundamental cycle i.e. including an edge which forms a cycle.
- All spanning trees of a graph have |V|-1 edges.
- Fundamental cut set i.e. deleting one edge of a spanning tree forms two disjoints sets.

B. Combinatorial Optimization problems and structures

Optimization problems are concerned with determining the best solution from several alternative solutions. Combinatorial optimization is an optimization problem that deals with finding an optimal solution from a finite set of solutions. It is the process of identifying the maxima or minima of an objective function whose domain is discontinuous with a large configuration space. Examples of combinatorial problems include, The traveling salesman problem, Job-shop scheduling and Boolean satisfiability.

A combinatorial optimization problem is defined formally as a quadruple I, s, h, f, where

- I is a set of instances.
- s(x) contains all possible solutions, given an instance $x \in I$
- x is an instance and y is a viable solution, x, h (x, y) is the measure of y which is usually a positive real.
- f is the goal function either minimum or maximum.

The goal is to find an optimal solution y for some instance x with

$$h(x, y) = f\{h(x, y') \mid y' \in s(x)\} \quad (2)$$

II. Problem definition

Consider a business with several branches in different state capitals in Nigeria. A driver must make different deliveries to each branch exactly once during the trip. He must start from a state and end up in a destination state with a minimum total distance. The purpose of this study is to construct the MST of the paths using Kruskal and Prims algorithm to provide a minimal spanning tree with the least distance in kilometre. In the following section, a short introduction of the algorithms used is presented. In section 4 and 5, we present and discuss the computational results of the implementation of both algorithms and make conclusions.

The 'Lat/long' software was used to compute the distances in km from the latitude and longitude of each state capital in Nigeria. Figure 1 shows the screenshot of the road network of the states in Nigeria.

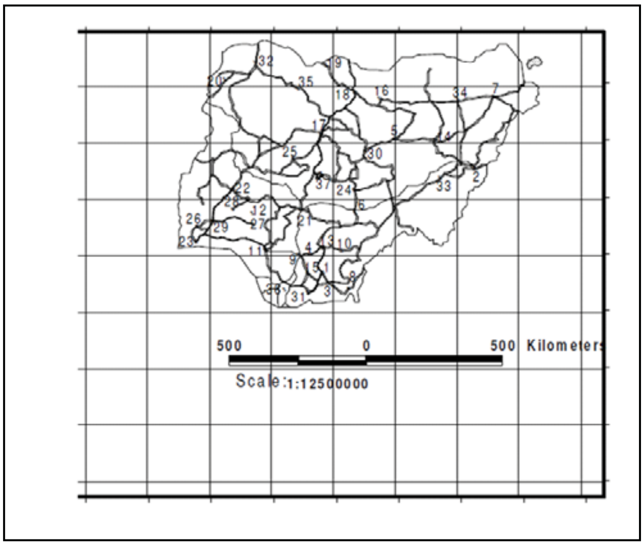

Fig. 1. Road network map of the 36 state capitals in Nigeria by Zhong et al. [6]

III. Methodology

Kruskal and Prim's algorithm are classic greedy algorithms used to produce the minimal spanning tree of an undirected connected weighted graph.

A. *Kruskal's Algorithm*

This algorithm, designed by Joseph Kruskal, appeared for the first time in the Proceedings of the American Mathematical Society in 1956. The algorithm starts by creating an ordered set of edges by weights and proceeds through the ordered set by adding an edge to the partial MST provided the new edge does not form a cycle. The algorithm employs a greedy approach whereby in each iteration stage, it finds a path with the least weight and includes it in the growing spanning tree. The steps in Kruskal's algorithm is presented as:

```
KRUSKAL(Graph):
T = Empty;
For every node n ∈ G.N:
   CreateSet(n)
For every path (m, n) ∈ G.E arranged by increasing weights(m,n):
   if NewSet(m) ≠ NewSet(n):
T = T ∪ {(m, n)}
   UNION (m, n)
return T
```

B. *Prim's algorithm*

Prim's algorithm was developed by a Czech mathematician, Vojtěch Jarník in 1930 and in 1957, it was rediscovered by Robert C. Prim and republished by Edsger W. Dijkstra in 1959. Prim's algorithm starts by picking any node r to be the root of the tree, finds the shortest path going out of the tree, adds the node to the tree if the tree does not contain all the node in the graph. Unlike Kruskal's algorithm where a path is added to the spanning tree, the node is included in the increasing tree, one at a time in Prim's algorithm. Prim's algorithm is shown below:

```
PRIM (Graph, c, t)
P ← N[Graph]
For every m Є P
            do key[m] ← ∞
            key[t] ← 0
            Π [t]←null
   while P ≠ Ø
do m ← EXTRACTMINNODE(P)
for every n Є Adjacent[m]
 do if n Є P and c (m,n ) < key[n]

          Π [n] ← m
key[n] ← c(m,n)
```

C. *Network graph of the major roads in Nigeria.*

In figure 2, the digitized graph of the road network map of the 36 states in Nigeria, and their respective distances has been presented

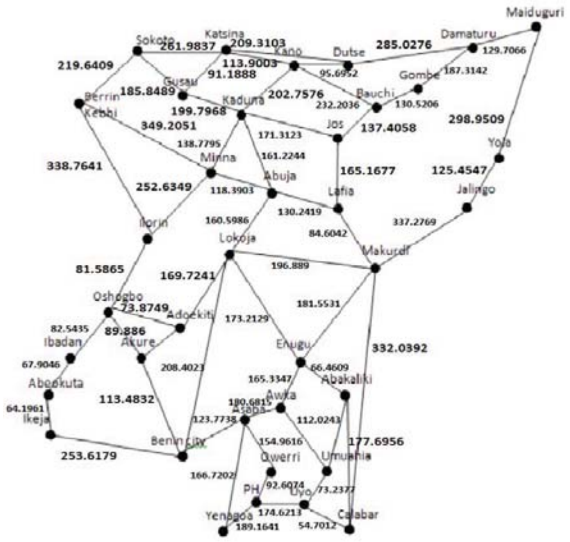

Fig. 2. Digitized graph showing the road network map with the corresponding distance in kilometers

In table 1, the longitude and latitude of state capitals is shown. This information helps to determine the distances between every state capital and Ilorin was chosen as the start node.

TABLE I. LONGITUDE AND LATITUDES OF STATE CAPITALS

State capital	Latitude	Longitude
Ilorin	8.5000N	4.5500E
Ilorin	8.5000N	4.5500E
Oshogbo	7.7660N	4.5667E
Ibadan	7.3964N	3.9167E
Abeokuta	7.1608N	3.3483E
Ikeja	6.5833N	3.3333E
Benin city	6.3176N	5.6145E
Akure	7.2500N	5.1950E
Ado-Ekiti	7.6211N	5.2214E
Lokoja	7.8167N	6.7500E
Minna	9.6139N	6.5569E
Sokoto	13.0667N	5.2333E
Katsina	12.2500N	7.5000E
Kano	12.000N	8.5167E
Dutse	11.7011N	9.3419E
Damaturu	11.7444N	11.9611E
Maiduguri	11.8333N	13.1500E
Gombe	10.2500N	11.1667E
Bauchi	10.5000N	10.0000E

State capital	Latitude	Longitude
Jos	9.9333N	8.8833E
Lafia	8.4917N	8.5167E
Makurdi	7.7306N	8.5361E
Yola	9.2300N	12.4600E
Jalingo	8.9000N	11.3667E
Enugu	6.4527N	7.5103E
Awka	5.0000N	7.8333E
Asaba	6.1978N	6.7285E
Abakaliki	6.3333N	8.1000E
Uyo	5.0500N	7.9333E
Owerri	5.4850N	7.0350E
Calabar	4.7500N	8.3250E
Umuahia	5.5333N	7.4833E
Port-Harcourt	6.3176N	7.0000E
Yenagoa	4.7500N	6.3333E
Gusau	12.1500N	6.6667E
Kaduna	10.5167N	7.4333E
FCT Abuja	9.0667N	7.4833E
Birnin Kebbi	11.5000N	4.0000E

IV. RESULTS AND SUMMARY

The entire graph consists of 37 vertices representing the state capitals and 55 edges representing the roads. Results of the implementation show that Kruskal's algorithm produces an MST with a minimum distance of 4,728.2768km while Prim's algorithm produces an MST with a distance of 4,908.45148km. Figure 3 and 4 show the minimal spanning trees produced by both Kruskal's and Prim's algorithm respectively.

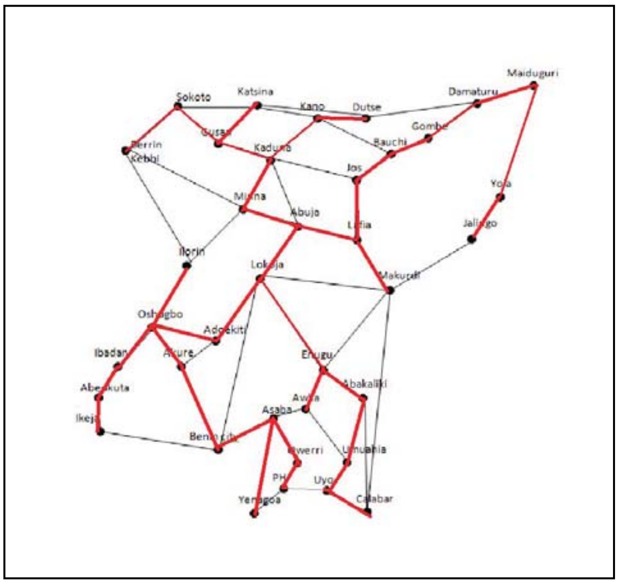

Fig. 3. Digitized graph showing the road network map with the corresponding distance in kilometers

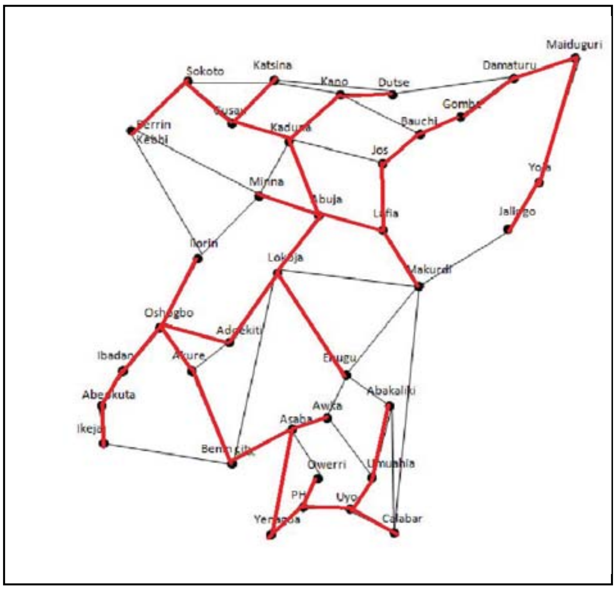

Fig. 4. Digitized graph showing the road network map with the corresponding distance in kilometers

V. Conclusion

From the results above, Kruskal's algorithm produces a minimal spanning tree with less distance in kilometre compared to Prim's algorithm, suggesting that Kruskal's algorithm has produced a better result. However, in our ongoing research we intend to investigate the behavior of both algorithms on dense graphs and compare the results, since this implementation was carried out on a sparse graph.

Acknowledgement

The authors would like to thank the reviewers for their constructive input and contributions to the study.

References

[1] O. T. Arogundade, B. Sobowale, & A. T. Akinwale, "Prim Algorithm Approach to Improving Local Access Network in Rural Areas," *International Journal of Computer Theory and Engineering*, vol. 3, issue 3, pp. 413–417, June 2011.

[2] E. O. Effanga & U. E. Edeke, "Minimum Spanning Tree of City to City Road Network in Nigeria," *IOSR Journal of Mathematics,* vol. 12, issue 4, pp. 41–45, July 2016.

[3] M. Saravanan & M. Madheswaran, "A Hybrid Optimized Weighted Minimum Spanning Tree for the Shortest Intrapath Selection in Wireless Sensor Network," *Mathematical Problems in Engineering,* vol 2014, pp. 1 – 8, July 2014.

[4] T. Yamada, S. Kataoka, & K. Watanabe, "Listing all the minimum spanning trees in an undirected graph," *International Journal of Computer Mathematics,* vol. 87, issue 14, pp. 3175-3185, Nov 2010.

[5] D.T. Akomolafe, F.O. Adekayode, J.A. Gbadeyan, & T.S. Ibiyemi, "Enhancing road and monitoring and safety through the use of Geo-spatial technology," *International journal of physical sciences*, vol. 4, issue 5, pp. 343 – 348, May 2009.

[6] C. Zhong, M. Malinen, D. Miao, & P. Franti, "A fast minimum spanning tree algorithm based on K-means," *Information Sciences*, vol. 295, pp. 1 – 7, Feb. 2015.