Research Article

Journal of Electrical Electronics Engineering

Innovative Matrix Algorithm to Address the Minimal Cost-Spanning Tree Problem

K.P.O.Niluminda[1]*, E.M.U.S.B.Ekanayake[2]

[1],[2]Department of Physical Sciences, Faculty of Applied Sciences, Rajarata University of Sri Lanka, Mihinthale, Sri Lanka

***Corresponding Author**
K.P.O.Niluminda, Department of Physical Sciences, Faculty of Applied Sciences, Rajarata University of Sri Lanka, Mihinthale, Sri Lanka.

Submitted: 28 Nov 2022; **Accepted:** 05 Dec 2022; **Published:** 14 Dec 2022

Citation: Niluminda K.P.O*, Ekanayake E.M.U.S.B. (2022). Innovative Matrix Algorithm to Address the Minimal Cost-Spanning Tree Problem. *J Electrical Electron Eng,* 1(1), 148-153.

Abstract
A spanning tree in a connected graph is a subgraph that forms a tree by connecting all the nodes. The multiple spanning trees may exist in the same graph. Additionally, by computing the total cost of each edge in a spanning tree, we can assign an expense to a spanning tree, which measures how unfavorable it is, and do the same for each edge. A tree with the fewest total edge costs is the Minimal Cost Spanning Tree (MCST). To obtain MCST, several techniques have been developed. Kruskal's and Prim's algorithms, which are widely respected and recognized practices, can be used to find the MCST. Prim's method is node-based as opposed to Kruskal's, which is based on arcs (edges). It is preferred to utilize Kruskal's technique when the graph is sparse, and Prim's approach when it is solid. The term "finite graph" in graph theory refers to a square matrix called an adjacency matrix. The matrix elements display the proximity of vertex pairs in a graph. When discussing an adjacency matrix for a weighted graph, it is sometimes referred to as the cost (weight) adjacency matrix. This work suggested a matrix technique that uses the cost adjacency matrix to determine the MCST of a given undirected connected graph. The recommended method is then used to present illustrative examples and achieve outcomes comparable to those of the Prim and Kruskal algorithms.

Keywords: Adjacency Matrix, Connected Graph, Minimal-Cost Spanning Tree, Multiple Spanning Tree, Subgraph

Introduction
The study of graphs, which serve as mathematical models for pairwise relationships between objects, is known as graph theory. Graph G (V, E) is a collection of nodes (also known as vertices) linked by edges (also known as links). A tree, also known as an attached acyclic undirected graph, is an undirected graph in which any two vertices are connected by precisely one route [1]. A subgraph of the graph called a spanning tree visits every vertex and is shaped like a tree. There can be n (n-2) spanning trees in a fully connected undirected graph, where n is the total number of nodes. When the weight of the edges added together is as tiny as feasible, the spanning tree is said to be the MCST. In design networks, such as transportation, water supply, electrical grids, and computer network, MCST are used directly [2]. For a problem like a phone network design, the standard application is used. You wish to rent phone lines to connect your firm's many offices; however, the phone provider charges varying rates to connect various city pairs. To link all of your offices at the lowest possible cost, you need a set of lines. Since you can always cut back on costs by removing certain edges from a network if it is not a tree, it should be a spanning tree. A graph can be represented as a matrix of Booleans (0 and 1) using an adjacency matrix. A square matrix can describe a finite network, with the Boolean value of the matrix indicating whether there is a direct path between any two vertices. A weighted graph's adjacency matrix is also known as the cost (weight) adjacency matrix [3]. In 2019, A. Khan developed a new algorithmic approach to finding minimum spanning trees [1].

Review of Related Studies
The MCST can be determined using different classic algorithms. The first algorithm for finding MWST was developed by Czech researcher [4]. Prim's algorithm is second; it was created by [2]. Joseph Kruskal proposed the third algorithm in 1956, known as Kruskal's algorithm [5]. The reverse-delete algorithm, which is the opposite of Kruskal's method, is a fourth algorithm that is less frequently utilized [6]. In an efficient algorithm for finding minimum spanning trees in undirected and directed graphs [7]. The Negligence Minimum Spanning Tree Algorithm was published by [8]. In 2016, P. Biswas developed an efficient greedy minimum spanning tree algorithm based on the vertex associative cycle detection method [9]. In the literature, there are few methods to find MCST using matrix algorithms. proposed a matrix algorithm to find [10]. A new algorithm for finding minimum spanning trees with undirected neutrosophic graphs was proposed by [11]. Ant colony optimization is a meta-heuristic algorithm. It uses to find the shortest path between two points. Using this algorithm in found an approach for solving the minimum spanning tree problem and transportation problem using a modified ant colony algorithm [2]. P. Ayegba did a comparative study of the minimal spanning tree algorithm in 2020 [12]. In 2019, U. Paradigm researched the multi-objective minimum spanning tree problem under an uncertain paradigm [13]. In studied the locality behavior of the minimum spanning tree algorithm [14]. The new algorithm involving a minimum spanning tree for a computer network in a growing company was done by [15]. An optimal minimum spanning tree algorithm surveyed multiple objective minimum spanning tree problems

in 2009 [16, 17].

Objectives of the Study

This paper, suggests a unique method that requires fewer iterations than the existing approach, utilizing the cost adjacency matrix to determine the MCST of an undirected graph G. In graph theory, a square matrix known as an adjacency matrix is used to describe a finite graph. The matrix's components show whether a graph's vertex pairs are adjacent. It is also known as the cost adjacency matrix when referring to an adjacency matrix for a weighted graph. Finally, we used the recommended strategy to solve an illustrative example and compared it to Prim and Kruskal's methods [18].

Methodology

This section describes a cost adjacency matrix approach that can use to determine the MCST of an undirected connected graph. The steps to solve the minimal spanning problem are as given below. There are five steps in this algorithm. This algorithm was developed using matrix representation.

Proposed Novel Matrix Algorithm for Find MCST

Step 1: If any vertex has a loop (the edge that starts and ends with the same vertex), then remove that loop, and if there is a loop between two vertices, then remove this loop by deleting parallel edges, which have the highest, edge cost. Otherwise, go to step 2.

Step 2: Create the n×n cost matrix, $C(G)=[c_{ij}]_{n\times n}$ for the given undirected weighted connected graph G.

Step 3: Select the smallest element (expect 0's) of each row and column, and mark them separately.

Step 4: Draw selected edges in step 3 in ascending order. If selected edges create a loop, then ignore that edge.

Step 5: If the resulting graph is disconnected, then connect those subgraphs by using edges that have a minimum cost.

The flow chart of the proposed novel algorithm is represented in Chart 1. This algorithm has five steps. Applying this algorithm, any undirected connected graph's minimal spanning tree can generate.

Results and Discussion

In this part, use the newly presented model to determine the MST of an undirected graph using two examples. Finally, the results obtained in instances will compare with Kruskal's and Prim's algorithms.

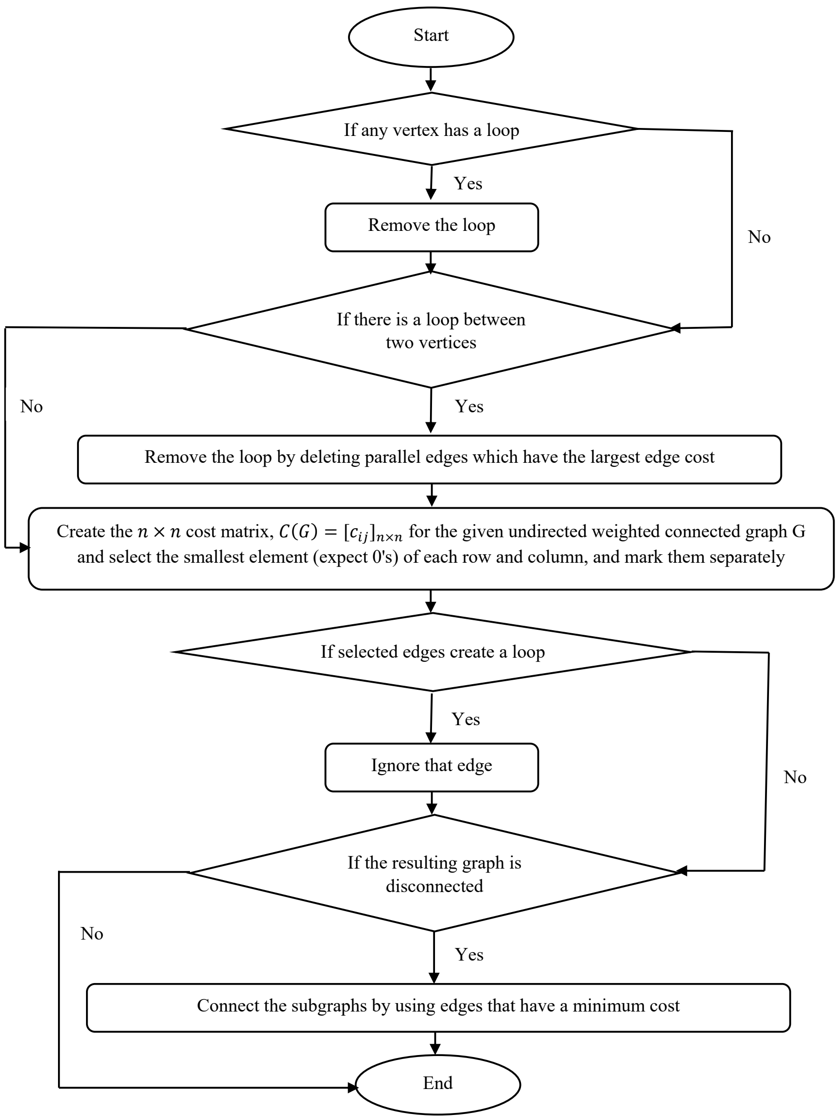

Figure 1: The Flow Chart of New Proposed Matrix Algorithm

Example 1 [18]:

Step 1:

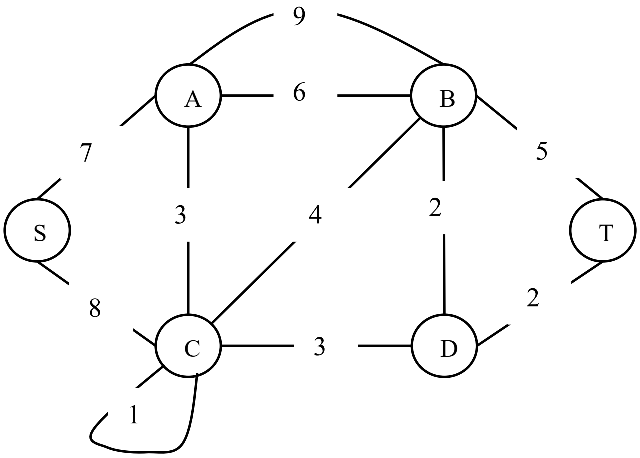

Figure 2: Connected Undirected Graph 1

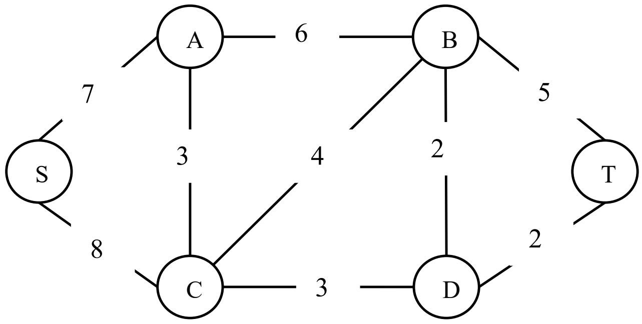

Figure 3: Reduced graph after step 1

$$C(G) = \begin{bmatrix} \ddots & \vdots & \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} & \boldsymbol{S} & \boldsymbol{T} \\ \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{A} & \vdots & 0 & 6 & 3^{*+} & 0 & 7^{+} & 0 \\ \boldsymbol{B} & \vdots & 6 & 0 & 4 & 2^{*+} & 0 & 5 \\ \boldsymbol{C} & \vdots & 3^{*+} & 4 & 0 & 3^{*} & 8 & 0 \\ \boldsymbol{D} & \vdots & 0 & 2^{*+} & 3^{+} & 0 & 0 & 2^{*+} \\ \boldsymbol{S} & \vdots & 7^{*} & 0 & 8 & 0 & 0 & 0 \\ \boldsymbol{T} & \vdots & 0 & 5 & 0 & 2^{*+} & 0 & 0 \end{bmatrix}$$

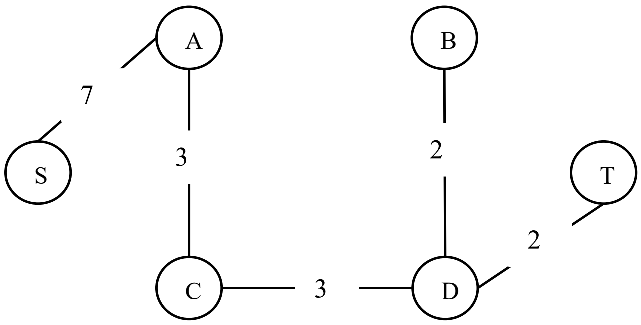

Figure 4: Minimal Spanning Tree of Graph 1

The order of selecting edges:
BD – DT – AC – CD – AS
The total cost of MCST = (2 + 2 + 3 + 3+ 7) = 17
The total cost of MCST using Prim's algorithm = 17
The total cost of MCST using Kruskal's algorithm = 17

In Figure 1 connected undirected graph has six nodes and eleven edges. Node C has a loop itself, and a loop between nodes A and B. In step 1, If any vertex has a loop, then remove that loop, and if there is a loop between two vertices, then remove this loop by deleting parallel edges which have the highest edge cost. The loop between nodes A and B has two parallel edges so the deleted edge which has the highest expense (i.e., 9). After applying step 1, the reduced graph is shown in Figure 2. Then applying the algorithm step by step, the MCST of graph one can be obtained. It shows in Figure 3. The total cost of MCST using the proposed method is equal to 17. By comparing Prim's and Kruskal's algorithms, the same outcome can be obtained.

Example 2 [18]:

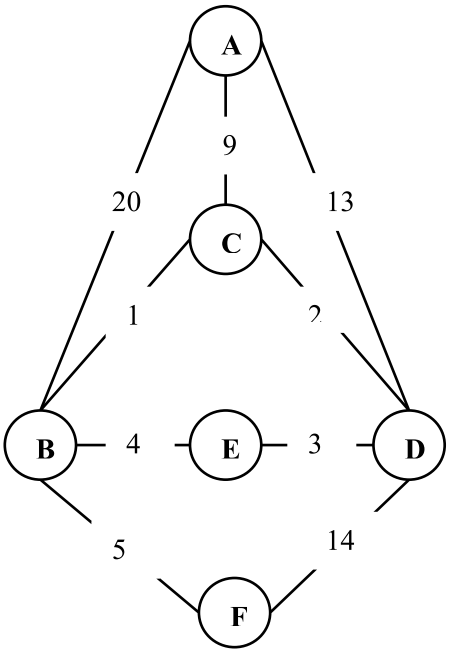

Figure 5: Connected Undirected Graph 2

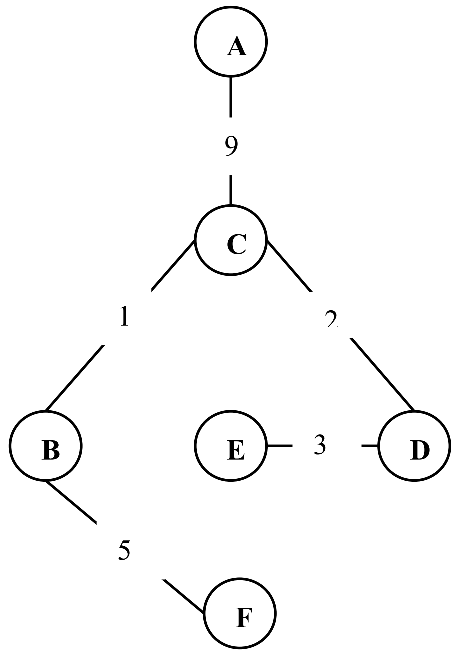

Figure 6: Minimal Cost Spanning Tree of Graph 2

$$C(G) = \begin{bmatrix} \ddots & \vdots & \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} & \boldsymbol{E} & \boldsymbol{F} \\ \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{A} & \vdots & 0 & 20 & 9^{*} & 13 & 0 & 0 \\ \boldsymbol{B} & \vdots & 20 & 0 & 1^{*+} & 0 & 4 & 5^{+} \\ \boldsymbol{C} & \vdots & 9^{+} & 1^{*+} & 0 & 2^{+} & 0 & 0 \\ \boldsymbol{D} & \vdots & 13 & 0 & 2^{*} & 0 & 3^{+} & 14 \\ \boldsymbol{E} & \vdots & 0 & 4 & 0 & 3^{*} & 0 & 0 \\ \boldsymbol{F} & \vdots & 0 & 5^{*} & 0 & 14 & 0 & 0 \end{bmatrix}$$

The order of selecting edges:
BC – CD – DE – BF – AC
The total cost of MCST = (1 + 2 + 3 + 5 + 9) = 20
The total cost of MCST using Prim's algorithm = 20
The total cost of MCST using Kruskal's algorithm = 20
Figure 4 shows a connected undirected graph with six nodes and nine edges. There are no loops in between two vertices or itself in this example. So that step 1 can ignore, and the algorithm can start in step 2. Then applying the algorithm step by step, the MCST of connected undirected graph, two can be obtained. It shows in Figure 5. The total cost of MCST using the proposed method is equal to 20. By comparing Prim's and Kruskal's algorithms, the same outcome can obtain.

Comparative Analysis

In this section, the outputs achieved in examples 1 and 2 are compared with other existing methods, whereas Prim's and Kruskal's algorithms.

Table 1: Comparative analysis of examples 1 and 2 with Prim's, Kruskal's, and New Matrix Algorithm (NMA)

Example No	Prim's Algorithm	Kruskal's Algorithm	NMA
1	17	17	17
2	20	20	20

Interpretation of Table 1.

There are no value changes between Prim's, Kruskal's, and New Matrix Algorithm (NMA). In examples 1 and 2, the same findings are achieved using NMA by comparing them to the existing algorithms.

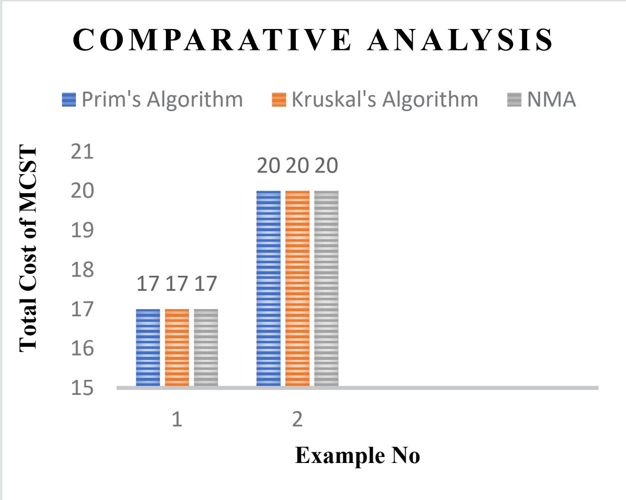

Figure 7: Showing the comparative analysis of example 1 and 2 with Prim's, Kruskal's and NMA algorithms

Figure 7 and Table 1 above demonstrate the effectiveness of the new approach. Similar results may be obtained compared to Prim's and Kruskal's methods. Example 1's minimal cost-spanning tree has 17 units overall, whereas Example 2's cost is 20 units. The Prim and Kruskal algorithms can generate an identical result. In figure 7 above, Example 1 and 2 bar graphs have the same heights. That means the proposed new matrix algorithm method also will give the same results as compared to the other existing methods.

Conclusion

This research represents a novel definite approach for computing the MCST of the weighted undirected graph G utilizing the cost (weight) matrix on a given connected graph. The MCST starts with a single node and determines all of its reachable nodes and even the set of connections that link them with the least amount of cost. For computing the MCST of a given connected graph G, the well-known greedy algorithms Prim and Kruskal are used to find the minimal-spanning tree. It is possible to determine if a direct path exists between any two vertices using the adjacency matrix of a weighted graph, generally referred to as the cost adjacency matrix. The proposed approach has been created utilizing a cost adjacency matrix. Compared to other current methods, this new algorithm is more straightforward and less challenging to implement. This paper used two examples to illustrate the new proposed approach, and when compared to Prim's and Kruskal's algorithms, it can able to get the same outcome.

References

1. Khan, A., Aesha, A. A., & Sarker, J. (2018, September). A new algorithmic approach to finding minimum spanning tree. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 590-594). IEEE.
2. U. Ekanayake, W. Daundasekara, and P. Perera. (2020). "Research An Approach for Solving Minimum Spanning Tree Problem and Transportation Problem Using Modified Ant Colony Algorithm," North Am. Acad. Res., vol. 3, no. 9, pp. 28–45.
3. Busato, F., & Bombieri, N. (2017). Graph algorithms on GPUs. Adv. GPU Res. Pract, 163-198.
4. Nešetřil, J., Milková, E., & Nešetřilová, H. (2001). Otakar Borůvka on minimum spanning tree problem Translation of both the 1926 papers, comments, history. Discrete mathematics, 233(1-3), 3-36.
5. Salahddine, K. (2021). The Implementation of Kruskal's Algorithm for Minimum Spanning Tree in a Graph. In E3S Web of Conferences (Vol. 297). EDP Sciences.
6. Choi, M. B., & Lee, S. U. (2013). An Efficient Implementation of Kruskal's and Reverse-Delete Minimum Spanning Tree Algorithm. The Journal of The Institute of Internet, Broadcasting and Communication, 13(2), 103-114.
7. Gabow, H. N., Galil, Z., Spencer, T., & Tarjan, R. E. (1986). Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica, 6(2), 109-122.
8. SAYLI, A., & ALKHALİSSİ, J. H. (2018). Negligence Minimum Spanning Tree Algorithm. Avrupa Bilim ve Teknoloji Dergisi, (14), 70-76.
9. Biswas, P., Goel, M., Negi, H., & Datta, M. (2016). An efficient greedy minimum spanning tree algorithm based on vertex associative cycle detection method. Procedia Computer Science, 92, 513-519.
10. Vijayalakshmir, D. (2014). Minimum cost spanning tree using matrix algorithm.
11. Dey, A., Broumi, S., Son, L. H., Bakali, A., Talea, M., & Smarandache, F. (2019). A new algorithm for finding minimum spanning trees with undirected neutrosophic graphs. Granular Computing, 4(1), 63-69.
12. Ayegba, P., Ayoola, J., Asani, E., & Okeyinka, A. (2020, March). A comparative study of minimal spanning tree algorithms. In 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS) (pp. 1-4). IEEE.
13. S. V. Kamble and B. G. Kore. (2019) "A New Method to Obtain an Initial Basic Feasible Solution of Transportation Problem with the Average Opportunity Cost Method," Int. J. Eng. Adv. Technol., vol. 9, no. 2, pp. 206–209.
14. Cong, G., & Sbaraglia, S. (2006, December). A study on the locality behavior of minimum spanning tree algorithms. In

International Conference on High-Performance Computing (pp. 583-594). Springer, Berlin, Heidelberg.
15. KİRİSCİ, M., & Öncel, D. (2017). The new algorithm involving minimum spanning tree for computer networks in a growing company. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 16(32), 1-14.
16. Pettie, S., & Ramachandran, V. (2002). An optimal minimum spanning tree algorithm. Journal of the ACM (JACM), 49(1), 16-34.
17. Ruzika, S., & Hamacher, H. W. (2009). A survey on multiple objective minimum spanning tree problems. In Algorithmics of Large and Complex Networks (pp. 104-116). Springer, Berlin, Heidelberg.
18. P. S. Reddy and A. M. Derdar. (2017). "Minimal Spanning Tree", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 6 Issue 03, March-2017.