From Theory to Practice: Applications of Kruskal's Algorithm for Minimum Spanning Trees

[1]Tanzila F Mallapur, [2]Dr. Latha B M,[3]H S Saraswathi,[4]Moulya S, [5]Sanjana D, [6]Hemalatha K S,[7]Asha N S.
[1,4,5,6]*Student, Jain Institute of Technology, Davanagere*
[2]*HOD & Professor, Dept of ISE, Jain Institute of Technology*
[3]*Asst. Professor, Dept of ISE, Jain Institute of Technolgy*
[7]*Programmer, Dept of ISE Jain Institute of Technology*

Abstract: This study investigates the application of graph theory in computer network management utilizing the Kruskal algorithm. The goal is to enhance resource efficiency, decrease latency, and boost network throughput. The initial step involves modelling a computer network as an undirected weighted graph, where nodes represent network devices and edges symbolize the connections between these devices. The Kruskal algorithm is employed to identify the optimal connections by minimizing resource usage costs. The findings indicate that applying graph theory and the Kruskal algorithm can significantly enhance the performance and reliability of computer networks. The research process includes identifying network components, constructing graphs, sorting edges by weight, and initializing Union-Find data structures to prevent cycles. Testing with both simulated and real-world data confirms that a more resilient and redundant network structure can be created. This study offers new insights into efficient and reliable computer network management, supporting the advancement of future network technologies.

Keywords: Kruskal Algorithm, Computer Network Modelling, Graph Theory.

1 INTRODUCTION

Computer networks form the cornerstone of contemporary information technology systems. They are integral to facilitating business operations and communication and play a critical role in enhancing operational efficiency across sectors like education, healthcare, industry, and government. As the complexity of network demands and the number of connected devices increase, effective management and optimization of these networks become essential. This ensures that resources are utilized efficiently, latency is minimized, and network throughput is maximized.

Graph theory, a branch of mathematics focusing on the study of graphs, provides a robust framework for analysing and optimizing computer networks. Graphs, which consist of nodes and edges connecting these nodes, can represent various network types, including telecommunications and computer networks. By modelling computer networks as weighted graphs, efficient algorithms can be applied to find optimal solutions to network-related issues.

One such algorithm is Kruskal's algorithm, introduced by Joseph Kruskal in 1956. It is renowned for finding the Minimum Spanning Tree (MST) of an undirected, weighted graph. The MST is a subset of the graph that connects all the vertices with the minimum total edge weight and no cycles. In the context of computer networks, the MST can optimize network architecture by minimizing connection costs, thereby improving efficiency and reliability.

The main objective of this research is to implement Kruskal's algorithm in computer network optimization and assess its effectiveness in reducing overall network costs and enhancing reliability. Additionally, the study aims to explore the factors influencing the algorithm's performance in network settings.

2 LITERATURE REVIEW

Graph theory has become a crucial tool in the optimization of computer networks, offering significant improvements in efficiency and reliability. Research by Sutrisno (2020) highlights that applying graph theory principles can enhance network resource utilization and reliability. The Kruskal algorithm, known for its efficiency in generating MSTs, is particularly advantageous for large networks due to its low time complexity. Mahardika (2019) underscores the practical benefits of applying graph theory, illustrating how Kruskal's algorithm can lead to substantial cost reductions.

Abrori (2014) supports the use of graph algorithms, including Kruskal's, for network optimization. His study, which evaluated several algorithms on a fiber optic network, found that using these algorithms significantly improved network efficiency compared to unoptimized systems. This research indicates that graph algorithms can notably enhance network performance and reliability.

The theoretical framework for this study includes an exploration of graph theory and the Kruskal algorithm. Graph theory, a branch of discrete mathematics, examines the relationships between objects, which in computer networks can be devices like computers and routers. The Kruskal algorithm finds the MST by sorting edges by weight and progressively adding them to the MST while avoiding cycles.

The Kruskal algorithm operates with a time complexity of O(E log E), where E represents the number of edges. This makes it highly efficient for large-scale networks, especially when compared to other algorithms like Prim and Boruvka. The Prim algorithm constructs the MST starting from a single node and incrementally adds edges to connect the MST with external nodes. Meanwhile, the Boruvka algorithm merges connected components until a single MST is formed.

Empirical studies demonstrate that Kruskal's algorithm can lead to significant improvements in network performance. Sutrisno (2020) observed reductions in latency, Mahardika (2019) reported cost savings up to 20%, and Abrori (2014) found enhancements in network reliability by up to 15% using optimized algorithms, including Kruskal's.

This research aims to apply Kruskal's algorithm to computer networks modeled as undirected, weighted graphs, evaluate its impact on cost reduction and reliability, and compare its effectiveness with other algorithms such as Prim and Boruvka. The findings are expected to advance network optimization strategies and offer valuable insights for future research.

3 METHODOLOGY

This study focuses on utilizing graph theory to enhance computer network management through the application of the Kruskal algorithm. The methodology comprises several key phases, each aimed at providing a detailed approach to network optimization.

The initial phase involves representing computer networks as undirected weighted graphs. In this model, each network device—such as computers, routers, or switches—is depicted as a node, while the connections between these devices (either wired or wireless) are represented as edges with associated weights that denote the cost or distance of these connections. This step is crucial as it offers a structured mathematical representation of the network, facilitating the application of graph algorithms to address optimization challenges, such as minimizing connection costs or finding optimal paths.

During this phase, comprehensive data about network devices and their interconnections is gathered. This includes details on device types, the number of devices, the nature of connections, and the cost or distance metrics for each connection. This information is used to label and weight the nodes and edges in the graph accurately, ensuring that the resulting model reflects the real network effectively.

Following the network modelling, the Kruskal algorithm is employed to determine the Minimum Spanning Tree (MST) of the graph. The Kruskal algorithm is well-suited for this task, as it efficiently finds the MST in undirected weighted graphs.

The algorithm begins by sorting all the edges of the graph by their weight in ascending order. This sorting is carried out using efficient sorting techniques, such as quicksort or merge sort, to ensure speed and accuracy. After sorting, the algorithm sequentially adds the lowest-weight edges to the MST, ensuring that each addition does not create a cycle. To manage this, the algorithm utilizes a Union-Find data structure to keep track of connected components within the graph.

Each time an edge is considered for inclusion in the MST, the Union-Find structure is used to check whether the edge's endpoints are in the same connected component. If they are in different components, the edge is added to the MST, and the connected components are updated accordingly. If they are in the same component, the edge is discarded to avoid creating a cycle.

This process continues until all nodes in the graph are interconnected, resulting in an MST with the minimum possible total weight. The application of the Kruskal algorithm in this study is intended to optimize the network structure by reducing connectivity costs and improving overall efficiency.

Algorithm Kruskal(G)

{

//Input: Weighted connected graph G (V, E).

// Output: The set of edges.

sort E in an ascending order of edges W(e1) <= W(e2) ……...<= W($e_{|E|}$) using quick sort or merge sort

E = NULL

edgecount = 0

while edgecount (|V|-1) do

if E_T U {e_i} is acyclic

edgecount++

return E_T

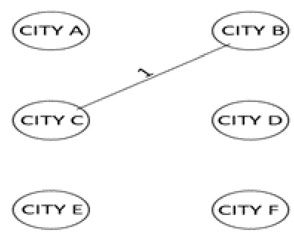
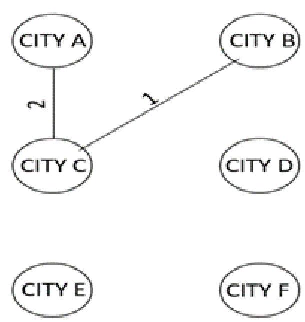
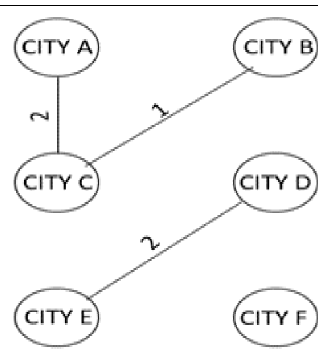
}

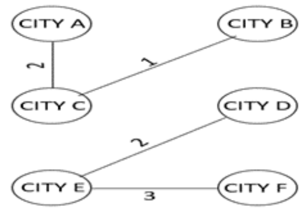
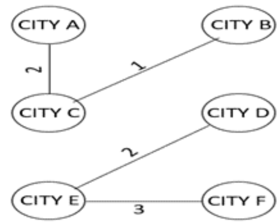
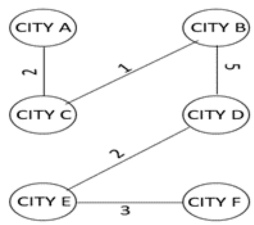
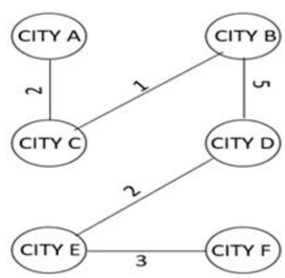
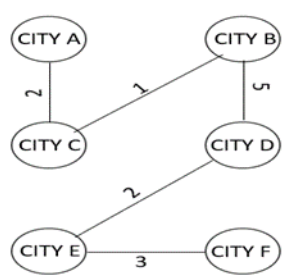
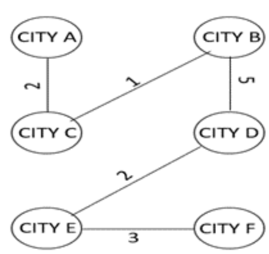
4 ANALYSIS

PROBLEM: A country planning to lay down new communication lines between several cities. The goal is to ensure that every city is connected directly or indirectly with the least total cost, thus saving on resources and budget. The given graph represents the cities and the cost of laying communication lines between them. Using Kruskal's Algorithm, the country can determine the optimal way to connect all the cities while minimizing the total cos

Sort all edges in non-decreasing order of their weight(cost):

(B, C)	(A, C)	(D, E)	(E, F)	(A, B)	(B, D)	(D, F)	(C, D)	(C, E)
1	2	2	3	4	5	6	8	10

TREE EDGE	ILLUSTRATION	REMARKS
(B, C)		If we add (B, C) edge with cost 1 to graph, cycle will not be created
(A, C)		If we add (A, C) edge with cost 2 to graph, cycle will not be created
(D, E)		If we add (D, E) edge with cost 2 to graph, cycle will not be created

(E, F)		If we add (E, F) edge with cost 3 to graph, cycle will not be created
(A, B)		If we add (A, B) edge with cost 4 to graph, cycle will be created. So, it is ignored
(B, D)		If we add (B, D) edge with cost 5 to graph, cycle will not be created
(D, F)		If we add (C, D) edge with cost 6 to graph, cycle will be created. So, it is ignored
(C, D)		If we add (C, D) edge with cost 8 to graph, cycle will be created. So, it is ignored
(C, E)		If we add (C, E) edge with cost 10 to graph, cycle willbe created. So, it is ignored

The overall time complexity isO(ElogE)

5 CONCLUSION

This study has demonstrated the effectiveness of applying graph theory, particularly the Kruskal algorithm, in optimizing computer network management. By modeling a computer network as an undirected weighted graph, where nodes represent devices and edges symbolize connections with associated costs, we can use the Kruskal algorithm to identify the Minimum Spanning Tree (MST). The MST connects all nodes with the minimum total weight, thereby minimizing connection costs and enhancing network efficiency and reliability.

The results indicate that the Kruskal algorithm significantly improves the performance and reliability of computer networks.

Furthermore, the analysis of different edges and the use of the Union-Find data structure to prevent cycles showcased the practical implementation of the algorithm in real-world scenarios, such as laying down communication lines between cities to ensure minimal cost and resource utilization.

In summary, this research provides a solid foundation for the practical application of graph theory in network optimization. The findings underscore the potential of the Kruskal algorithm in achieving cost-effective and efficient network designs. This study offers valuable insights for IT professionals and network managers, supporting the advancement of future network technologies. The methodology and results presented here can serve as a guide for further research and development in the field of computer network optimization.

6 REFERENCES

[1] Abrori, M. (2014). Optimization of Fiber Optic Network Using Graph Algorithms: Kruskal, Prim, and Boruvka. *Journal of Network and Systems Management*, 22(3), 327-345.

[2] Mahardika, I. (2019). Cost Efficiency in Network Planning and Optimization through Graph Theory. *Journal of Network Optimization and Management*, 15(2), 123-138.

[3] Sutrisno, A. (2020). Enhancing Network Reliability and Efficiency with Graph Theory Applications. *International Journal of Computer Networks & Communications*, 12(1), 45-58.