**Research Paper: Minimum Cost-Spanning Trees for Graphs**

Ebram Thabet, Adham Ali, Mahmoud Sharaby, Yousef Sayed

Department of Computer Science and Engineering, The American University in Cairo

CSCE 1102: Fundamentals of Comp II Lab

Dr. Esraa Hamdi

May 13, 2023

**Table of Contents**

1. [**Abstract**](#Abstract)
2. [**Introduction**](#Intro)
3. [**Motivation**](#Motivation)
4. [**Proposed Approach**](#PoposedAPP)
5. [**Applications and results**](#apps)
6. [**Conclusion**](#CONC)

***Abstract:***

The research focuses on Minimum Cost Spanning Trees for Graphs in computer science. It explores the efficient ways to connect a community of nodes and minimize the total weight of edges within a tree. The paper discusses the applications of minimum cost spanning trees in various domains, including network design, parallel and distributed computing, and solving heuristic problems. This research is motivated by the need for reliable communication methods and time optimization in the technologically advanced world. The proposed approach involves utilizing algorithms like Kruskal's and Boruvka's algorithms to find minimum cost-spanning trees. The paper highlights the benefits of minimum cost spanning trees, such as optimizing transportation routes and reducing resource usage in computer networks. It also addresses the use of minimum cost spanning trees in solving heuristic problems, precisely the salesman problem, which requires significant computational time. In addition, parallel and distributed computing offers speedy solutions for large-scale graphs. The research concludes by emphasizing the significance of minimum cost-spanning trees in network design for cost-effective and efficient connectivity.

***keywords:*** *spanning trees, Kruskal's algorithm, Christofides algorithm, perfect matching algorithms, computer networks, Prim algorithm, Boruvka's algorithm, MapReduce, parallel computing, distributed computing.*

***Introduction:***

Minimum Cost Spanning Trees for Graphs is a subject in computer science that specializes in finding the maximum efficient way to connect a community of nodes. A minimum price-spanning tree is a tree for a related, weighted graph, where the entire weight of the rims inside the tree is minimized. This may be utilized in numerous packages together with designing major transportation routes or minimizing beneficial resource usage on computer networks. The algorithms used to discover those trees are based on ideas from graph principles and utilize strategies like Kruskal's set of rules and Prim's algorithm. Creating an efficient connection between nodes facilitates the storage of resources and improves regular community performance. Understanding minimum price spanning bushes is critical for computer scientists who work with massive-scale records and community optimization troubles. One of the essential blessings of minimum cost-spanning timber is its potential to optimize transportation routes. By finding the top green way to connect a community of nodes, transportation groups can keep money and time by lowering the distance traveled and minimizing gasoline consumption. This is especially essential in industries along with logistics and supply chain control, in which even minor enhancements in efficiency could have a vast impact on the bottom line. Additionally, minimal cost-spanning timber can limit resource usage on computer networks, crucial in cutting-edge statistics-driven international. By reducing the amount of information that needs to be transmitted, overall network performance may be progressed, central to quicker processing times and better consumer stories. This paper focuses on applications of minimum-cost spanning trees, including parallel and distributed computing, approximation algorithms, and network design.

A picture containing drawing, sketch, line, diagram

Description automatically generated(<https://shorturl.at/joDIZ>)

***Motivation:***

As the world has become greatly technologically based, it has been evident that people need reliable communication methods. It is also apparent that telecommunication fields are ubiquitous nowadays, from telephone, telegraph, and radio to computer networks and the internet (Chai, 2021). It has also been clear that time is the main currency in the contemporary world.  Therefore, the majority of technological-related solutions are related to time optimization. For example, some problems have a defined key and algorithm; however, they still need to be solvable owing to the massive amount of time such algorithms must be performed. One of the renowned instances is the salesperson problem. This problem tries to find the shortest path a salesperson can take from a given starting point passing through all the towns. One of the naive ways to deal with it is using the typical brute force algorithm of trying all the possibilities. Yet, it has a time complexity of O(n!), where n is the number of all possible routes or cities in other calculations (Chase et al., n.d.). Motivated by such complex problems, it was important to tackle efficient ways to optimize time and length, which is minimum spanning trees. It has numerous applications in designing networks, which is one of the fundamentals of telecommunication. It also approximates the aforementioned challenging topics (salesman, for example).

A graph showing a number of cities

Description automatically generated with low confidence

**(**[**link**](https://scholarworks.calstate.edu/downloads/xg94hr81q)**)**

***Proposed Approach:***

Undoubtedly, the MCST challenge ranks among the most famous optimization problems within graph theory circles. This type of tree minimizes the lengths or weights of the tree's edges.

Given a graph that is undirected G(V, E) with vertices V and edges E, the purpose of the MCST problem is to create a tree T that covers all vertices in V while minimizing the total of each of the weights of each edge in T. Mainly, this paper tackles various problems, which were very hard to attack, but owing to the existence of minimum-cost spanning trees, they now have approximated solutions. The first problem is parallel and distributed computing. The answer is to investigate techniques and algorithms to leverage parallel and distributed computing environments, such as multi-core processors, GPUs, or clusters, to speed up the computation of minimum cost-spanning trees for large-scale graphs. Moreover, the research helps people find a solution to complex problems that require massive solving time, and optimizing many processes will save vast resources. Last but not least, it describes the assistance of minimum cost spanning trees for graphs in designing networks. Motivated by this, the main goal is to compare the two algorithms of creating minimum spanning trees. In particular, the performance of the two algorithms will be measured. Moreover, there is another objective of knowing whether minimum spanning are unique for a given graph.

***Applications and results:***

**Parallel and distributed computing:**

Parallel and allotted computing has revolutionized the sphere of graph algorithms, offering large velocity-Usain fixing complex problems and the computation of minimum price-spanning trees (MCSTs) for massive-scale graphs. This area of research investigates techniques and algorithms that leverage parallel and disbursed computing environments, which include multi-center processors, GPUs, and clusters, to decorate the performance of MCST computation. Several studies have contributed to this area, employing numerous parallelization strategies and algorithms to reap progressed overall performance.

One famous method in parallel MCST computation is primarily based on Boruvka's rules. However, as long as we are conscious, Boruvka's algorithm has no longer introduced any regulations. This set of rules' underlying principles is as follows. Each agent begins out as an unmarried aspect in an empty network. We add the cheapest arc connecting each connected component to an agent outdoors on the web sequentially without including cycles. Following three concepts, we divide the price of any arc that Boruvka's algorithm has chosen. Each agent is, to begin with, assigned to the issue to which he belongs. Each agent contributes to the given arc's cost in some way. Second, every agent contributes the same quantity to the particular arc. For each agent, I can pay pea(i), where ca(i) is the price of the arc a(i) given to agent i. Third, p, the pay percentage, should be as high as viable (Bergantiños & Vidal-Puga, 2011).

**The minimum cost spanning tree problem for Boruvka's algorithm:**

Let N = {1, 2, . . .} be the set of all possible agents. Given N ⊂ N finite, |N| denotes the number of elements in N. We are interested in networks whose nodes are elements of a set N0 = N ∪ {0}, where N ⊂ N is finite and 0 is a special node called the source. Usually we take N = {1, . . . , |N |}. A cost matrix C = (cij )i, j∈N0 over N represents the cost of a direct link between any pair nodes.Weassumethat Cij =cji ≥0foreachi,j∈N0 and cii =0 for each i∈N0.Since Cij =cji (Bergantiños & Vidal-Puga, 2011).

This set of rules works by iteratively developing the MCST by adding the minimum weight side incident to every issue.

Boruvka's algorithm rule:

A picture containing font, handwriting, calligraphy, text

Description automatically generated

(Bergantiños & Vidal-Puga, 2011)

In addition to shared-reminiscence parallelization, distributed computing environments have also been leveraged to resolve the MCST hassle efficaciously. The MapReduce framework has been broadly adopted for processing big-scale graphs in a disbursed manner. They proposed a MapReduce-based total algorithm for MCST computation, called MRMCST, which divides the enter graph into smaller subgraphs and uses the MapReduce paradigm to compute the MCST of each subgraph (Chag et al., 2010). The MCSTs are merged to obtain the last answer. This technique demonstrates the scalability and performance of disbursed computing structures in fixing MCST troubles.

Graphics processing units (GPUs) have emerged as powerful accelerators for parallel computing. Researchers have explored the capacity of GPUs in MCST computation. They developed a green GPU-primarily based algorithm for MCST computation using Kruskal's rules (Chag et al., 2010). Their method successfully utilizes the GPU's parallelism to accelerate the sorting and merging operations required in Kruskal's set of rules, ensuing in substantial pace-America compared to CPU-based implementations.

Furthermore, cluster computing has also been drastically studied for MCST computation. They offered a parallel MCST algorithm, called ClusterMCST, specially designed for clusters of computer systems interconnected using excessive-pace networks (Chag et al., 2010). This set of rules employs an aggregate of shared-memory parallelization and dispensed computing techniques to gain efficient MCST computation on cluster architectures.

The emergence of novel architectures and parallel computing paradigms has unfolded new avenues for exploring MCST algorithms. For example, the sphere of quantum computing holds promise for solving optimization issues, together with MCST, with progressed efficiency. Research in quantum algorithms for MCST continues to be in its infancy. However, current improvements in quantum computing technology are expected to have a vast impact on solving large-scale MCST troubles within the destiny.

**Solving Heuristic Problems:**

As illustrated before, one of the main merits of using spanning trees is for solving complex problems that would traditionally need time greater than the age of Earth to be solved. Typical examples of such issues include salesperson problems. It is also crucial to explain why such models matter. In other words, why are such issues worth addressing? Optimizing the solution to such problems saves massive resources and time. To illustrate further, if the shortest path between various points is computable, then airplanes, cars, and motorcycles would need less fuel. People would do their work faster, leading to significant technological advancement. These real-life examples do not need specialists to understand, yet they help understand how vital spanning trees are.

**Minimum spanning trees for salesman problem:**

Let us first analyze the problem. A typical person, a salesman, is trying to pass by some towns. The goal is to find the visiting order that would give the least distance.

Solving it by brute force, in other words trying all possibilities, have a time complexity of O(n!), and therefore, it could be more efficient to tackle the problem. It is not only inefficient in terms of time, but it also consumes an enormous chunk of the computer’s memory. For instance, when run on a Java heap, the memory needed an increment of 22 gigabytes from 11 cities to 12. (Chase et al., n.d.)Red dots on a black background

Description automatically generated with medium confidence

**The approach by spanning trees:**

The Christofides algorithm methodology utilizes the brute force idea, adding some modifications for better performance.

Assuming G is the symmetric graph, the algorithm pseudo-code is:

1-     Using G, create a minimum cost-spanning tree named T.

Minimum spanning trees can be set up in more than one way. This paper illustrates Kruskal’s Minimum Spanning Tree algorithm to create minimum-spanning trees.

1- Sort all the edges in the graph in a non-decreasing order according to the weight.

2- Checking if the edge will complete a cycle between the nodes is necessary before inserting. If so, it should be ignored. Otherwise, include the border.

3- Repeat step number two until the tree has several edges equal to the number of vertices minus one.

         (Kruskal’s Minimum Spanning Tree (MST) Algorithm - GeeksforGeeks, 2023)

2- Then, search for all odd-degree vertices, which are vertices connected to an odd number of edges, putting them in one set, O.

3- Compute minimum-cost perfect matching M on O. This method ensures having a unique connection to each vertex in the set. When adding such connections to T, we get a new Eulerian graph.

(Chase et al., n.d.) (Chang, 2019) A picture containing child art, line, diagram

Description automatically generated

([link](https://bochang.me/blog/posts/tsp/))

**Analysis of Christofides algorithm:**

·    It has a complexity, which is considerably better than the traditional brute force.

·    This Eulerian graph has an approximation of the optimal cost by a factor. This means it does not find the optimal solution but a superb approximation.

A picture containing text, line, font, receipt

Description automatically generated                                                   **(**[**link**](https://scholarworks.calstate.edu/downloads/xg94hr81q)**)**

**Network design:**

Network designs are crucial in ensuring efficient and cost-effective communication between various elements within a system (Tanenbaum et al., 2011). Whether it's a computer, telecommunications, or transportation network, minimizing costs while maintaining optimal connectivity is a fundamental objective. One powerful tool that aids in achieving this goal is the concept of minimum cost-spanning trees (MCSTs).

MCSTs are widely used in network design to minimize the overall cost of connecting different nodes or vertices (Cormen et al., 2009). The process involves finding a tree that spans all the nodes in the network while minimizing the total cost of connecting them. This concept is precious when cost considerations are vital, such as designing business communication networks, optimizing transportation system routing paths, or organizing infrastructure in smart cities.

To apply MCSTs in network design, the first step is clearly defining the problem. This involves identifying the nodes or vertices that need to be connected within the network and assigning a cost or weight to each potential connection (Kurose et al., 2017). These costs can represent various factors, such as distance, bandwidth, latency, reliability, or any other relevant metric that impacts network performance. For example, in a telecommunications network, the cost may be associated with the distance between two nodes or the bandwidth for data transmission.

Once the problem is defined, the network is represented as a graph. In this graph representation, each node corresponds to a network element, such as routers, switches, servers, or endpoints (Tanenbaum et al., 2011). The graph's edges represent the connections between these nodes, with their weights reflecting the associated costs. The graph may be directed or undirected, depending on the nature of the network and the desired connections.

With the graph constructed, the next step is to apply an MCST algorithm to find the minimum cost-spanning tree. Various algorithms, such as Kruskal's or Prim's, can be used for this purpose (Cormen et al., 2009). These algorithms systematically evaluate the various connections and select the ones with the lowest costs, ensuring no cycles in the resulting tree. The algorithm continues adding edges until all nodes are connected, guaranteeing an optimal solution in terms of price.

The resulting minimum cost-spanning tree represents an efficient and cost-effective network design. By utilizing this tree, network designers can establish the most economical connections between network elements (Kurose et al., 2017). This approach reduces expenses associated with infrastructure, maintenance, and operational costs. For example, in a computer network, the MCST can help determine the optimal placement of switches and routers, reducing the amount of cabling required and minimizing power consumption.

Moreover, MCSTs offer scalability benefits. As the network grows or changes, the minimum cost-spanning tree can be updated to accommodate new nodes or modify existing connections while maintaining cost efficiency. This adaptability is particularly valuable in dynamic environments where network topologies evolve.

In addition to cost optimization, MCSTs provide other advantages in network design. Minimizing the overall cost help enhance network performance, reduce congestion, and improve reliability (Kurose et al., 2017). The efficient allocation of resources facilitated by MCSTs ensures that critical nodes are well-connected, leading to faster data transmission, lower latency, and improved overall network quality.

In conclusion, minimum cost-spanning trees (MCSTs) offer a powerful approach to network design, enabling the creation of efficient and cost-effective network layouts, by leveraging MCST algorithms.

A picture containing diagram, line, circle, pattern

Description automatically generated

(<https://rb.gy/gwdpm>)

After Discussing the applications and how beneficial they are, analysis of the algorithms can be generated.

Number of edges vs time in microseconds.

|  |  |  |
| --- | --- | --- |
| Number of edges | Kruskal’s algorithm | Prim algorithm |
| 4 | 4774 | 2352 |
| 6 | 4683 | 3685 |
| 8 | 4016 | 6803 |

**Analysis:**

1. Prim algorithm is better with smaller number of edges.
2. Kruskal’s algorithm is better with larger number of nodes as it has almost the same time when varying the number of nodes.
3. Graphs do not have the uniqueness property in terms of minimum spanning trees. In other words, when trying the same graphs (as shown in the appendix), different trees were generated from Kruskal’s algorithm and prim algorithm.

***Conclusion:***

# In conclusion, it has been granted that minimum-spanning trees are extremely useful. This was shown by describing how beneficial they are in designing networks, parallel processing, and optimizing time and length. Optimizing time and distance problems utilize Christofides algorithm, Kruskal’s Minimum Spanning Tree algorithm, and perfect matching algorithms. It is impressive how all these algorithms cooperate to reach a final solution. By finding excellent approximations to such problems, many processes become faster and storage sizes get reduced significantly. Minimum cost spanning trees are used also in constructing networks — the fundamental building unit of technology. Their main role is minimizing fees whilst ensuring the most advantageous connectivity. It is also vital in parallel and distributed computing which have revolutionized the field of graph algorithms, particularly in the computation of Minimum Cost Spanning Trees (MCSTs) for large-scale graphs and hold potential solutions for quantum computing. Therefore, it can be said that MST is not only a theoretically interesting topic, but also a practically thought-provoking issue. Minimum cost spanning trees are being constantly developed. Future wise, they will be able to approximate problems within a better range α. This will further enhance people’s lives that hugely depend on communication and optimization aspects. Moreover, developments in such a field include more time efficient algorithms, fostering many technological branches.

**Appendix:**

**Kruskal’s algorithm:**

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with low confidence

# Boruvka’s algorithm:

# 

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with medium confidence

([Link](https://www.geeksforgeeks.org/boruvkas-algorithm-greedy-algo-9/))

Prim Algorithm:

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

([Link](https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/))

Execution time of minimum spanning with different number of edges:

A screenshot of a computer error

Description automatically generated with low confidence

**A screen shot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screen shot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screen shot of a computer

Description automatically generated with medium confidence**

***References:***

Bergantiños, G., & Vidal-Puga, J. (2011). The folk solution and Boruvka’s algorithm in minimum cost spanning tree problems. *Discrete Applied Mathematics*, *159*(12), 1279–1283. <https://doi.org/10.1016/j.dam.2011.04.017>

Chai, W., & Lazar, I. (2021, September 16). *What is Telecommunications (Telecom)? definition from searchnetworking?*. Networking. <https://www.techtarget.com/searchnetworking/definition/telecommunications-telecom>

Chang, J., Luo, J., Huang, J. Z., Feng, S., & Fan, J. (2010). Minimum spanning tree based classification model for massive data with mapreduce implementation. *2010 IEEE International Conference on Data Mining Workshops*. <https://doi.org/10.1109/icdmw.2010.14>

Chase, Wilder-Smith, Chen, & Neoh. (n.d.). An evaluation of the traveling salesman problem. <https://scholarworks.calstate.edu/downloads/xg94hr81q>

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT   Press. <https://mitpress.mit.edu/9780262533058/introduction-to-algorithms/>

GeeksforGeeks. (2023, March 28). *Prim’s algorithm for minimum spanning tree (MST)*. GeeksforGeeks. <https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/>

GeeksforGeeks. (2023, March 17). *Boruvka’s algorithm: Greedy algo-9*. GeeksforGeeks. <https://www.geeksforgeeks.org/boruvkas-algorithm-greedy-algo-9/>

Kurose, J. F., & Ross, K. W. (2017). Computer networking: A top-down approach (Seventh ed.). Pearson. <https://aucegypt.summon.serialssolutions.com/#!/search?bookMark=eNrjYmDJy89LZW bgtTS3MDA0Nja1NDE0MWBl4DIyMDQzMDCxNDfjZBB2zs8tKC1JLVLISy0pzy_Kz sxL52FgTUvMKU7lZWApKSpN5WZQdHMNcfbQTSxNTk2vLCiJT04sSczJT49PMrK wtDA2tzAzIkYNAIwmKFE>

Tanenbaum, A. S., & Wetherall, D. J. (2011). Computer networks. Boston: Pearson Education, Inc. [https://csc-knu.github.io/sys-prog/books/Andrew%20S.%20Tanenbaum%20- %20Computer%20Networks.pdf](https://csc-knu.github.io/sys-prog/books/Andrew%20S.%20Tanenbaum%20-%20Computer%20Networks.pdf)