Optimal Electric Network Distribution. A Case study in Caracolí, Colombia.

Samuel Rico, Felipe Gómez and Gregorio Bermúdez.

EAFIT University
Mathematical Engineering
Applied Sciences and Engineering Department

Abstract

In this article, a problem of linear programming and optimization in graphs is posed by the method of minimum spanning tree (MST). The graph to be optimized is an electrical network in Caracoli, Antioquia with a total distance of more or less 23 km. Three different algorithms were used to solve the minimum spanning tree problem. The algorithms used are Prim, Kruskal and Boruvka. The distance of the MST was reduced in about 10% of the original total distance. All three are Greedy algorithms that basically imply that they are optimizing at each instance or step of the algorithm.

***Key Words**: Minimum Spanning Tree, MST, Forest, Electrical Network, Prim Algorithm, Kruskal Algorithm, Boruvka Algorithm, Graph, Vertices, Edges, Caracolí, Antioquia, Colombia.*

Introduction

In today's interconnected world, reliable and efficient communication networks are essential for enabling seamless connectivity and information exchange. At the heart of these networks lies the intricate web of cabling that carries data, voice, and video signals across vast distances. Cable connections form the backbone of telecommunication infrastructure, facilitating communication between individuals, businesses, and institutions [1].

The purpose of cable connections in telecommunication networks is to establish robust and high-speed communication channels. These channels enable the transmission of various types of data, including internet traffic, phone calls, multimedia content, and other digital communications. By interconnecting devices and systems, cable connections empower individuals and organizations to access and share information, collaborate remotely, and conduct business operations efficiently [2].

In the context of Colombia, optimizing cable connection on utility poles is of paramount importance for enhancing the nation's communication infrastructure. By ensuring efficient and reliable cable connections, Colombia can foster economic growth, bridge the digital division between urban and rural areas, and improve access to essential services for all its citizens [3]. For these reasons, an electric distribution network in Caracolí, Antioquia was chosen. The newtork contains 170 poles and a total distance of cable of about 22987.262 meters, connecting the whole network.

A minimum spanning tree "is a graph consisting of the subset of edges which together connect all connected nodes, while minimizing the total sum of weights on the edges" [4]. It is desirable to have a minimum spanning tree in a electric network distribution, because it provides the most cost-effective way to connect all poles, reduce emissions and increase reliability over time [5, 6].

Mathematical approach

In order to find the minimum spanning tree of a graph, there are several mathematical programming forms to write the problem's formulation: Subtour Elimination formulation, cutset formulation or Martin's formulation. Each of these approaches are written differently but all of them are posed as linear programming problems. But regardless of the fact that the approach of each of them is different, it must be concluded that the minimum spanning tree is the same.

In our case, we chose to formulate the problem with the Subtour Elimination formulation. An approach that diminishes the cost of the minimum spanning tree until it finds the optimal solution. Nevertheless when it was tested with the graph of the electric network in a solver, this formulation was too expensive computationally and it was not possible to find the MST using this model [7]. Due to this we used the Kruskal's, Boruvka's and Prim's algorithms to find the MST for our case study and compare their outputs.

For a better understanding of the formulation, these are the descriptions of each of the elements:

- Let x_{ij} be defined as:

$$x_{ij} = \begin{cases} 1 & \text{if edge (i, j) is in the tree} \\ 0 & \text{otherwise} \end{cases}$$

- ϕ_{ij} = the cost of edge(i, j).
- n = number of vertex in the graph.
- E = the set of edges.
- V = the set of vertex.
- S = subset of V.

Now let's check the mathematical formulation:

$$\min_{x} \quad \sum_{(i,j)\in E} \phi_{ij} x_{ij}$$

$$\text{s.t.} \quad \begin{cases} \displaystyle\sum_{(i,j)\in E} x_{ij} = n - 1 \\ \displaystyle\sum_{(i,j)\in E: i,j\in S} x_{ij} \leq |S| - 1, \forall S \subset V, S \neq V, S \neq \emptyset \\ x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in E \end{cases}$$

Above is the Subtour Elimination formulation. It is a minimization problem that reduces the final cost of the minimum spanning tree of a graph. This formulation is based on the fact that MSTs have no simple cycles and have n - 1 edges. The first constraint is to ensure that the MST has exactly one edge less than the number of vertices in the graph, a characteristic of minimum spanning trees. The second constraint checks every possible subset (S) of the set of vertices (V), and makes sure that in the final tree there are no cycles between the nodes in S. Finally, the last constraint establishes that x_{ij} is a binary variable, and for every i and j in the set of edges, this variable will only be either 0 or 1, depending if it is or not in the tree. [8, 9, 10, 11, 12]

Methodologies

The electric network was given in a kml file, which helped to visualize the network via Google Earth. To be able to implement the algorithms, the exact location of each pole was extracted from the kml file, and stored in a Python Dataset.

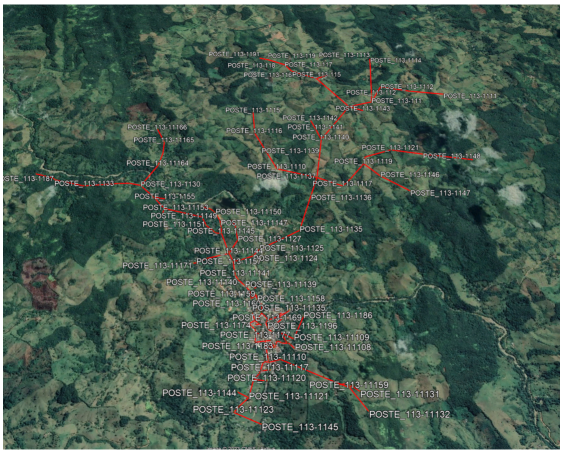

Figure 1: *Current Electrical Distribution Network in Caracolí*

In order to optimize the electrical network using the minimal spanning tree, the following three algorithms were implemented.

Kruskal's Algorithm

Kruskal's algorithm finds a minimum spanning tree of a given weighted graph. The algorithm can be implemented following the steps shown below:

1. Sort all edges of given graph $G = (V, E)$ in increasing order of their edge weights.
2. Set the expansion tree $T = (V, \emptyset)$.
3. For each edge (u, v), if $T \cup (u, v)$ doesn't form a cycle, then include (u, v) in T.
4. Return the Minimum Spanning Tree T.

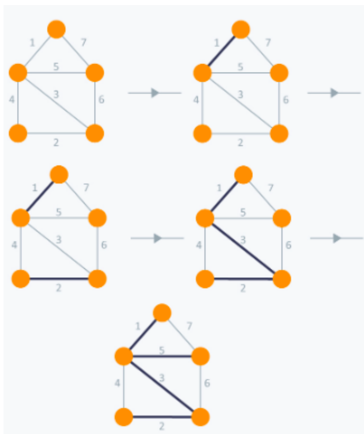

Figure 2: *Kruskal ´s Algorithm*

The algorithm stops when the minimum spanning tree is found. To find the minimum cost spanning tree, Kruskal's algorithm uses a greedy approach. This approach chooses the smallest weighted edge in each iteration without creating a cycle in the MST that has been constructed so far. [13, 14]

Boruvka's Algorithm

Like Kruskal's algorithm, Boruvka's algorithm is a Greedy algorithm and it's used to find the minimum cost spanning tree of a given weighted and undirected graph [15]. The steps to implement the algorithm are shown below:

1. The input is a connected, weighted and undirected graph.
2. Initialize all nodes (vertices) is individual components or sets.
3. Initialize the MST as an empty graph.
4. If there is more than one component in the original graph, then the algorithm continues as it follows:
    (a) Find the minimum cost edge that connects the component to any other component.
    (b) Add the chosen edge to the MST is has not been already added.
5. Return the minimum spanning tree.

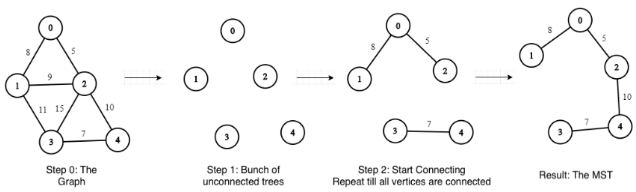

Figure 3: *Boruvka's Algorithm.*

Prim's Algorithm

Like Kruskal's algorithm and Boruvka's algorithm, Prim's algorithm is also a Greedy algorithm and it's used to find the minimum cost spanning tree of a given weighted and undirected graph. Prim's algorithm starts with a single node and moves to adjacent nodes in the graph in order to explore the all of the connected edges in the graph.

Similar to the Boruvka's algorithm, Prim's algorithm starts with an empty MST. This algorithm keeps track of two sets of vertices; one set containing the vertices in the MST and the other set containing the vertices that had not been used yet. At every iteration, the algorithm starts in one node and chooses the minimum weighted edge form this node to another in the set of vertices that have not been used yet. Then, it moves the chosen vertex to the set containing the edges used in the MST. This is called the cut in graph theory. At every step of Prim's algorithm, find a cut, pick the minimum weighted edge from the cut and then include the vertex in the set of the vertices used in the MST [16]. The steps to implement this algorithm are shown below.

1. Determine an arbitrary vertex as the starting node of the MST.
2. Find all edges connecting the vertex in the MST to a vertex that is not yet included in the MST.
3. Find the minimum weighted edge among all the edges found in the step before.
4. Add the edge to the MST if it does not form any cycle and the vertex to the set of vertices included in the MST.
5. Return the minimum spanning tree (MST).
    (a) Find the minimum cost edge that connects the component to any other component.
    (b) Add the chosen edge to the MST is has not been already added.
6. Return the minimum spanning tree.

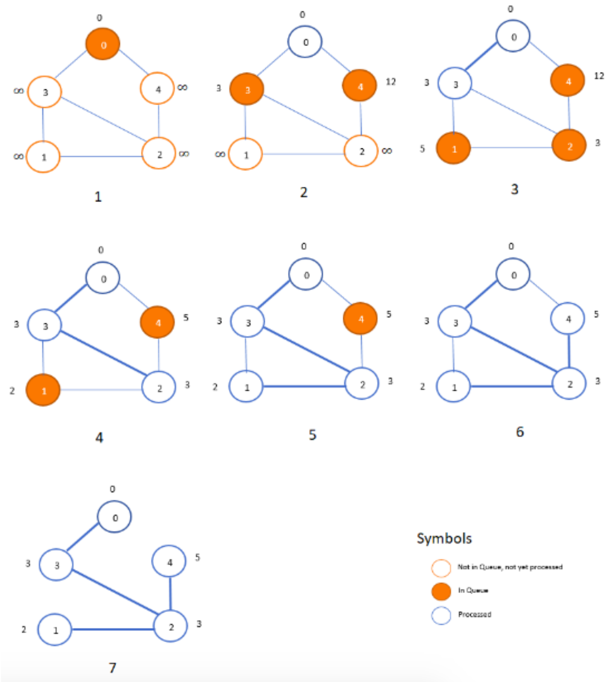

Figure 4: *Prim's Algorithm.*

Results Analysis

All the algorithms used in this project were implemented to find the minimum spanning tree of a given graph, in this case an electric network in Caracoli, Antioquia. We are aware that the minimum spanning tree did not take into account constraints such as maximum lengths of power lines, objects or structures between poles, and other possible factors that could affect the outcome of the algorithms. It was done in this way for the sake of time and ease of understanding how the algorithms mentioned previously work.

After implementing all three algorithms and applying them to the original problem, the distance of the minimum spanning tree was the same, with a precision up to 10 decimal units. The total distance of the MST is 20923.2739675691 meters. The original distance in the former electric network is 22987.261711137093 meters.

With these results, it can be concluded that the minimum spanning tree of the network was found, and that the existing electrical network is not the optimal one.

	Algorithms		
	Kruskal	Boruvka	Prim
Distance (m)	20923.278	20923.278	20923.278
Execution Time (ms)	159.92	33.38	12.31
Complexity	O(E(log(EV))	O(ElogV)	O(ElogV)

Table 1: *Algorithm's MST final distance, execution time, and complexity*

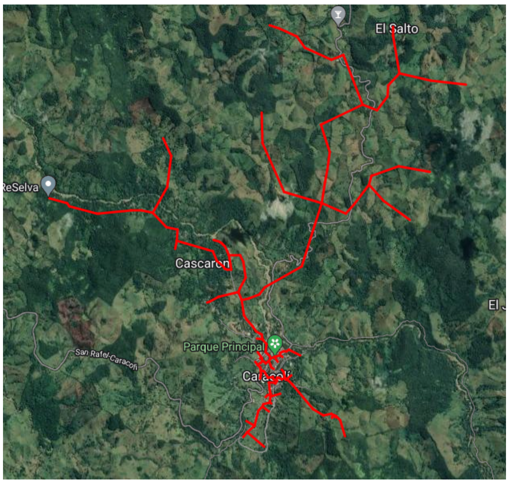

Figure 5: *Optimal Electical Distribution Network found*

Conclusion

In a country such as Colombia, it is very important to have optimal cable connection networks in order to improve communication and infrastructure development. Even so, it is common to find that wiring in rural territories is not optimal and sometimes even inefficient. This article explored the optimization of electric networks, with a specific focus on the Caracolí region in Antioquia. Through algorithmic analysis and application of optimization techniques, the optimal electric network for Caracolí was determined. The findings of this study highlighted the importance of optimizing electric networks to cut costs of constructions, ensure better information security and improve the data loss through cabling.

The optimal distribution of the electric network in Caracolí was obtained by finding the minimum spanning tree of the complete graph formed by the energy poles in the region. In order to obtain the MST of the network, algorithmic techniques such as Kruskal's, Boruvka's, and Prim's algorithms were utilized .

By employing these algorithmic approaches, the study aimed to identify the most efficient and cost-effective connections within the network, enabling the establishment of an optimized infrastructure. The application of Kruskal's, Boruvka's, and Prim's algorithms facilitated the identification of the minimum spanning tree, which effectively minimized the cost of construction while ensuring optimal data transmission and communication across the network.

When comparing the actual electrical network established in Caracoli with the optimal distribution found with the algorithmic approach, it can be seen that the savings in the amount of cable are high. The distance of the MST was a reduction of around 10% of the original network's distance, which is about 2063 meters. As mentioned above, cutting the distances of a cabling and communication network benefits both the cost of infrastructure and the security of the information flowing over the network.

The difference in distances between the current network and the optimal one, represents a big change in both cost and safety. However, this study was done simply by taking into account the distances between poles. This means that it is uncertain whether any constraints that were not related to distance were overlooked.

It is known that the electrical network is already built and it is somewhat difficult to change the configuration of its cables, but with this project the minimum spanning tree of any network can be obtained if the location of the poles is known.

References

[1] *Computer Network: Top Five Important Things You Need to Know*. https://www.dotcommagazine.com/2023/05/computer-network-top-five-important-things-you-need-to-know/. Accessed on May 21, 2023. DotCom Magazine, May 2023.

[2] Europacable. *Introduction to Telecommunication Cables*. Accessed on May 21, 2023. Mar. 2013. URL: https://europacable.eu/wp-content/uploads/2021/01/Europacable_Introduction_to_Telecommunication_Cables_20_March_2013.pdf.

[3] *Launch of Going Digital in Colombia Review - Bogota, October 2019*. https://www.oecd.org/about/secretary-general/launch-of-going-digital-in-colombia-review-bogota-october-2019.htm.

[4] *scipy.sparse.csgraph.minimum_spanning_tree*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.minimum_spanning_tree.html.

[5] *Todo lo que deberías saber sobre las redes inteligentes*. https://blogs.iadb.org/energia/es/todo-lo-que-deberias-saber-sobre-las-redes-inteligentes/.

[6] *Optimización del sistema de distribución de energía eléctrica mediante el algoritmo de Prim*. Accessed on May 21, 2023. URL: https://www.tesisenred.net/handle/10803/129631#page=11.

[7] Alex Brou. *Minimum Spanning Tree with Linear Programming*. https://github.com/AlexBrou/Minimum-Spanning-Tree-with-Linear-Programming/blob/master/main.py. Accessed on May 21, 2023.

[8] D. Glickenstein. *Graph Theory: Minimum Spanning Trees*. Accessed on May 21, 2023. 2014. URL: https://www.math.arizona.edu/~glickenstein/math443f14/golari.pdf.

[9] D. Glickenstein. *Graph Theory Course Materials*. Accessed on May 21, 2023. URL: https://www.math.arizona.edu/~glickenstein/math443f14/.

[10] *Introduction to Minimum Spanning Trees*. https://www.youtube.com/watch?v=8YsaHENoaRo&list=PL9NlTZRdFADfOvq26T-5SMXsCKhfNyy05&index=4.

[11] *An Efficient Mixed Integer Linear Programming Model for the Minimum Spanning Tree Problem*. https://www.mdpi.com/2227-7390/6/10/183.

[12] *Minimum Spanning Tree: Linear Programming Formulation*. http://www.columbia.edu/~cs2035/courses/ieor6614.S16/mst-lp.pdf. Accessed on May 21, 2023. Columbia University.

[13] *Kruskal's Minimum Spanning Tree Algorithm - Greedy Algo 2*. https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/.

[14] *Spanning Trees*. https://www.tutorialspoint.com/discrete_mathematics/discrete_mathematics_spanning_trees.htm.

[15] *Boruvka's Algorithm - Greedy Algo 9*. https://www.geeksforgeeks.org/boruvkas-algorithm-greedy-algo-9/.

[16] *Prim's Minimum Spanning Tree (MST) - Greedy Algo 5*. https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/.