Research Paper

"Minimal Spanning Tree and Prim's Algorithm"

Aftab Alam Masjidi

MAT-253: Discrete Mathematics II

Barry University

Professor Singh

October 18, 2025

Research Proposal

Title: *Minimal Spanning Tree and Prim's Algorithm*

Student: Aftab Alam Masjidi

Introduction and Motivation

I plan to learn the topic ***of Minimal Spanning Tree (MST) and Prim's Algorithm in Discrete Mathematics***. Spanning tree theory is one of the most elegant and practical applications of graph theory. A spanning tree connects all the vertices in a weighted graph using the minimum total edge weight possible. Prim's Algorithm is one of the most efficient and basic methods to compute such a tree, and it shows how mathematical optimization and algorithmic thinking go hand in hand.

I chose this topic because it so beautifully demonstrates how abstract mathematical ideas can lead to real-world relevance; especially in network design, computer science, and operations research. The topic connects theory to implementation and shows how mathematical simplicity leads to algorithmic power.

Research Objectives

- Define and discuss the mathematical concept of the Minimal Spanning Tree (MST) and its fundamental properties.
- Analyze Prim's Algorithm step-by-step, including its proof of correctness and computational complexity.
- Compare Prim's Algorithm with Kruskal's Algorithm and understand where each performs best.
- Explore key theorems (cut and cycle properties) that guarantee MST optimality.
- Investigate modern applications of MSTs in data analysis, AI, and communication systems.

Mathematical Focus Areas

- Graph theory, combinatorial optimization, algorithmic design and analysis, complexity theory, and mathematical proofs involving weighted graphs.

Expected Outcomes

- A clear, formal understanding of the mathematical properties of MSTs and Prim's Algorithm; practical applications of these concepts; and an organized presentation of theorems, proofs, and algorithmic analysis leading to an 8-page scholarly research paper.

References

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.
- Prim, R. C. (1957). Shortest connection networks and some generalizations. Bell System Technical Journal, 36(6), 1389–1401.
- Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, 7(1), 48–50.

“**Minimal Spanning Tree and Prim’s Algorithm**”

Abstract

A Minimum Spanning Tree (MST) of a connected, weighted graph is a spanning subgraph of minimum total edge weight. Among the numerous algorithms for constructing an MST, Prim's Algorithm remains one of the most widely used due to its simplicity and efficiency. It is a greedy algorithm, repeatedly choosing the minimum edge that expands the tree. This essay discusses MSTs' and Prim's Algorithm's mathematical foundations, correctness proofs, computational complexity, and real-world applications. Algorithm comparisons and future prospects such as distributed and quantum computing implementations are also taken into account.

Keywords: Graph Theory, Prim’s Algorithm, Minimal Spanning Tree, Optimization, Combinatorics, Algorithms

MSC 2010 Subject Classification: 05C85 (Graph algorithms), 68R10 (Graph theory), 68W05 (Nondeterministic algorithms).

Introduction

Graphs are a basic construct in discrete mathematics. They model relationships among objects in the real world, e.g., cities in a road network or computers in a network. Often, we would like to find the most economical way to connect all the points with minimum total cost. This leads to the Minimal Spanning Tree problem.

An MST of a connected, weighted graph ***G= (V, E)*** is a subset of edges that connect all vertices without cycles and with minimum total edge weight. Prim's Algorithm provides a simple greedy solution to compute the MST. It begins at any vertex and adds repeatedly the minimum-weight edge that expands the current tree.

I chose this topic because it lies at the intersection of computational efficiency and mathematical rigor. It shows how locally optimal choices lead to globally optimal solutions, a frequent paradigm in optimization and computer science.

Background

Let $G = (V, E)$be a connected, weighted graph where each edge $e = (u, v)$has an associated weight $w(u, v) \geq 0$. A **spanning tree** is an acyclic subset of edges that connects all vertices in V. Among all spanning trees, an **MST** minimizes the total sum of edge weights.

$$\text{Find } T \subseteq E \text{ such that } T \text{ is a tree and } \sum_{(u,v)\in T} w(u, v) \text{ is minimal.}$$

Two key principles guide MST formation:

1. **Cut Property:** For any partition (cut) of vertices, the smallest edge crossing the cut must belong to the MST.

2. **Cycle Property:** In any cycle, the largest edge cannot be part of the MST.

These properties justify the greedy selection process used by both **Prim's** and **Kruskal's Algorithms**.

Mathematical Formulation and Algorithm

Prim's Algorithm Steps:

1. Choose any starting vertex v_0.
2. Initialize the MST as empty.
3. At each step, find the smallest-weight edge connecting a vertex in the MST to a vertex outside it.
4. Add this edge to the MST.
5. Repeat until all vertices are included.

Proof of Correctness

Prim's Algorithm meets the cut property in each iteration. Suppose a partial MST T. The minimum edge crossing the cut between T and V−T must be in the MST since any larger edge would contradict minimality. By induction, the algorithm keeps a tree that grows but remains part of some MST, which guarantees correctness.

Time Complexity

- Using an adjacent matrix: $O(V^2)$
- Using adjacency lists with a min-heap: $O(E\log V)$

The heap-based implementation is more efficient for dense graphs and forms the basis for modern MST algorithms.

Examples

Consider the graph $G = (V, E)$with vertices

$$V = \{A, B, C, D, E\}$$

and edges:

A–B (2), A–C (3), B–C (1), B–D (4), C–E (5), D–E (2).

Applying Prim's Algorithm (starting from A):

Step	*Action*	*Edge Chosen*	*Current MST*
1	Start	–	A
2	Add smallest edge	A–B (2)	{A–B}
3	Next smallest edge	B–C (1)	{A–B, B–C}
4	Add next	B–D (4)	{A–B, B–C, B–D}
5	Final edge	D–E (2)	{A–B, B–C, B–D, D–E}

Total weight: 9

Thus, the MST contains edges {A–B, B–C, B–D, D–E}, connecting all vertices with the minimal total cost.

Comparison with Kruskal's Algorithm

While Prim's Algorithm grows a single tree, **Kruskal's Algorithm** sorts edges and adds them one by one, skipping those that create cycles.

Feature	*Prim's Algorithm*	*Kruskal's Algorithm*
Approach	Expands one tree	Builds forest, merges components
Data structure	Priority queue (heap)	Union–Find (disjoint sets)
Best for	Dense graphs	Sparse graphs
Complexity	$O(E\log V)$	$O(E\log E)$

Both algorithms produce identical MSTs but differ in their strategies and performance trade-offs.

Applications and Future Directions

MSTs and Prim's Algorithm are used across diverse fields:

- **Network Design:** Constructing minimal-cost electrical or communication networks.
- **Data Clustering:** Forming natural groupings in data (single-linkage clustering).
- **Transportation:** Designing efficient road, rail, or airline networks.
- **Image Processing:** Simplifying and segmenting visual data.

Future research focuses on:

- **Parallel and Distributed MST algorithms** for massive datasets.
- **Dynamic MST updates** for changing networks.
- **Quantum MST computation** using emerging quantum-optimized graph algorithms.

Conclusion

The Minimum Spanning Tree problem is among the most elegant outcomes of graph theory. Prim's Algorithm, through its simplicity and efficiency, illustrates how making local decisions found on mathematical insights can give rise to global optimality. The MST continues to be a cornerstone of optimization, network design, and theoretical computer science, a bridge between discrete mathematics and computational practice.

Bibliography

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal, 36*(6), 1389–1401.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society, 7*(1), 48–50.

Tarjan, R. E. (1983). *Data Structures and Network Algorithms.* SIAM.

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications.