Smart Optimization of Fiber Optic Network Design Using Prim-Dijkstra Algorithm

Emerson D. Cumlat Jr.
Access Planning & Engineering
Telecom Fiji Limited
Suva, Fiji Islands
emerson.cumlat@telecom.com.fj

Marloun P. Sejera
School of Electrical, Electronics &
Computer Engineering
Mapua University
Manila, Philippines
mpsejera@mapua.edu.ph

***Abstract*—The increasing demand for high-speed and reliable communication infrastructure has intensified the need for efficient fiber optic network design. This study presents a smart optimization approach that integrates Prim's and Dijkstra's algorithms to enhance the planning and deployment of fiber optic networks. The proposed hybrid algorithm leverages Prim's algorithm for constructing a minimum spanning tree (MST) to ensure cost-effective backbone layout, while Dijkstra's algorithm is employed to determine the shortest paths for optimal routing. The system is implemented using a custom simulation environment that models real-world urban topologies. Results demonstrate significant improvements in network efficiency, reduced total cable length, and minimized latency compared to traditional design methods. This approach offers a scalable and intelligent solution for next-generation fiber optic infrastructure planning, particularly in smart city applications.**

Keywords — *Fiber Optic Network, Prim's Algorithm, Dijkstra's Algorithm, Network Optimization, Smart Cities, Minimum Spanning Tree, Shortest Path*

I. INTRODUCTION

In today's hyper-connected world, fiber optic networks are the backbone of modern telecommunications, offering unmatched speed, reliability, and bandwidth. These networks have transformed how data is transmitted, enabling everything from high-speed internet to cloud computing. A recent example is PLDT's rollout of its Hyper-Gig fiber service, which underscores how commercial fiber offerings are intensifying competition and reshaping the telecom landscape [1] .

To improve the design and deployment of fiber networks, researchers have explored various optimization techniques. Traditional heuristic methods, such as those proposed by Smith, improved deployment efficiency but struggled with scalability [2]. More advanced approaches, including genetic algorithms and particle swarm optimization, have shown promise in reducing costs and fiber length [3]. Gonzales, for instance, applied Dijkstra's algorithm to enhance routing performance [4]. However, many of these studies focus on single-objective outcomes. Addressing this gap, a hybrid approach that combines Prim's and Dijkstra's algorithms has been proposed to optimize both cost and performance in fiber network design.

This study applies that hybrid optimization framework to urban and semi-urban environments, such as Metro Manila's central business districts. It aims to enhance route planning, reduce deployment costs, and improve scalability using algorithmic techniques. While the study does not cover next-generation technologies or rural deployments, it contributes to sustainable infrastructure development and aligns with global goals for innovation and urban connectivity. The findings offer a practical, data-driven model for telecom providers navigating the complexities of modern network planning.

II. RELATED WORKS

A. Copper to Fiber Optics Network Transition

The transition from copper-based to fiber-optic networks marks a significant technological advancement in the telecommunications industry. Traditionally, copper and coaxial cables were used for data transmission and internet services. While effective for lower bandwidth needs, these cables struggle to meet the high-speed demands of modern applications

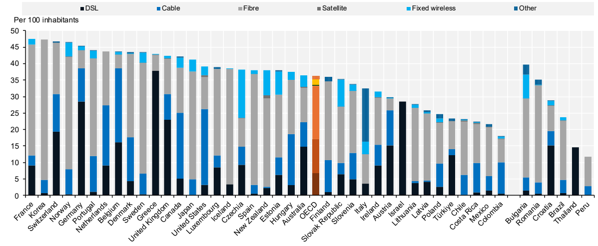


Figure 1: OECD Broadband Statistics Update June 2024 [5]

Fiber-optic technology emerged as a far superior alternative, offering significantly higher bandwidth and faster data transmission speeds. Unlike copper cables, which rely on electrical signals, fiber-optic cables use light to transmit data. This key difference enables fiber networks to handle much larger data volumes over longer distances with minimal signal loss or degradation. [6]

B. Basic Steps Implementing Fiber Network

To ensure long-term stability and optimal performance of fiber optic networks, operators begin with a thorough planning phase that encompasses key decisions regarding communication protocols, geographic layout, and transmission equipment. This foundational stage involves defining the network's infrastructure, selecting appropriate hardware, and outlining the services to be delivered. Effective planning requires collaboration among stakeholders to align with technical standards, regulatory frameworks, and business goals. Considerations such as permits, easements, and inspections are addressed early to facilitate smooth deployment. Geographic Information Systems (GIS) play a vital role in this phase, enabling precise mapping of routes and infrastructure, which enhances cost-efficiency and operational transparency.

Following initial planning, operators conduct detailed assessments including site surveys and feasibility studies to evaluate environmental and logistical constraints. These insights inform the design of network topology, guiding decisions on cable routing, equipment placement, and redundancy strategies. Equipment selection is based on compatibility, scalability, and performance metrics, while infrastructure planning ensures reliable power supply and environmental controls. The final stages involve documenting the network architecture, installing components, and performing rigorous testing to validate connectivity and performance. Continuous monitoring and troubleshooting are essential to maintain network reliability and support future upgrades.

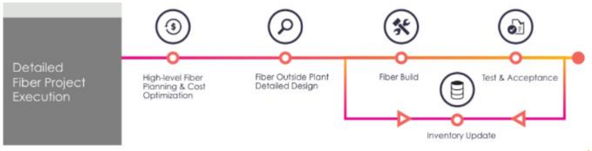


Figure 2: Steps in Implementing Fiber Network

C. Prim's Algorithm

Prim's algorithm is a fundamental technique used to determine the minimum spanning tree (MST) of a connected, weighted graph, ensuring that all vertices are connected with the least total edge weight and without forming cycles. It begins with an arbitrary starting node and iteratively adds the smallest edge that connects a new vertex to the growing tree, avoiding any closed loops. This step-by-step process—starting from selecting an initial vertex, identifying the minimum edge, and expanding the tree—ensures that the most cost-effective connections are made at each stage. The algorithm is especially useful in network design applications, such as cable television or fiber optic networks, where minimizing installation costs and ensuring efficient connectivity are critical. By consistently choosing the lightest available edge, Prim's algorithm constructs an optimal network layout that balances performance and cost-effectiveness. [7].

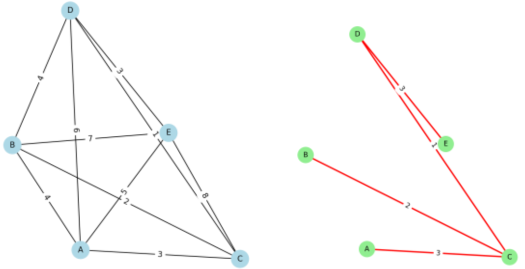


Figure 3: Implementation of Prim's Algorithm

D. Djikstra Algorithm

Dijkstra's Shortest-Path Algorithm is a foundational method in graph theory used to compute the shortest paths from a single source node to all other nodes in a weighted graph, making it highly applicable in areas such as network routing and geographic navigation. The algorithm operates systematically by selecting the node with the smallest known distance, updating the distances of its adjacent nodes, and marking it as processed, thereby constructing a shortest-path tree [8]. This approach is particularly effective in fiber optic network design, where identifying the most efficient routing paths can significantly enhance performance and reduce latency [8]

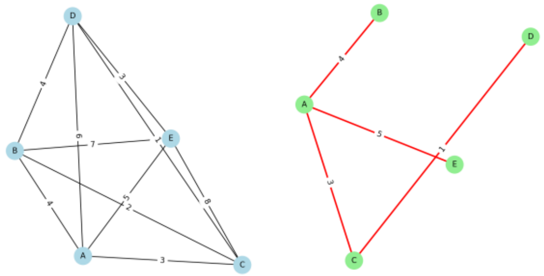


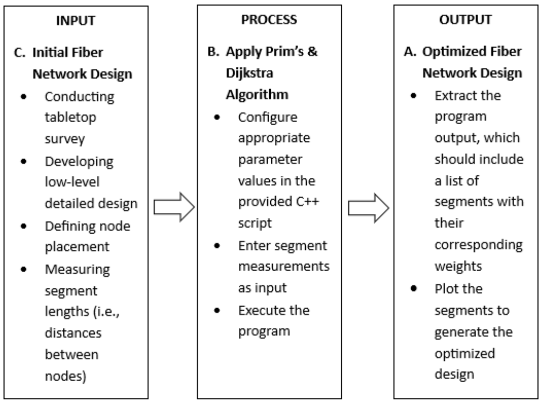
Figure 4: Implementation of Dijkstra Algorithm

E. .Optimized Parameters in Fiber Network Design

In fiber optic network design, optimizing key parameters is essential to achieving both cost-efficiency and high performance. These parameters include route selection, cable length minimization, equipment placement, and redundancy planning. Designers use algorithms and modeling tools to determine the most efficient paths for fiber deployment, reducing trenching and material costs while ensuring robust connectivity. Additionally, considerations such as signal attenuation, bandwidth capacity, and scalability are factored into the design to support current and future data demands. By carefully balancing technical requirements with economic constraints, network planners can create resilient and cost-effective fiber infrastructures that meet service quality expectations and operational goals.

III. PROPOSED METHOD

The framework illustrates the structured approach taken to develop the smart optimization of fiber network design using Prim's and Dijkstra's algorithms. The Input phase includes network requirements, node data, and segment measurements. The Process involves applying algorithm-based logic through a custom C++ program, which calculates optimized paths and network connections. Finally, the Output yields an efficient, cost-effective fiber layout, minimizing redundant paths and optimizing resource use. This framework highlights the systematic flow of data and operations that led to the development of the optimized fiber network design.



INPUT

C. Initial Fiber Network Design
- Conducting tabletop survey
- Developing low-level detailed design
- Defining node placement
- Measuring segment lengths (i.e., distances between nodes)

PROCESS

B. Apply Prim's & Dijkstra Algorithm
- Configure appropriate parameter values in the provided C++ script
- Enter segment measurements as input
- Execute the program

OUTPUT

A. Optimized Fiber Network Design
- Extract the program output, which should include a list of segments with their corresponding weights
- Plot the segments to generate the optimized design

Figure 5: Conceptual Framework

A. Case Study Description

This scenario outlines the strategic design and deployment of a cost-effective and high-performance fiber optic network intended to connect 100 Fiber Network Access Points for an Internet Service Provider operating in Pasig City, specifically within Rosario and Maybunga Village. Each access point is equipped with a 16-way FTTH splitter, capable of serving up to 16 residential or commercial users with high-speed internet. These nodes are carefully positioned to maximize coverage within optimal distances and leverage existing utility poles for mounting and cable support. The primary goal is to minimize the total cost of fiber cable installation while ensuring full interconnectivity and efficient data transmission across all nodes. A graph-based model illustrates the network layout, where each node represents a connection point and each edge indicates a potential fiber link, weighted by cost or distance.

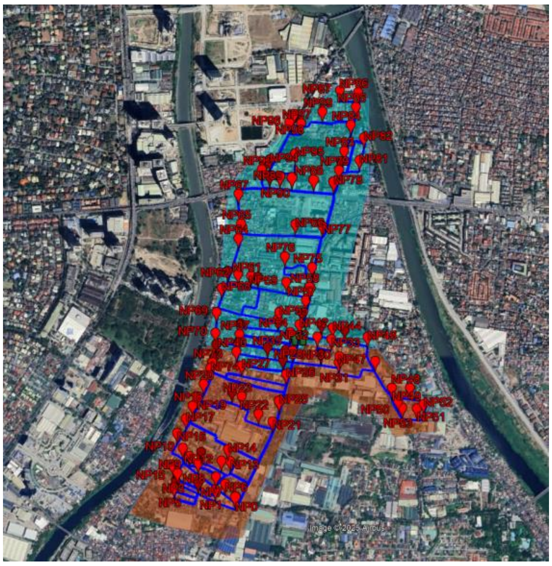


Figure 6: Fiber Optic Network Design with 100 Nodes

B. Equipment Specifications for Deploying 100 Nodes

The equipment specifications for deploying 100 Network Access Point (NAP) nodes and the associated Multi-Service Access Node (MSAN) cabinet requirements are comprehensively detailed. Each NAP is defined as an outdoor enclosure that houses optical splitters and termination points, typically serving 8 to 16 premises, with robust weatherproofing, splicing trays, SC/APC adapters, and cable management features. For efficient data aggregation, the MSAN cabinet is specified to accommodate multiple fiber distribution panels, power supplies, and network interface modules, supporting GPON or XGS-PON architectures. The MSAN is designed with scalability in mind, offering support for up to 2,000 subscribers, ensuring flexibility for current and future demand. These specifications align with the FTTH deployment standards and ensure reliable connectivity, signal integrity, and operational safety across the entire access network.

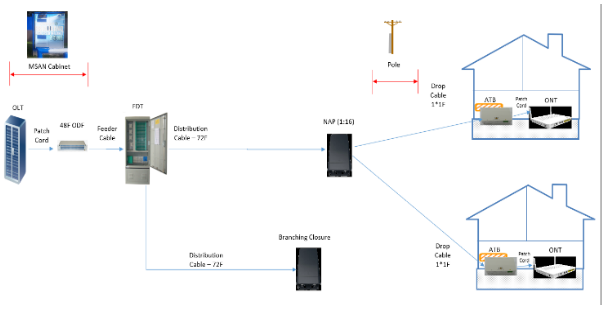
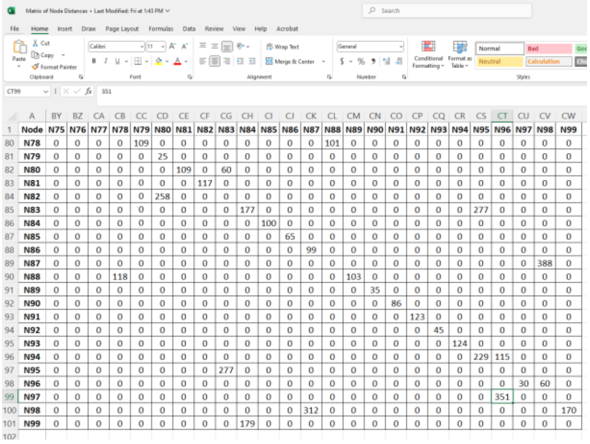


Figure 7: Fiber Network Components

C. Input of Edge Weights

Using Microsoft Excel, we capture and organize the distances between all defined nodes within the network. The matrix includes embedded formulas that were designed to be easily translated into a C++ script, allowing seamless integration between data preparation and algorithm implementation. Once the distance matrix is set up, the weights of each segment—representing the actual link cost between node pairs—are computed by the program and exported back into the same Excel file. This structured approach ensures accuracy and efficiency in managing network parameters for route optimization.



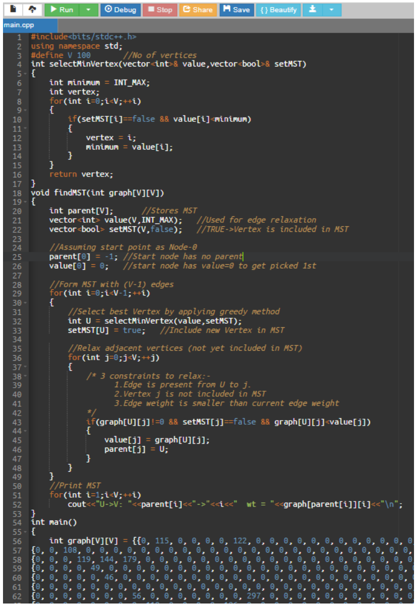
	A	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW
1	Node	N75	N76	N77	N78	N79	N80	N81	N82	N83	N84	N85	N86	N87	N88	N89	N90	N91	N92	N93	N94	N95	N96	N97	N98	N99
80	N78	0	0	0	0	109	0	0	0	0	0	0	0	0	101	0	0	0	0	0	0	0	0	0	0	0
81	N79	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	N80	0	0	0	0	0	0	109	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
83	N81	0	0	0	0	0	0	0	117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	N82	0	0	0	0	0	258	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	N83	0	0	0	0	0	0	0	0	0	177	0	0	0	0	0	0	0	0	0	0	277	0	0	0	0
86	N84	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
87	N85	0	0	0	0	0	0	0	0	0	0	0	65	0	0	0	0	0	0	0	0	0	0	0	0	0
88	N86	0	0	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0
89	N87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	388	0
90	N88	0	0	0	118	0	0	0	0	0	0	0	0	0	0	103	0	0	0	0	0	0	0	0	0	0
91	N89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0	0	0	0	0	0	0
92	N90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	0	0	0	0	0	0	0	0
93	N91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	0	0	0	0	0	0
94	N92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0
95	N93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	0	0	0	0	0
96	N94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	229	115	0	0	0
97	N95	0	0	0	0	0	0	0	0	277	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
98	N96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	60	0
99	N97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	351	0	0	0
100	N98	0	0	0	0	0	0	0	0	0	0	0	0	312	0	0	0	0	0	0	0	0	0	0	0	170
101	N99	0	0	0	0	0	0	0	0	0	179	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
102																										

Figure 8: Matrix of Node Distances

For instance, the distance between Node 80 and Node 81 is 109 units, shown in the matrix at Row 82 and Column CE, while a zero in other cells indicates no direct connection between those nodes. A formula is applied in a specific column to concatenate all the required data. This concatenated output is then copied and used to generate a C++ script essential for running our program.

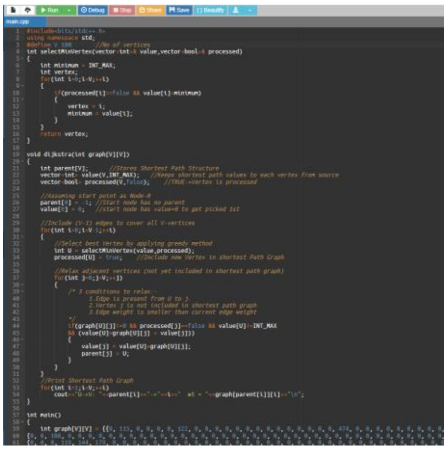
D. Implement the Prim-Djikstra Algorithm

We will implement the algorithm using a C++ script, which will automatically compute the optimal solution based on the defined parameters and input data. The expected output is a set of segments, each with its own length, which collectively indicate the route chosen by the algorithm.



```cpp
#include<bits/stdc++.h>
using namespace std;
#define V 100        //No of vertices
int selectMinVertex(vector<int>& value,vector<bool>& setMST)
{
    int minimum = INT_MAX;
    int vertex;
    for(int i=0;i<V;++i)
    {
        if(setMST[i]==false && value[i]<minimum)
        {
            vertex = i;
            minimum = value[i];
        }
    }
    return vertex;
}
void findMST(int graph[V][V])
{
    int parent[V];       //Stores MST
    vector<int> value(V,INT_MAX);   //Used for edge relaxation
    vector<bool> setMST(V,false);   //TRUE->Vertex is included in MST

    //Assuming start point as Node-0
    parent[0] = -1; //Start node has no parent
    value[0] = 0;   //start node has value=0 to get picked 1st

    //Form MST with (V-1) edges
    for(int i=0;i<V-1;++i)
    {
        //Select best Vertex by applying greedy method
        int U = selectMinVertex(value,setMST);
        setMST[U] = true;   //Include new Vertex in MST

        //Relax adjacent vertices (not yet included in MST)
        for(int j=0;j<V;++j)
        {
            /* 3 constraints to relax:-
                  1.Edge is present from U to j.
                  2.Vertex j is not included in MST
                  3.Edge weight is smaller than current edge weight
            */
            if(graph[U][j]!=0 && setMST[j]==false && graph[U][j]<value[j])
            {
                value[j] = graph[U][j];
                parent[j] = U;
            }
        }
    }
    //Print MST
    for(int i=1;i<V;++i)
        cout<<"U->V: "<<parent[i]<<"->"<<i<<"  wt = "<<graph[parent[i]][i]<<"\n";
}
int main()
{
    int graph[V][V] = {{0, 115, 0, 0, 0, 0, 122, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
{0, 0, 108, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
{0, 0, 0, 119, 144, 179, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
{0, 0, 0, 0, 49, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
{0, 0, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
{0, 0, 0, 0, 0, 0, 0, 56, 0, 0, 0, 0, 0, 0, 0, 297, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Figure 9: Prim's Algorithm in C++ Compiler

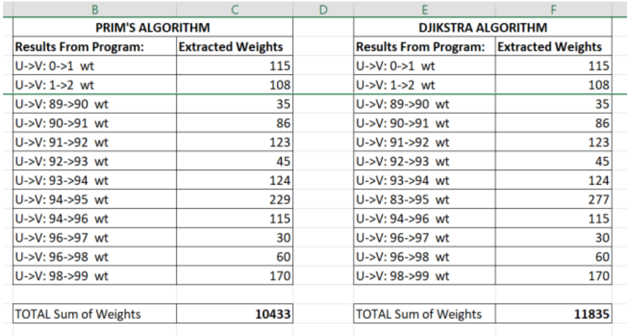


```cpp
#include<bits/stdc++.h>
using namespace std;
#define V 100        //No of vertices
int selectMinVertex(vector<int>& value,vector<bool>& processed)
{
    int minimum = INT_MAX;
    int vertex;
    for(int i=0;i<V;++i)
    {
        if(processed[i]==false && value[i]<minimum)
        {
            vertex = i;
            minimum = value[i];
        }
    }
    return vertex;
}

void dijkstra(int graph[V][V])
{
    int parent[V];       //Stores Shortest Path Structure
    vector<int> value(V,INT_MAX);   //Keeps shortest path values to each vertex from source
    vector<bool> processed(V,false);    //TRUE->Vertex is processed

    //Assuming start point as Node-0
    parent[0] = -1; //Start node has no parent
    value[0] = 0;   //start node has value=0 to get picked 1st

    //Include (V-1) edges to cover all V-vertices
    for(int i=0;i<V-1;++i)
    {
        //Select best Vertex by applying greedy method
        int U = selectMinVertex(value,processed);
        processed[U] = true;    //Include new Vertex in shortest Path Graph

        //Relax adjacent vertices (not yet included in shortest path graph)
        for(int j=0;j<V;++j)
        {
            /* 3 conditions to relax:-
                  1.Edge is present from U to j.
                  2.Vertex j is not included in shortest path graph
                  3.Edge weight is smaller than current edge weight
            */
            if(graph[U][j]!=0 && processed[j]==false && value[U]!=INT_MAX
            && (value[U]+graph[U][j] < value[j]))
            {
                value[j] = value[U]+graph[U][j];
                parent[j] = U;
            }
        }
    }
    //Print Shortest Path Graph
    for(int i=1;i<V;++i)
        cout<<"U->V: "<<parent[i]<<"->"<<i<<"  wt = "<<graph[parent[i]][i]<<"\n";
}

int main()
{
```

Figure 10: Dijkstra Algorithm in C++ Compiler

E. *Extract Results from Program*

We will extract the output generated by the program and overlay the results onto our fiber network design using a drawing software tool. In this case, we are utilizing Google Earth Pro to visually map the optimized fiber routes and access point locations, ensuring alignment with real-world geography and infrastructure.



PRIM'S ALGORITHM			DJIKSTRA ALGORITHM	
Results From Program:	Extracted Weights		Results From Program:	Extracted Weights
U->V: 0->1 wt	115		U->V: 0->1 wt	115
U->V: 1->2 wt	108		U->V: 1->2 wt	108
U->V: 89->90 wt	35		U->V: 89->90 wt	35
U->V: 90->91 wt	86		U->V: 90->91 wt	86
U->V: 91->92 wt	123		U->V: 91->92 wt	123
U->V: 92->93 wt	45		U->V: 92->93 wt	45
U->V: 93->94 wt	124		U->V: 93->94 wt	124
U->V: 94->95 wt	229		U->V: 83->95 wt	277
U->V: 94->96 wt	115		U->V: 94->96 wt	115
U->V: 96->97 wt	30		U->V: 96->97 wt	30
U->V: 96->98 wt	60		U->V: 96->98 wt	60
U->V: 98->99 wt	170		U->V: 98->99 wt	170
TOTAL Sum of Weights	**10433**		TOTAL Sum of Weights	**11835**

Figure 11: Extracted Results from the Program

Figure 12: Plotted results based from the algorithm output

IV. RESULTS & DISCUSSIONS

The results of this research demonstrate the effectiveness of combining Prim's Minimum Spanning Tree and Dijkstra's Shortest Path algorithms in designing a cost-efficient and high-performance fiber optic network.

A. *Visual Comparison of Cable Layout*

The optimized fiber network design significantly enhances cable dimensioning by leveraging intelligent algorithms such as Prim's and Dijkstra's. These techniques reduce total cable length while maintaining full connectivity, resulting in lower material costs and faster installation. Compared to manual methods, the optimized layout eliminates redundant paths, improves signal quality, and ensures network resilience through preserved redundancy links—all while serving the same number of users efficiently.

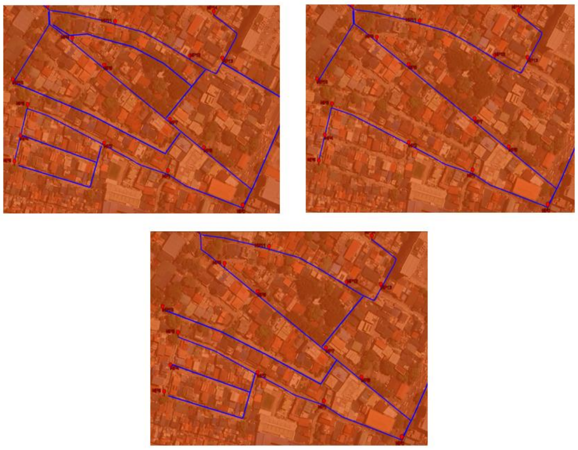


Figure 13: Visual Comparison of Cable Layout of the Three Design Solution Approach. Left-Heuristic; Right - Prim; Bottom - Djikstra

B. *Cost Reduction Analysis*

For this scenario, we assume a cost of PHP 650 per meter for laying the fiber optic cable. In comparing the heuristic approach and the optimized model using Prim's and Dijkstra's algorithms, a substantial cost saving is observed through reduced total fiber length and associated installation costs.

Table 1: Cost Comparison

Design Solution	Fiber Length	Total Costs
Heuristic Approach	15,648	PHP 10,171,200.00
Djikstra Algorithm	11,835	PHP 7,692,750.00
Prim's Algorithm	10,433	PHP 6,781,450.00

C. *Utility Pole Usage*

The analysis shows that applying Dijkstra's and Prim's algorithms to the fiber network design notably reduced both the number of cable segments and the required utility poles compared to the heuristic method.

Table 2: Pole Usage

Design Solution	No. of Segments	No. of Poles Used
Heuristic Approach	104	313 poles
Djikstra Algorithm	79	237 poles
Prim's Algorithm	70	209 poles

D. *Route Efficiency Improvements*

A significant outcome of this study is the measurable improvement in route efficiency achieved by applying algorithmic optimization techniques in the design of fiber optic networks. The implementation of Prim's Minimum Spanning Tree (MST) and Dijkstra's Shortest Path algorithms allowed for the generation of optimized routes that significantly reduced total fiber length and minimized redundant or inefficient cable paths.

Table 3: Route Efficiency Improvements

Design Solution	Fiber Length	Improvement
Heuristic Approach	15,648 m	0%
Djikstra Algorithm	11,835 m	24.37%
Prim's Algorithm	10,433 m	33.33%

E. *Link Loss Calculation*

There is a 0.15 dB difference in link loss between the heuristic and optimized fiber network designs. This indicates that applying optimization algorithms results in improved link performance, as the network follows a shorter and more efficient path. Although both calculated link losses remain within the acceptable threshold of below 28 dB, having a lower loss is advantageous. It reduces the likelihood of future maintenance and provides greater margin for operational flexibility and reliability.

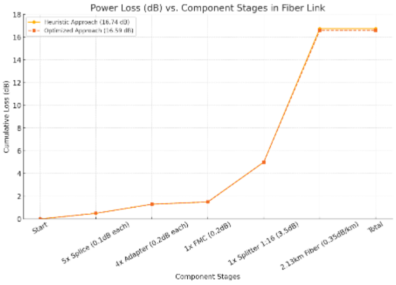


Figure 14: Power Loss Components

F. *Network Redundancy*

By applying Prim's and Dijkstra's algorithms to the fiber network design, redundancy is significantly enhanced through the strategic creation of loop or ring topologies, ensuring that each critical node has an alternative data path. This not only minimizes total fiber length and installation costs but also strengthens fault tolerance by enabling traffic rerouting during outages. Such smart optimization is especially beneficial in urban and semi-urban environments where high service reliability is essential.

Figure 15: Redundancy Path After Optimization

G. *Comparison for Heuristic and Algorithm Based Planning*

Overall, while manual and heuristic-based planning remains relevant for smaller or resource-limited projects, algorithm-based approaches provide superior efficiency, cost savings, and scalability. The integration of algorithm-driven methods is the preferred choice for large-scale and complex fiber deployments.

Parameter	Heuristic /Manual	Prim's Algorithm	Dijkstra's Algorithm
Redundancy	High	Moderate	Moderate
Propagation Delay (ms)	15	11	9
Link Loss (dB)	14.8	10.2	9.1
Number of Segments	120	82	94
Deployment Time (days)	60	40	45
Time to Design (hrs)	50	10	20
Household Coverage	2,000	2,000	2,000
Scalability	Low	High	High
Sustainability	Basic	Strong	Strong

V. CONCLUSION

This practicum research delves into advanced strategies for enhancing fiber optic network deployment, emphasizing the use of modern algorithms and innovative design approaches. By integrating cutting-edge methodologies, the study offers practical solutions that lower deployment costs while boosting network efficiency and scalability. It underscores the importance of aligning technological innovation with strategic planning to meet the increasing demand for high-speed internet and data-driven services. The research also traces the shift from traditional copper systems to fiber optics, which now form the backbone of telecommunications due to their superior performance in bandwidth, reliability, and security—key enablers for technologies like cloud computing, IoT, and high-definition streaming.

A central focus of the study is the application of graph theory, particularly Prim's Minimum Spanning Tree and Dijkstra's Shortest Path algorithms, to optimize fiber network design. A hybrid model—Prim-Dijkstra—is introduced to combine the strengths of both algorithms, addressing cost and performance simultaneously. This model was tested in a real-world scenario in Pasig City, Metro Manila, involving 100 network nodes. The results demonstrated significant cost savings and improved routing efficiency, with a total deployment cost of PHP 6,781,450.00. The study concludes by highlighting the practical value of algorithmic optimization in network planning and sets the stage for future research aimed at scaling these methods for broader, more complex deployments.

VI. RECOMMENDATIONS

Based on the findings, analysis, and limitations of this study, several key recommendations are proposed to enhance future fiber optic network design and implementation. Integrating real-world GIS data, utility maps, and environmental constraints into the optimization model will improve both the practicality and cost-efficiency of deployment. Expanding the algorithmic approach with metaheuristic techniques—such as Genetic Algorithms, Ant Colony Optimization, or Particle Swarm Optimization—can further refine accuracy and adaptability, especially in large-scale networks. Incorporating AI and machine learning for predictive modeling will enable dynamic topology adjustments and better demand forecasting. Field validation of simulation results is essential to confirm real-world feasibility; while planning for scalability and future technologies like 5G and smart city integration ensures long-term viability. Regulatory alignment and collaboration with local authorities can streamline deployment, and the development of user-friendly tools and training programs will support broader adoption of intelligent design practices across the telecom industry.

VII. REFERENCES

[1] "FIber Systems," PLDT, 29 11 2024. [Online]. Available: https://www.fibre-systems.com/news/pldt-accelerates-ftth-roll-out-philippines.

[2] L. B. a. K. L. J. Smith, "Heuristic-Based Framework for Fiber Network Access Planning," *Journal of Optical Communications and Networking,* vol. 14, no. 3, p. pp. 145–152, Mar. 2022.

[3] M. A. a. Y. Zhao, "Application of Evolutionary Algorithms in Fiber Optic Network Design," *IEEE Access,* vol. 9, p. 87123–87135, 2021.

[4] T. C. a. M. L. R. Gonzales, "Optimizing FTTH Layouts Using Dijkstra's Algorithm," *Proc. IEEE Int. Conf. Communications (ICC), Dublin, Ireland,* pp. 1-6, Jun. 2020.

[5] O. (. f. E. C.-o. a. Development), "OECD (Organisation for Economic Co-operation and Development)," OECD (Organisation for Economic Co-operation and Development), June 2022. [Online]. Available: https://www.oecd.org/en/topics/sub-issues/broadband-statistics.html.

[6] M. M. a. J. N. D. K. A. Ogudo, "Comparative analysis of fibre optic and copper cables for high-speed communication: South African context," in *International Conference on Advances in Big Data, Computing and Data Communication Systems*, pp. 1–8, 2019.

[7] A. P. U. S. N. E. a. D. P. M. Iqbal, "Prim's algorithm for optimizing fiber optic trajectory planning," *International Journal of Scientific Research in Science and Technology,* vol. 3, pp. 504-509, 2017.

[8] N. &. B. A. S. &. H. B. &. A. S. K. A. F. Mohd Sabri, "The Utilisation of Dijkstraâ Algorithm to Assist Evacuation Route in Higher and Close Building," *Journal of Computer Science,* pp. 330-336, 2015.

[9] J. Li, "The application of optical fiber in network communication," Applied and Computational Engineering," vol. 35, p. pp. 141–146, 2024.

[10] I. U. E. N.-O. a. F. N. V. Matthews, "Analysis and implementation of fiber to the home network using Peace Estate, Lagos State, Nigeria," 2018.

[11] M. Z. a. J. Á.-T. Miguel Vargas-Bustamante, "Planning and Management of Fiber Optic Networks Based on a Geographic Information System: A Case in Ecuador,," *Human-Centered Design, Operation and Evaluation of Mobile Communications,Lecture Notes in Computer Science,,* vol. 15823, pp. 158-170, June 2025.

[12] D. P. a. T. Taber, "Managing A Fiber Optic Network with GIS," Esri User Conference Proceedings," in *Montana Dakota Utilities*, 2007.

[13] J. M. H. E. a. H. T. Mouftah, "Topological and Architectural Designs for Pre-ISDN Integration of Fiber Optics," *IEEE Transactions on Communications,* Vols. COM-35, no. 12, p. 1343–1351, Dec. 1987.

[14] I. &. J. A. &. W.-S. B. Szcześniak, "Generic Dijkstra for optical networks. Journal of Optical Communications and Networking," p. 11, 2019.