Optimal Tree Topology for a Submarine Cable Network With Constrained Internodal Latency

Tianjiao Wang, Xinyu Wang, Zengfu Wang, Chao Guo, Bill Moran, Moshe Zukerman, *Life Fellow, IEEE*

***Abstract*—This paper provides an optimized cable path planning solution for a tree-topology network in an irregular 2D manifold in a 3D Euclidean space, with an application to the planning of submarine cable networks. Our solution method is based on total cost minimization, where the individual cable costs are assumed to be linear to the length of the corresponding submarine cables subject to latency constraints between pairs of nodes. These latency constraints limit the cable length between any pair of nodes. Our method combines the fast marching method (FMM) and a new integer linear programming (ILP) formulation for minimum spanning trees (MST) where there are constraints between pairs of nodes. For cable systems for which ILP is not able to find the optimal solution within an acceptable time, we propose two polynomial-time heuristic methods based on Prim's algorithm, which we call PRIM I and PRIM II. PRIM I starts with an arbitrary initial node, while PRIM II iterates PRIM I over all nodes. A comprehensive comparative study is presented that demonstrates that PRIM II achieves results for the total cable length that are on average only 2.98% in excess of those obtained by the ILP benchmark. In addition, we apply our method, named FMM/ILP-based, to real-world cable path planning examples and demonstrate that it can effectively find an MST with latency constraints between pairs of nodes.**

***Index Terms*—Integer linear programming, Prim-based algorithm, minimum spanning tree, cable path planning, latency constraints.**

This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China under Project CityU8/CRF/13G, in part by the City University of Hong Kong under Project 9667193, and in part by the Shenzhen Municipal Science and Technology Innovation Committee under Project JCYJ20180306171144091. *(Corresponding author: Zengfu Wang.)*

Tianjiao Wang, Xinyu Wang, Chao Guo and Moshe Zukerman are with the Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: {tianjwang6-c; xywang47-c; chaoguo6-c}@my.cityu.edu.hk; m.zu@cityu.edu.hk).

Zengfu Wang is with the Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China, and also with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China. (e-mail: wangzengfu@nwpu.edu.cn).

Bill Moran is with the Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: wmoran@unimelb.edu.au).

I. Introduction

We have experienced an explosive growth of internet traffic over the last several decades that is expected to continue with the rapid development of 5G, IoT and AI technologies, especially considering the current COVID-19 outbreak. Cisco's latest report that predates the COVID-19 outbreak states that global annual IP traffic will reach 4.8 ZB per year by 2022 [1]. As the COVID-19 pandemic places many countries in lock down and many people are working (and learning) from home, the consumption of internet services increases dramatically. Generally, as the result of the pandemic, internet traffic is 25% to 30% higher than usual [2].

Submarine cables form a critical component of the international data transmission system, carrying more than 99% of global IP traffic [3]. As IP traffic is growing larger, the construction of additional submarine cables and their path planning optimization is key for meeting the ever-increasing internet traffic demands and provision of cost-effective and reliable internet services.

An important factor in cable path planning optimization is the cost of cable construction. While the cost may depend on several factors, such as future cost consequences of cable breakage associated with earthquakes or fishing activity, and the legal requirements to avoid certain areas, as discussed in [4], for this paper, for simplicity, we regard it as a linear function of cable length. That is, the cost of the cable between two nodes is assumed to only be based on the length of the geodesic in an irregular 2D manifold in a 3D Euclidean space. Our simplified assumption will be applicable to areas where the above mentioned factors are not applicable. In Section V-A we give an example for a region the Mediterranean, to show that our solution using this assumption is almost identical to a solution based on risk consideration where all of the risk factors are taken into account in the cable path design.

Based on the fast marching method (FMM) [5, 6], we find an optimal cable path between two nodes and its

optimal length and cost (linear in the length). Currently, the cost of submarine cable construction is estimated at around 24,000 USD per kilometer, indicating a significant cost of a long-haul submarine cable that may be in the tens or even hundreds of millions of dollars. Accordingly, a procedure to find the minimum length and/or cost of laying a cable path network becomes an important part of constructing a submarine cable system [7, 8]. As mentioned above, in this paper, we focus on the case, where only the cable length affects the cost. Henceforth, we will use the term *FMM – length only* to refer to this approach.

Another important criterion considered in cable path planning is *latency*; that is, the time it takes for a data packet to travel from the sender to the receiver. Latency includes transmission time as well as propagation, queuing delays and processing time. Propagation delay, which can be assumed to be linearly proportional to the cable length is a significant source of latency [9].

With technological advances, the number and importance of latency-critical applications are increasing. Different latency-critical applications have different latency requirements, as shown in Table I. In many cases, cable propagation delay is a major part of total latency. From [10], each 1,000 kilometers of cable length produces approximately 10 milliseconds of round trip delay. Clearly, limiting the length of submarine cables can reduce that latency up to a point. Ultra-low latency requirements of the order of 1 milliseconds, applicable to, for instance, autonomous vehicles and high frequency trading applications, entail cable lengths under 100 km may be difficult or impossible to meet using submarine cables, which are typically longer. However, a medium latency requirement of over 50 milliseconds is potentially achievable, as cable lengths of under 5000 km and even under 2000 km are quite common. We observe that latency can have other components beyond cable length and propagation delay.

In addition, for a cable network, some end-nodes are not connected directly by a single length of cable, so that data transmitted between such end-nodes needs to pass through intermediate nodes. Here we assume that optical bypass [17, 18] is used in intermediate nodes, so that the delay incurred is negligible, because the traffic can pass transparently through the node. Accordingly, we consider latency to be linear in the length of cable.

According to the Submarine Cable Almanac of 2020 [19], of all 266 submarine cable systems in the world, 246 have a tree topology (152 out of these 246 systems are point-to-point topology, and the remaining 94 cable systems use trunk-and-branch topology). In 36 cable systems that are now in the planning stage, 12 of them use point-to-point topology and 12 of them use trunk-and-branch topology. A tree topology (including both point-to-point and trunk-and-branch) is the most commonly used topology in submarine cable systems.

TABLE I: Latency-critical application.

Applications	Latency requirements and considerations
Video Interaction	About 500 milliseconds of latency in each direction is the upper limit, which is enough to allow for smooth conversation without awkward pauses [11].
Online Gaming	"A latency of more than 100 milliseconds can affect the experience of the gamer." [12].
E-commerce	"Amazon found every 100 ms of latency cost them 1% in sales." [13]
Autonomous -vehicle	With 5G and edge computing technologies, a target latency of 1 ms is envisioned for latency-critical applications such as self-driving cars [14, 15].
High-frequency Trading	"In today's world of high-frequency, algorithmic trading, a delay of just mere microseconds can cost an individual or company millions of dollars." [16]

In this paper, we aim to limit the time latency between pairs of nodes according to their requirements while minimizing the overall construction cost of the cable network. Nodes with strict latency requirements are either located near data centers or are heavy users of latency-critical applications that require limited latency in their communications with data centers. The contributions of this paper are as follows.

- We propose a new perspective to optimize cable network planning. We regard minimizing the cost of cable network problem as a minimum spanning tree (MST) problem and consider the latency constraints between pairs of nodes. To our best knowledge, latency constraints between pairs of nodes have not been considered in the research of cable network planning.
- We provide, for the first time, a new method for the MST problem over an irregular 2D manifold in a 3D space with constraints on the total length of the cables that constitute a path between a pair of nodes, for a prespecified set of pairs of nodes. Our new method, named FMM/ILP method, combines the FMM – length only which is used to find the optimized cable path between pairs of two nodes and its cost, as well as a new integer linear

programming (ILP) method named ILP-based algorithm that provides a tree-topology cable network at minimal cost and also satisfies the latency constraint for any pair of nodes. In addition, we propose two alternatives to the ILP-based method; these are modifications of Prim's MST algorithm and named PRIM I and PRIM II. The two Prim-based algorithm provide approaches to the solution of the MST with constraints problem. A large number of numerical simulations enable us to describe the relative performance of the three methods.

The remainder of this paper is organized as follows. In Section II, we review related research on cable path planning. Section III focuses on the modeling of the problem of a submarine cable network with constrained latency between pairs of nodes as an MST with constraints problem. In Section IV we propose an ILP-based method to solve this problem for a wide range of existing cable systems, and describe the two Prim-based algorithms as alternatives to the ILP-based method. The performances of the ILP-based method and Prim-based algorithms are compared and discussed in the context of a large number of numerical simulations in Section V-B. A real-world example that uses the ILP-based method to achieve optimal cable network planning is shown in Section V-A. Finally, Section VI concludes this paper.

II. Related Work

Many research publications on cable path planning focus on minimizing the total cable cost under the survivability constraints [4, 20–27]. Msongaleli *et al.* [25] considered a set of possible routes between nodes and a set of disaster scenarios with a probability model for cable break and provided an ILP-based algorithm to design a submarine cable network to minimize the expected cost in case of a disaster. In [24], Zhao *et al.* proposed a path planning method that aims to obtain a path aiming to minimize cable cost and earthquake risk to the cable using a semi-supervised model based on raster graphics. They used the Dijkstra's algorithm to minimize cost and cable break risk as a result of an earthquake. In [4, 26, 27], the approach we call "FMM – length and other considerations" was used to provide a solution for a multi-objective (cost and risk) path-planning problem on a 2D manifold in a 3D space that models the earth's surface. In addition to considering the presence of earthquake-activities, various other design considerations (such as water depth, sediment hardness and human activities) were considered in the cable path design. Many other publications on network resilience in 2D are not concerned with path planning of the links, and are not discussed here.

Most of the existing work on cable path planning has focused mainly on point-to-point path optimization. Except for point-to-point cable design, there still remains the problem of choosing the optimal topology for the cable system, and as mentioned above, a tree is a widely used topology for cable systems. The paper closest to the present paper in terms of the cable path planning is [28] which optimizes the cable network in a tree topology. In [28], Wang *et al.* considered path planning for a cable system with a trunk-and-branch tree topology on the earth's surface and formulated the problem as a Steiner minimal tree problem. Tran *et al.* [29] presented a dynamic programming method to choose new links and routes for a given network. None of these cable path planning methods considered latency constraints for different pairs of nodes which is the main contribution of this paper.

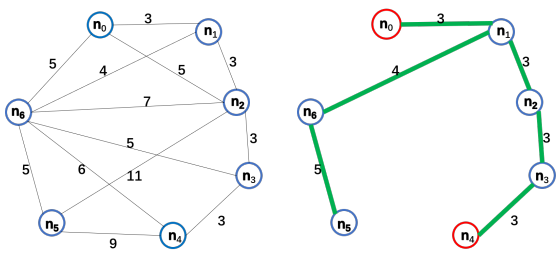
The MST problem is a well-known combinatorial optimization problem, and it is heavily considered in this paper. Related work on the MST problem includes work on the Kruskal's algorithm which generates forests in the process of obtaining the MST and finds an edge of the least possible weight that connects any two trees in the forests [30], and Prim's algorithm which is slightly different from Kruskal's algorithm. Prim's algorithm starts from a node n_0, then selects the least-cost edge $e(n_0, n_x)$ and adds it and node n_x to the spanning tree. For graphs stored in adjacency matrix, the computational complexity of Kruskal's and Prim's algorithms are $O(E * \log(N))$ and $O(N^2)$, respectively, where E and N represent the number of edges and nodes in the graph, respectively. Kruskal's and Prim's algorithms have been used in [31] in the context of distance constraints between a certain prespecified *central* node and all other nodes. Jaewon and Pedram [31] proposed an algorithm, named *bounded path length Kruskal* (BKRUS), to find the MST that satisfies this constraint. For pairs of nodes not involving the central node, BKRUS is unable to satisfactorily address distance requirements between them. To the best of our knowledge, no algorithm for constrained MST exists where the constraints can be between any pairs of nodes.

Over the last couple of decades, as a result of advances in high-performance computing and more efficient algorithms, ILP has achieved considerable success in obtaining optimal solutions to many combinatorial

optimization problems. Existing ILP formulations for the MST problem include Martin's Formulation, the Subtour Elimination Formulation, and the Cutset Formulation [32–34].

III. Problem Formulation and Modeling

In this paper, our objective is to construct a tree-topology cable network without additional Steiner nodes at minimal cost, while satisfying constraints on certain paths between specified nodes, possibly involving more than one hop between those nodes. To this end, we consider the cable network as a spanning tree (in the sense of graph theory) in which the cost of every edge is based on the length of the geodesic between its end-nodes. Such costs can be calculated using FMM – length only. Certain pairs of nodes need different requirements of latency, as discussed in Section I, where the constraints are on the total length of the cables that provide paths between prespecified pairs of nodes. In particular, while focusing on minimizing the cost of the entire cable network system, we ensure that cable length between any pair of nodes satisfies given (achievable) latency requirements.



(a) Input graph (b) A spanning tree

Fig. 1: Problem description

Let $\mathbb{D}$ be a closed and bounded path-connected region on the surface of the earth where we aim to lay the cable network, and $n_0, n_1, \ldots, n_x \in \mathbb{D}$ denote the nodes to be connected in a cable network with spanning tree topology.

Let $V = \{n_0, n_1, \ldots, n_x\}$. Let $(i, j), i, j \in V$ denote the edge connecting the nodes i and j on $\mathbb{D}$. Let E denote the set of edges, and $G = (V, E)$ the graph, as indicated in Fig. 1(a). As Fig. 1(b) illustrates, a spanning tree T of G is a connected subgraph containing no cycles. That is, $T = (V, E^*)$, $E^* \subseteq E$ and T is a tree. Let $l_{i,j}$ denote the length of the cable path between nodes i and j, which may include one or several individual cable edges (*e.g.*, $l_{n_0,n_4} = e_{(n_0,n_1)} + e_{(n_1,n_2)} + e_{(n_2,n_3)} + e_{(n_3,n_4)}$) in a spanning tree. We write $\mathbb{C} = \{(i \leftrightarrow j), i, j \in V\}$ to denote the set of pairs of nodes with constraints. Denote the total length of T by $l(T)$.

We aim to find the spanning tree T with minimal total length while satisfying the constraints given by $\mathbb{C}$. We formulate the problem in the following equations:

$$\underset{T}{\text{minimize}} \quad \sum l(T), \tag{1a}$$

$$\text{subject to}$$

$$l_{i,j} \le l^*_{i,j}, \ \forall\, (i \leftrightarrow j) \in \mathbb{C}, \tag{1b}$$

where $l^*_{i,j}$ is the threshold for the length constraint between node-pair i and j.

IV. Solution Methodologies and Algorithms

In this section, we first formulate a new ILP for our problem based on Martin's formulation. We assess its computational burden by calculating its number of decision variables (namely, $x_{ij}, y_{ij}, z^{ab}_{ij}$) and constraints. In addition, we propose two heuristic algorithms, namely PRIM I and PRIM II. These two algorithms are modified by Prim's MST algorithm, which can find MST with some constraints.

A. ILP-based method

Our ILP-based method is derived from Martin's formulation which expresses the MST problem in terms of a number of polynomial constraints [32, 34]. For the graph $G = (V, E)$, the formulation is given below.

$$\underset{x_{ij}}{\text{minimize}} \quad \sum_{(i,j)\in E} l_{ij} x_{ij}, \tag{2a}$$

$$\text{subject to}$$

$$\sum_{(i,j)\in E} x_{ij} = n - 1, \tag{2b}$$

$$y^k_{ij} + y^k_{ji} = x_{ij}, \quad \forall (i,j) \in E, \ k \in V \tag{2c}$$

$$\sum_{k \in V \setminus \{j\}} y^j_{ik} + x_{ij} = 1, \quad \forall (i,j) \in E, \ k \in V \tag{2d}$$

$$x_{ij}, y^k_{ij} \in \{0, 1\}. \tag{2e}$$

Here, l_{ij} represents the length of the edge (i, j). The variables x_{ij} and y^k_{ij} are all binary, where $x_{ij} = 1$ indicates that the edge (i, j) is included in the spanning tree. The statement $y^k_{ij} = 1$ indicates that edge (i, j) is

in the spanning tree and node k is on the side of j, i.e., the connection between nodes k and i must go through node j.

Constraint (2b) is derived from the properties required of the tree topology, and provides a guarantee that the number of edges is one less than the number of nodes. Constraint (2c) for $(i,j) \in E, k \in V$ establishes that, if $(i,j) \in E$ is chosen to be a member of the tree (that is, $x_{ij} = 1$), then any node $k \in V$ has either to be on the same side as j ($y_{ij}^k = 1$) or on the same side as i ($y_{ji}^k = 1$). If the edge $(i,j) \in E$ is not in the tree (i.e., $x_{ij} = 0$), then no node k is on the side of j or i ($y_{ij}^k = y_{ji}^k = 0$). Constraint (2d) for an edge $(i,j) \in E$ means that, if $(i,j) \in E$ is in the tree ($x_{ij} = 1$), edges (i,k) connecting to i have to be on the same side as i. If the edge $(i,j) \in E$ is not in the tree ($x_{ij} = 0$), then some edge (i,k) must exist so that node j is connected to node i through node k (i.e., $y_{ik}^j = 1$).

The use of Martin's formulation for the MST problem allows us to add the following inequalities (3) to enforce the cable length constraint (1b). For any $(a \leftrightarrow b) \in \mathbb{C}$, we have

$$\sum_{(i,j)\in E} \left(y_{ij}^a \cdot y_{ji}^b + y_{ij}^b \cdot y_{ji}^a\right) l_{i,j} \leq l_{a,b}^*. \tag{3}$$

As in the previous definition, $y_{ij}^a = 1$ means that edge (i,j) is in the spanning tree and node a is on the side of node j, $y_{ji}^b = 1$ means that edge (i,j) is in the spanning tree and node b is on the side of i. If both $y_{ij}^a = 1$ and $y_{ji}^b = 1$ then the edge (i,j) is included in the spanning tree and, moreover, in the path between node a and node b. In this case, node a is close to node j and node b is close to node i which gives a direction to edge (i,j). However, the cable path between node a and node b does not have a preferred direction, so we need also to analyze y_{ji}^a and y_{ij}^b in the same way. The four variables y_{ij}^a, y_{ji}^b, y_{ji}^a and y_{ij}^b decide whether the edge (i,j) is included in the path between a and b, without considering the direction.

Unfortunately, (3) is non-linear. To apply ILP techniques, we need to replace them by linear constraints. To this end, we introduce two new variables z_{ij}^{ab} and z_{ji}^{ab}, and make $z_{ij}^{ab} = y_{ij}^a \cdot y_{ji}^b$ and $z_{ji}^{ab} = y_{ji}^a \cdot y_{ij}^b$. This means that $z_{ij}^{ab} = 1$ only when both $y_{ij}^a = 1$ and $y_{ji}^b = 1$; otherwise, $z_{ij}^{ab} = 0$. Because y_{ij}^a, y_{ji}^b and z_{ij}^{ab} are binary variables representing Boolean values, the relationship between them can be regarded as a Boolean operation. So we can use four linear constraints to express a single Boolean constraint, as (4) shows. The analysis is the same for z_{ji}^{ab}.

$$\begin{cases} z_{ij}^{ab} \leqslant y_{ij}^a + y_{ji}^b, \\ z_{ij}^{ab} \geqslant y_{ij}^a + y_{ji}^b - 1, \\ z_{ij}^{ab} \leqslant 1 - y_{ij}^a + y_{ji}^b, \\ z_{ij}^{ab} \leqslant 1 + y_{ij}^a - y_{ji}^b. \end{cases} \tag{4}$$

Our problem is then reformulated as:

$$\underset{x_{ij}}{\text{minimize}} \quad \sum_{(i,j)\in E} l_{ij} x_{ij}, \tag{5a}$$

subject to

$$\sum_{(i,j)\in E} x_{ij} = n - 1, \tag{5b}$$

$$y_{ij}^k + y_{ji}^k = x_{ij}, \quad \forall (i,j) \in E,\ k \in V \tag{5c}$$

$$\sum_{k\in V\setminus\{j\}} y_{ik}^j + x_{ij} = 1, \quad \forall (i,j) \in E,\ k \in V \tag{5d}$$

$$\sum_{(i,j)\in E} (z_{ij}^{ab} + z_{ji}^{ab}) \cdot l_{i,j} \leq l_{a,b}^*, \quad \forall (a \leftrightarrow b) \in \mathbb{C} \tag{5e}$$

$$x_{ij}, y_{ij}^k, z_{ij}^{ab} \in \{0,1\}. \tag{5f}$$

Constraints (5b)-(5d) are the same as constraints (2b)-(2d) and enforce the tree topology. Constraint (5e) is used for the length constraint.

The variables in the ILP-based formulation include x_{ij}, y_{ij}^k and z_{ij}. For a complete graph, $|E| = 1/2 \cdot |V| \cdot (|V| - 1)$. For MST with one pair of latency constraint (i.e., $|\mathbb{C}| = 1$), the number of variables is $1/2 \cdot (|V|^3 - |V|)$. More generally, we analyze the complexity in the case of an incomplete but connected graph. In this case, the total number of variables in the formulation is the sum of the number of x (=$|E|$), the number of y (=$|E| \cdot (|V| - 2)$), and the number of z (=$2|E| \cdot |\mathbb{C}|$). The number of constraints in the ILP-based formulation is one plus the sum of two times the number of x (=$1 + 2|E|$), according to (5b)-(5d), and plus nine times the number of latency constraints requirements (=$9|\mathbb{C}|$), according to (4) and (5e).

The solution of the ILP formulation above provides a solution to the MST problem. To obtain numerical results for this ILP problem, we use the package python-pulp [35] that employs a method called "branch-and-cut", an exact algorithm based on a combination of the branch-and-bound algorithm and the cutting plane method.

B. Computational complexity analysis for ILP-based method

In this paper, we consider an extension of the MST problem. In addition to the well-studied problem of finding an MST in a weighted, undirected connected graph, there exist length constraints between pairs of nodes. The complexity of the full MST problem without constraints is known to be no worse than $O(|E|\alpha(|E|,|V|))$, where α is the Ackermann function [36]. Based on the computation times needed in our simulations, our algorithm seems to be more complex.

We have been unable to determine whether or not it is NP-hard and there appear to be no corresponding results in the literature. However, from the perspective of real-world cable routing problems, this is not a major issue as typically $|V|$ is relatively small. In [34], a detailed computational analysis of the different ILP formulations was described in terms of run-time. We adopt a similar approach here, and illustrate the computational complexity of our ILP-based method via run-times with graphs with various numbers of nodes and constraints.

In Fig. 2 and Table II, we provide run-time of our ILP algorithm as a function of the number of nodes and constraints. Fig. 2 demonstrates that the run-time of the ILP is exponentially increasing with the number of nodes and constraints. Nevertheless, the ILP solution is applicable to most existing cable systems because the number of nodes in such systems is normally not too large, which implies that the number of latency constraints is also not too large. This is evidenced by the information available in the Submarine Cable Almanac of 2020 [19]. According to this information, 93% of the cable systems have less than 10 nodes. In Fig. 3, we provide a histogram based on data from [19] of the distribution of the number of cities (nodes) in each submarine cable systems. We note that the highest number of nodes for any existing cable system is 39 which is the number of nodes in the so-called *South East Asia-Middle East-Western Europe 3 cable system.* Therefore, our ILP solution is applicable to most realistically sized cable systems. Especially, we can still use some commercial solvers, e.g., AMPL/Gurobi [37] to find some feasible solutions of LIP problem with provable bounds, as the example shown in Section V-B.

C. Heuristic Prim-based algorithms

Compared with our ILP-based method, the proposed Prim-based algorithm, a heuristic method, can find a relatively small spanning tree that satisfies the constraints in complete graphs, but optimality is not guaranteed. Our Prim-based algorithm applies the Prim algorithm while iteratively checking that the constraint is satisfied. It is described in Algorithm 1.

Our proposed Prim-based algorithm takes as input a graph, stored as an adjacency matrix. Its computational complexity is a combination of the computational complexity of the original Prim algorithm, namely, $O(|V|^2)$ and that of constraint verification, which is $O(3|V|)$.

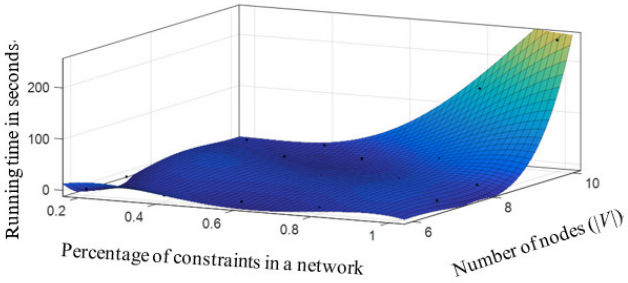


Fig. 2: Run-time in seconds of ILP on partially and fully constrained network.

TABLE II: Run-time in seconds of ILP on partially and fully constrained network.

T \ P / V	1	0.8	0.6	0.4	0.2
6	0.5323	0.4232	0.3771	0.3043	0.1732
7	2.747	1.9854	1.3217	0.8337	0.4237
8	9.8977	8.9833	2.3104	1.5235	0.71715
9	31.2542	25.3531	12.78992	4.668	0.964
10	245.0578	138.1458	16.3513	3.2174	1.223

V: Number of nodes; T: Run-time in seconds; P: Percentage of constraints in a network.

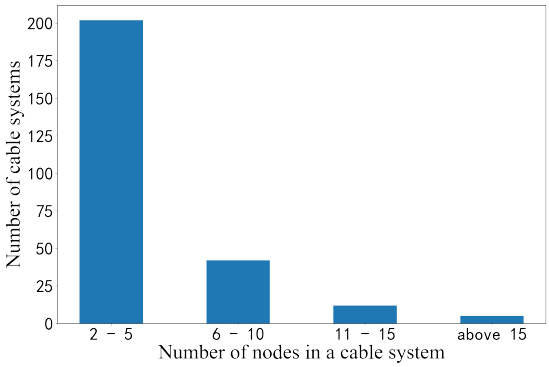


Fig. 3: Histogram of the number of nodes in a cable system.

Algorithm 1 Prim-based method for MST with constraints (PRIM I).

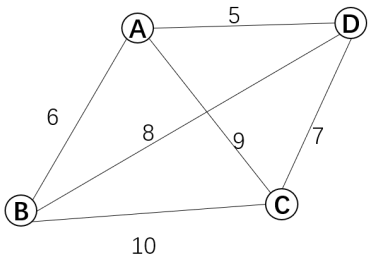
Input:
The graph $G = (V, E)$, and the set of constraints requirements $\mathbb{C}$.
Output:
A spanning tree T.

Let $U = \{i\}$, i, an arbitrary node in V;
Let $F = \emptyset$, used to store edges in the spanning tree;
while $U! = V$ **do**
  Find the smallest edge (i, j), where the node $i \in U$ and $j \in V \setminus U$.
  if $T = (U, F \cup \{i, j\})$ satisfies $\mathbb{C}$, **then**
    $U = U \cup \{j\}$ and $F = F \cup (i, j)$;
    Return to Step 3;
  else
    Eliminate this edge from E, and return to Step 3;
  end if
end while
return T;

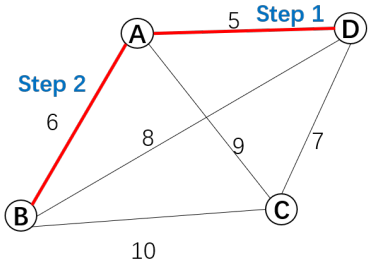
The Prim-based algorithm, therefore, has polynomial complexity — in fact, $O(|V|^2)$. This should be compared with the ILP-based method, for which, in simulation, run-time grows exponentially with the size of the problem (see our demonstration in Section IV-B). This is an especially important consideration for large-scale networks.

Unfortunately, the Prim-based algorithm is not guaranteed to find a feasible solution (that is an MST satisfying the length constraints) even if one exists. This is demonstrated in the example presented in Fig. 4. For the complete graph of Fig. 4(a), our aim is to find an MST with two constraints ($l_{C,B} \leq 12$, and $l_{C,D} \leq 12$). The first two steps of the Prim-based Algorithm starting at node A are shown in Fig. 4(b). The Prim-based algorithm cannot find a suitable edge at step 3, though there is a feasible solution as Fig. 4(c) shows.

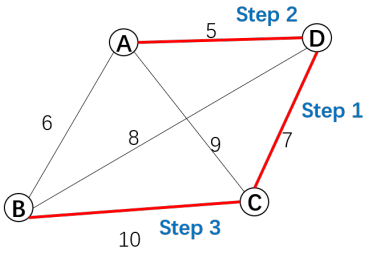
This problem can be addressed by reapplying the Prim-based algorithm starting at each time with a different initial node and then selecting the result that gives the least cost among the various spanning trees obtained. Note that, unlike the original Prim's algorithm where the resulting MST is independent of the start node, the Prim-based algorithm may give different results if the start node is different. For example in Fig. 4, we can obtain a feasible solution by executing the Prim-based algorithm with start node C.



(a) A complete graph with four nodes.



(b) The first two steps.



(c) A feasible solution.

Fig. 4: Prim-based Algorithm missed a feasible solution.

This approach leads to an improved heuristic algorithm described in Algorithm 2. Our first heuristic algorithm described in Algorithm 1 that aims to minimize a spanning tree starting from an arbitrary initial node based on Prim's Algorithm will henceforth be called *PRIM I*, and the improved Prim's based algorithm that starts from the best initial node will be called *PRIM II*.

As PRIM II repeats PRIM I $|V|$ times, each with a different node in V being chosen as the starting node and selecting the least cost of the resulting spanning trees, the computational complexity of PRIM II is $O(|V|^3)$.

We have observed in numerical simulations that the failure of Prim I was related to the choice of the initial node. Accordingly, we have proposed to use Prim II, an improved version of Prim I, that iterates over all nodes

Algorithm 2 PRIM II

Input:
The graph $G = (V, E)$, and the set of constraints requirements $\mathbb{C}$.

Output:
A spanning tree T_{min}.

for Every node k in the Graph. **do**
  Let $U = \{k\}$;
  Let $F = \emptyset$, used to store edges in the spanning tree;
  while $U! = V$ **do**
    Find the smallest edge (i, j), where the node $i \in U$ and $j \in V \setminus U$.
    if $T_k = (U, F \cup \{i, j\})$ satisfies $\mathbb{C}$, **then**
      $U = U \cup \{j\}$ and $F = F \cup (i, j)$;
      Return to Step 3;
    else
      Eliminate this edge from E, and return to Step 3;
    end if
  end while
  if $T_k \leqslant T_{min}$ **then**
    $T_{min} = T_k$;
  end if
end for
return T_{min};

as initial node. We have performed many simulations as described in Section V, and we did not find a single case where Prim II failed to find a feasible solution if such a solution exists.

V. Numerical Results

This section has three parts. We begin by presenting numerical results for real-life applications of our ILP-based approach. Next, we present comprehensive results for comparison of quality and run-times of the three algorithms, namely, the ILP-based and the two Prim-based algorithms. Finally, we present results that demonstrate the applicability of our two Prim's based algorithms for a large network of 100 nodes (in this case the ILP does not provide a solution) and compare the costs of the resulting spanning trees and run-times. All simulations in this section are run on a platform with a 2.30-GHz Intel Core i5-6200U CPU, except that the 50 node network example solved by a commercial solver Gurobi [37] is run on a platform with a 4.20-GHz Intel Core i7-7700K CPU.

A. Applications of the ILP-based method in realistic scenarios

In this section, we apply the ILP-based method to several 3D realistic scenarios. We use bathymetric data from the Global Multi-Resolution Topography synthesis [38]. The object region $\mathbb{D}$ is from a northwest corner (44.000°N, 2.000°E) to the southeast corner (36.000°N, 9.000°E), as shown in Fig. 5. We plan a submarine cable communication network using a spanning tree topology between these six cities: Barcelona (41.386°N, 2.190°E), Marseille (43.297°N, 5.359°E), Alghero 40.557°N, 8.312°E), Annaba (36.928°N, 7.760°E), Algiers (36.761°N, 3.074°E) and Palma (39.576°N, 2.632°E), and denoted as A, B, C, D, E, F, in Figs. 5-7, respectively. We assume these six locations and the

Fig. 5: Region $\mathbb{D}$. Source: Google Earth.

available links between them form a complete graph. Knowing the cost and length of each edge, we can use our ILP-based method to construct a submarine cable network with minimal cost satisfying the given constraints. Firstly, we use FMM – length only to find the optimal cable path for every pair of nodes. In order to compare the difference between, on the one hand, the optimal cable paths taking account of risk and, we run FMM – length and other considerations again for every pair of nodes to find the optimal path with minimal cost. In addition, we also calculate the great-circle distance between pairs of nodes using their geographic coordinate which are the length of smooth curves. The length and cost of these cable edges calculated by different approaches are recorded in Table III. The result of MST for submarine cable network based on the result of FMM

TABLE III: Length and cost of edges between pairs of nodes.

Edges	FMM – length and other considerations		FMM – length only		great-circle distances	
	Length (km)	cost (million $)	Length (km)	cost (million $)	Length (km)	cost (million $)
AB	304.11	8515.86	303.96	8511.12	302.75	8480.11
AC	440.44	12328.47	433.62	12142.55	428.95	12015.11
AD	637.73	1785.86	623.64	17456.69	622.96	17449.34
AE	651.82	15451.18	653.64	15602.13	533.02	14930.04
AF	216.04	6049.20	219.51	6146.24	213.14	5970.06
BC	379.01	10613.87	373.53	10459.96	361.13	10115.40
BD	718.34	20105.46	727.92	20382.83	703.02	19691.71
BE	715.08	20022.31	715.15	20024.38	715.66	20045.79
BF	395.19	11084.32	395.64	11078.96	395.69	11083.49
CD	379.45	10625.66	375.27	10508.75	375.13	10507.55
CE	483.41	13536.97	482.06	13498.93	482.06	13501.63
CF	310.76	8701.67	309.71	8671.92	308.16	8631.69
DE	281.95	7875.72	245.53	6875.06	244.52	6849.17
DF	446.62	12506.61	412.89	11561.92	411.87	11536.59
EF	337.17	9440.81	337.60	9452.99	333.90	9352.55

– length and other considerations is shown in Fig. 6. Here, we suppose there are no constraints between any pairs of nodes. The results of MST by the other two approaches are the same. From the comparison of the records in the table, the differences among FMM – length and other considerations, FMM – length only and great-circle distance are small. However, FMM – length and other considerations and FMM – length only are more close to realistic results, and in this area, the consideration of the risk in cable contribution cost makes a small contribution. Accordingly, in this area, for simplicity, we can use FMM – length only to find the optimal cable path.

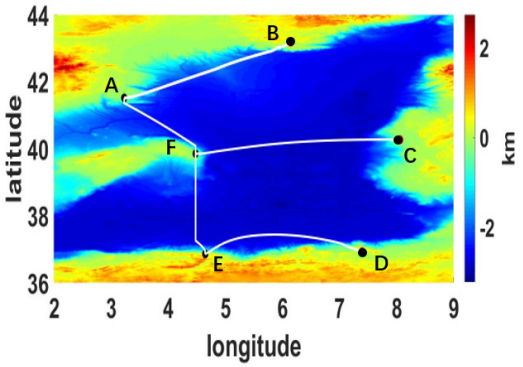


Fig. 6: Minimum cost cable network without constraints.

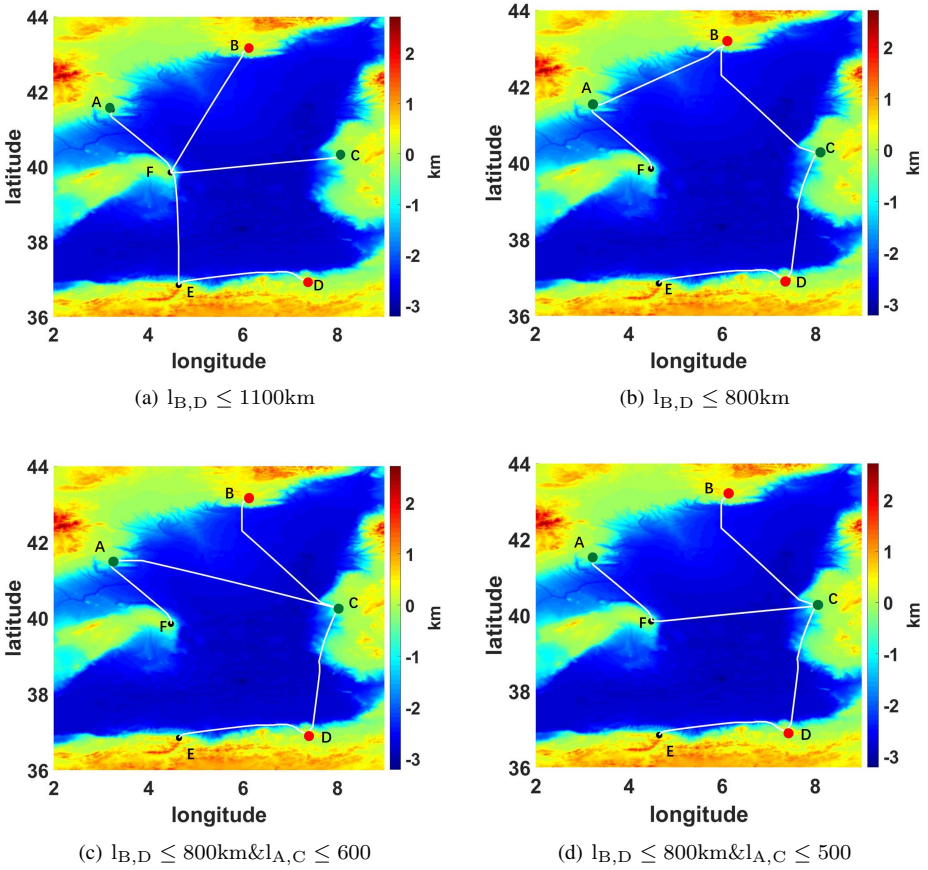
As discussed in Section I, submarine cable network construction often needs to take account of the latency constraints for communication between different pairs of nodes. To demonstrate the effectiveness of our method with respect to the constraints, we have done experiment simulations to illustrate the effect of different internodal latency constraints on the resulting MST for a submarine cable network. By including various numbers of inequalities (5e), we produce different MSTs with different length constraints as shown in Table IV and Fig. 7. With the increase in the number of constraints and the strengthening of restrictions, the total length of the MST increases, resulting in increased costs for submarine cable network construction. This provides quantification of an expected behavior.

B. Comparisons of the three algorithms

We will now provide comprehensive numerical results to compare the ILP-based method, PRIM I and PRIM II, in terms of the solution quality and run-times. We begin with three complete graphs of different sizes: 25 node graphs, 30 node graphs and 40 node graphs. The positions of the nodes are generated by sampling from a uniform distribution in the region $[0, 100] \times [0, 100]$. The lengths of the edges connecting two nodes are just the Euclidean distances between them. Each size of graph has 10 randomly generated realizations. This gives a total of 30 different graphs in which we apply all three algorithms to find the spanning tree with constraints. The plots in Figs. 8 (a)–(c) show the comparison for the three methods in terms of their lengths of the spanning trees

TABLE IV: Different results of spanning tree with different constraints requirements by ILP-based method

Constraints (km)	No constraint	$l_{B,D} \leq 1100$	$l_{B,D} \leq 800$	$l_{B,D} \leq 800 \& l_{A,C} \leq 600$	$l_{A,C} \leq 800 \& l_{A,C} \leq 500$
$l_{B,D}$ (km)	1106.6	978.77	748.8	748.8	748.8
$l_{A,C}$ (km)	529.22	529.22	677.49	529.22	433.62
Length of tree (km)	1416.31	1507.99	1517.8	1523.55	1647.46
Tree topology	Figure 6	Figure 7a	Figure 7b	Figure 7c	Figure 7d



(a) $l_{B,D} \leq 1100$km (b) $l_{B,D} \leq 800$km

(c) $l_{B,D} \leq 800$km&$l_{A,C} \leq 600$ (d) $l_{B,D} \leq 800$km&$l_{A,C} \leq 500$

Fig. 7: Different results of spanning tree with different constraints requirements by ILP-based method

with different numbers of nodes, namely, plots (a), (b), and (c), for 25, 30, and 40 nodes, respectively.

The results show that the performance of PRIM II is clearly superior to that of PRIM I. They also demonstrate that PRIM II's results are close to the optimal solution obtained by the ILP-based method. By calculating the difference between the PRIM II and ILP-based method, we can see that the PRIM II achieves results that are on average 2.98% above those obtained by the ILP benchmark. In Table V, we present the results obtained for the run-time comparison of the ILP-based method and PRIM I and PRIM II. As expected, the run-times of the Prim-based algorithms for the various cases are significantly smaller than those of the ILP-based method.

The run-time of the ILP-base method grows exponentially with the increase of the network size, which may prohibit us from obtaining the optimal results. However, we can still use the ILP to obtain near optimal solutions, so that ILP can be used not only for the optimal solution, but also as a heuristic algorithm that provides good quality solutions with provable bounds. Using a com-

TABLE V: Run-time comparisons (seconds)

Scenarios	25 node graph			30 node graph			40 node graph		
	ILP	PRIM I	PRIM II	ILP	PRIM I	PRIM II	ILP	PRIM I	PRIM II
1	131.06	2.25	0.17	134.22	3.81	0.17	11306	3	0.13
2	68.38	3	0.12	234.39	2.43	0.15	34638	3.2	0.23
3	62.76	2.25	0.13	132.13	33	0.14	24268	3	0.19
4	74.65	3.25	0.10	138.09	3.66	0.11	11165	3.77	0.12
5	115.53	2.25	0.09	382.74	2.07	0.09	9353	2.97	0.14
6	102.25	3.75	0.15	967.97	4.98	0.12	7025	2.28	0.20
7	104.35	1.5	0.16	314.32	2.79	0.09	9935	4.08	0.52
8	114.22	3.5	0.14	64.34	3.29	0.12	31322	4.14	0.11
9	64.45	2.6	0.10	132.16	3.54	0.15	4265	4.76	0.15
10	45.06	2	0.11	328.09	2.52	0.12	11456	3.44	0.17

TABLE VI: Results of PRIM II and ILP for a 50 node network

	PRIM II	ILP bounded approximation	ILP optimal
Total length	509.03	502.15	487.34
Run-time	15.05 s	24,248 s	26,093 s

mercial solver, namely, AMPL/Gurobi [37], we obtain optimal results as well as bounded approximation results for a 50 node network using ILP and compare them with PRIM II. The results are presented in Table VI. As shown in Table VI, the PRIM II result is 509.03 obtained in 15.05 seconds, the ILP bounded approximate result (502.15) is obtained in $24,248$ seconds when the gap [37] between the lower and upper bounds of the optimal solution is 3.35%, while the ILP optimal result (487.34) is obtained in $26,093$ seconds.

The run-time for the 50 node network is somewhat surprisingly low if we consider the average run-times for the 25, 30 and 40 node networks in Table V and the expected exponential run-time increase as discussed in Section IV-B. However, not only the number of nodes affects the run-time, but also the strength of the constraints. In addition, considering the large standard deviations presented in Table V we can realize that some low run-times for a 50 node network are possible.

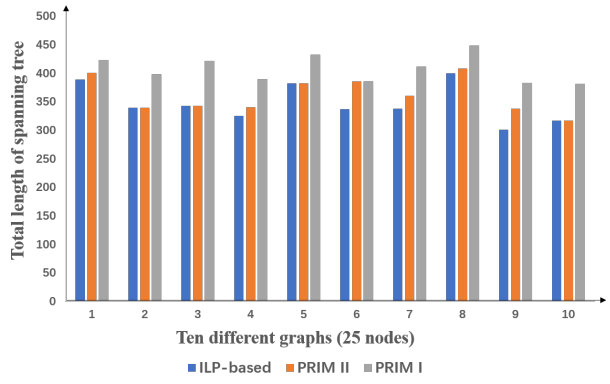
C. The Prim-based algorithms for a 100 node network

We now demonstrate in Figs. 9 and 10 the improvement achieved by PRIM II relative to the results obtained by PRIM I. For the 100 node input graph, the positions of the 100 nodes are randomly generated in the region $[0, 100] \times [0, 100]$. The length of the edge connecting two nodes is defined as the Euclidean distance between them. The constraints requirements are $l_{n1,n5} \leq 90$ and $l_{n2,n3} \leq 100$. Fig. 9 shows the result of PRIM I with an arbitrary initial node $n0$, while Fig. 10 shows result of PRIM II with the best initial node $n5$. As expected, PRIM II provides a better quality solution than PRIM I. For the result of PRIM I, the total length of the spanning tree is 758.87, and the constraint requirements are satisfied ($l_{n1,n5} = 69.23$ and $l_{n2,n3} = 66.80$). For the result of PRIM II, the total length of the spanning tree is 684.98, and the constraint requirements are also satisfied ($l_{n1,n5} = 82.54$ and $l_{n2,n3} = 98.40$). The run-times for PRIM II and PRIM II are 0.55 s and 62.37 s, respectively.

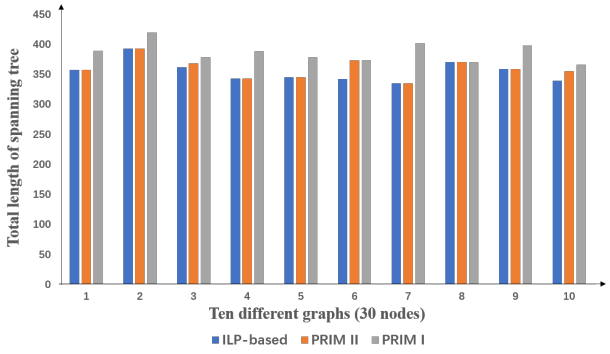
VI. Conclusion

We have provided a method, called FMM/ILP, for optimizing a tree-topology cable network with latency constraints in an irregular 2D manifold in a 3D Euclidean space. Specifically, in the submarine cable application, while focusing on minimizing the cost of the entire network, we ensure that cable length between any pair of nodes satisfies latency requirements. Our FMM/ILP method is based on finding a cable path and cost between any pair of nodes using FMM and then finding an MST with latency constraints (i.e., constraints on cable length) for any pair of nodes, based on ILP.

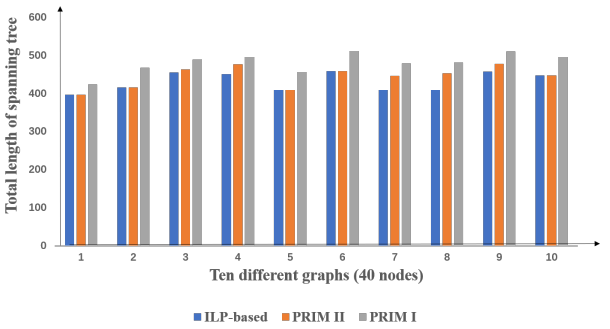
We have proposed a new ILP-based method to solve constrained MST problems. Although, in general, ILP is not scalable, we have shown that it can find an MST with latency constraints for most realistic cable systems in a few minutes. Two alternative heuristic algorithms,



(a)



(b)



(c)

Fig. 8: Length comparisons

PRIM I and PRIM II, based on Prim's algorithm to finding an MST with constraints, has been demonstrated to provide approximate solutions for large scale cable systems. We have provided explanations of the weakness of PRIM I, and how they are overcome by the PRIM II algorithm. We have presented comprehensive numerical results that demonstrate the accuracy of PRIM II. In particular, the PRIM II results for the total cable length have been on average only 2.98% worse than those obtained by the ILP benchmark.

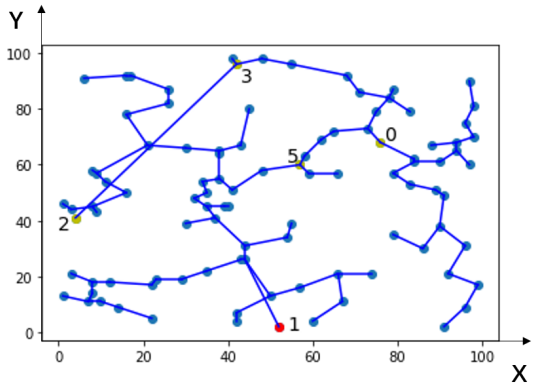


Fig. 9: The spanning tree with constraints obtained by PRIM I.

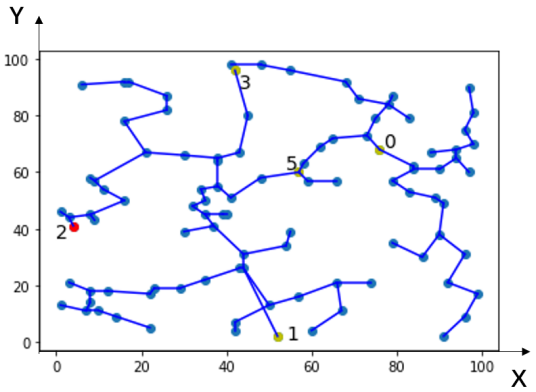


Fig. 10: The spanning tree with constraints obtained by PRIM II.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their important comments and corrections that helped to improve the paper.

REFERENCES

[1] Cisco, "2019 Annual Report: Defining the Future of The Internet," December 2019. [Online]. Available: https://www.cisco.com/c/en/us/about/annual-reports.html

[2] M. Branscombe, "The Network Impact of the Global COVID-19 Pandemic," April 2020. [Online]. Available: https://thenewstack.io/the-network-impact-of-the-global-covid-19-pandemic/

[3] A. McCurdy, A. Shelton, B. Burns, C. Bayly, E. Handa, G. Gerstell, J.-M. Fortain, J.-P. Joensuu, L. Duvernay, R. Lingle, and S. Ash, "Submarine telecoms industry report," Submarine Telecoms Forum, Inc., Tech. Rep., 2018. [Online]. Available: https://subtelforum.com/products/submarine-telecoms-industry-report/

[4] Q. Wang, J. Guo, Z. Wang, E. Tahchi, X. Wang, B. Moran, and M. Zukerman, "Cost-effective path planning for submarine cable network extension," *IEEE Access*, vol. 7, pp. 61 883–61 895, 2019.

[5] R. Kimmel and J. A. Sethian, "Fast marching methods on triangulated domains," *Proc. Nat. Acad. Sci.*, vol. 95, pp. 8431–8435, 1998.

[6] J. A. Sethian, "Fast marching methods," *SIAM Review*, vol. 41, pp. 8431–8435, 1998.

[7] D. R. Burnett, R. Beckman, and T. M. Davenport, *Submarine Cables: the handbook of Law and Policy*. Leiden, Netherlands: Martinus Nijhoff Publishers, 2013.

[8] E. Tahchi, personal communication, Oct. 2016.

[9] X. Xiao, *Technical, commercial and regulatory challenges of QoS: An internet service model perspective*. MA, USA: Morgan Kaufmann, 2008.

[10] TeleGeography, "Trans-atlantic network latency reduced," October 2015. [Online]. Available: https://www.thebroadcastbridge.com/content/entry/3988/trans-atlantic-network-latency-reduced

[11] W. M. Systems, "What Is Low Latency and Who Needs It?" July 2020. [Online]. Available: https://www.wowza.com/blog/what-is-low-latency-and-who-needs-it

[12] P. Mastin, "How latency is killing online gaming," April 2016. [Online]. Available: https://venturebeat.com/2016/04/17/how-latency-is-killing-online-gaming/

[13] Y. Einav, "Amazon Found Every 100ms of Latency Cost them 1% in Sales," January 2019. [Online]. Available: https://www.gigaspaces.com/blog/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/

[14] Gigabyte, "An Autonomous Vehicles Network with 5G URLLC Technology." [Online]. Available: https://www.gigabyte.com/Solutions/Networking/urllc

[15] S. Panwar, "Breaking the millisecond barrier: Robots and self-driving cars will need completely reengineered networks," *IEEE Spectrum*, vol. 57, no. 11, pp. 44–49, 2020.

[16] J. Kornegay, "Why Is Latency So Important In Financial Networks?" July 2018. [Online]. Available: https://www.gigabyte.com/Solutions/Networking/urllc

[17] B. Guo, Y. Shang, Y. Zhang, W. Li, S. Yin, Y. Zhang, and S. Huang, "Timeslot switching-based optical bypass in data center for intrarack elephant flow with an ultrafast DPDK-enabled timeslot allocator," *Journal of Lightwave Technology*, vol. 37, no. 10, pp. 2253–2260, 2019.

[18] B. Guo, S. Li, S. Yin, and S. Huang, "TDM based optical bypass for intra-rack elephant flow with a DPDK based online timeslot allocator," in *Proc. Optical Fiber Communication Conference*, Los Angeles, USA, 2017, pp. W1H–4.

[19] K. Nielsen, "Submarine telecoms industry report," Tech. Rep., 2020. [Online]. Available: https://subtelforum.com/products/submarine-cable-almanac/

[20] A. Agrawal, V. Bhatia, and S. Prakash, "Network and risk modeling for disaster survivability analysis of backbone optical communication networks," *Journal of Lightwave Technology*, vol. 37, no. 10, pp. 2352–2362, 2019.

[21] W. Wu, B. Moran, J. H. Manton, and M. Zukerman, "Topology design of undersea cables considering survivability under major disasters," in *Proc. International Conference on Advanced Information Networking and Applications Workshops 2009*, Bradford, UK, May 2009, pp. 1154–1159.

[22] C. Cao, M. Zukerman, W. Wu, J. H. Manton, and B. Moran, "Survivable topology design of submarine networks," *Journal of lightwave technology*, vol. 31, no. 5, pp. 715–730, 2013.

[23] C. Cao, Z. Wang, M. Zukerman, J. Manton, A. Bensoussan, and Y. Wang, "Optimal cable laying across an earthquake fault line considering elliptical failures," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1536–1550, Sept. 2016.

[24] M. Zhao, T. W. S. Chow, P. Tang, Z. Wang, J. Guo, and M. Zukerman, "Route selection for cabling considering cost minimization and earth-

quake survivability via a semi-supervised probabilistic model," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 502–511, Apr. 2017.

[25] D. L. Msongaleli, F. Dikbiyik, M. Zukerman, and B. Mukherjee, "Disaster-aware submarine fiber-optic cable deployment for mesh networks," *Journal of Lightwave Technology*, vol. 34, no. 18, pp. 4293–4303, 2016.

[26] Z. Wang, Q. Wang, M. Zukerman, J. Guo, Y. Wang, G. Wang, J. Yang, and B. Moran, "Multiobjective path optimization for critical infrastructure links with consideration to seismic resilience," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 836–855, 2017.

[27] Q. Wang, Z. Wang, J. Guo, E. Tahchi, X. Wang, B. Moran, and M. Zukerman, "Path planning of submarine cables," in *Proc. 2019 21st International Conference on Transparent Optical Networks*, Angers, France, 2019, pp. 1–4.

[28] Z. Wang, Q. Wang, B. Moran, and M. Zukerman, "Optimal submarine cable path planning and trunk-and-branch tree network topology design," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1562–1572, 2020.

[29] P. N. Tran and H. Saito, "Enhancing physical network robustness against earthquake disasters with additional links," *Journal of Lightwave Technology*, vol. 34, no. 22, pp. 5226–5238, 2016.

[30] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[31] J. Oh, I. Pyo, and M. Pedram, "Constructing minimal spanning/steiner trees with bounded path length," *INTEGRATION, the VLSI journal*, vol. 22, no. 1-2, pp. 137–163, 1997.

[32] R. K. Martin, "Using separation algorithms to generate mixed integer model reformulations," *Operations Research Letters*, vol. 10, no. 3, pp. 119–128, 1991.

[33] N. Fan and M. Golari, "Integer programming formulations for minimum spanning forests and connected components in sparse graphs," in *Proc. International Conference on Combinatorial Optimization and Applications*, Hawaii, USA, 2014, pp. 613–622.

[34] M. A. Dara, A. Fischera, J. Martinovica, and G. Scheithauera, "Integer linear programming formulations for the minimum connectivity inference problem and model reduction principles," Dresden University of Technology, Tech. Rep. Preprint MATH-NM-03-2019, 2019.

[35] S. Mitchell, M. OSullivan, and I. Dunning, "PuLP: a linear programming toolkit for python," *The University of Auckland, Auckland, New Zealand*, 2011.

[36] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," *Journal of the ACM*, vol. 49, no. 1, pp. 16–34, 2002.

[37] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[38] Marine Geoscience Data System, "Global Multi-Resolution Topography Data Synthesis," October 2019. [Online]. Available: https://www.gmrt.org/