Suboptimal PON Network Designing Algorithm for Minimizing Deployment Cost of Optical Fiber Cables

Akira Agata, and Kosuke Nishimura

KDDI R&D Laboratories Inc., 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
Email: agata@kddilabs.jp

Abstract—In order to meet the bandwidth requirements for the access network, many network operators nowadays have been rapidly expanding their fiber-to-the-home (FTTH) service area. In such an area, a passive double star (PDS) network, which shares one optical fiber with multiple subscribers by using a power splitter, is commonly deployed as an infrastructure for the passive optical network (PON) systems. One of the main focuses for PON network planning is to determine the locations of the optical splitters and optical fiber cable routes that connect every splitter to the central offices (COs) based on the forecasted demands, within a limited deployment cost under realistic restrictions. In this paper, we propose and demonstrate a novel suboptimal design algorithm of PON optical distribution network (ODN). Based on the forecasted demand, it can automatically generate a suboptimal PON network in terms of the total cable deployment construction length under realistic restrictions.

Index Terms—Network topology, Network theory, Optimization method

I. INTRODUCTION

The rapid increase in demand for broadband access services has led to widespread use of passive optical networks (PONs), such as Ethernet PON (EPON) [1] [2] or gigabit PON (GPON) [3] [4], as fiber-to-the-home (FTTH) networks. As the FTTH deployments have been accelerating, a large number of optical fiber cables are being installed day by day to meet the demand. Since PON is a point-to-multipoint fiber optical network with no active components in the optical distribution networks (ODNs), it has been considered as a cost-effective and reliable optical access system. Compared to a point-to-point system however, optimal design of a PON ODN requires considering many complex factors, such as optical splitter positioning and its splitting ratio, locations of the central offices (COs), and the set of given possible paths on which the optical fiber cable could be placed.

There have been only a few reports studying the suboptimal point-to-multipoint network planning algorithm [5] [6]. However, these algorithms assumed that it is possible to establish additional optical fiber routes other than the given possible paths when connecting any pair of nodes [5], or when avoiding obstacles [6]. In actual use, the PON ODN should be designed based on the set of given possible paths. This is due to the fact that most paths where optical fiber cables can be deployed are limited for example to the paths among existing power poles or those along a road especially in urban areas of many Asian countries.

We previously proposed an algorithm that can generate a suboptimal ODN that connects every optical network unit (ONU) at the subscriber side to the optical line terminal (OLT) through a power splitter within a total optical fiber length that is close to the shortest value, when the locations of the COs, subscribers, and set of paths where the optical fiber cable can be installed are given [7]. However, so as to describe the basic idea of the algorithm, we have intentionally ignored the construction cost and have demonstrated employing a relatively simple model (277 possible fiber paths, 100 subscribers, and 2 COs).

In this paper, we propose a novel ODN planning algorithm for PON systems that takes into account the construction costs which are generally several times higher than the cost of the cable itself, while almost independent on the total number of deployed cables. The proposed algorithm was implemented in prototype software, and we have confirmed its effectiveness by testing it using a realistic model involving a real road map in Japanese urban area and a realistic number of forecasted demand.

II. PROPOSED PON ODN DESIGN ALGORITHM

In many cases, a network designer of PON systems has to design the point-to-multipoint optical fiber network under many restrictions such as locations of COs where OLTs are installed, those of subscribers, and the paths where the optical fiber cable could be placed. In this section, we propose a PON network planning algorithm for solving this issue. The proposed algorithm uses two graph related algorithms as key elemental techniques. One is the graph clustering algorithm. The other is the construction method of “Steuber tree” in graph.

A. Elemental Technology (1) Graph Clustering

Cluster analysis is based on the assignment of a large number of data into multiple groups (called “clusters”) so that data in the same cluster are similar with each other in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields. Among many clustering techniques, k-means algorithm, which classify n data into k clusters (k<n) based on Euclidean distance, is the most well known and commonly used clustering algorithm [8] [9]. The k-means algorithm is a fairly simple data clustering technique which minimizes the average distance between data and its cluster center. The number of clusters to form, k, is an input parameter to the algorithm. Given a set of points \( x = \{x_1, x_2, ..., x_d\} \), where each point is a \( d\)-dimensional real vector, k-means clustering aims to partition the n points into k clusters \( k \leq n \) \( S = \{S_1, S_2, ..., S_k\} \) so as to
minimize the within-cluster sum of squares (WCSS):

$$\arg \min_S \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2$$  \hspace{1cm} (1)$$

Where $\mu_i$ is the center of the $i$-th cluster $S_i$.

The graph clustering algorithm is an extension of the normal clustering algorithm to the graph [8] [10]. It is obtained from replacing an Euclidean distance in a plane with a shortest-path distance in a graph. As a result, the graph clustering procedure becomes as follows:

**Step 1:** Randomly designate $k$ instances to serve as “seeds” for the $k$ clusters.

**Step 2:** Assign the remaining data points to the cluster of the nearest seed using Dijkstra shortest-path algorithm [11].

**B. Elemental Technology (2) Steiner Tree in Graphs**

When the network designer plans to deploy PON services in a new area, the basic ODN architecture including the locations of optical splitters and the routes of optical fiber cable that connect every splitter to the CO must be determined based on the forecasted demand prior to starting the services. In many cases, the network designer has to design such an optical fiber network under many restrictions such as the CO location (i.e. the OLT location), the locations where the optical splitter could be placed, and the paths where the optical fiber cable can be deployed. Particularly in many Asian countries, the both ends of these paths are restricted by the locations of existing power poles. The network designer must also consider that the installation construction cost of optical fiber cable(s) is higher than the cost of the cable itself per unit length. In view of these restrictions, the network designer has to design a basic ODN plant whose total cable deployment construction length is as short as possible.

We previously showed that the location of the optical splitter can be determined by using a modified graph clustering algorithm when the set of paths where the optical fiber cable can be installed are modeled as a graph $G(N,E)$ [7]. In this graph $G$, the set of nodes $N$ and edges $E$ represent the locations of existing power poles and the possible cable paths, respectively. Once the locations of all optical splitters are determined, the optical fiber cable network that connects every splitter to the CO should be designed. An optimum network in terms of total optical fiber length can be designed by using a shortest path search algorithm [7], such as Dijkstra shortest-path algorithm, as shown in Fig. 1(a). However, the construction cost for installing optical fiber cable per unit length is several times higher than the cable cost while it is almost independent on the total number of deployed cables. Therefore, in order to design a network that takes the construction costs into account, the ODN planning algorithm should contrive an optical fiber cable network that connects every splitter to the CO with the sum of the cable deployment construction length as short as possible, as schematically illustrated in Fig. 1(b).

This type of problem can be treated as a Steiner tree problem in graphs [12]. In the Steiner tree problem in graphs, we are given an undirected graph $G = (N,E)$ with positive edge costs and a set $T \subseteq N$ of terminals, and the goal is to find the minimum-cost tree connecting all the terminals. This problem is known to be a non-deterministic polynomial time hard (NP hard) problem [12]. The most popular algorithm for obtaining a Steiner tree in polynomial time is the Dreyfus-Wagner's algorithm [13].

![Fig. 1. Optimum ODN architecture in terms of: (a) Total fiber length, (b) Total construction length](image)

<table>
<thead>
<tr>
<th>ODN 1</th>
<th>Total Fiber Length</th>
<th>Total Construction Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODN 1</td>
<td>$2 \times L_1$</td>
<td>$2 \times L_1$</td>
</tr>
<tr>
<td>ODN 2</td>
<td>$(2 \times L_2)+(2 \times L_3)$</td>
<td>$L_2+(2 \times L_3)$</td>
</tr>
</tbody>
</table>
Step 4: Check the total number of $2^{nd}$ stage optical splitters in the $i$-th cluster ($N_i$). If $N_i$ is larger than the maximum splitting ratio of the $1^{st}$ stage optical splitter, then increment $N_i$ and go back to the step 3.

Step 5: Set the cluster medoid as the location of the $1^{st}$ stage optical splitter, and define the cable route between the $1^{st}$ and $2^{nd}$ stage optical splitters.

Step 6: Assuming all the $1^{st}$ stage optical splitters and CO as the terminals, construct a Steiner tree in $G = (N,E)$, where edge weight is set to the installation construction cost, by using an algorithm for Steiner tree in graph problem, such as the Dreyfus-Wagner's method.

Step 7: Set the calculated Steiner tree as the cable route between the CO and $1^{st}$ stage optical splitters.

D. Numerical Simulation

To evaluate the feasibility of the proposed algorithm, we conducted a numerical simulation. In this evaluation, we first used a Delaunay triangulation graph [14] of randomly deployed 1,000 nodes in the square area of 5 km per side to approximate a road network in one service area. In this process, we generated a road network with 2,930 roads and 1,000 intersections. Since no two edges intersect with each other in Delaunay triangulation, the graph is assumed to represent a realistic road network. The assumed road network is shown in Fig. 2(a). Next, we placed power poles along the road. In this process, we assumed that: (1) each intersection of the road network has one power pole and (2) the average spacing between two power poles is 30m. We then obtained a graph $G(N,E)$ consisting of the possible optical fiber paths ($E$), which are placed along the road, between the 15,650 power poles ($N$). In order to reduce the complexity of the figure, only power poles on the crossections are illustrated in Fig. 2(a).

First, we randomly selected one node in the network as the location of the CO (the square at the center of Fig. 2). We next applied our algorithm to the network, assuming the estimated demand $d(x,y)$ is constant and the total number of estimated subscribers in the area is 1,000. After the step 2 and 3, the power poles are clustered and the locations of $2^{nd}$ stage optical splitters are determined. The total number of the required splitters to accommodate all estimated subscribers was calculated to be 157. In these processes, the splitting ratio of the splitter and the maximum length of an aerial lead-in line were set to 1:8 and 400m, respectively. Next, the location of $1^{st}$ stage optical splitters was determined by applying the step 4 and 5. In these processes, we assumed the maximum splitting ratio of $1^{st}$ stage optical splitters as 1:4. The total number of the $1^{st}$ stage splitters was calculated to be 40.

Fig. 2(b) shows the calculated locations of the $1^{st}$ stage optical splitters and their cover area. Finally, the cable route between CO and every $1^{st}$ stage optical splitters are calculated by applying the step 6 and 7, as shown in Fig. 2(c). In our evaluation, we used the GOBLIN Graph Library [15] to solve the Steiner tree problem in graph.

![Fig. 2 Designing process of PON ODN](image)

(a) Given road network and CO, (b) Determining the location of the $1^{st}$ stage optical splitters and their cover area, (c) Designing the cable routes between CO and the optical splitters

<table>
<thead>
<tr>
<th></th>
<th>Total Fiber Length [km]</th>
<th>Total Construction Length [km]</th>
<th>Relative Cost [km$^{-1}$]</th>
<th>Total Relative Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>134.5</td>
<td>27.5</td>
<td>8</td>
<td>354.5</td>
</tr>
<tr>
<td>Conventional</td>
<td>79.9</td>
<td>49.9</td>
<td>8</td>
<td>479.1</td>
</tr>
</tbody>
</table>
To confirm the feasibility of the proposed algorithm, we compared the total deployment cost of the PON ODN between the proposed algorithm and the method in [7]. Since the installation construction cost of optical fiber cable(s) are generally several times higher than the cost of the cable itself, in this evaluation, we assumed that the construction cost to deploy optical fiber cable(s) per unit length is 8 times higher than the cable cost and does not depend on the total number of deployed cables for simplicity. The results are summarized in Table 2. The results indicate that the proposed algorithm can generate more cost-effective PON ODN under the situation where a relatively higher construction cost and a realistic complexity of power pole network in urban area is assumed.

### III. IMPLEMENTATION OF THE PROTOTYPE

#### A. Acceleration of the Steiner Tree Algorithm

When network designers plan to deploy PON services in a new area, sometimes they have to design a large scale PON ODN involving several tens of optical splitters. In such a case, the main drawback will be the calculation time of the Steiner tree which connects all the optical splitters and the CO with the minimum total cost. Assuming an undirected graph $G = (N,E)$ with positive edge costs and a set $T \subseteq N$ of terminals, it has been known that the Dreyfus-Wagner's algorithm can compute the Steiner tree with order of $O(t^3)$, where $t$ is the total number of the terminals ($t = |T|$) [13]. As a result, the computational time drastically increases as the total number of optical splitters increases. Therefore, in order to implement a realistic PON ODN designing tool, it is indispensable to reduce the complexity of the algorithm.

In order to overcome this issue, we took notice of the fact that the shortest path between two terminals in a Steiner tree is often the same as the shortest path in an original graph $G$. By using this characteristic, suboptimal Steiner tree would be calculated by nesting Dijkstra shortest-path algorithm with the order of polynomial time. Based on this assumption, we propose a novel algorithm to calculate suboptimal Steiner tree as shown below:

**Step 1:** In the graph $G = (N,E)$, let $P_{ij}$ be the shortest path between the $i$-th terminal ($t_i$) and the $j$-th terminal ($t_j$), where $1 \leq i, j \leq |T|$.

**Step 2:** Let $G' = (N', E')$ be an empty graph.

**Step 3:** Calculate $P_{ij}$ for all $(i, j)$ pairs by using Dijkstra shortest-path algorithm in the graph $G$.

**Step 4:** Find the $(i, j)$ pair which gives the minimum positive (greater than zero) path-cost which is defined as a sum of the weights along the path.

**Step 5:** Add all the nodes and edges in the path $P_{ij}$ to the graph $G'$.

**Step 6:** Change the weight of all the edges in the path $P_{ij}$ to zero in the graph $G$.

**Step 7:** Check the total number terminals in the graph $G'$. If all the terminals have been added to $G'$, return the graph $G'$ as the suboptimal Steiner tree. Else, go back to the step 3.

Although we used nested Dijkstra method, this proposed algorithm can compute the suboptimal Steiner tree with order of $O(t^2)$ which should be much faster than conventional Dreyfus-Wagner's method.

![Fig. 3. Total CPU time for both the Dreyfus-Wagner and our proposed algorithm as a function of the number of terminals with the error bar of 99% confidence intervals.](image)

**Table 3: Comparison of total relative cost**

<table>
<thead>
<tr>
<th>Number of terminals ($t$)</th>
<th>Relative cost of the suboptimal Steiner tree (Dreyfus-Wagner’s algorithm = 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
</tr>
<tr>
<td>2</td>
<td>100.0</td>
</tr>
<tr>
<td>5</td>
<td>100.0</td>
</tr>
<tr>
<td>8</td>
<td>102.4</td>
</tr>
<tr>
<td>11</td>
<td>100.0</td>
</tr>
<tr>
<td>14</td>
<td>101.0</td>
</tr>
<tr>
<td>15</td>
<td>100.0</td>
</tr>
<tr>
<td>16</td>
<td>100.1</td>
</tr>
</tbody>
</table>

To confirm the effectiveness of the proposed algorithm, we compared the total cost and calculation time with the Dreyfus-Wagner’s algorithm by changing the total number of terminals. In this evaluation, we used the same road network as shown in Fig. 2(a). After randomly select 2 to 16 nodes as terminals, we calculated the Steiner tree by using both the algorithms. The CPU time was evaluated on an Intel Core2 Quad 3.0 GHz processor with 2.0 GByte of main memory, under the Windows XP operating system. In order to improve accuracy, we randomly changed the location of terminals and repeated the calculation 10 times for each $t$ value. Fig. 3 shows total CPU time for both the Dreyfus-Wagner and the proposed algorithm as a function of the number of terminals with the error bar of 99% confidence intervals. As shown in this figure, CPU time of the Dreyfus-Wagner algorithm rapidly increases as a function of the number of terminals. On the other hand, the proposed algorithm can drastically reduce the calculation time. Table 3 summarizes the relative cost of the suboptimal Steiner tree calculated by the proposed algorithm compared to the Steiner tree obtained by the Dreyfus-Wagner algorithm.
The results indicate that the proposed algorithm can generate the large scale PON ODN the processing time practically short enough, while with almost the same deployment cost compared to the Dreyfus-Wagner’s algorithm.

B. Implementation and Evaluation of the Designing Tool

Finally, we implemented the suboptimal PON ODN designing tool which generates a suboptimal network in terms of the total cable deployment construction length based on the forecasted demand and a real road map. In order to reduce the calculation time, we used the proposed suboptimal Steiner tree algorithm in place of the Dreyfus-Wagner’s method, as described in the previous section.

The graphical user interfaces (GUIs) of the prototype software are shown in Fig. 4. The network designer should take the following three steps by making use of the GUIs:

**Step 1:** First, the user chooses the target area in square by dragging on the real road map, and separates it into \( n \times m \) square mesh (Fig. 4(a)).

**Step 2:** Next, the user forms several sub-areas which encompasses the target area by concatenating the square meshes, and assigns the forecasted demand for each sub-area (Fig. 4(b)).

**Step 3:** Based on these inputs, the system automatically generates the suboptimal PON ODN in terms of the total cable deployment construction length (Fig. 4(c)).

In the example depicted in Fig. 4, we selected the square area of 2 km per side from a real road map in an urban area, and assumed the total number of forecasted demand in the area is 1,000. The maximum splitting ratio, maximum number of splitters per power pole, and maximum drop fiber length were 1:64, 4, and 200 m, respectively.

Throughout this study, we confirmed that the proposed algorithm can automatically generates the suboptimal PON ODN in terms of the total cable deployment construction length, even when the area consists of mixed sub-area of high and low forecasted demand and road density.

IV. CONCLUSION

We proposed and demonstrated the novel PON ODN planning algorithm. By assuming the set of possible fiber paths as a graph, and applying the combination of Steiner tree and clustering technique, our proposed algorithm can automatically generate the suboptimal ODNs based on the forecasted demand with the sum of the cable deployment construction length as short as possible. Our method also considers the splitting ratios of the optical splitters. Throughout the numerical simulations and prototype test, we confirmed that the proposed algorithm are applicable to the design of a suboptimal PON ODN in terms of the sum of the cable deployment construction length especially when a realistically complex road and power pole network in an urban area was assumed.
ACKNOWLEDGMENT

The authors wish to thank Drs. Y. Nakajima, M. Suzuki, and Y. Takeuchi in KDDI R&D Laboratories Inc. for their continuous encouragements.

REFERENCES


