Paper

FTTH Network Optimization

Hoang Nghia Le

Suntech S.A., Warsaw, Poland

Abstract—Fiber To The Home (FTTH) is the most ambitious among optical technologies applied in the access segment of telecommunications networks. The main issues of deploying FTTH are the device price and the installation cost. Whilst the costs of optical devices are gradually decreasing, the cost of optical cable installation remains challenging. In this paper, the problem of optimization that has practical application for FTTH networks is presented. Because the problem is Non-deterministic polynomial-time hard (NP-hard), an approximation algorithm to solve it is proposed. The author has developed the algorithm in a C# program in order to analyze its performance. The analysis confirms that the algorithm gains near-optimal results with acceptable time consumption. Therefore, the algorithm to be applied in a network design tool for FTTH network planning is proposed.

Keywords—cost optimization, development optimization, FTTH, optical technology, telecommunications network.

1. Introduction

Unlike in other technologies, in FTTH the digital data stream is transmitted through the optical medium directly to the subscriber's terminal. Thanks to this approach, FTTH allows for data transmission with higher speed and better quality than other networks based on radio, copper, coaxial and optical-coaxial mixed technologies. Moreover, FTTH is future proofed – a greater transmission speed requires only faster terminals and routers, with the fibers remaining unchanged.

However, nowadays the FTTH development in the world (also in Europe) is unequal. In annual ranking of FTTH Council based on FTTH coverage (the percentage of Internet subscribers using FTTH) in particular countries [1], apart from about 20 countries (led by South Korea, United Arab Emirates and Japan) with FTTH technology widely developed (FTTH coverage larger than 25%), for the rest of the world this key factor is still less than 5%. The problem is due to economic issues – FTTH requires high cost of devices and optical cable installation. Below a brief overview of the technology FTTH is presented.

The FTTH network starts from an Optical Line Terminal (OLT), an endpoint of the core network (Fig. 1). The optical signals that carry the digital data streams to the subscribers are firstly transmitted in a common cable. This is possible thanks to Wave Division Multiplexing technology (WDM), which allows a fiber to simultaneously transmit several tens of optical signals, each of them in a separate waveband. Later, the signals are split into different routes, by means of splitters, and end in the Optical Network Units (ONUs) installed in the subscribers' homes. Unlike other networks, in an FTTH network copper cables are not used. Therefore, optical to electrical conversion, which severely lowers the limit of data transmission speed for a single user, is not needed.

Whilst the prices of optical devices (OLTs and ONUs), the most problematic issue so far, are gradually decreasing, the cost of optical cable installation remains unchanged. The cost is made up of several factors as:

– the duct digging,

– the drilling the conduits in new or existing ducts,

– the laying the cables in new or existing conduits,

– the cable.

In this paper, a new method for FTTH network optimization with the focus on cable installation minimization is proposed. After analyzing the previous work, the FTTH network optimization issues as a mathematical problem is presented, called the problem of FTTH Network Optimization (FNO). An exact algorithm to solve FNO is proposed, modeling it as an Integer Linear Programming (ILP) problem. Due to the NP-hardness of FNO, the exact algorithm allows to solve the problem only for small FNO instances (with less than ten subscribers). Therefore, for larger FNO instances, the author proposes an approximation algorithm. Although the algorithm does not guarantee the ideal solution, it allows for finding the solution closed to optimum. The algorithm computation complexity is polynomial, which allows for its application in computer system with acceptable cost for FTTH network planners. In order to evaluate presented algorithms, the result of their operations in examples is presented.

2. Related Work

The optimization issues in FTTH network planning are considered in various surveys, among others [2]. Detailed studies about FTTH optimization focus on 3 areas:

– device installation optimization [3],

– cable installation optimization [4]–[7],

– bandwidth utilization optimization in an FTTH existing network [8].

In [3], the network cost to be optimized is composed of two parts: CAPEX – cost of device installation, and OPEX – cost of network element maintenance. The net-

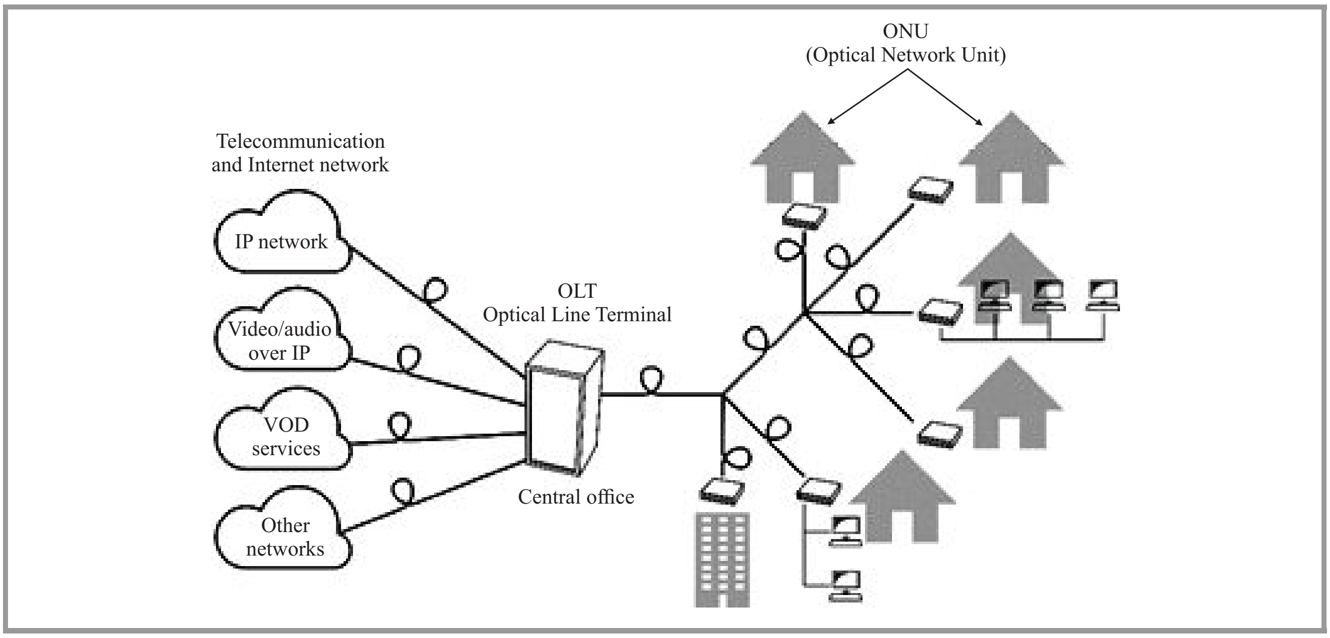

Fig. 1. FTTH network structure.

work optimization is modeled as an Integer Linear Programming (ILP) problem. Due to the NP-hardness property of the problem, an approximated solution is proposed. Cable installation cost optimization is not considered in the paper.

In [4]–[7], the cable installation cost optimization issues are considered. However, the authors have limited the research scope to the problem of selection of the best cable routes from among the predefined ones. Thus, this approach makes the result strongly depended on the subjectively predefined cable routes. In [7], the robustness of the network is also considered. The network should operate without loss after a failure of a single cable. In [8], another category of optical network optimization is considered: optimization of bandwidth distribution in an existing FTTH network.

After thorough analysis of the previous studies, it could be concluded that the problem of FTTH network optimization in the full format of geographical optimization – finding the optimal allocation of devices and cables in a given area, with the geographical aspects considered has not been resolved yet. Therefore, the aim of this study is to fill this research gap.

Geographical optimization is applied in various areas, including energy, transport, industry, and also in telecommunications (especially for core networks). The basic geographical optimization problems are the problems of Euclidean Traveling Salesman (ETS) and Euclidean Steiner Tree (EST) [9], [10]. The first of these problems is to find the shortest tour to visit a set of points in a plane, whilst the later problem is to find the shortest tree-network that connects a set of points in a plane.

The author has analyzed the attempts to extend EST problem with geometrical conditions in previous studies [11], [12]. In those studies, the algorithms to find the EST that avoid given obstacle areas are proposed. In this paper, the problem further by considering the following aspects is extended:

- The geographical aspect. In EST, the optimization objective is to find the shortest network. In practice, it may not be the network with the lowest cost, if some of its links have to be provided across areas, where cable installation is difficult or impossible. In this work, the cable installation cost resulting from the terrain condition is assigned to each point of the plane and the algorithm finds the network with the lowest total cost.
- The cable capacity limitation.
- The existing network resources. This extension is essential for using the algorithm in frequent practical situation, in which the network planner task is rather to develop an existing network than to create a new network for a "green-field".

3. FTTH Network Optimization Problem

In this section, the formal description of the FTTH network optimization problem (FNO) is presented.
It starts with geometrical input data:

- finite set of source points in the plane $S \subset R^2$ (a source point represents an OLT);
- finite set of destination points in the plane space $D \subset R^2$ (a destination point represents an ONU);
- finite set of existing transit points in the plane $T_1 \subset R^2$ (a transit point represents a splitter);

- finite set of lines connecting the points in the sets S, D and $T_1 : E_1 \subset (S \cup D \cup T_1)^2$ (a line represents a linear segment of a cable).

In Fig. 2 the model of an FTTH network is presented.

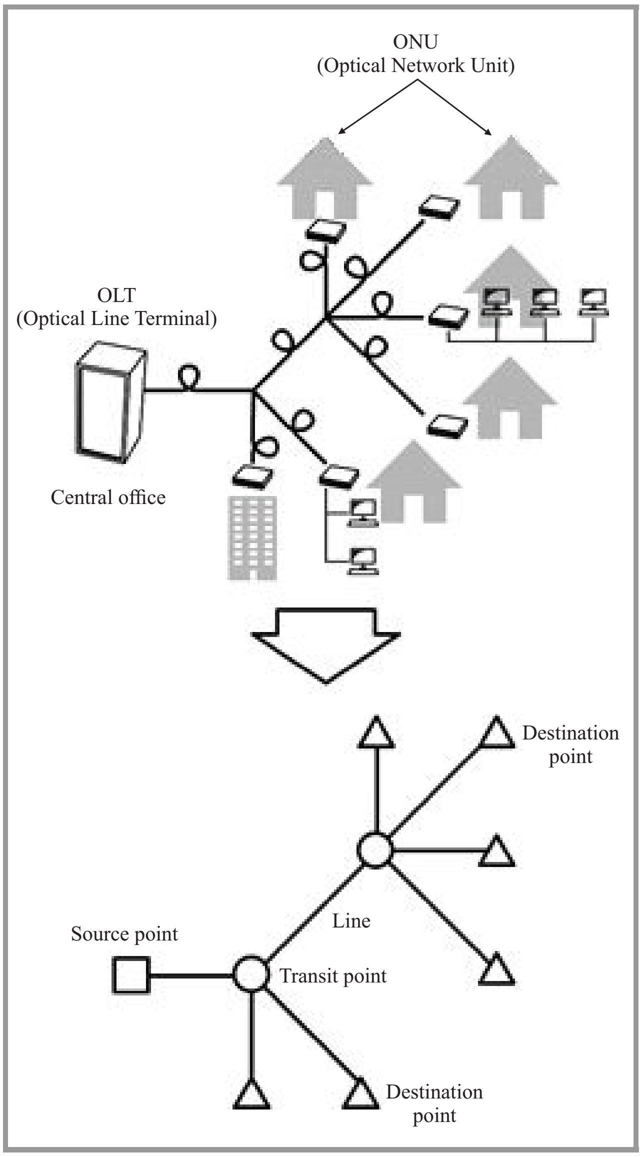

Fig. 2. Modeling FTTH network in FNO problem.

Other input data:

- Cable capacity C. It represents the maximum number of WDM channels that can be transmitted via the cable;
- Density function of cable installation cost in the area g: $R^2 \rightarrow R$. The cost of cable installation depends on the terrain of the area and the infrastructure existing in the area. The cost of cable installation can be: quite low if the cable is laid along a road, quite high if the cable is laid across a road, or infinite if the cable has to cross a building or a housing estate (impossible to install).

Function $g(x)$ describes the installation cable cost with a unit length at the point $x \in R^2$. Having the function g, the installation cost of a cable e by means of the following integral can be calculated:

$$f(e) = \int_{r=0}^{1} g\big((1-r)\cdot \text{in}(e) + r \cdot \text{out}(e)\big)\text{d}r, \qquad (1)$$

where $\text{in}(e)$ denotes the input point of line e, $\text{out}(e)$ denotes the output point of line e.
It is assumed that function g is described by means of a finite and sorted collection of polygons, each of which has an assigned cost value, and called the g-polygons. The value $g(x)$ for point x in R^2 is the cost of the first g-polygon that contains x. If x does not belong to any g-polygon, the $g(x) = 1$. A g-polygon represents an area with the same cost. For example: a roadside (low cost), a road (high cost), a building (infinite cost).
In Fig. 3, the cost of cable installation from A to B is $f(\text{AB}) = ||\text{AT}_1|| + ||\text{T}_1\text{T}_2|| \cdot m_1 + ||\text{T}_2\text{T}_3|| \cdot m_2 + ||\text{T}_3\text{T}_4|| \cdot m_1 + ||\text{T}_4\text{B}||$, where m_1, m_2, m_3 are the costs of the highlighted areas, respectively.

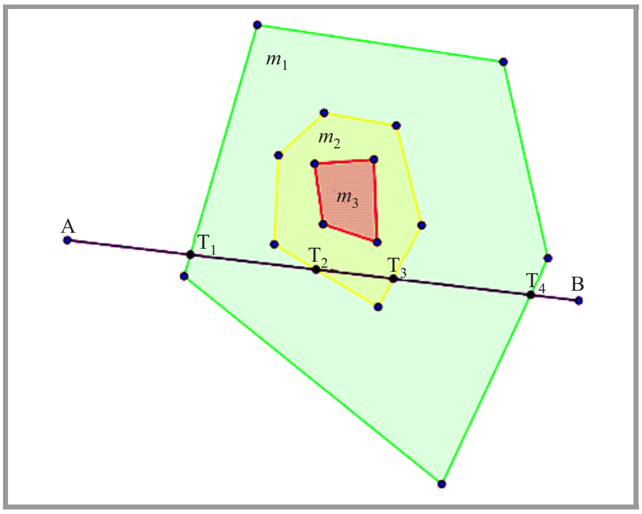

Fig. 3. The cost of installing cables through g-polygons.

The next step is to find:

- finite set of transit points $T \subset R^2$,
- finite set of lines connecting the points in the sets S, D and T: $E \subset (S \cup D \cup T)^2$,
- finite set of paths (sequences of connected lines) $P = \{p : p = (e_1, e_2, \ldots, e_n);\ \forall i = 1 \ldots n,\ e_i \in E$, where $\forall j = 1 \ldots n-1,\ \text{out}(e_j) = \text{in}(e_{j+1})\}$.

Conditions:

- The **connectivity condition**. Each destination point is connected to at least one source point, either directly via one line or a sequence of connected lines. In other words, for each point $d \in D$, there exists a path $p \in P$ so that $\text{in}(p) \in S$, $\text{out}(p) = d$, where in(p) denotes the input point of the first line of path p, and out(p) denotes the output point of the last line of path p;

- The **capacity condition**. The number of path containing a line should not exceed the predefined capacity of the line: $\forall e \in E$, $\sum_{p\in P:e\in p} 1 \leq C$. In the context of WDM technology, C is the number of channels that can be transmitted through an optical cable.

Optimization goal:

- The optimization objective is to get seek solution with the least total cost of installation of the cables:

$$\text{Minimize} \sum_{e\in E\setminus E_1} f(e) . \quad (2)$$

4. Exact Algorithm for FNO

In this paper, the two algorithms for FNO are proposed. The main one is an approximation algorithm that allows for effective finding of a near optimal solution of the problem. In addition an exact algorithm modeled as an Integer Linear Programming (ILP) is proposed, which is effective only for small problem instances. The exact algorithm is used to evaluate the quality of the solution obtained from the main algorithm.
FNO is strongly related to the family of Steiner Tree Problems (STP). Apart from the EST already mentioned earlier, FNO is also related to the topological version of the STP. The descriptions of the two problems are as follows:

- the Topological Steiner Tree (TST): given a graph consisting of node set V and link set E (each link has a cost value assigned), interconnect a given subset T of V by a sub-graph with the shortest total link cost;
- the Euclidean Steiner Tree (EST): given n points in the plane (called the terminal points), connect them by line segments of the minimum total length in such a way that any two terminal points may be interconnected by the line segments either directly or via other points (called the Steiner points).

TST will be used multiple times as a subroutine of the algorithms for FNO, whilst EST is a special case of FNO when S is a one-single-element set, g is a constant function, and C is larger than $|D|$.
Because EST has been intensively studied in the last century, the author will construct the exact algorithm using the knowledge collected in the studies on EST. EST is proved to be an NP-hard problem [13].
Transferring the EST and its derived problems, like FNO, into an ILP form is difficult, because the Steiner points to be found are derived from a continuum and unconstrained set, whilst an ILP is assumed to have discrete and limited set of variables and equalities. Therefore, in order to transfer FNO into an ILP, there is need to "discretize" the problem data space. In other words, the aim is to construct a limited and discrete set of points that contains the sought Steiner points. The author calls these points the candidate points.

4.1. Candidate Points for EST and FNO

There are two approaches for discretizing the space of EST deriving problems:

- by means of a grid of pixels on the plane – the candidate points are defined as the vertices of the grid;
- by generating the candidate points, based on geometrical properties of EST.

The first approach is practical for an approximation algorithm. However, it does not allow for finding the exact solution. The quality of the solution depends on the grid granularity. The second approach is used in presented exact algorithm.
Firstly, the set of candidate points for EST is found, then the set of candidate points for FNO is extended. The symbol EST(n) is used to denote the EST of the given value of n.
EST(2) is trivial. The solution consists of the line segment connecting the two given points.
EST(3) is equivalent to the Fermat Point Problem – given a triangle ΔABC, find a point F in the plane such that the total distance from the three vertices of the triangle to the point is minimum. The problem was first raised in year 1643 by the famous French mathematician Fermat, as a challenge to the Italian mathematician Torricelli. Fermat resolved the problem (Fig. 4) by:

- constructing 3 equilateral triangles (ΔBCA′, ΔACB′ and ΔABC′), each of which shares an edge of the given triangle;
- constructing the line segments AA′, BB′ and CC′, which meet at one point. If this point belongs to the ΔABC area, it will be the Fermat point.

Torricelli announced another, equivalent, solution (see Fig. 5):

- construct a point P, so that ΔBCP is an equilateral triangle which does not cut ΔABC. P is called the Torricelli substituting point of the pair of points B and C;
- construct the circumscribed circle of ΔBCP. We call the short arc BC of this circle the Torricelli arc of BC;
- if line AP will meet the Torricelli arc of BC at a point, this point will be the Fermat point.

The Torricelli solution is more useful for further solving the EST(n) problem with $n > 3$. The solution is based on the fact that for any point T lying on the Torricelli arc, the following equation holds: $|\text{TP}| = |\text{TB}| + |\text{TC}|$. There-

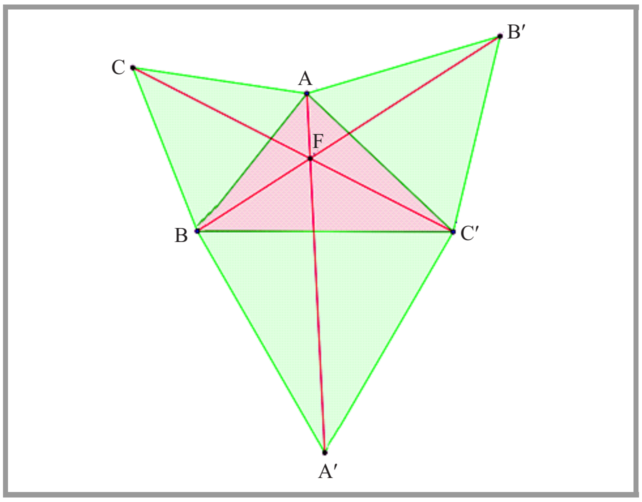

Fig. 4. Fermat Point Problem and its original solution.

fore, the sum of $|TA| + |TB| + |TC|$ will be minimized if $|TA| + |TP|$ is minimized, which will happen if T belongs on AP.

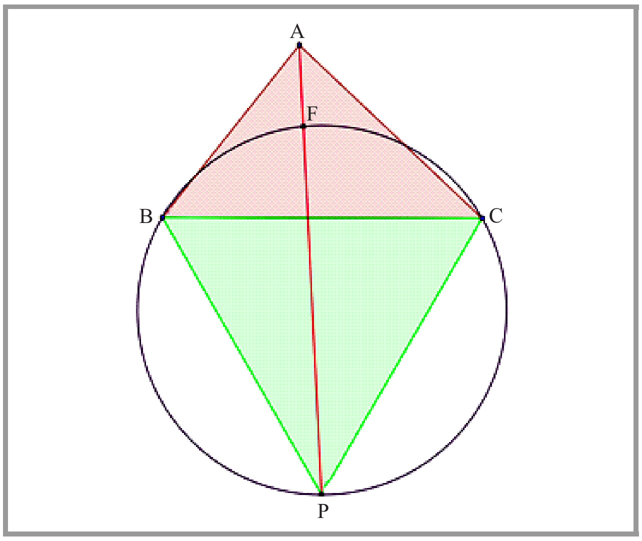

Fig. 5. Torricelli solution for Fermat Point Problem.

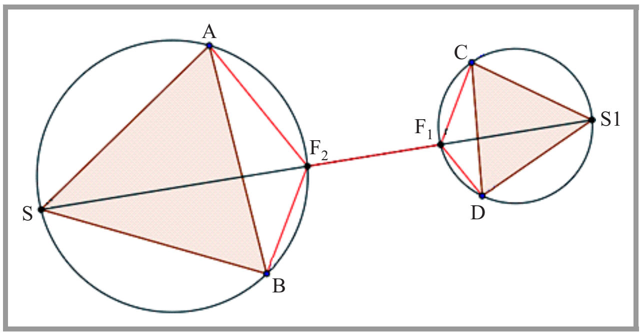

Fig. 6. Solution of EST(4).

Both Fermat and Torricelli solutions are valid only for a triangle with angles less than 120°. Otherwise, F will be the vertex of the triangle with the widest angle.

EST(4) problem can be solved as follows (Fig. 6):

- choose two points from the 4 points (the chosen points are called A and B, and the remaining points C and D);
- construct the Torricelli substituting point and Torricelli arc of A, B: S = S(A,B) and $\alpha = \alpha$(A,B);
- construct the Fermat point of C, D and S: F_1 = F(C,D,S);
- if α cuts the line segment SF_1 at F_2, then we have the EST solution consisting of AF_2, BF_2, F_1F_2, CF_1 and DF_1;
- otherwise, return to the beginning of the procedure, and choose another pair A, B. It has been proven that the solution will be found for at least one A, B pair.

The method proposed for EST(4) can be generalized for EST(n) with $n > 4$ as follows:

- choose two points from the n points (we call the chosen points A and B);
- construct the Torricelli substituting point and Torricelli arc of A, B: S = S(A,B) and $\alpha = \alpha$(A,B);
- find the solution of the EST($n-1$) problem for the set of points consisting of S and the remaining points. Let F_1 be the first point, through which S is connected to the remaining points of the EST($n-1$) solution.
- if α cuts the line segment SF_1 at F_2, then we have the solution of the EST(n) problem consisting of AF_2, BF_2, and the element of the EST($n-1$) problem solution reduced by SF_2;
- otherwise, return to the procedure beginning, and choose another pair A, B.

The given algorithm has the complexity of $O(n!)$. It has application in practice for small value of n. For large value of n, it is impossible to use this algorithm to find the EST in a sensible time. However, an essential property of the algorithm will be reused in proposed algorithm, in order to construct the set of candidate points. Thus, a candidate point should be either: the Torricelli substituting point of a pair of points, each of which is a terminal point or another candidate point, or the Fermat point of a triple of points, each of which is a terminal point or another candidate point. Another proven fact is that the number of Steiner points cannot excess $n-2$ [9].
Therefore, the set of candidate points can be generated as shown in Fig. 7.
Up to now, we have been finding the candidate points for EST. The FNO, however, has a larger set of candidate points, due to the impact of g function, described by means of a collection of g-polygons. Each g-polygon has an assigned cost value. Providing the g-polygons causes

```
procedure GenerateESTCandidates(in
  terminals; out candidates)
begin
  candidates = terminals
  n = Cardinality(terminals)
  for i=1 to n-2 do
  begin
    candidates += {set of points, each of
      which is a Torricelli substituting
      point of a pair of points in
      candidates}
    candidates += {set of points, each of
      which is a Fermat point of a triple
      of all triple of points in
      candidates}
  end
  candidates -= terminals
```

Fig. 7. Method of generating set of candidate points.

the straight line between two points to not always be the least cost line.

Let us consider a situation in Fig. 8, in which the plane is divided into two half-planes with different costs (m_1 and m_2). The goal is to find the lowest-cost line that connects two points A and B belonging to the two half-planes.

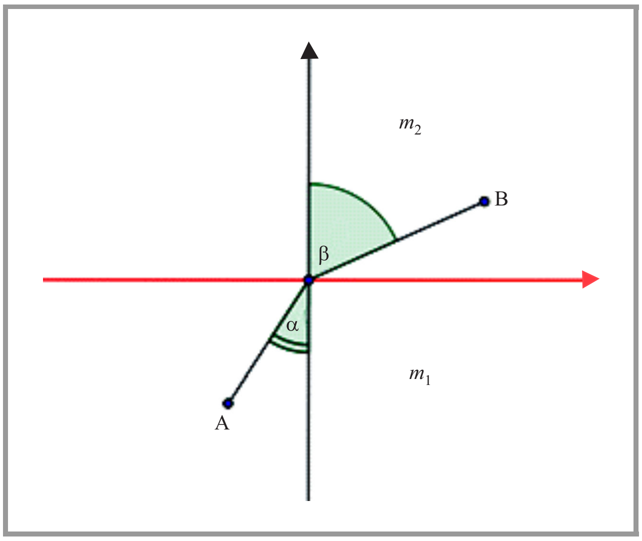

Fig. 8. Fermat Principle.

According to Fermat Principle, the line with lowest cost is the two-segment polyline A-O-B, where O belongs to the interface between the two half-planes and fulfills following condition:

$$\frac{\sin\alpha}{\sin\beta} = \frac{m_1}{m_2}. \tag{3}$$

The Fermat Principle has wide applications in physics; among others, it explains the phenomenon of light refraction. The point O is called the refraction point of A, B through the interface line.

In FNO, the Fermat Principle allows for finding a new subset of candidate points, each of which is the refraction point of a pair of other candidate points through an edge of a g-polygon. Taking refraction points into consideration, the candidate points of FNO can be generated by means of procedure shown in Fig. 9.

```
procedure GenerateFNOCandidates(in sources,
  destinations, g-polygons; out candidates)
begin
  candidates = sources + destinations +
    {set of vertices of the g-polygons}
  n = Cardinality(sources + destinations)
  for i=1 to n-2 do
  begin
    candidates += {set of points, each of
      which is the Torricelli substituting
      point of a pair of points in
      candidates}
    candidates += {set of points, each of
      which is the Fermat point of a triple
      of all triple of points in
      candidates}
    candidates += {set of points, each of
      which is the refraction point of a
      pair of points in candidates through
      an edge of a g-polygon}
  end
  candidates -= sources + destinations;
end
```

Fig. 9. Generating procedure of NFO candidate points.

The cardinality of the set of the candidate points grows according to $O\big((n+m)^{3n}\big)$, where n is the total number of source and destination points, whilst m is the total number of the g-polygons vertices.

4.2. Transformation of FNO into ILP

In FNO problem, let K denote the set of candidate point generated for S, D and the given g-polygons. Let V denote the sum of S, D, T_1 and K.

For each pair of $u, v \in V$ a binary variable $x_{uv} \in \{0,1\}$ is defined. In addition the cost of the line segment u-v is calculated by means of the given cost density function g:

$$c_{uv} = \int_{r=0}^{1} g\big((1-r)u + rv\big)\mathrm{d}r. \tag{4}$$

Because the exact algorithm is not the main goal of this work, the author leaves the capacity constraints to future work.

The FNO problem can be transferred into ILP format, denoted by ILP_FNO(V, S, D, c), as follows:

Minimize

$$\sum_{u,v \in V} c_{uv} x_{uv}$$

Subject to:

$$\sum_{u \in M, v \in V \setminus M} x_{uv} \geq 1$$

for each set $M \subset V$, $M \cap S \neq \emptyset$ and $V \setminus M \cap D \neq \emptyset$.

ILP_FNO can be resolved by means of an integer linear programming package. Furthermore, ILP_FNO has a very similar form to the ILP of the TST problem [14]. In particular, the ILP_FNO into the ILP form of TST is derived by creating an artificial "super source node" connected to all

nodes of S by artificial links with zero cost. Hence, the algorithms proposed for TST to find the exact or approximate result of ILP_FNO can be reused.
For a FNO instance presented in Fig. 10 (Example 1), with the only g-polygon is the brown triangle with assigned cost = 2, the exact algorithm provides the optimal result presented by the bold polylines: AS_2, BS_3, CS_1, S_1S_2, S_1S_3, with the total cost = 10.59. If the g-polygon was not considered in the algorithm, the total cost would be 10.96 (3.5% more expensive).

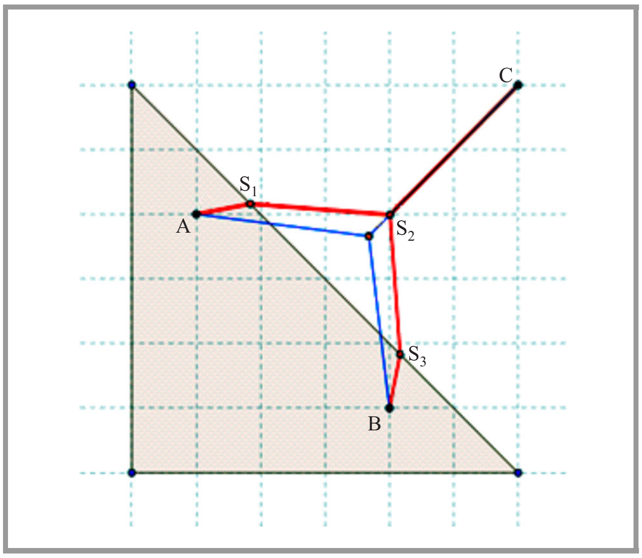

Fig. 10. FNO instance resolved by exact algorithm.

Since the FNO problem is NP-hard, the exact algorithm allows to find exact solutions only for small instances of FNO. The algorithm has been developed and conducted performance tests, which have shown that for typical configurations of the computer systems used by FTTH network planners, it is possible to apply the exact algorithm for a network with less 10 users. This limitation is unacceptable in practice, since a typical FTTH network is to serve several hundred users.

5. Approximation Algorithm for FNO

In the previous section, the mathematical properties of the EST have been applied in order to find the candidate points for optimization. This approach allows for finding the exact optimal solution, but due to the enormous cardinality of the candidate points set, it is impossible to use it for a larger instance of FNO in a sensible time. However, parts of the work on the exact algorithm can be reapplied in construction the approximation algorithm.

5.1. Approximation Algorithm Strategy

In the approximation algorithm, instead of starting already with the full set of candidate points, as the exact algorithm, the author starts the algorithm with a basic candi-

```
procedure CalculateCostVector
(in candidates; out cost_vector)
begin
  foreach v1,v2 in candidates do
  begin
    cost_vector[v1,v2] = the cost of
      the g-polygons containing v1 and v2.
  end
end

procedure CalculateILP_FNO(in candidates,
  sources, destinations, cost_vector; out
  steiner_points, steiner_lines)
begin
  Resolve the ILP by means of CPLEX, MS
  Solver, or Steiner algorithms
end

procedure FNO_Main(in PEC, LVR; in sources,
  destinations, existingTransits,
  g-polygons; out steiner_points,
  steiner_lines)
begin
  candidates = sources + destinations +
    existingTransits + vertices(g-polygons)
  for i=1 to PEC do
  begin
    foreach a, b in candidates do
    begin
      if a, b belongs to different
        g-polypons then
      begin
        Construct the cut points of line
          segment (a,b) with the
          g-polygons. Call them x1, x2, ..
          xn
        candidates += {x1, x2, .. xn}
      end
    end

    for j=1 to LVR do
    begin
      foreach a, b, c in candidates do
      begin
        if a, b, c belong to the same
          g-polygon then
        begin
          Construct the Fermat point for
            the triple a, b, c. Call it f
          candidates += {f}
        end
      end
    end

    cost_vector =
      CalculateCostVector(candidates)
    CalculateILP_FNO(candidates, sources,
      destinations, cost_vector,
      steiner_points, steiner_lines)
    foreach s in steiner_points do
    begin
      if s belongs to the boundary of a
        g-polygon then
      begin
        Find the nearest (clockwise and
          anti-clockwise) candidate
          points to s in the g-polygon
          boundary: v1 and v2

        Establish the midpoints of line
          segments: v1-s and
          v2-s and call them x1 and x2
        candidates += {x1,x2}
      end
    end
  end
end
```

Fig. 11. Algorithm details.

date point set consisting only of given points: the sources, destinations, existing transit points and the vertices of the g-polygons. After resolving the ILP_FNO for those points, the result will be improved in iterations. In each iteration, new points are added to the candidate point set. The number of the iterations is preset by means of the algorithm configuration parameters.
The main configuration parameters of the algorithm are:

- **Loose Vertices Resolution (LVR)** – the number of iterations, in each of which we find the Fermat point for each triple of candidate points lying on the same g-polygon. This way, we can locally improve the solution quality;

- **Polygon Edge Cut (PEC)** – the number of iterations, in each of which we find the quasi-refraction points and add them to the set of candidate points.

For the ILP_FNO implementation, beside using the commercial optimization packages, the author also created his own algorithm based on the known algorithms for the TST) [14]–[16].
The details of the algorithm are presented in Fig. 11.
The complexity of the approximation algorithm is $O(n^4) \cdot PEC \cdot LVR + S(n^{4 \cdot PEC}) \cdot PEC$, where n is the cardinality of the starting candidate point set (consist of S, D and the g-polygon vertices), and $S(k)$ is the complexity of the algorithm that resolve the TST problem for a k-element graph.

6. FNO Algorithm Evaluation

6.1. Comparison with Exact Algorithm

In order to compare the results of the approximation and exact algorithms, the FNO instance in Example 1, which has been resolved by the exact algorithm (Fig. 8) is considered. The approximation algorithm operates as follows:

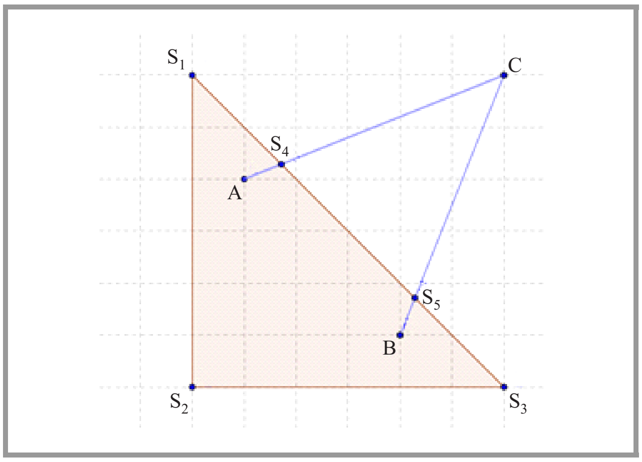

Fig. 12. Approximate algorithm for Example 1 (step 2).

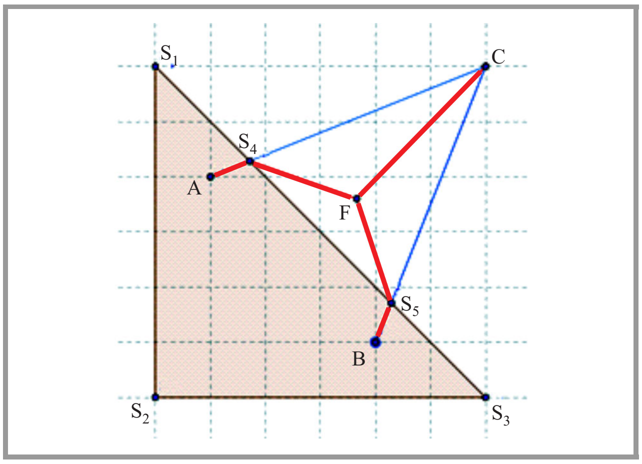

Fig. 13. Approximate algorithm for Example 1 (step 4).

1. It starts with the basic candidate points A, B, C (the given points), S_1, S_2 and S_3 (the vertices of the g-polygon).

2. Next, we add to the candidate point set the cut points of AC and AB with the boundary of the g-polygon: S_4 and S_5 (Fig. 12).

3. The Fermat point of all point triples that belong to the same g-polygon is constructed. In this example, the most interesting is the Fermat point of (C, S_4 and S_5).

4. The ILT_FNO for (A, B, C, F, $S_1, \ldots, S_5$) is resolved. The result is the red (bold) polyline presented in Fig. 13.

5. The midpoints of the line segments of the candidate points belonging to the g-polygon edges (S_1S_4, S_4S_5, etc.) is constructed. The constructed midpoints are treated as the quasi-refraction points, and added to the candidate point set.

6. The PEC is decreased by one, and repeat step 2 until PEC is zero.

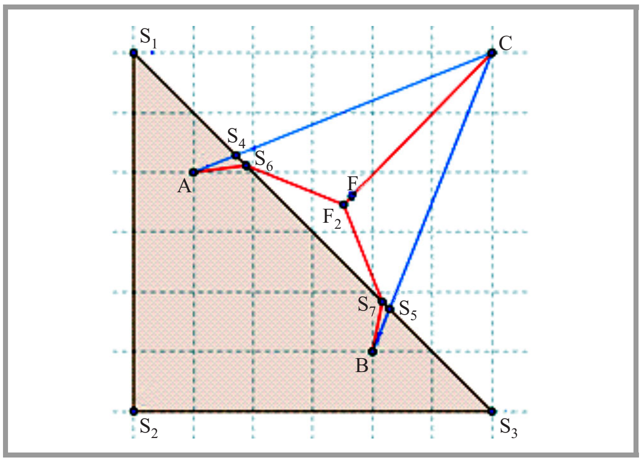

Fig. 14. Approximate algorithm for Example 1 (final result).

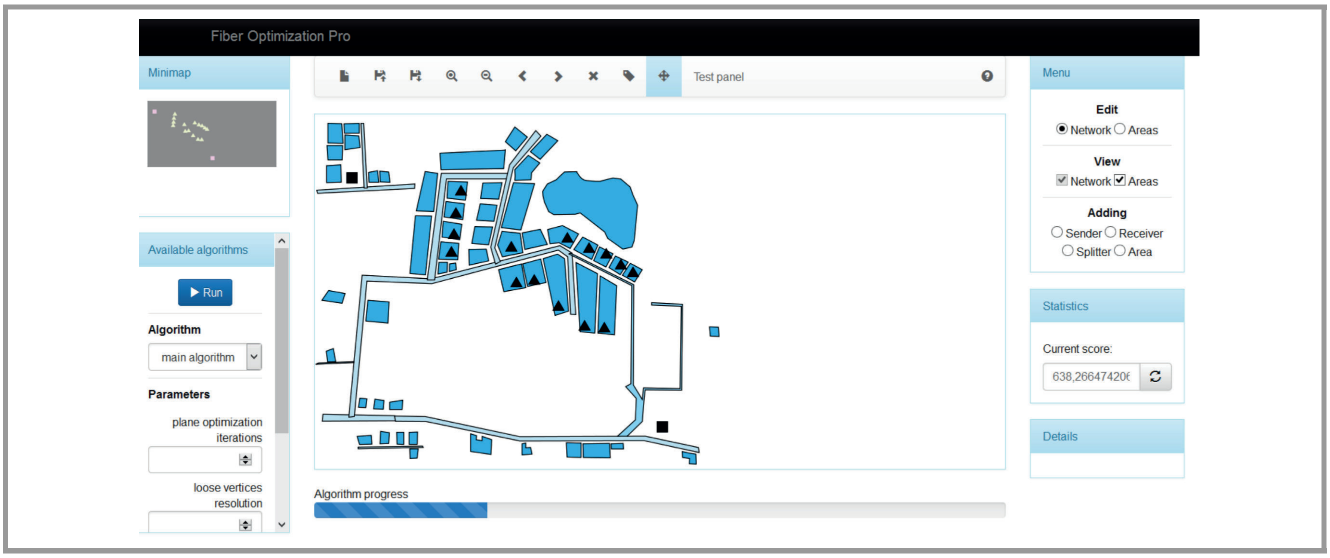

Fig. 15. FNO testing Web application – main page.

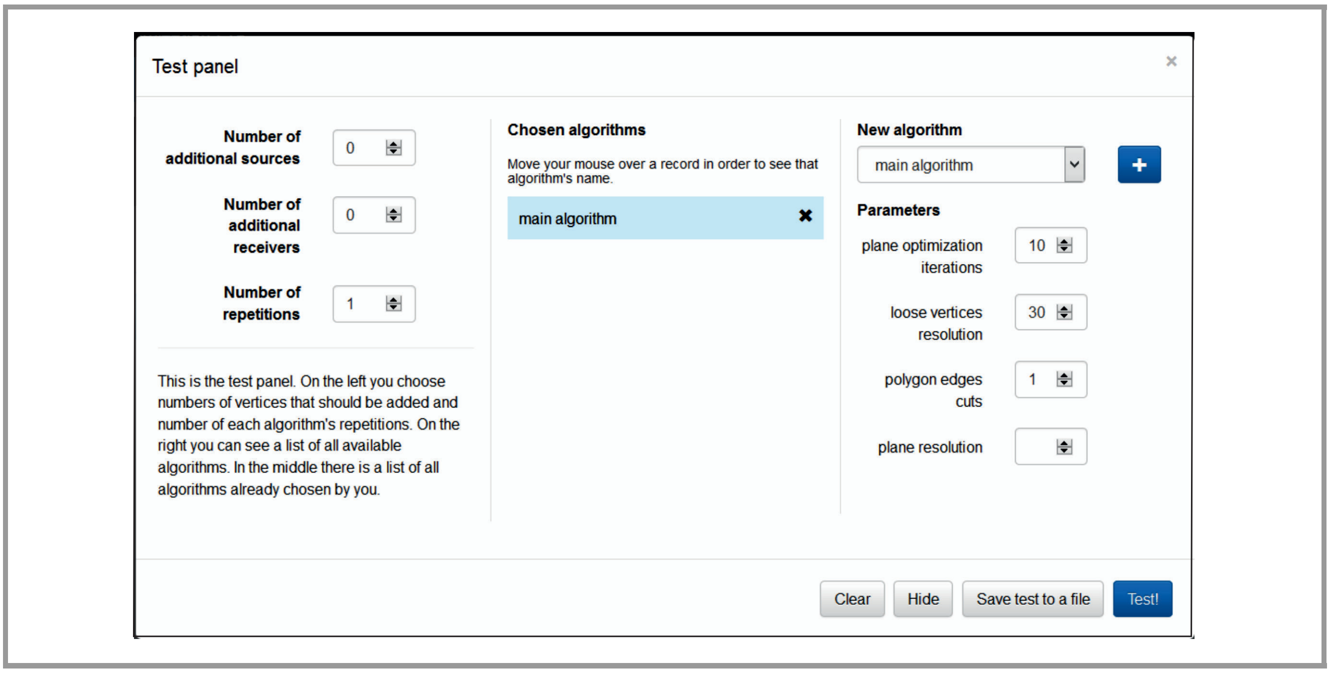

Fig. 16. FNO testing Web application – mass test panel.

The final result for PEC = 20 is presented in Fig. 14. It is very close to the exact algorithm result (0,001% cost difference).

6.2. Analysis of Algorithm Performance in Practical Example

In order to evaluate the effectiveness of the algorithm, the author has developed it in a .net C# program (Fig. 15). The program has been run in a PC based on Intel Core i5 2.66 GHz and 8 GB RAM.

The Web application includes a test panel that allows for mass testing of the algorithm problem for different values of configuration parameters PEC and LVR (Fig. 16).

In order to explain how the configuration parameters impact the result quality and the time consumption, let us consider an FNO instance (Example 2), in which we have 1 source (the square symbol), 3 destinations (triangles) to be connected through 3 high-cost polygons.

For PEC = 1 and LVR = 1 (Fig. 17), the cost is quite high (644.9) but the algorithm runs exceptionally fast (0.7 s duration).

When LVR is increased to 10 (Fig. 18), the cost is improved significantly (638.3). However, the algorithm runs slower (6 s), though the duration is still acceptable (near real-time).

Additional increasing LVR up to 20 (Fig. 19), causes minimal cost improvement (638.2). However, the algorithm

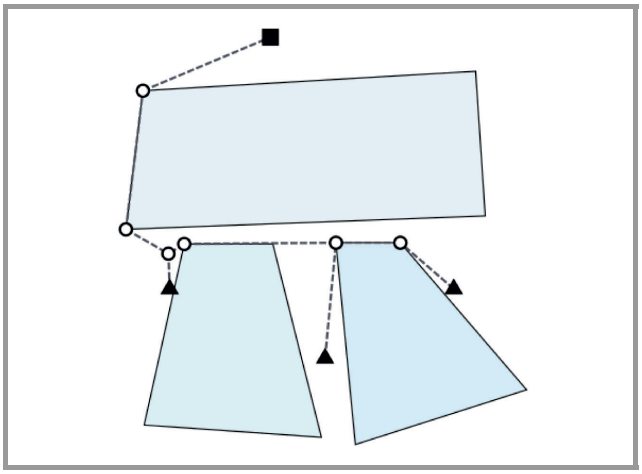

Fig. 17. Algorithm result for Example 2, PEC = 1, LVR = 1, cost = 644.9.

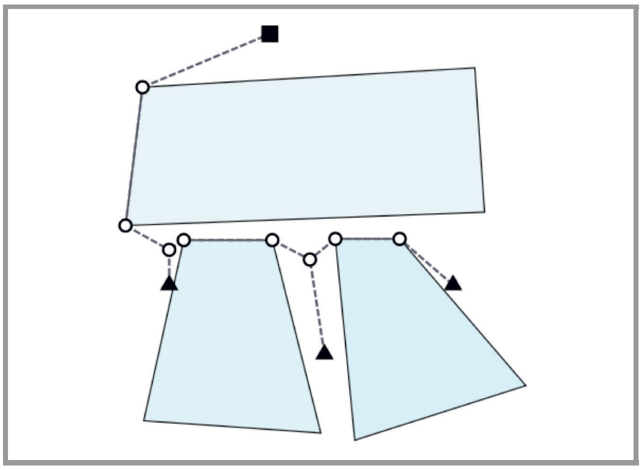

Fig. 18. Algorithm result for Example 2, PEC = 1, LVR = 10, cost = 638.3.

runs much slower (374 s), and the duration is unacceptable for a network design tool.

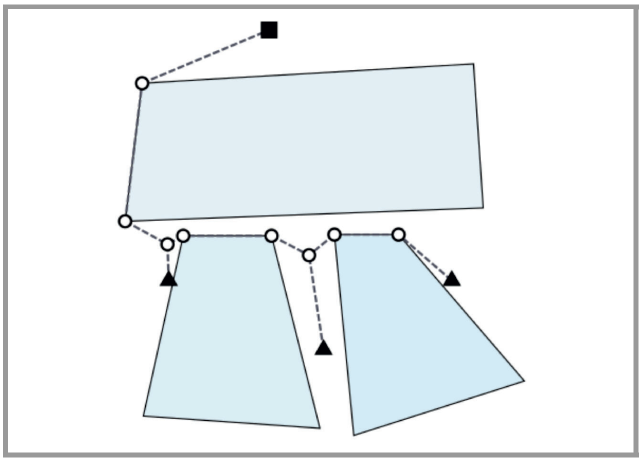

Fig. 19. Algorithm result for Example 2, PEC = 1, LVR = 20, cost = 638.2.

More detailed impact analysis of the configuration parameters PEC and LVR on the algorithm operation has been conducted by means of the mass testing on a practical example of FTTH network optimization for a housing estate in Warsaw (Example 3, Fig. 20). The mass testing relies on running the algorithm multiple times, changing the configuration parameters for each iteration.

In Fig. 21, the tendency regarding the result quality when PEC and LVR increase is presented. It can be observed that increasing LVR from 0 to 1 causes significant improvement of the result quality. Increasing LVR from 1 to 2 gives only modest improvement, whilst further increment of this parameter does not causes any noticeable effect. This tendency is logical, because parameter LVR only decides on local improvement of the algorithm result relying on finding the Fermat points.

The PEC impact on the algorithm is different. The author has observed that, a number of iteration is needed (PEC = 20 for LVR = 0 and PEC = 7 for LVR = 1), in order to gain a significant improvement of the result quality. This tendency is logical, because parameter PEC decides on the cardinality of the set of candidates points, whilst the algorithm needs the set large enough in order to return the converging the optimum.

The performance of the algorithm is presented in Fig. 22. It can be observed that each increment of LVR causes a significant growth of operation time, whilst the LVR rise causes rather modest increment of operation time.

The conclusion of mass testing analysis is that in practical situation, it is recommended to run the algorithm with LVR to be 1, 2 or 3, whilst PEC should be set to be as large as possible. The mass testing analysis confirms that, although the algorithm does not guarantee the ideal solution, it allows for finding the solution closed to the optimum. This allows the algorithm to be applied in computer systems used by FTTH network planners.

7. Conclusion

In the paper, a new method for FTTH optimization focused on minimizing the cost of cable installation is presented. This method is a development of the previous work result on geometric optimization. The author added new aspects not considered in previous work, in particular geographical aspect and the aspect of existing network resources.

The optimization problem FTTH has been formulated as a mathematical problem called FTTH Network Optimization (FNO). FNO problem has been transformed into an Integer Linear Programming. Since FNO problem is NP-hard, its exact algorithm allows to find optimal solutions only for small FNO instances (for less than ten end-users). The algorithm has been developed and performance tests were conducted, which have shown that for typical configurations of the computer systems used by FTTH network planners, it is possible to apply the exact algorithm for a network with less than 10 users. This limitation is un-

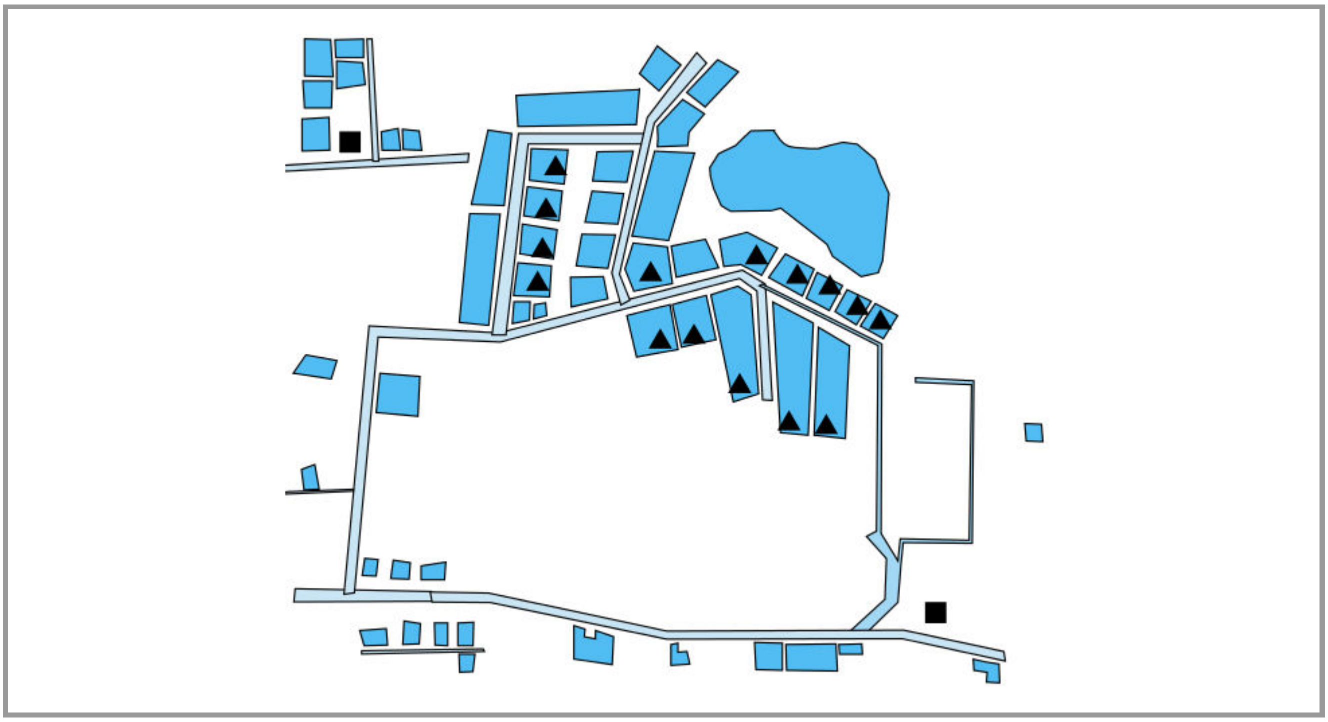

Fig. 20. Example 3, testing for housing estate in Warsaw.

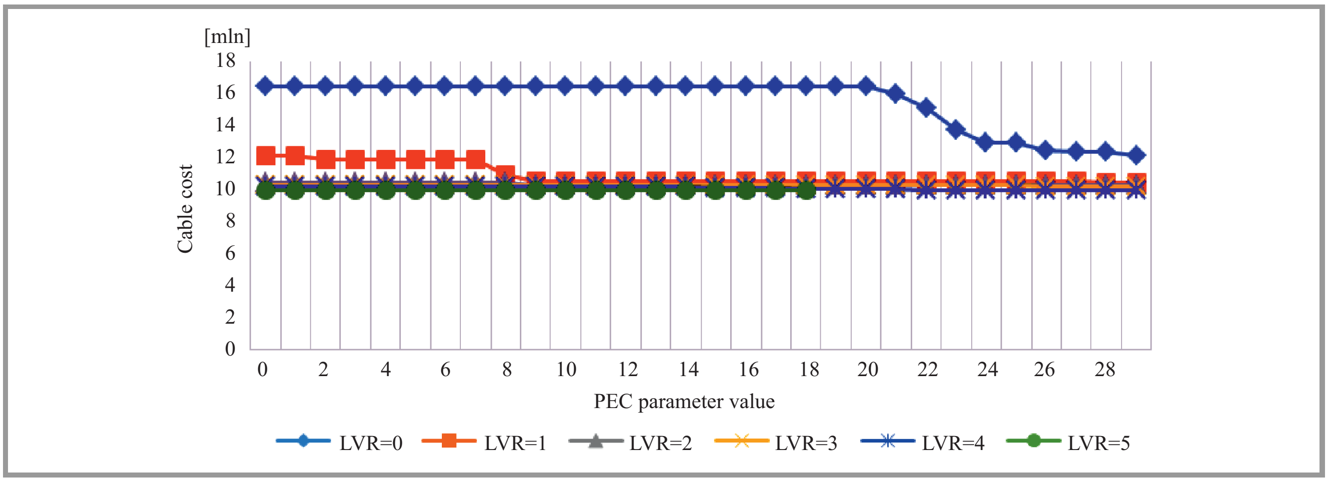

Fig. 21. Result quality dependence on parameters PEC and LVR for Example 3.

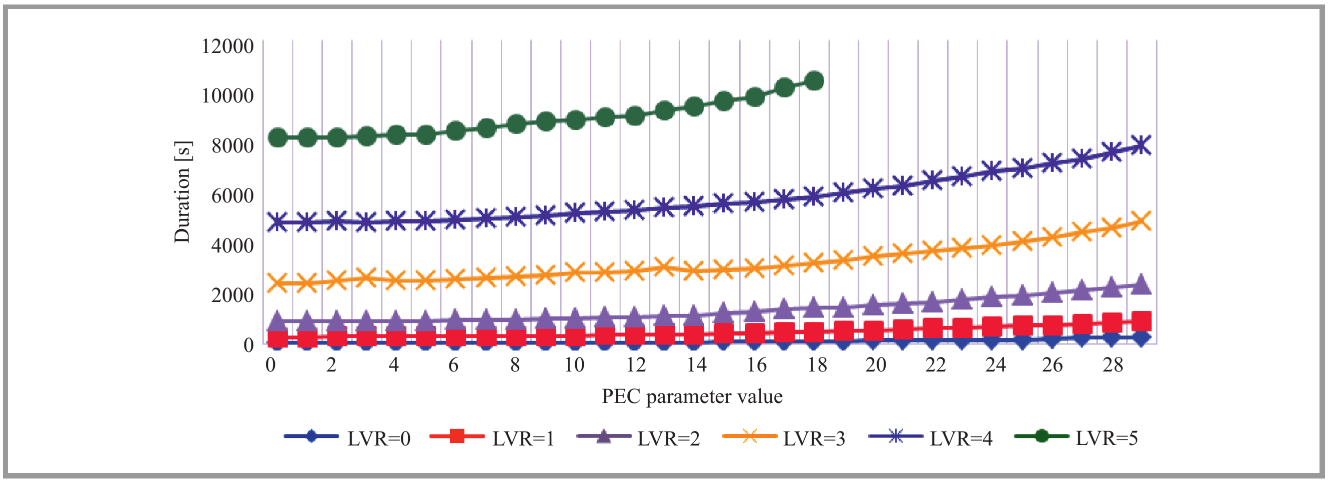

Fig. 22. Performance dependence on parameters PEC and LVR for Example 3.

acceptable in practice, since a typical FTTH network is to serve several hundred users.

Therefore, for larger FNO instances, an approximation algorithm have been proposed. Although the algorithm does not guarantee the ideal solution, it allows for finding the solution closed to optimum. The algorithm computation complexity is polynomial. It can be applied in typical computer systems used by FTTH network planners.

In the paper, the approximation algorithm has been described. In order to confirm its performance in practice, the author has developed the algorithm in a .net C# program. In order to verify the result quality and evaluate the impact of the algorithm configuration parameters, the mass testing on a practical example was conducted. As a conclusion after analyzing the test results, the recommendation for algorithm usage in practice has been presented.

References

[1] B. Williams, "FTTH council global alliance updates", in *Proc. 2013 FTTH LATAM Ann. Conf.*, Sao Paulo, Brazil, 2013.

[2] B. Deutsch, R. Whitman, and C. Mazzali, "Optimization of FTTH passive optical networks continues", *Lightwave Mag.*, Jan. 2005 [Online]. Available: http://www.lightwaveonline.com/articles/2005/01/optimization-of-ftth-passive-optical-networks-continues-53911092.html

[3] M. Chardy and C. Hervet, "FTTH network design under OA&M constraints", in *Proc. 5th Int. Conf. on Netw. Optimiz. INOC'11*, Hamburg, Germany, 2011, pp. 98–104.

[4] S. van Loggerenberg, L. Grobler, and F. Terblanche, "Optimization of PON planning for FTTH deployment based on coverage", in *Proc. Southern African Telecommun. and Netw. Access Conf. SATNAC 2012*, Fancourt in George, Sth. Africa, 2012.

[5] M. Grötschel, C. Raack, and A. Werner, "Towards optimizing the deployment of optical access networks", *EURO J. Comput. Optimiz.*, vol. 2, no. 1–2, 2013.

[6] M. Chardy, M.-C. Costa, A. Faye, and M. Trampont, "Optimizing splitter and fiber location in a multilevel optical FTTH network", *Eur. J. Operat. Res.*, vol. 222, no. 3, 2012.

[7] C. Hervet, A. Faye, M. Costa, M. Chardy, and S. Francfort, "Solving the two-stage robust FTTH network design problem under demand uncertainty", *Elec. Notes in Discrete Mathem.*, vol. 41, 2013.

[8] D. Kadhim and N. Hussain, "Link and cost optimization of FTTH network implementation through GPON technology", *Commun. and Netw.*, vol. 5, 2013.

[9] M. Brazil, "On the history of the Euclidean Steiner tree problem", *Archive for History of Exact Sciences*, vol. 68, no. 3, 2014.

[10] S. Arora, "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems", *J. of the ACM*, vol. 45, no. 5, 1998.

[11] L. Chung-Wei *et al.*, "Efficient obstacle-avoiding rectilinear Steiner tree construction", in *Proc. Int. Symp. on Physical Design ISPD'07*, Austin, TX, USA, 2007.

[12] M. Zachariasen and P. Winter, "Obstacle-avoiding Euclidean Steiner trees in the plane: an exact algorithm", in *Algorithm Engineering and Experimentation*, M. T. Goodrich, C. C. McGeoch, Eds., *LNCS*, vol. 1619, pp. 286–299. Springer, 1999.

[13] M. Garey, "The complexity of computing Steiner minimal trees", *SIAM J. of Appl. Mathem.*, vol. 32, no. 4, 1977.

[14] J. W. Van Laarhoven. "Exact and heuristic algorithms for the Euclidean Steiner tree problem", PhD thesis, University of Iowa, 2010.

[15] K. Mehlhorn. "A faster approximation algorithm for the Steiner problem in graphs", *Inform. Process. Lett.*, vol. 27, no. 3, 1988.

[16] V. V. Vazirani, *Approximation Algorithms*. Berlin: Springer, 2001.

Hoang Nghia Le received his M.Sc. degree from Warsaw University of Technology, Faculty of Electronics and Information Technology, Institute of Telecommunications in 1998. Currently he is a product manager at Suntech S.A., a software company specialized in producing OSS (Operations Support Systems) for telecommunication networks. His scientific interests cover optimization issues in optical and wireless networks.

E-mail: NghiaLe.Hoang@suntech.pl
Suntech S.A.
Puławska st 107
02-595 Warsaw, Poland