

Profitable areas in large-scale FTTH network optimization

Mateusz Żotkiewicz¹ · Mariusz Mycek¹ · Artur Tomaszewski¹

Published online: 27 March 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We present an optimization platform for Fiber-to-the-Home network design. The platform is capable of minimizing the capital expenditures (CAPEX) of network deployment by optimizing locations of optical equipment, signal splitters and cable cabinets, optimizing cable routes and types of cables as well as the number and types of optical cards and splitters. We present the architecture of the platform, the design process it implements, and the algorithms it deploys. The platform is used to indicate the parts of the design process that require complex optimization with dedicated algorithms and those that can be left to appropriately crafted engineering rules. We indicate that while keeping the computation time acceptable, much of the CAPEX savings can be obtained when locations of optical equipment are thoroughly optimized, cable routes are determined with plain engineering rules, and finally, signal splitting patterns are optimized carefully to lower the fiber count and thus the cost of cables.

Keywords FTTH · Access networks · Network design · Simulated annealing

1 Introduction

The ever increasing need for broadband Internet access leads fixed network providers to consider replacing their legacy

copper-based access infrastructure with a new one that would employ a fiber-based technology. The process seems to be inevitable due to the increasing pressure from mobile network providers and their Long-Term Evolution (LTE) networks—to cope with that threat, fixed network providers have to substantially increase the quality of offered services, and this can be achieved only by replacing the copper-based technology and bringing the fiber as close to the customer as possible. From this perspective, the Passive Optical Network (PON) technology seems to be the most suitable choice, especially in the long run: As fiber-based technologies are capable of providing much greater bandwidth than their copper-based alternatives, they make it easier to cope with a potential increase in demand volumes in the future.

The terminology used in this research is as follows. A considered network is called Optical Access Network (OAN), and it consists of a number of Optical Line Terminals (OLTs) feeding with optical signals a set of Optical Network Units (ONUs). An OAN can be implemented in either point-to-point (P2P) or point-to-multipoint (P2M) architectures. The P2P architecture is suitable for ONUs serving a large number of non-optical customers or serving a single customer that requires a full-fiber (broadband or a dark-fiber) access. On the other hand, the P2M architecture is used for ONUs with just a single optical customer—such ONUs are called Optical Network Terminals (ONTs).

The FTTH standardization is currently a predominant solution for deployment of P2M OANs. It exploits the PON technique with up to 128 of ONTs sharing a bidirectional optical signal to an OLT using an enhanced time division multiplexing paradigm. The sharing is realized using optical splitters and is scheduled by the OLT, which delimits time slots in the optical signal and assigns these time slots to particular ONTs. Generally, PON OANs exploit different wavelengths for uplink and downlink traffic and can achieve

✉ Mateusz Żotkiewicz
mzotkiew@tele.pw.edu.pl

Mariusz Mycek
mariusz@tele.pw.edu.pl

Artur Tomaszewski
artur@tele.pw.edu.pl

¹ Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

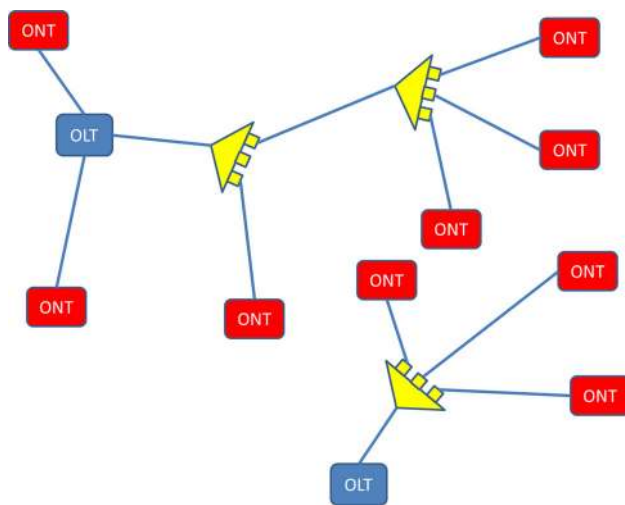


Fig. 1 P2M OAN

capacities of up to 10 Gbit/s in the forward channels and up to 2.5 Gbit/s in the backward channels. An exemplary P2M OAN is shown in Fig. 1.

In our research, we address a problem of optimizing P2M OAN networks in the FTTH standard. We have selected the P2M architecture, as the solutions based on it can be up to 50% cheaper than solutions based on the P2P architecture [10]. When existing infrastructure is not sufficient, i.e., all trenches have to be dug, it is indicated in [24] that selecting P2P architecture does not significantly impact the total cost of deployment. However, as stated in [6], in the majority of urban areas, underground infrastructure is rather abundant and allows to install the FTTH network without digging new trenches, which justifies using the P2M architecture.

We start by optimizing OLT locations and end up equipping access points that group a number of individual ONTs. We use complex industry-acceptable network models that encompass, in particular, the following detailed elements: attenuation of cables, splitters, and optical plugs; power budget of the demand; output power of the PON card; available telecommunication infrastructure; costs of trenching; costs of cable rollout; and costs of splicing fibers. As our research aims at facilitating large deployment projects with up to one million ONTs grouped into up to 100k access points, we must make use of both hard optimization methods and non-exact approaches.

This paper is organized as follows. In Sect. 2, the available literature in the FTTH network optimization domain is summarized. In Sect. 3, the problem we are facing is described in detail, thus allowing the reader to understand the specific features of our research. Our optimization method is described in Sect. 4. In Sect. 5, the numerical results are presented, which give insight into the importance of different parts of

the optimization process for the obtained CAPEX savings. The paper is concluded in Sect. 6.

2 Related work

The FTTH network design problem has been intensively studied, and it is well covered in the literature. However, many of the proposed models lack some important features that make them impractical from the point of view of the industry. Approximately 10 years ago, Khan proposed a simple 2-factor approximation to a PON design problem [19]. His model, for instance, does not take an intermediate (between OLT and ONT) split into account, which limits the flexibility of obtained solutions. In addition, the model cannot utilize knowledge about existing infrastructure, which, in the case of an access network and high labor costs, can easily lead to solutions of overestimated cost. However, from the industrial point of view, the biggest disadvantage of [19] lies in its objective. A 2-factor approximation for a problem in which a small percentage reduction in costs means huge savings is not sufficiently convincing.

Given the mentioned costs savings as a priority, a vast majority of the approaches presented in the literature can be divided into two main groups. In the first group, we find all the approaches that concentrate on the optimality of a returned solution. They usually model considered problems using integer programs (IPs) and use commercial solvers to handle them. The biggest disadvantage of this group of approaches is their tractability. They are efficient either for comparably small test cases, or the models they are based on are unacceptably simplified from our point of view. For instance, in [3], only one split is allowed, and its ratio is given as an input. On the other hand, in [32], trenching was considered separately for each cable; thus, two parallel connections cannot be realized in one trench. Similar simplifications were considered in [27], in which trenching was also considered separately for each connection. Moreover, only one OLT location was considered in their model. An additional example is [9], in which the split ratio was not considered. Obviously, even IP models can be very detailed. Consider a model presented in [21], in which even cost and attenuation of splices are considered. Unfortunately, such models can be utilized for only comparatively small use cases, such as a 28-node network considered in [21]. A very good summary of the Integer Programming approach to the FTTH network design can be found in [13].

In the second group, which is larger in size and closer to our research, we find all heuristic approaches that concentrate on returning solutions to be as reasonable as possible but without any warranty on their optimality. Approaches from this group are more tractable; thus, they can be used to solve bigger use cases. However, we also see a number of simplifications that we find unacceptable in our research.

Consider a very good model presented in [25]. It seems to be complete. However, taking a closer look, we notice that it lacks, for example, the costs related to OLTs. One may claim that those costs are constant because the number of served clients is constant. However, there is a difference if the same number of clients is served by a minimal number of line cards or not. In fact, in our research, we indicate that the cost of line cards is significant, and it can substantially vary during the optimization process (allowing for the cable cost reduction). An additional optimization approach is presented in [28], in which the authors propose an algorithm that mixes a genetic approach with a spanning tree algorithm, but in their model, only one split is considered. A similar assumption is taken in [23]. In addition, in [23], the problem of parallel connections is also not adequately addressed. The closest to our research is the work published by Orange Labs, France [6, 14, 15]. However, the assumptions taken there result from the special features of the areas considered in their research (densely populated urban areas), i.e., always sufficient underground infrastructure or neglected effective reach of FTTH technologies (which does not exceed 10 km in practice for 1:64 split [31]) simplify their model to an extent that we cannot accept.

As indicated above, the nearly infinite variety of different features that can be taken into account while modeling an FTTH network design problem makes virtually any model extendable. Obviously, the model presented in our research does not encompass all possible features that can be considered while designing an FTTH network. For instance, in our research, uncertainty in demand volumes is considered in the pre-optimization phase, and thus, it is not taken into account in the optimization model itself as in [14]. Second, the operations & management (O&M) rules considered in our research are not as tight as in [6]. Moreover, we do not consider reliability as in [7, 16, 17]. Still, reliability is taken into account in our research, but failures of single fibers are only considered. Finally, in our research, we directly focus on reducing capital expenditures. However, the O&M rules applied in our research are designed in such a way that operational expenditures (OPEX) are also minimized, although indirectly. Still, some researchers state that limiting the optimization to a direct CAPEX minimization (and indirect OPEX minimization) does not always give the best results. As an example, consider the issue of reliability. It is costly to provide, and the profit it generates cannot be directly measured, making it difficult to compare the gain resulting from increased reliability to the capital expenditures [7]. An additional example is maximizing gains. In [25], the authors do not have to connect all the clients to an access network. On the contrary, they want to connect only those clients that will generate a solid profit in the future. An additional interesting field of research is presented in [4], in which a game theoretical approach is employed to address an issue of competition

between operators—an interesting but very rarely studied problem faced by the industry. Although our model does not encompass all features considered in the literature, we still believe it is practical and according to our experience, it can be successfully used in industrial FTTH network design.

Finally, to make our survey complete, we have to mention publications that address optimization of future passive optical access networks, which are supposed to employ wavelength-division multiplexing (WDM) technology. First steps to model and solve problems resulting from introducing WDM technology to PON can be found in [7] or [8]. In both cases, they sacrifice details of a PON model to cope with novel features that result from introducing WDM technology to a problem.

3 Problem statement

In this section, we present in detail a problem we are facing. The description is divided into two parts. First, we name all data taken into account. Second, we list requirements for a solution that indirectly defines a model of a feasible solution.

3.1 Data model

First, we present a data model used in our research. In this section, a reader can find all equipment, objects, and parameters that are taken into account in the optimization process. The data are divided into three groups. The first group is called *Equipment* and contains all information about the types of equipment considered in the research.

In the second group, called *Objects*, we gather all elements that cannot be classified as equipment or are tightly connected to a given location. An additional strict definition of these two groups is that the elements of *Equipment* can be used many times in many different locations, while the elements of *Objects* are location specific and can be used only once.

Finally, the third group is called *Parameters* and contains all values that are constant and common for the whole execution of the optimization process but can vary between consecutive executions. The values in *Parameters* are neither *Equipment* nor *Object* specific, and they represent general assumptions, e.g., attenuation caused by a fiber per kilometer or a splice cost.

In Table 1 (*Equipment*), all types of equipment considered in our research are listed. All of the equipment are characterized by their cost; thus, we can ignore the cost issue in the explanations that follow. As mentioned in the introduction, we dimension a network starting from the central offices and ending as far as the access points that group a number of subscribers. Therefore, we do not consider individual clients, which makes cost considerations for a segment connecting an access point to a client out of the scope of this research. This

Table 1 Equipment

CableType	fiberCount::= Integer capex::= Double
SplitterType	loss::= Double [dB] outputCount::= Integer capex::= Double
ClosureType	spliceCountLimit::= Integer capex::= Double
CardType	outputPower::= Double (dBm) outputCount::= Integer capex::= Double
SiteType	oltWeightLimit::= Integer cabinetWeightLimit::= Integer capex::= Double
CabinetType	weight::= Integer portCountLimit::= Integer capex::= Double
OltType	weight::= Integer cardCountLimit::= Integer capex::= Double

results in a situation in which only considered active pieces of equipment are OLTs and line cards that they accommodate.

We understand the term OLT to mean a device that is capable of holding `cardCountLimit` line cards and needs some space expressed by `weight`. We assume that operating OLTs have to be accommodated in appropriately prepared sites.

We understand the term site to mean all equipment and arrangements needed to accommodate OLTs of total weight not exceeding `oltWeightLimit` and cabinets (to be described later) of total weight not exceeding `cabinetWeightLimit`. In practice, only sites allowing for OLT accommodation will have a positive cost, as in our research solely OLTs are active equipment needing additional care to work (power, air-conditioning, etc.). Still, we will usually have to pay for locating all sites, even those of zero cost, in particular nodes. Notice that sites are not location specific—all location specific objects are placed in Table 2.

As written earlier, each OLT can accommodate a number of line cards. Each line card is characterized by the number of outputs it can support (`outputCount`) and the output power it provides at each port (`outputPower`). The former parameter indicates how many fibers can be connected to the line card, while the latter expresses the power of a transmitted signal impacting in this way a range of a line card.

Fibers are grouped in cables. Each cable is characterized by its fiber capacity represented by `fiberCount`.

If two consecutive cables are to be connected or one cable is to be split into two independent cables, the appropriate fibers of the considered cables have to be spliced to ensure

Table 2 Objects

Node	x::= Double y::= Double
AdmissibleSite	node::= Ref(Node) siteType::= Ref(SiteType) capex::= Double
Edge	length::= Double nodeA::= Ref(Node) nodeB::= Ref(Node)
Demand	node::= Ref(Node) signalCount::= Integer requiredPower::= Double [dBm]
EdgePreparation	edge::= Ref(Edge) capex::= Double
CableRollout	edgePreparation::= Ref(EdgePreparation) cableType::= Ref(CableType) capex::= Double

continuity of fiber paths. Such a set of splices has to then be secured using a closure. In our model, we distinguish different closures depending on the number of splices they can cover (`spliceCountLimit`).

Finally, in passive FTTH networks, splitters are used. In our model, splitters are characterized by the loss they inflict on a signal (`loss`) and the number of outputs they feed (`outputCount`).

We consider a splitter a point of elasticity of a network; thus, we need to provide easy access to its outputs and the possibility to replace and rearrange fibers that are connected to it. This can be done by locating splitters in appropriately designed cabinets equipped with slots for splitters and optical distribution frames (ODFs) to facilitate management of fibers. We characterize cabinets by their `weight` and the total number of ports of splitters they can support (`portCountLimit`). Notice that to locate a cabinet, as in the case of OLTs, we need to provide an appropriate site that can accommodate it.

In Table 2 (Objects), all location specific elements are stored. Obvious elements of this table are nodes and edges.

By the term node, we understand a point defined by its coordinates (x,y) . From a formal viewpoint, a node can represent any place, but in practice, we use the nodes to represent: potential locations for sites accommodating OLTs or cabinets, endings of available or potential ducts, or locations of access points.

Each entry of the admissible sites specifies a `node` and a `siteType` that can be located in this particular node together with a `capex` cost of installing the site. Notice that some nodes can allow for a different type of sites. Still,

Table 3 Parameters

<code>fiberLossPerKm::= Double (dB/km)</code>
<code>pigtailLoss::= Double (dB)</code>
<code>pigtailPortCapex::= Double</code>
<code>spliceCapex::= Double</code>
<code>maximumSplit::= Integer</code>

we assume that in a feasible solution, only one site can be located in one node.

The second obvious elements of Objects are edges. Each edge is described by a pair (`nodeA`, `nodeB`) of nodes that it connects and its `length`. Notice that the length does not have to be equal to a distance on a plane between the points represented by the connected nodes but should rather represent a length of the cable needed to connect the related nodes. Each edge represents solely a relation between a pair of nodes and alone does not allow for rolling a cable out. It has to be appropriately prepared first—admissible edge preparation objects are used to model the issue.

An admissible edge preparation allows for rolling out cables at a given edge after paying a `capex` set up price. The object represents all activities necessary to prepare an edge to accommodate a cable, such as digging a trench or removing old copper cables from an already existing duct. Obviously, for some edges that represent recently built underground infrastructure, a cost of an edge preparation can be zero. Finally, some of the admissible edge preparation objects can represent technologies that do not allow for use of all available cable types. For instance, an operator can follow a policy that does not allow for using aerial cables of sizes greater than a given threshold, or the threshold is imposed by a diameter of micro tubes used in a selected edge preparation.

The above issue is modeled by admissible cable rollouts that specify a cable of which `cableType` can be used on an edge that implements given `edgePreparation`.

By now, we have presented all the equipment and infrastructure (current or potential) that can be used to realize our network. However, we still do not know where our clients are located. This issue is modeled by demand entries that, for a given `node`, express the number of clients (`signalCount`) to be served. The last feature of a demand is `requiredPower`, which expresses the minimum power of a signal that has to reach a client to satisfy the demand.

In Table 3 (Parameters), all values that are constant and common for a single execution of the optimization process are stored. They are attenuation of a fiber (`fiberLossPerKm`) and loss inflicted by a single detachable connection (`pigtailLoss`); to provide points of elasticity, we assume one detachable connection for each input or output of a splitter or two detachable connections

to provide a point of elasticity for a fiber that is not split in a considered location; the cost of detachable connections (`pigtailPortCost`); the cost of splicing one fiber (`spliceCost`); the maximum number of clients that can be served by a single card port (`maximumSplit`).

3.2 Solution requirements

In this section, we present requirements for a solution—they define what constitutes a feasible solution. Alternatively, the requirements could be presented using an integer program. However, we believe that the complexity of the problem justifies a less formal but more readable presentation that will follow.

Let us emphasize that a formal mathematical definition of the problem would require nearly 100 different names for constants and variables. In addition, each requirement listed below would require at least one but in many cases two or more different mathematical constraints in a formal model, resulting in more than 50 inequalities. Therefore, we decided to skip a mathematical formulation in the article.

The requirements are as follows:

1. Three types of logical points are distinguished: Central Office (CO), Distribution Point (DP), and Access Point (AP).
2. OLTs can be only located in COs.
3. There is an AP in each node associated with a demand.
4. Two types of logical connections are distinguished: Trunk Connections (TCs) from COs to DPs and Distribution Connections (DCs) from DPs to APs.
5. CO is always connected to a DP using a TC, and DP is always connected to an AP using a DC; some of the TCs or DCs can be of zero length when a DP is collocated with a CO or an AP.
6. Each logical point is realized by elements of Equipment and Objects. Elements of Objects can be shared between different logical points, while elements of Equipment cannot be shared except with Site elements. Notice that TCs and DCs of zero length do not require any physical elements.
7. TCs and DCs are characterized by the number and types of signals they are supporting.
8. Each signal is characterized by an already experienced split ratio and by a power at the end of a TC/DC that supports the signal.
9. An output of a line card can be connected either to a splitter in the same CO or can create a signal for a TC originating in the CO (assume TCs are directed from COs to DPs).
10. Each output of a splitter in a CO can create a signal for a TC originating in the same CO.

11. Each signal from a TC ending at a given DP has to either be connected to a splitter in this DP or create a signal for a DC originating in this DP (assume DCs are directed from DPs to APs).
12. Each output of a splitter in a DP can create a signal for a DC originating in the same DP.
13. Each signal from a DC ending at a given AP has to be either connected to a splitter in this AP or designated to feed a demand associated with this AP.
14. Each output of a splitter in an AP can be designated to feed a demand associated with this AP.
15. Each signal and each splitter output designated to feed a demand cannot experience a greater split ratio than `maximumSplit`, and its power cannot be smaller than the power threshold required by the demand.
16. Each demand has to be satisfied.
17. In each CO, there should be sufficient OLTs to accommodate all line cards that are used in it.
18. In each logical point, there should be cabinets to support all splitters located in it.
19. In each node hosting logical points, there should be a site that can accommodate all OLTs and cabinets that are placed in those logical points.
20. TCs and DCs have to be realized on consecutive edges starting and ending at appropriate logical points.
21. At each edge there should be cables of adequate capacity to accommodate signals of logical connections realized by the edge.
22. TCs cannot share cables with DCs.
23. If an edge is shared by many TCs and those TCs use different following edges in the direction of their respective DP, the same cable can be used only on one of those following edges, and on the remaining edges, the Trunk Branch (TB) should be located.
24. The above applies also to DCs and their APs. On DCs, Distribution Branches (DBs) are located.
25. Each TB (DB) requires as many splices as a sum of signals of TCs (DCs, respectively) crossing it.
26. TBs and DBs need a closure on a node in the direction of CO of adequate capacity to accommodate all required splices.
27. A closure can cover either a number of TBs or a number of DBs, but closures cannot be shared between a TB and a DB.

To let the reader better understand the requirements, they are followed by a number of examples and supporting figures. The first six points explaining the general architecture of a solution are covered by Fig. 2. In the figure, one CO, two DPs, and four APs are displayed. Notice that a CO shares a site with one of the DPs. In addition, the second DP shares a site with one of the APs. Therefore, one TC and one DC are

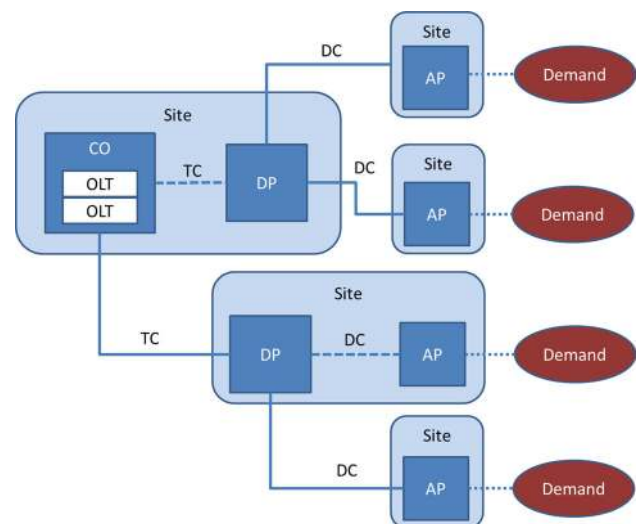


Fig. 2 General architecture of a solution

depicted by dashed lines and represent logical connections of zero length.

Let us examine what happens inside the logical points. The following 13 points of the description specify what is allowed inside logical points. An example of those rules is divided into three parts (the CO part, DP part, and AP part) and is presented in Fig. 3. Remember that the purpose of the example is to let the reader understand the model. Therefore, a solution presented in the example does not have to be optimal. In addition, the data and parameters assumed in the example do not have to be realistic—an FTTH network based solely on 1:2 and 1:4 splitters accompanied by 4-port line cards and cabinets that cannot support more than 10 outputs is hardly realistic. However, it is perfect for illustrative purposes.

The CO part of a solution is presented in Fig. 3a, in which a CO equipped with two OLTs is displayed. The OLTs host three line cards in total. Ports of two of the line cards are directly connected to a TC (notice that one port is not used at all), while two ports of the third line card are connected to 1:2 splitters, which are also connected to the same TC.

Assume that the TC of Fig. 3a ends at the DP of Fig. 3b, where the signals are split and divided into two different DCs (DC1 and DC2) feeding two different APs. The incoming signals in Fig. 3b are 4 signals with split 1:2 and signal power of 15 dBm and 3 signals with 1:1 split (no split) and signal power of 18 dBm. DC1 will be of interest in this example. It is fed by three outputs of a 1:4-splitter (notice that one output of this 1:4-splitter is not used at all), which alone is fed by a signal with split 1:2 and signal power of 15 dBm.

The example ends with an AP part displayed in Fig. 3c. The AP presented in the example is connected to the DP of Fig. 3b with DC1 accommodating three signals with 1:8 split and signal power of 7 dBm. All of them are connected

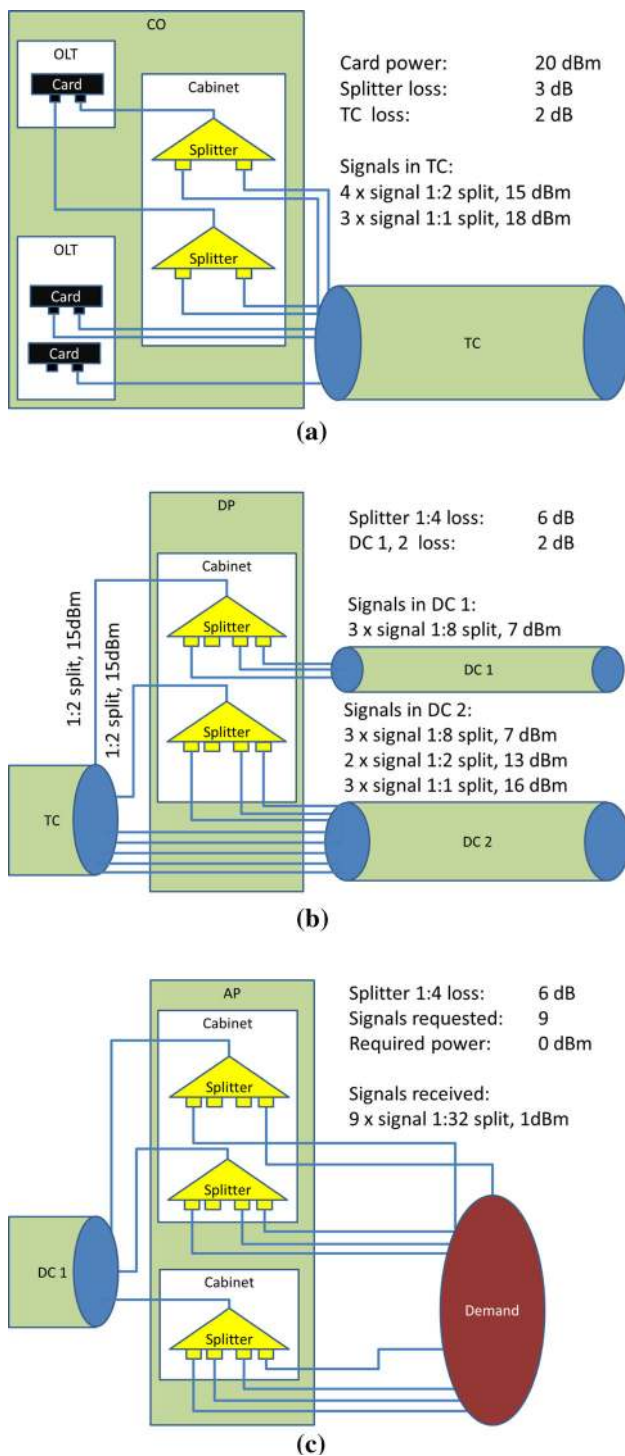


Fig. 3 Examples of logical points in a solution

to 1:4-splitters that can provide 12 signals in total. However, a demand associated with the AP requires solely 9 signals. Therefore, three (out of twelve) splitter outputs are not used. The power of the provided signals is 1 dBm, which is greater than the required threshold 0 dBm for this demand. Notice also that in the example, the splitters are

too large to be accommodated in one cabinet (assume that portCountLimit of available and installed cabinets is equal to 10, which is insufficient to accommodate three 1:4-splitters). Therefore, two cabinets have to be used. The provided signals have experienced a 1:32 total split; thus the solution is feasible only if maximumSplit is greater than or equal to 32.

Let us observe what happens inside logical connections. An example is displayed in Fig. 4, in which one CO, one DP, and two AP are depicted. In general, TCs and DCs are nearly independent. However, as shown in the figure, if an edge is shared by a TC and a DC, an edge preparation object installed on the edge is also shared by them. Assume that DC1 and DC2 serve the same signals as in the example of Fig. 3b. They share a distribution cable on the first edge leaving DP but use different edges leaving the inner node of the figure. Therefore, a distribution branch has to be used there. It is located on the edge used by DC1, as DC1 carries less signals than DC2; thus less splicing is needed. The branch is covered by the closure. Notice that the closure is associated with a node, not with an edge; thus, having a cable that is divided into tree cables in a node, we need only one closure to cover all needed fiber splices (branches). In addition, notice that the same cable is used at the whole path of DC2. Such an approach has both advantages and disadvantages. On one hand, using the same cable reduces splicing costs, but on the other hand, the capacity of the utilized cable has to be constant (and large enough); thus, the price can be greater than the price of two independent cable segments of different capacity but the same total length. Obviously, we are not constrained to use the same cable at the whole path of DC2, and a DB can also be located on the edge used only by DC2.

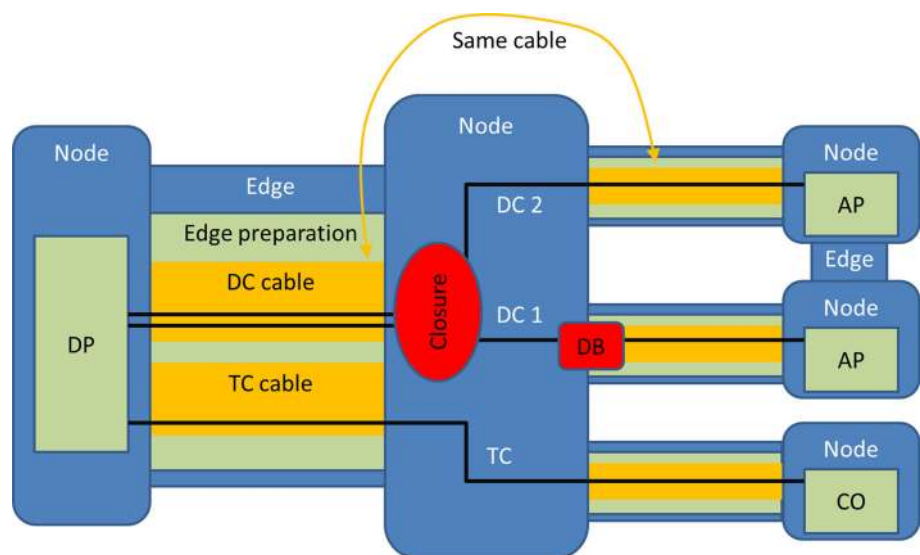
3.3 Simplifications

To make the problem more tractable, we introduce the following simplifications.

1. Edges used by all TCs have to form a forest of trees rooted at COs. In addition, edges used by all DCs have to form a forest of trees rooted at DPs.
2. Splitters in a logical point are treated as a group and do not have to be explicitly associated to particular cabinets.
3. Signals on an edge are treated as a group and do not have to be explicitly associated with particular cables. Still, trunk signals and distribution signals cannot share a cable.

Simplification 1 is crucial from the viewpoint of the network optimization theory, as it allows for employing aggregated node-link formulation [26], which in this context means that once a connection leaves its source node, i.e., AP (DP, respectively), it can be treated as any other connection

Fig. 4 Example of a cable part of a solution



heading for the same DP (CO, respectively). Although the aggregated node-link formulation alone has not been directly used in our optimization algorithms, we have taken advantage of its fundamentals while designing our approach. This approach is clearly shown in Sect. 4, in which we describe our optimization procedures in detail. By accepting the simplification, we agree that some solutions (although highly unlikely, possibly even optimal from the global point of view) are not feasible anymore.

Simplification 2 has a much smaller impact on our methods than Simplification 1. Still, we feel it is worth mentioning how it affects the plausibility of a solution, and on the other hand, how it simplifies the optimization. Its sense can be seen when more than one cabinet is needed in a logical node to accommodate all requested splitters. If the splitters are treated as a group, we cannot guarantee that there will be a valid way of dividing them between the chosen cabinets. Consider an example with one 64-output splitter accommodated in two 40-output cabinets. In such a situation, although the solution is unpractical, it is still feasible as the only constraint imposed on the cabinets, i.e., the sum of its outputs cannot be smaller than the sum of the outputs of the splitters that they accommodate, is satisfied. On the other hand, the main advantage of the simplification is a much more tractable problem. When we do not have to model relations between splitters and cabinets but only relations between a number of supported fibers and a number of cabinets, the time needed to compute a set of cabinets capable of supporting a given number of fibers can be substantially limited. The simplification reduces this problem to a knapsack problem that can be efficiently solved using dynamic programming [11].

Simplification 3 is similar to Simplification 2; however, it addresses signals and cables instead of splitters and cabinets. The sense of the simplification is only seen when more than

one cable of a single type (TC or DC) is needed to accommodate a requested number of signals. Treating all signals as a group, we lose the possibility to always keep fibers of the same AP-CO connection in the same cable, which can be seen as a threat from the maintenance point of view. However, the main drawback of the simplification lays in modeling branching and the required number of splices. When information on the number of active fibers in each cable on an edge is lost, the possibilities of modeling branching are greatly reduced.

We have listed the main simplifications, and have detailed their pros and cons. In general, all of them allow for solving the problem much faster while sacrificing either the feasibility of some solutions or the plausibility of a model. However, the analysis would not have been completed without judging how often the simplifications are employed in practice. Simplification 1 can have an important impact on a solution only if two DPs or two COs are to be located near each other, and there is a point in dividing traffic between them not with respect to distances, which are quite similar due to the proximity of the considered logical points, but concentrating more on efficient load balancing. In our research, we assume that the cabinets available in a market tend to follow the scale law, i.e., the bigger they are, the lower their single output cost is; thus, two cabinets will tend to merge if they are located next to each other, making Simplification 1 justified. Simplification 2 can impact a solution only when two or more cabinets are needed in one location. Here, the cabinet scale law is employed again; thus, two cabinets can be located in one logical point only if the amount of splitters to be supported by them is greater than the capacity of the biggest available cabinet. Although such situations may happen, they are rather rare, and in many cases, they result from inadequate data with too few considered admissible locations for cabinets. Finally, Simplification 3 impacts

a solution when more than one cable is needed on an edge to realize TCs or DCs. Although we also here assume that the equipment, i.e., cables, follow the scale law, it still can happen in practice that one cable is incapable of accommodating all the fibers on an edge, and a number of parallel cables have to be used instead. On the other hand, we were designing our method for a large-scale FTTH projects. In this case, when a total length of all rolled out cables is large, we can hope for custom cable sizes of greater fiber capacities if they are needed.

4 Optimization

This section is devoted to optimization techniques used in this research. First, we analyze the problem and justify the applied methodology. Second, we present the whole optimization process in general. As the process is decomposed into phases, in the third and following subsections, we describe each of the resulting optimization phases in detail.

4.1 Analysis of the problem

Our goal in this research is to minimize the total CAPEX of a returned FTTH network design. The problem is obviously \mathcal{NP} -hard because, for instance, it encompasses the Steiner tree problem [18]. As briefly justified in Sect. 2, the problem cannot be approached with exact methods, due to its size, and thus heuristic methods should be utilized instead. To better explain the enormous size of a formal formulation of the problem, let us present a simple example. If the problem is to be modeled using a straightforward node-link formulation, it would require two variables for each combination of APs, edges, and splitter types. Having 30k APs, 80k edges, and 7 splitter types, we end with more than 3×10^{10} variables. Our preliminary research has indicated that solving even a linear relaxation of a simplification of our problem for networks of sizes considered in this project using state-of-the-art commercial LP solvers requires tens of seconds of computations. This was the main reason behind decomposing the problem.

Decomposition can be applied in a twofold manner. Either a problem is decomposed into a number of smaller problems solved independently, and the resulting solutions of the smaller problems are then combined into a full solution of the original problem, or an optimization process is decomposed into phases. We opt for the second option for two reasons. The only possibility to decompose the problem into independent subproblems is to consider each CO independently. To do so, areas of coverage and a number of COs have to be specified at first. Unfortunately, such a division cannot be done efficiently without applying complex optimization methods, which in fact would mean decomposing the optimization process. The second reason behind the decomposition of the optimization

process is a reasonable size of a coverage area of a single CO, which is still too large to be handled without further decomposition.

4.2 General optimization process

We decided to divide the whole optimization process into a number of steps starting at the decisions of global reach, such as selecting the locations for COs, and ending at easily distributable local search decisions, such as selecting optimal splitting patterns for different demands. The optimization phases are as follows:

1. Find locations of COs and DPs (intermediate phase)
2. Find routes for TCs (intermediate phase)
3. Find routes for DCs (intermediate phase)
4. Find splitting patterns (final phase)

Obviously, this is not the only way the process can be divided. However, we believe that as a result of dividing the process in this way, some interesting conclusions can be drawn from this research. Here, it is important to remember that the main goal of this research is not to demonstrate a novel FTTH optimization method that is better than all other methods published so far but rather to indicate the most important parts of the FTTH optimization process from the CAPEX viewpoint and when specialized optimization algorithms are needed. This is the reason the process is decomposed into phases that significantly differ in the ways in which they can be addressed, except Phases 2 and 3, which are quite similar and are not merged into one phase just to strengthen the conclusions of the research—they independently proved that selecting routes for cables is not the main concern in the FTTH network design.

In addition, the goal of the research also slightly constrained us in selecting optimization techniques for our method. To compare the phases on equal terms, we decided to use reasonable and popular techniques that are available in the literature but avoided very specialized approaches, as this could easily result in favoring one of the phases.

We refer to three first optimization phases as intermediate phases, while the last optimization phase is called the final phase. In each phase, entities optimized in other phases are fixed. If the entity has already been optimized, e.g., locations of COs and DPs in Phases 2, 3, and 4 or routes for DCs in Phase 4, the optimized value for it is taken. On the other hand, when the entity has not been subject to optimization, e.g., splitting patterns in Phases 1, 2, and 3 or routes for TCs in Phase 1, simple engineering rules are used to fix them at hand. Those engineering rules accompany detailed descriptions of the optimization phases that will be described in this section.

A vigilant eye would notice that we do not write about optimizing the equipment selected in logical points. More-

over, we do not optimize types of cables used to realize TCs and DCs. The reason is that, as a result of the simplifications in Sect. 3.3, the mentioned parts of the optimization process can be greatly reduced. In fact, we are able to precompute optimal sets of equipment or cables, satisfying each single possible need for them. Using dynamic programming, we can efficiently compute all minimal sets of cabinets accommodating a given number of ports and all minimal sets of cables accommodating a given number of signals (fibers). In this way, the whole process of selecting equipment reduces to selecting one possibility out of a few satisfying a given number of ports for each logical point located in a node and finding a site capable of hosting the selected equipment. This can be solved by comparing all possibilities because the number of options is relatively small. The same approach can be used to optimize cables. In this case, it reduces to selecting a triple: trunk cable type, distribution cable type, and edge preparation, which is even simpler, as, in practice, the quantity of those options for each edge hardly reaches a two-digit number.

Before addressing the description of the phases, we would like to describe an additional important feature of our optimization process. *None of the intermediate phases (Phases 1, 3, and 2) returns a single solution but rather a set of promising solutions.* While solving any intermediate phase, we assume that the following phases do not exist, i.e., their optimization algorithms are substituted by the engineering rules. However, when the considered phase is finished, a returned solution obtained using the previously employed engineering rules will be again subject to optimization in the following phases. This approach does not guarantee that a better solution of an intermediate phase will still be better after executing the following optimization phases. In practice, this is usually not the case. Therefore, we decided to follow the idea of Beam Search [2] and in each intermediate phase generate a number of the most promising solutions instead of just the best one. These assumptions justify a number of decisions taken while designing particular phases of the optimization process.

4.3 Find locations of COs and DPs

In this phase, locations for COs and DPs are selected from sets of available locations. The sets are not given directly but can be deduced from data describing admissible sites. If a node allows for an admissible site of a type characterized by `oltWeightLimit` greater than zero, the node is a potential host for a CO. Similarly, if a node allows for an admissible site of a type characterized by `cabinetWeightLimit` greater than zero, the node is a potential host for a DP. Although we were designing our method for problems with proportionally small sets of available locations for COs and DPs in comparison to a size of a set of nodes, the method can easily cope

with test cases characterized by proportionally large sets of available locations for COs and DPs.

The problem faced in this optimization phase can be seen as a very complicated version of a Facility Location Problem or Warehouse Location Problem [1]. Based on the results of the research published in [29], we decided to use Simulated Annealing [5, 20] as a main methodology for this phase.

Having the methodology selected, the second question we had to answer while designing an optimization method for this phase was how detailed the network model should be. We could either stick to the full model presented in Sect. 3.1 or simplify the model and sacrifice its accuracy to increase tractability. Our preliminary research indicated that simplifying the model was not profitable in general. If the simplification is not sufficiently significant, it will not allow for changing the methodology (heuristics based on Simulated Annealing); thus, it is irrelevant from this perspective. On the other hand, if it is too substantial, the problem is to an extent reduced to a version of the Warehouse Location Problem that can be solved by B&B, and then the accuracy of the model is reduced to unacceptable levels. In other words, if we are able to solve the simplified problem to optimality, an obtained solution is usually by no means near to optimality when the original model is taken into account. These results convinced us to stay with the original model and work on improving fast evaluation methods for it.

The second fundamental problem we faced was the decomposition, as one of the priorities of our research was to design our optimization process in such a way that it can take advantage of multi-threading. The problem of locating COs and DPs cannot be easily decomposed, as areas covered by particular COs or DPs are not defined in advance. On the contrary, they are defined by a solution itself at the end of the process. To geographically decompose the problem, the considered APs have to be clustered, as in [22], or any other method defining partitioning of the area has to be employed. However, such methods have too significant an impact on the final solution—greater than any optimization method could have had working on a given partitioning. This observation led us to the conclusion that the optimization process should be decomposed instead of the problem alone. Taking into account that our goal is to provide a number of good solutions that are close to each other in terms of an objective value but as distant as possible in terms of their positions in the feasibility region, we decided to run a number of hardly dependent executions of Simulated Annealing that exchange their solutions only if one of the executions has problems with finding a solution with an objective value within a given threshold from the best objective value among all the executions.

Assume that N good solutions are needed for the following phase—in this research, we assume that N equals a number of CPU cores available for computations. To return N good

but diverse solutions, each execution maintains a database of the best distinct $N/2 + 1$ (for simplicity, assume that N is even) solutions found in this execution. When the time elapses (time limit is set by a user), $N/2$ best executions are selected, and their respective best solutions, one for each execution, are included in the final result. The remaining good solutions maintained by the executions are then compared with each other, and the remaining $N/2$ spots in the final result are taken by the best among them. The final result is then passed to the second optimization phase.

4.4 Find routes for TCs

Having locations of DPs and COs given, we start to work on routes of TCs. This problem can be seen as a more complicated version of Steiner Tree Problem or, to be more precise, Euclidean Steiner Tree Problem. As shown in [12] also this problem can be efficiently handled by Simulated Annealing, thus we use this metaheuristic also in this case to make sure that all the phases are treated in a similar way.

Our engineering rules for this phase, used when locating COs and DPs, are described in detail in [33] and are a compromise between a shortest path algorithm and a minimum spanning tree algorithm. In general, the engineering rules search for shortest paths between all CO and all unserved DP and select the shortest path among them. The selected path is used in a solution, the DP it connects is marked as served, and all edges on the path that have not been used before have their costs reduced (multiplied by constant α , where $0 < \alpha < 1$). The procedure is repeated as long as there are DPs that have not been served. The method seems to be algorithmically difficult, and in theory, it is much more complex than the algorithms for which the method is a compromise. However, appropriately implemented and working in practice with real-world networks, it is only slightly more time consuming than, for example, a shortest path algorithm.

The optimization method for this phase is Simulated Annealing based on the algorithm described above. A single solution is determined by a set of preferred pairs of DPs (or a CO and a DP). An occurrence of a pair in the set of preferred pairs means that costs of all edges on one of the shortest paths between nodes of the pair are set to zero. Having the set of preferred pairs, we modify costs of edges accordingly and run the above algorithm on the modified graph. In this way, we significantly reduce a feasible region for Simulated Annealing, leaving only a fraction of available solutions. The size of the region is still huge ($2^{|DP| \cdot (|DP| + |CO| - 1)}$), where $|DP|$ is a number of selected DPs and $|CO|$ is a number of selected COs), so we limit it more severely by carefully selecting candidates for preferred pairs. The selection is done as follows. We compute shortest paths from each DP and CO to all other nodes. If there is an edge connecting two nodes having different nearest COs or DPs, a pair of those nearest COs or

DPs is a candidate for a preferred pair. This approach leaves us with a number of candidates for preferred pairs bounded by $O(2^{|DP|})$, which is a reasonable value.

Following the same observations as in the previous phase, we decompose this method into a number of Simulated Annealing executions and eliminate all dependencies between parallel executions to allow for more efficient computations. Having N good solutions returned by the previous phase, we run N independent runs of this phase. Selecting N good solutions for the following phase is done in the same way that it is done for this phase, i.e., $N/2$ best executions are selected and their respective best solutions are included. The remaining $N/2$ spots are taken by the best solutions that have not been included in the final result yet.

4.5 Find routes for DCs

In this phase, routes for DCs are computed. The problem is very close to the problem considered in the previous phase; thus, the methodology to be used here is the same as the methodology used for finding routes for TCs.

The procedure is similar to the procedure of the previous phase. However, this time it works on a much bigger scale with a great deal more simultaneously optimized routes. The importance of this phase arises not only from an enormous size of the feasibility space but also from an impact a simple change in a solution can have on the global objective function. One may claim that changes of routes for DCs have smaller effect than changes of routes for TCs, as their impact is more local. However, this is not always the case. Each change in routes of TCs can impact: the lengths and capacities of trunk cables and equipment needed in COs. On the other hand, each change in the routes of DCs can impact the above, excluding the lengths of trunk cables, but it also impacts: the lengths and capacities of distribution cables and equipment needed in DPs. Therefore, by no means can this phase be neglected or be given lower priority than the priority of the previous phase. In fact, as indicated in Sect. 5, this phase is more important than the phase of finding routes for TCs.

As mentioned before, the methodology used here is similar to the methodology used in the previous phase. However, this time, we introduce dependencies between the parallel executions. Although the executions work with slightly different networks, i.e., differing in the locations of COs and DPs, and the routes of TCs, there exists a correlation between impacts of the same changes in parallel executions. In other words, the same change in the set of preferred pairs usually impacts a global objective value in a similar way. We take advantage of this feature, as in this phase the size of the feasibility space and the fact that we are able to provide a starting solution that is near the optimum justifies using relatively low temperatures for Simulated Annealing, making it close to Local Search, in which the greatest difficulty lies in

finding candidate solutions that improve the global objective value. Therefore, when finding a promising modification of the set of preferred pairs in one execution, we also immediately consider it in all other executions. This approach greatly improves the efficiency of the method.

4.6 Find splitting patterns

In the last phase of the optimization process, all locations of COs and DPs are given. Moreover, all routes for TCs and DCs have been already computed. All equipment and cables will be selected based on the precomputations; thus, the only thing left for optimization are splitting patterns used by clients. We understand a splitting pattern to be a triple of splitters (for which “no split” is considered as a 1:1 splitter) used in: AP, DP, and CO, in addition to a card used in a CO.

The problem faced in this phase cannot be easily classified in any group of well-known combinatorial problems. In our opinion, it is nearest to a type of multi-layer bin-packing, as the biggest challenge in this phase is to select splitting patterns in such a way that neither splitter outputs nor ports of line cards are wasted. As Simulated Annealing is used for all three previous optimization phases and it can also be efficiently used for bin-packing problems [30], we decided to also take advantage of this strategy in the final phase of the optimization process.

An engineering rule for this phase is as follows. First, we select line cards for each client assuming that there will be one dominant card type of the first choice for each CO. The first choice card is supposed to be able to serve a majority of clients of the CO using the maximum allowable split. Having the TCs and DCs given, we can compute how many clients connected to a given CO cannot be served using the first choice card with a maximum split and at least a two-level split (notice that due to power constraints, a two-level split has bigger requirements than a one-level split). If this number justifies using more expensive card types, we assign a number of such cards to clients starting from those located in the most distant APs. The next step is to select the splitters to be located in each AP. As long as a number of unserved clients in an AP is greater or equal to the greatest number of outputs among all available splitters for this AP (notice that the already selected card type can make some splitters unavailable for this AP), one of such splitters is located in the AP. After that, the choice is made between two extremities. Either the smallest splitter capable of serving all remaining clients is selected or the smallest set of splitters that serves exactly all remaining clients is employed. The choice is made based on a simple rule that calculates a fraction of wasted outputs for the former option. If the waste is smaller than a given number (0.2 in our research), the former option is selected. In the other case, the latter possibility is chosen. We will call this rule the “waste” rule in the rest of this section.

Having line cards selected and splitters in APs located, we start to compute splitters to be allocated in DPs using a following rule: If a splitter is located in a DP, all its outputs have to be used, and all served clients connected to it have to be served efficiently, i.e., the signal they receive should be split in such a way that the number of clients served by a single fiber connected to a card in a CO is maximized. This rule allows us to locate splitters in DPs quite efficiently and leaves us with a very simple problem of locating splitters in COs, which we solve with the trivial rule: Starting from a splitter with the highest number of outputs, we locate it if it can be efficiently used, i.e., if there are more TC fibers that can be connected to it without violating the power budget rules than the number of outputs of any smaller splitter. This simple operation ends the process and leaves us with a reasonable splitting pattern allocated to each client. Having described the engineering rules, let us now move to the optimization methods applied in this phase.

As in the previous section, the feasibility region of the problem considered in this phase is also tremendous. We limit it by allowing only two options for each AP, which leaves us with $2^{|AP|}$ cases to consider. The two options are tightly connected with the “waste” rule described above. Instead of using the threshold in the rule, we subject the decision of the “waste” rule to optimization. All other decisions stay as in the engineering rule. The methodology behind this phase is also Simulated Annealing. However, this time, the decomposition process is much more complex.

This is the first phase that can be easily decomposed with respect to physical locations, as the areas covered by given COs cannot change. Therefore, the problem can be solved independently for each CO. We use this feature in our method but not from the very beginning. The previous phase leaves us with N promising solutions, as each thread has returned one. We start the current phase optimizing all those returned solutions independently and sharing promising changes between threads, as in the previous phase. However, finding splitting patterns is the last accord of the optimization process; thus, the method described in this section should return a single best solution—not a set of equally good promising solutions, as in the previous phases. Therefore, here, we gradually shift computing power from worst solutions to their better counterparts. We can do so, as the problem can be easily decomposed, and $|CO|$ independent threads can work toward optimizing a single solution without interferences.

4.7 Complexity, scalability, and optimality

All algorithms used in the above phases are polynomial. The most time-consuming part of the whole optimization process consists of executing the engineering rules of [33]. The rules are used in three first phases of the process, and their complexity dominates the complexity of other computations.

Although a theoretical complexity of applying the engineering rules is $O(|\mathcal{V}|^3)$, according to [33], for real-world access network topologies, the complexity in practice is very close to the complexity of a single shortest path algorithm, which on the other hand is intuitively the smallest computational complexity we can obtain in access network design.

In addition, we emphasize that the presented algorithm is based on Simulated Annealing—it constantly evaluates neighboring solutions that only slightly differ from solutions evaluated previously. The most severe modifications of a solution consist of removing or adding one CO. In such a situation, affected is the recently removed or added CO together with all COs that serve areas adjacent to the area served by the removed (added) CO. The complexity of evaluating such a local modification, if appropriately implemented, does not increase if a certain size of a network is reached. Therefore, as far as we are able to provide and store an initial solution, the size of a considered area is not a factor seriously limiting the scalability of the algorithm.

As for the optimality of obtained solutions, the complexity of the formal formulation does not allow for meaningful conclusions concerning the optimality gap—lower bounds for considered real-world networks equal only a small fraction of costs obtained using the presented methods. However, using a simplified model, we are able to judge optimality of the last (fourth) phase that optimizes splitting patterns. In this case, the optimality gap is less than 11 % on average.

5 Numerical results

In this section, we indicate the parts of the optimization process described in Sect. 4 that are the most relevant from the capital expenditures viewpoint. We will indicate where potential savings are significant and which phases of the optimization process can be neglected while judging profitability of deployment because of their lower expected impact on the final solution. We indicate where appropriately designed engineering rules are capable of providing a valuable solution, and on the other hand, the parts of the design process that should definitely be supported by carefully crafted optimization methods.

We ran our tests on a Fujitsu RX200 server equipped with two Intel Xeon E5-2620v2 6C/12T 2.10GHz processors with 8 cores (16 threads) and 64 GB of RAM dedicated exclusively for our research. We implemented our methods in C# using Visual Studio 2010 and ran them under Windows Server 2012.

Networks and demand sizes used in our research were based on the available real-world data. We selected four areas for the research: two large cities of more than one million inhabitants (resulting in nearly one million potential clients) and two smaller towns with hundreds of thousands of poten-

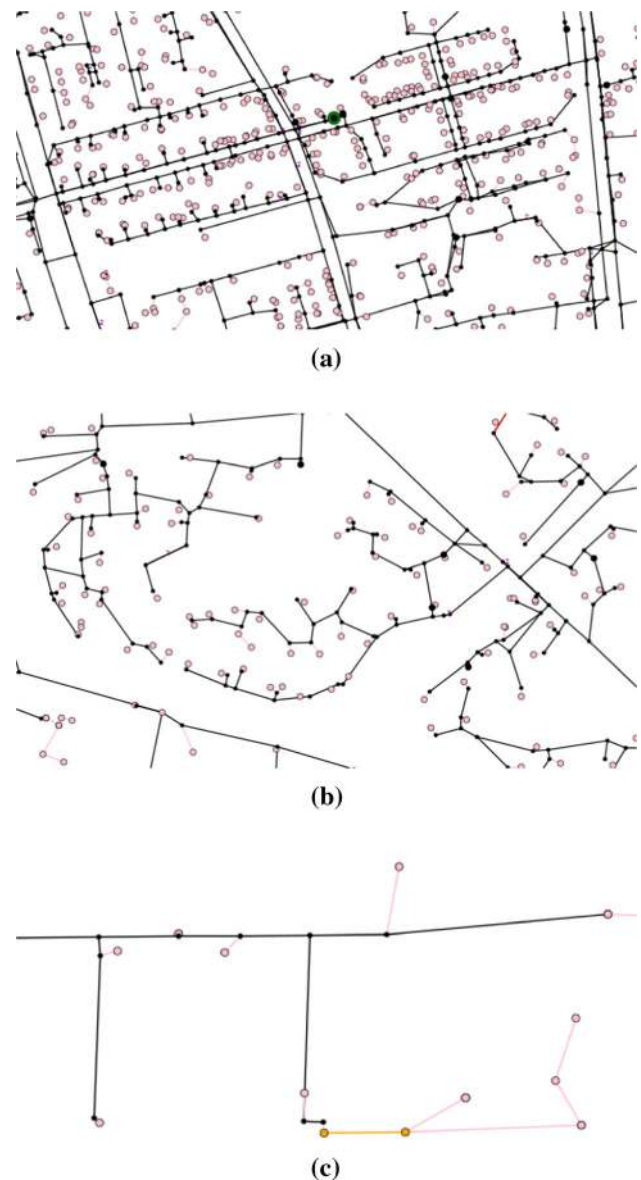


Fig. 5 Examples of building layouts **a**, City center. **b**, Blocks of flats. **c**, Detached houses

tial clients. The areas mix a number of different building patterns ranging from densely packed medieval Old Town-like quarters to allegedly erratic arrangements of housing projects consisting of blocks of flats and vast areas of sparsely populated districts of detached houses. Exemplary topology screenshots can be found in Fig. 5, in which Fig. 5a is a city center, Fig. 5b is an area of block of flats, and Fig. 5c is a street of detached houses.

We considered eight networks in total: four based on the large city areas and an additional four based on the town areas. For the city areas, we considered cases for which all possible clients have to be provided with the service (referred to as *full coverage (f)*) and only parts of a city are targeted based on external data (referred to as *partial coverage (p)*). An additional four networks are based on the town areas

Table 4 Networks

Name	Nodes	Edges	Clients	Ducts (km)	Area (km ²)
City1-f	79,768	86,604	822,633	2681.3	517.5
City1-p	79,768	86,604	245,297	2681.3	517.5
City2-f	70,815	77,321	741,270	2366.1	475.7
City2-p	70,815	77,321	195,591	2366.1	475.7
Town1-b	8424	8851	113,496	286.1	36.7
Town1-g	8424	8851	113,496	0.0	36.7
Town2-b	24,009	26,852	256,393	757.7	76.4
Town2-g	24,009	26,852	256,393	0.0	76.4

and differ in terms of available infrastructure. We consider two different cases: *greenfield* (*g*) and *brownfield* (*b*). In the former case, we do not take into account any data concerning existing infrastructure and assumed that to roll out any cable, a trench must first be dug. On the other hand, in the latter scenario, the existing infrastructure can be reused; thus, the expected costs of rolling out a network are much smaller. We based the choice of scenarios on two assumptions. First, in larger cities, some type of existing infrastructure has to be present; thus, the greenfield scenarios were considered only for the smaller towns. Second, in cities, competing businesses have to be expected; thus, the partial coverage scenarios are justified there.

In summary, we end up with eight different representative networks in total. In *City1-f* and *City2-f*, all potential clients in the large cities have to be connected, while in *City1-p* and *City2-p*, only some districts of those cities are targeted. In *Town1-b* and *Town2-b*, all potential clients in two small towns have to be connected, while in *Town1-g* and *Town2-g*, the same towns are considered; however, the infrastructure is unavailable in those cases. Details concerning the networks considered in the research can be found in Table 4.

Considered equipment price lists were created by compiling a number of different offers from different vendors and are depicted in Table 5.

As far as labor cost is concerned, we considered two scenarios—one for *developing* countries and one for *developed* countries. In the former, we assumed the labor cost to be four times smaller than in the latter. Therefore, in Table 6, in which labor costs are gathered, expenditures are depicted by ranges and not by single values.

Before drawing conclusions from the results, it is important to restate one of the assumptions of the research. This analysis is a case study. That is, strictly speaking, the findings apply only to the reference networks used for the analysis. However, because the reference networks are typical for European cities, the results are believed to be transferable to other areas with abundant urban infrastructure and demand densities similar to those observed in Europe.

In Table 7, the efficiency of our implementation is summarized. It contains the average running times needed to

Table 5 Equipment cost

Equipment	Cost (EUR)
OLT with 8 card slots	6000
OLT with 4 card slots	4000
Card with 8 B-class lasers	8000
Card with 8 C-class lasers	11,000
Equipment for hosting a CO (indoor)	4000
Equipment for hosting a CO (outdoor)	16,000
Splitter 1:2	10
Splitter 1:64	120
Pigtail	8
Single splice	2
1 km of a 6-fiber cable	3000
1 km of a 60-fiber cable	6000
Closure accommodating up to 12 splices	20
Closure accommodating up to 500 splices	100
Cabinet serving up to 12 fibers	250
Cabinet serving up to 500 fibers	1500

Table 6 Labor cost

Activity	Cost (EUR)
1 km of trenching	7500–30,000
Building 1 km of ducts	20,000–80,000
Rolling out 1 km of cable	500–2000
Preparing a location for a cabinet	500–2000
Setting up a cabinet (prepared location)	100–400
Setting up a central office	25,000–100,000

Table 7 Average evaluation time (ms)

	Full	CO/DP	TC	DC	Splitting patterns
Min	53.1	40.7	29.0	46.7	9.2
Max	821.0	492.0	328.9	429.4	43.9

evaluate a solution in each phase of the optimization process. In column *full*, the time needed to evaluate a full solution having locations of COs and DPs given is depicted. In column

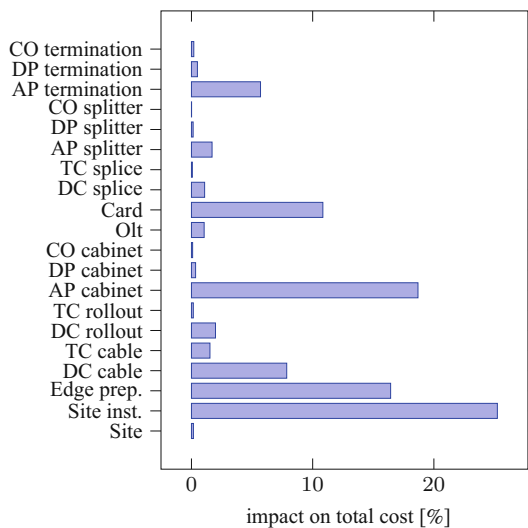


Fig. 6 Impact of different parts of a solution on a total cost

CO/DP, the time needed to evaluate an impact of a change in CO or DP locations is given. The following two columns contain the average times needed to evaluate changes in routes for TCs and DCs, while in column *splitting patterns* the average time to evaluate a change in splitting patterns is depicted. The minimum average times were obtained for test cases *Town1-b* and *Town1-g*, while the maximum times were needed for test cases *City1-f* and *City2-f*. In summary, for the worst test case, we are able to evaluate more than 4k full solutions or more than 80k changes in splitting patterns during an hour.

A structure of total costs of optimized solutions is shown in Fig. 6. It indicates that nearly half of the total cost is the AP-related cost of cabinets, splitters, and installations. Those expenditures can hardly be optimized in our research, as locations and sizes of AP are given. Therefore, we decided to present all obtained optimization savings using absolute values, as it is difficult to judge what should be the reference total cost that will allow us to present savings in relative values. In addition, we do not compare our method to other methods available in the literature for two reasons. First, there are currently no published methods working with such a detailed network model and capable of handling such large test cases. Second, our main goal was not to develop the best novel optimization method but rather to indicate the design phases that the operational research effort should be concentrated on. The proof that such an effort pays off is indicated in Table 8, in which the general savings, reaching millions of euros, are presented.

The savings were obtained by running our method for 10h for each test case (2h for each optimization phase). We selected the time limit based on our industrial experience—it takes at least 1 working day to prepare data for a serious optimization; thus, the optimization process alone can be executed overnight. Obviously, this does not mean that

Table 8 Generated savings (EUR)

Name	Inexpensive labor	Expensive labor
City1-f	3,360,025	4,664,181
City1-p	1,925,977	2,496,654
City2-f	3,004,213	4,126,500
City2-p	1,589,168	2,105,121
Town1-b	420,884	514,598
Town1-g	588,577	1,494,697
Town2-b	1,008,831	1,290,655
Town2-g	1,876,465	5,153,098

the presented method needs 10h to return a solution. The methodology used here assures that a reasonable solution (based on engineering rules) is available after less than ten seconds even for large networks; thus, the method can also be successfully used for fast but approximate evaluation of numerous possible scenarios and selecting the best scenario for the overnight optimization.

The savings can be divided and attributed to different parts of the optimization process, which is shown in Fig. 7. In our experiments, we compare our method to a simple benchmark that implements the following rules:

- Each AP is connected to the nearest DP using a shortest path, instead of using the algorithm of [33].
- Each DP is connected to the nearest CO using a shortest path, instead of using the algorithm of [33].
- A splitter can be installed only if none of its outputs will be wasted, instead of using the “waste” rule of Sect. 4.6.

According to our experiments, just replacing those rules with the engineering rules presented in this paper leads to significant savings reaching millions of euros (combining the results of Table 8 with Fig. 7). A very interesting observation is that although the costs of cable-related parts of a solution, i.e., cables, roll-outs, splices, and edge preparations, constitute nearly a third of the total costs (see Fig. 6), the importance of trunk and distribution route optimization is not very significant. Our experiments indicate that as far as trunk and distribution route optimization is concerned, the engineering rules do their job, and there is not much space for the savings generated by sophisticated optimization techniques. The situation looks totally different when CO/DP locations or splitting patterns are taken into account. In the former case, even providing any reasonable engineering rules becomes a challenge, while in the latter case, making local decisions (and only this type of decision can be made using engineering rules) cannot lead to near-optimal solutions due to strong relations between different APs or DPs.

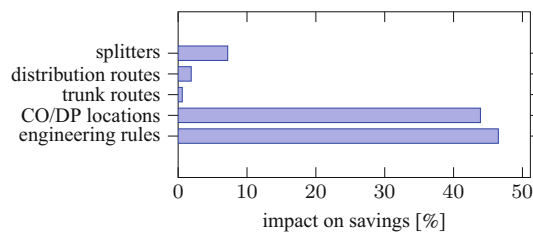


Fig. 7 Impact of different optimization phases on the gain

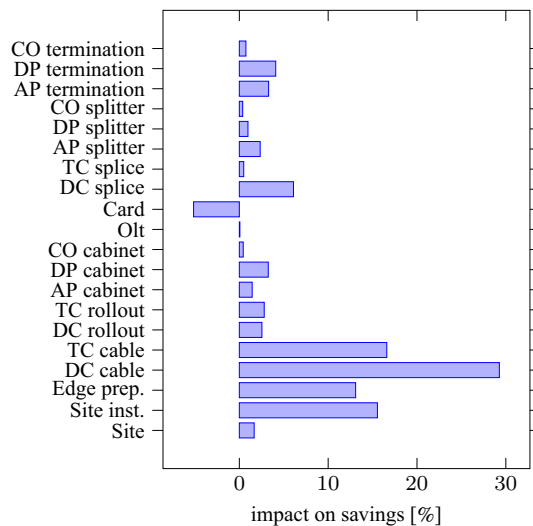


Fig. 8 Impact of different parts of a solution on savings

The relation is shown in Fig. 8 presenting how savings are divided between different components of the total cost on average. The figure indicates that more than 45 % of savings was generated by reducing the costs of TC and DC cable types. On the other hand, the optimization process increases on average a total cost of line cards by 5 % of the total savings. It leads us to the conclusion that, although the direct optimization of trunk and distribution routes does not bring significant savings (see Fig. 7), an indirect optimization of types (thickness) of cables from the appropriate selection of splitting patterns and an indirect optimization of lengths of cables from the appropriate selection of CO and DP locations seems to be a key to success in the FTTH network design. Notice that minimizing the costs of cables usually requires less efficient usage of OLT line card ports, thus leading to a visible increase in the total costs of the line cards. However, this increase does not have to be accompanied by the increase in the total cost of OLTs. On the contrary, the results of Fig. 8 indicate that, after the optimization, the total OLT cost is reduced. This can result either from more efficient packing of line cards in OLTs or from replacing cheaper B-class cards with their C-class counterparts to reach distant clients with more efficient splitting patterns.

6 Conclusion

In the paper, we presented an optimization platform for a FTTH network design capable of minimizing the CAPEX of network deployment. We described the methodology used in the platform and used it to evaluate the importance of different parts of the FTTH network design process. We conclude that the most important decisions from the viewpoint of CAPEX are those dealing with the locations of central offices and distribution points and those addressing splitting patterns. In our research, we indicate that designing appropriate routes for trunk and distribution cables is very important as far as total costs are concerned. However, moderately complicated engineering rules are capable of providing surprisingly good results in this field, allowing for correct evaluation of costs of deployment without using any other sophisticated methods for optimizing the cable routes. Still, although not critical from the viewpoint of cost evaluation, dedicated route optimization methods should be used to minimize the costs of the final detailed design.

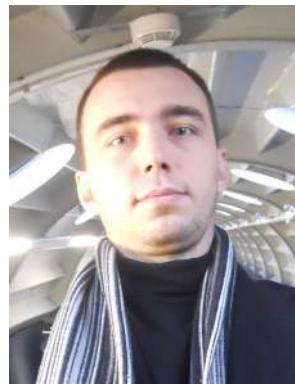
Acknowledgments This research was supported by the Ministry of Science and Higher Education (Poland) under Grant IP2012 065372 “Distributed optical access network optimization”.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Baumol, W. J., & Wolfe, P. (1958). A warehouse-location problem. *Operations Research*, 6(2), 252–263.
- Bisiani, R. (1987). *Encyclopedia of Artificial Intelligence*. Beam search New York, NY: Wiley.
- Bley, A., Ljubić, I., & Maurer, O. (2013). Lagrangian decompositions for the two-level FTTH network design problem. *EURO Journal on Computational Optimization*, 1(3–4), 221–252.
- Casier, K., Lannoo, B., Ooteghem, J. V., Verbrugge, S., Colle, D., Pickavet, M., et al. (2009). Game-theoretic optimization of a fiber-to-the-home municipality network rollout. *Journal of Optical Communications and Networking*, 1(1), 30–42.
- Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45, 41–51.
- Chardy, M., Costa, M. C., Faye, A., & Trampont, M. (2012). Optimizing splitter and fiber location in a multilevel optical FTTH network. *European Journal of Operational Research*, 222, 430–440.
- Chen, B., Shi, L., & Gan, C. (2010). Optimization model and simulation analysis of wavelength-shared WDM-PON. In: *Proceedings of the 2nd International Conference on Information Science and Engineering (ICISE)*, Hangzhou, pp. 2216–2219.
- Chowdhury, R., & Jaumard, B. (2012). A cross layer optimization scheme for WDM PON network design and dimensioning. In: *Proceedings of the IEEE International Conference on Communications (ICC)*, Ottawa, (pp. 3110–3115). IEEE

9. Cobo, L., Chamberland, S., & Ntareme, A. (2012). Low cost fiber-to-the-node access network design. In: *Proceedings of the 15th International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, Rome, pp. 1–4.
10. Effenberger, F., Clearly, D., Haran, O., Kramer, G., Li, R. D., Oron, M., et al. (2007). An introduction to PON technologies. *IEEE Communications Magazine*, 45(3), 17–25. topics in optical communications.
11. Garfinkel, R., & Nemhauser, G. (1972). *Integer Programming*. New York, NY: Wiley.
12. Grimwood, G.R. (1994) *The Euclidean Steiner Tree Problem: Simulated Annealing and Other Heuristics*. Ph.D. Thesis, University of Wellington, Victoria.
13. Grötschel, M., Raack, C., & Werner, A. (2013). Towards optimizing the deployment of optical access networks. *EURO Journal on Computational Optimization*. doi:10.1007/s13675-013-0016-x (available online).
14. Hervet, C., Faye, A., Costa, M.C., Chardy, M., & Francfort, S. (2013). Solving the two-stage robust FTTH network design problem under demand uncertainty. In: *Proceedings of the International Network Optimization Conference*, Costa Adeje.
15. Hervet, C., & Chardy, M. (2012). Passive optical network design under operations administration and maintenance considerations. *Journal of Applied Operational Research*, 222(3), 152–172.
16. Kadhim, D. J., & Hussain, N. A. (2013). Link and cost optimization of FTTH network implementation through GPON technology. *Communications and Network*, 5, 438–443.
17. Kantarci, B., & Moutfah, H. T. (2012). Availability and cost-constrained long-reach passive optical network planning. *IEEE Transactions on Reliability*, 61(1), 113–124.
18. Karp, R.M. (1972) Reducibility among combinatorial problems. In: *Complexity of Computer Computations*, pp. 85–103. Berlin: Springer
19. Khan, S. U. (2005). Heuristics-based PON deployment. *IEEE Communications Letters*, 9(9), 847–849.
20. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
21. Kokangul, A., & Ari, A. (2011). Optimization of passive optical network planning. *Applied Mathematical Modelling*, 35(7), 3345–3354.
22. Lakic, B., & Hajduczenia, M. (2007). On optimized passive optical network (PON) deployment. In: *Proceedings of the 2nd International Conference on Access Networks*, Ottawa, pp. 1–8.
23. Li, J., & Shen, G. (2009). Cost minimization planning for greenfield passive optical networks. *Journal of Optical Communications and Networking*, 1(1), 17–29.
24. Louchet, H., Richter, A., Patzak, E., & Schlosser, M. (2011). Impact of strategic decisions on the planning of FTTx-networks: A case study. In: *Proceedings of the 12th ITG Symposium on Photonic Networks*, pp. 1–5.
25. Mitcsenkov, A., Paksy, G., & Cinkler, T. (2011). Geography- and infrastructure-aware topology design methodology for broadband access networks (FTTx). *Photonic Network Communications*, 9(21), 253–266.
26. Pióro, M., & Medhi, D. (2004). *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Francisco, CA: Morgan Kaufman.
27. Poon, K.F., & Ouali, A. (2011). A MILP based design tool for FTTH access networks with consideration of demand growth. In: *Proceedings of the 6th International Conference on Internet Technology and Secured Transactions*, Abu Dhabi.
28. Poon, K.F., Mortimore, D.B., & Mellis, J. (2006). Designing optimal FTTH and PON networks using new automatic methods. In: *Proceedings of the 2nd Institution of Engineering and Technology International Conference on Access Technologies*, Cambridge.
29. Qin, J., Ni, L. L., & Shi, F. (2012). Combined simulated annealing algorithm for the discrete facility location problem. *The Scientific World Journal*, 2012, 576,392. doi:10.1100/2012/576392.
30. Rao, R. L., & Iyengar, S. S. (1994). Bin-packing by simulated annealing. *Computers & Mathematics with Applications*, 27(5), 71–82. doi:10.1016/0898-1221(94)90077-9.
31. Segarra, J., Sales, V., & Prat, J. (2012). Planning and designing FTTH networks: Elements, tools and practical issues. In: *Proceedings of the 14th International Conference on Transparent Optical Networks*, Coventry.
32. van Loggerenberg, S.P., Grobler, M.J., & Terblanche, S.E. (2012). Optimization of PON planning for FTTH deployment based on coverage. In: *Proceedings of the Southern Africa Telecommunications Networks and Applications Conference*, Fancourt.
33. Żotkiewicz, M.: Fast algorithms for access network design. In: *Proceedings of the 3rd International Symposium on Combinatorial Optimization (ISCO)*, Lisbon.



Matusz Żotkiewicz an assistant professor at Warsaw University of Technology (WUT), received his BEng at Coventry University, MSc at WUT, and PhD in informatics at Telecom SudParis and in telecommunications at WUT. He was a visiting researcher at Universitat Politècnica de Catalunya and at University of Luxembourg. His research interests concentrate on network planning and optimization. He is an author of over 40 technical papers published in international telecom and OR journals and conference proceedings.



Mariusz Mycek an assistant professor at Warsaw University of Technology (WUT), received his MSc and PhD degrees in telecommunications at WUT, Institute of Telecommunications. His research interests concentrate on network management, and network planning and routing. He is an experienced programmer. He has knowledge of the network management standardization and has both theoretical and practical experience related to the modelling of managed network resources, as well as to the analysis, designing and implementation of the network management interfaces. His professional interest includes also the application of advanced methods of the mathematical programming in the design and optimization of the telecommunications networks. He has participated in many research projects for telecom industry. He is an author of over 20 technical papers presented in international telecom and OR journals and conference proceedings. He is involved in European research, including active participation in the EU 6th and 7th Framework projects (such as Network of Excellence Euro NGI, Euro FGI and Euro NF), Celtic-Eureka (project MANGO) and COST Actions.



Artur Tomaszewski received MSc, PhD and DSc degrees from the Warsaw University of Technology (WUT) in 1990, 1993 and 2014, respectively. He is an associate professor at the Faculty of Electronics and Information Technologies of WUT. His research interests focus in the areas of telecommunications network architectures, network management and network optimization.