computers

MDPI

Article

Specialized Genetic Operators for the Planning of Passive Optical Networks

Oeber Izidoro Pereira [1], Edgar Manuel Carreño-Franco [1], Jesús M. López-Lezama [2,*] and Nicolás Muñoz-Galeano [2]

[1] Centro de Engenharias e Ciencias Exatas CECE, Universidade Estadual do Oeste do Paraná (UNIOESTE), Av. Tarquínio Joslin dos Santos, 1300, Foz do Iguaçu 85870-650, Brazil; edgar.franco@unioeste.br (E.M.C.-F.)

[2] Grupo de Investigación en Manejo Eficiente de la Energía (GIMEL), Departamento de Ingeniería Eléctrica, Universidad de Antioquia (UdeA), Calle 70 No. 52-21, Medellin 050010, Colombia; nicolas.munoz@udea.edu.co

* Correspondence: jmaria.lopez@udea.edu.co

Abstract: Passive Optical Networks (PONs) are telecommunication technologies that use fiber-optic cables to deliver high-speed internet and other communication services to end users. PONs split optical signals from a single fiber into multiple fibers, serving multiple homes or businesses without requiring active electronic components. PONs planning involves designing and optimizing the infrastructure for delivering fiber-optic communications to end users. The main contribution of this paper is the introduction of tailored operators within a genetic algorithm (GA) optimization approach for PONs planning. A three vector and an aggregator vector are devised to account, respectively, for physical and logical connections of the network, facilitating the execution of GA operators. This codification and these operators are versatile and can be applied to any population-based algorithm, not limited to GAs alone. Furthermore, the proposed operators are specifically designed to exploit the unique characteristics of PONs, thereby minimizing the occurrence of unfeasible solutions and accelerating convergence towards an optimal network design. By incorporating these specialized operators, this research aims to enhance the efficiency of PONs planning, ultimately leading to reduced costs and improved network performance.

Keywords: genetic algorithms; genetic operators; optimization; passive optical networks

check for updates

Citation: Pereira, O.I.; Carreño-Franco, E.M.; López-Lezama, J.M.; Muñoz-Galeano, N. Specialized Genetic Operators for the Planning of Passive Optical Networks. *Computers* **2024**, *13*, 259. https://doi.org/10.3390/computers13100259

Academic Editors: Paolo Bellavista and Lilatul Ferdouse

Received: 28 August 2024
Revised: 29 September 2024
Accepted: 8 October 2024
Published: 10 October 2024

1. Introduction

The rapid expansion of high-speed internet demand has highlighted the critical need for efficient and cost-effective telecommunications infrastructure. With increasing reliance on internet-based services, traditional copper-based networks often fail to meet the required bandwidth and speed, leading to a push towards deploying passive optical networks (PONs). This telecommunications technology uses fibre-optic cabling to provide internet and other services to end users. Unlike active optical networks, PONs do not require electrically powered components within the distribution network, instead relying on passive optical splitters to separate and route signals from a single optical fiber to multiple end users [1,2]. This results in lower operational and maintenance costs since there are no power requirements and fewer components that can fail. Additionally, PONs are simpler to deploy and more energy-efficient, which can lead to lower overall costs. However, PONs also have limitations, such as lower bandwidth scalability and higher latency compared to Active Optical Networks (AONs). These later employ active electronic equipment, such as switches and routers, to manage and direct the flow of data within the network. This allows for higher bandwidth and more flexible network management, as each connection can be individually monitored and optimized. AONs also generally offer lower latency and better performance scalability, making them suitable for high-demand applications.

However, AONs come with higher operational and maintenance costs due to the need for power and the complexity of the active components [3].

A typical PON consists of an Optical Line Terminal (OLT) at the service provider's central office and multiple Optical Network Units (ONUs) or Optical Network Terminals (ONTs) near end users. The connection between the OLT and ONUs/ONTs is established through a passive optical splitter (see Figure 1) [4]. PONs are applied in various contexts, primarily in residential broadband services, which deliver high-speed internet, television, and telephone services to homes. They also provide reliable and high-speed connectivity for businesses, including data, voice, and video services. Additionally, PONs support mobile network operators by connecting cell towers to the core network and enable high-speed connectivity within universities, schools, and municipal networks through campus and community networks.

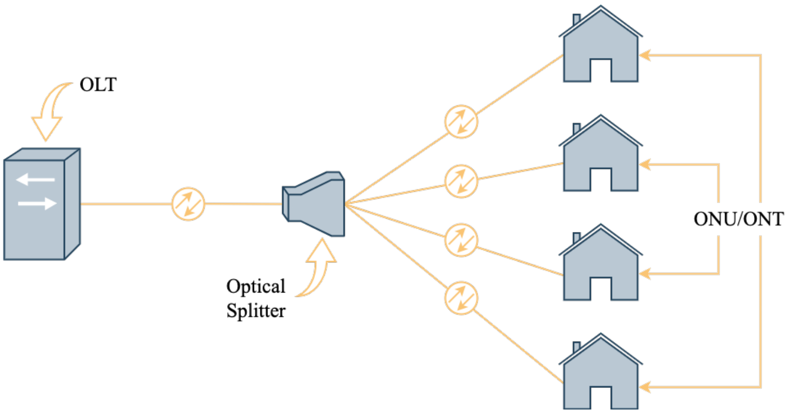


Figure 1. Illustration of a PON.

There are several studies in the specialized literature that deal with the design and optimization of PONs. In [5], the authors present the basic steps for converged network design and planning and evaluate algorithms for wireless backhaul optimization based on a PON architecture. They address capacity requirements per access segment and optical backhaul segment, physical layer limitations, and overall cost using both heuristic algorithms and stochastic optimization techniques. In [6], the significance of multi-layer planning in PON design is examined, and a PON planning tool is presented. The multi-layer mapping and routing mechanism, the automatic multi-layer planning algorithm, and related failure simulations are analyzed. Additionally, automatic planning and failure simulation based on the multi-layer model are experimentally demonstrated. A PON planning methodology which allows different alternatives to be compared and expected benefits to be analyzed is proposed in [7]. In this case, a particular use case of an access operator is taken as an example. Particle Swarm Optimization (PSO) is used in [8] for Optical Distribution Network (ODN) design in a Fiber-To-The-Home (FTTH) network based on a PON. In [9], the design of a metropolitan PON consisting of tree-type access networks is presented. In this case, a channel plan and network architecture design were developed to provide direct optical paths between users, facilitating classical and one-way communication.

In [10], the authors propose an all-optical processor to mitigate the limitations of high-speed PONs, addressing the growing demand for higher data rates and improved performance; such a processor enhances system efficiency and reduces power consumption. In [11], a combined optimization method is proposed for a transmitter and receiver using probabilistic shaping techniques to reduce the error rate in PON signal transmissions. A cost analysis of rural roll-out using a long-reach PON is presented in [12]. In this case, a holistic framework was developed to examine real-world FTTH deployment scenarios, using a rural county in Ireland as a case study. An in-depth techno-economic analysis was conducted to identify the most applicable methods for rural deployment.

A network planning for dual residential-business exploitation of PON to provide symmetrical 1 Gb/s services is presented in [13]. In this case, the authors compare various types of next-generation PONs. In [14], the authors propose an energy-efficient PON that uses orthogonal frequency division multiple access and wavelength division multiplexing to facilitate the deployment of radio units in a communication network. An approach to enhance the efficiency and capacity of PONs using dynamic spectrum allocation is proposed in [15]. In this case, the proposed methodology relies on wavelength division multiplexing.

The high costs associated with implementing PONs, particularly in infrastructure and maintenance, present significant challenges for service providers. This requires the development of optimized planning methodologies to ensure economical and scalable network deployment [16,17]. PON planning is also explored in [18,19]; nonetheless, the authors primarily focused on standard genetic algorithms (GAs) without delving into the customization of operators to exploit the unique characteristics of PONs, which presents an opportunity to enhance the efficiency and feasibility of GAs in network planning. This paper aims to address this gap by developing specialized operators for GAs in PON planning. The main contribution of this research is the proposal of versatile codification and operator schemes and can be applied to any population-based algorithm, not limited to GAs. It is worth mentioning that the comparison of metaheuristics for PONs planning is not the objective of this study. Instead, this paper proposes specialized operators within a GA optimization framework that are tailored to leverage the specific attributes of PONs, reducing the number of unfeasible solutions and accelerating the convergence towards an optimal network design. By integrating these specialized operators into GA optimization, this research seeks to provide a more effective tool for PON planning, ultimately contributing to cost reduction and improved network performance.

2. Mathematical Modeling

The mathematical model used in this paper is given by Equations (1)–(15). This model is inspired by the multi-level approach presented in [20], being adjusted to the design of PON networks.

$$Min\left(\sum_{\mathfrak{p}\in P}\left(\sum_{\ell=1}^{ml}\left(\sum_{i\in M}\sum_{j\in N}x_{\ell_{ij}}m_{\ell_{ij}}+\sum_{j\in N}t_{\ell_j}n_{\ell_j}\right)\right)+\sum_{k=0}^{n(F)}cfm_k\right) \tag{1}$$

Subject to

$$P=\{\mathfrak{p}\in P|n(P)\geq qp\} \tag{2}$$

$$1\leq\ell\leq ml \tag{3}$$

$$x_{\ell=1_{ij}}>0 \tag{4}$$

$$x_{\ell_{ij}}>0\ \forall\ m_{\ell_{ij}}=1 \tag{5}$$

$$n(N_\ell)=n(M_\ell) \tag{6}$$

$$\sum_{i\in M}m_{\ell_{ij}}\leq ms \tag{7}$$

$$At = \left(\sum_{\substack{\ell=1 \\ i \in M \\ j \in N}}^{L} ax_{\ell_{ij}} m_{\ell_{ij}} + at_{\ell_j} n_{\ell_j} \right) + at_{ac} \leq pb \tag{8}$$

Equations and constants relevant to PONs technology:

$$qp = \left\{ {}^{M_{\ell=1}}/_{qcp} \right\} \tag{9}$$

$$qcp = 128 \tag{10}$$

$$ml = {}^{apb}/_{3\text{ dB}} \tag{11}$$

$$pb = 28\text{ dB} \tag{12}$$

$$at_{ac} = 3\text{ dB} \tag{13}$$

$$apb = pb - at_{ac} = 28 - 3 = 25\text{ dB} \tag{14}$$

$$ms = 32 \tag{15}$$

Equation (1) represents the objective function (OF), which minimizes the total cost of building the network for customer service, considering one or more concentrator nodes ($\mathfrak{p}$) needed to meet the users. This equation brings together the cost of the path between the demand and facility nodes ($x_{\ell_{ij}}$) plus the cost of the aggregating node (t_{ℓ_j}) for all levels (ℓ). These two components of the OF depend on the value of the binary variables associated to the trajectory i-j ($m_{\ell_{ij}}$) and aggregator j (n_{ℓ_j}), respectively. The third component of the OF accounts for the cost of the lines in the network that are not used by any demand (cfm_k).

In the processing of each level (ℓ) of the PON network, the aggregator nodes of the previous level ($\ell - 1$) become the demand nodes of the current level (ℓ) and so on, until reaching the last level (ml), where the facilitating or aggregator node is elected as the main node. The cost of the aggregator node (t_{ℓ_j}) varies according to the number of demands. This value is counted only once through the n_{ℓ_j} control variable, regardless of how many demands are met by these nodes, provided that the limit defined by (ms) is respected.

Equation (2) indicates that the concentrator node $\mathfrak{p}$ belongs to the group of available concentrator nodes (P), but it emphasizes that the cardinality (size) of the set of concentrator nodes $[n(P)]$ must be equal or greater than the variable of the PON technology (qp). The qp variable indicates the number of central network equipment (OLT) that will be needed to meet all end customers ($M_{\ell=1}$ or level demands $\ell = 1$).

The constraint given by Equation (3) points out that there must be at least one level (ℓ) for the calculation of interconnection costs, where ℓ varies from 1 to ml levels. The constraint given by Equation (4) checks whether there are lines that connect the end clients to the network; that is, whether there is at least one edge value (line cost) associated with the path that goes from the demand node i to the facilitator node j to the first level ($\ell = 1$).

Similarly to the previous constraint, Equation (5) checks if there is an edge value (line cost) associated with the path demand facilitator, but now it is checked on all paths and for all levels ($x_{\ell_{ij}}$), validating all sections ($m_{\ell_{ij}} = 1$). This constraint verifies whether there is a continuity of the tree between the demand node i and the facilitator node j. The constraint given by Equation (6) verifies if the quantity (cardinality) of facilitating nodes for level $\ell[n(N_\ell)]$ is sufficient to meet the number of clients of the same level $[n(M_\ell)]$.

Equation (7) evaluates whether the total amount of demands i met by a single facilitator node j at level ℓ is within the maximum port limit; this is defined by ms, through adding the values "1" of the $m_{\ell_{ij}}$ binary variable.

Equation (8) is a key parameter of PON technology that must be verified with each calculation of the OF, as an optimal solution in terms of cost may also be technically unfeasible due to excessive optical signal attenuation. In this case, At is the total attenuation between the aggregator node and the client and at_{ac} is the maximum value of accessory losses. This constraint checks the total attenuation between the customer and the concentrator node for all customers. The worst-case scenario (highest attenuation) is stored for further processing, such as penalizing the solution during the optimization process.

Equations (9) to (15) describe the main characteristics or premises of PON networks and are related to the objective function or its constraints. The first characteristic is the minimum quantity of concentrating nodes (qp), indicated in Equation (9) and which is formed by the ceiling, resulting from the division of the number of final customers by qcp. Where $qcp = 128$, given by Equation (10), is the maximum number of clients per OLT port, which is a technology definition.

The maximum number of levels (ml) given in Equation (11) is defined by dividing the available optical power (apb) by 3 dB. This variable evaluates the maximum 1:2 optical divisions (one input for two outputs), which implies a drop of 50% (or 3 dB) optical signal power for each splitter output, which can be carried out until the optical power runs out.

The total optical power, available to calculate the attenuation along the way, is indicated in the constant pb power budget given by Equation (12) and has its value defined by the technology. For accessory losses at_{ac}, given by Equation (13), which occur in connectors, splices and mergers, a market standard was adopted.

The available optical power (apb) is defined by the power provided by the OLT, which for the PON standard is 28 dB, minus the total at_{ac} accessory losses, as indicated in Equation (14). Finally, the maximum number of ports (output) of splitter was limited to 32 as indicated by Equation (15), because it is a standard value adopted by the market. Nonetheless, there is no impediment for this value to be changed, provided that the proportion 2^n and the technology limit is respected.

The PON planning described by Equations (1)–(15) features integer decision variables and topology constraints. Problems of such a nature are better handled by metaheuristic techniques than by classical mathematical approaches. One of the main drawbacks of population-based metaheuristics is that many of the individuals generated or created are unfeasible, and a lot of computational time is wasted verifying those solutions; therefore, in this paper, a special codification is used that only represents feasible individuals, and a set of specialized operators are proposed that manipulate these feasible individuals to create other feasible individuals. GAs are known for their proven flexibility and efficacy in dealing with complex engineering problems. A GA is used to showcase the effectiveness of the codification and operators; nonetheless, the codification and special operators developed in this paper can be used on any population-based metaheuristics and similar optimization algorithms suited to solve mixed integer non-linear problems.

3. Solution Approach

The optimization problem given by Equations (1)–(15) was solved using a specialized GA. These techniques have proven effective in solving complex optimization problems in engineering [21,22]. By simulating the process of natural evolution, GAs can efficiently explore a wide solution space and identify optimal or near-optimal configurations for network design.

3.1. Notation

The following notation was used in this paper:

- "Final clients" or "first level demands": these are clients of the service provider, which correspond (graphically) to the nodes at the end of the tree (or level 1).
- "Level ℓ demands": these correspond to the nodes that had the function of "aggregator" at the previous level ($\ell-1$) but are linked to other aggregators or to the concentrator node at level ℓ.
- "Aggregators", "facilities" or "facilitators" and "splitters": these correspond to nodes that receive two or more clients or demands, up to the limit defined by ms. Although the term "splitter" is characteristic and represents PON technology equipment, functionally it acts as a node that receives the termination of several logical connections, aggregating them into a single output.
- "Concentrators" or "Principals": these correspond to the last demand aggregator node, at the ml level, where the active equipment of the PON network (OLT) will be installed.
- "Passing nodes": their function is to maintain network connectivity. They do not have a splitter or other PON equipment installed.

Figure 2 illustrates the nomenclature of each element and its arrangement in a tree. In this case, the node numbers correspond to the quantity of clients associated with each node.

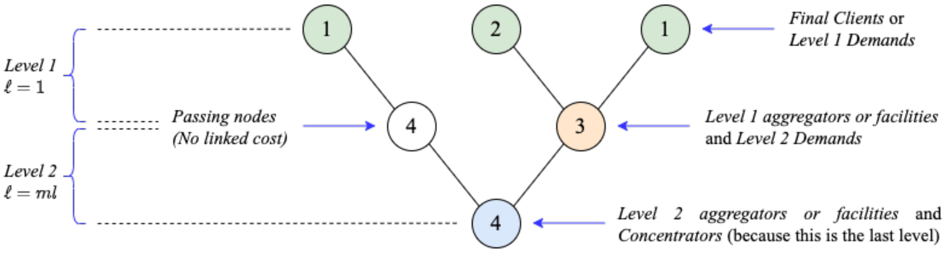

1
2
1
Final Clients or
Level 1 Demands
Level 1
ℓ = 1
Passing nodes
(No linked cost)
4
3
Level 1 aggregators or facilities
and Level 2 Demands
Level 2
ℓ = ml
4
Level 2 aggregators or facilities and
Concentrators (because this is the last level)

Figure 2. Notation used in this work.

3.2. Codification

A proper representation of solutions (encoding or codification) of a metaheuristic is key to its success, as it can either facilitate or complicate operations between individuals [23]. The encoding implemented in this paper is similar to that used in power system problems [24,25]. Conventional GAs typically use a single vector to encode the individuals in their population. In contrast, the algorithm proposed in this paper employs two encoding vectors that deal with different information, which is a significant departure from traditional approaches.

- Tree vector: Contains a sequence of identifiers for the branches (or lines) that make up the tree. It has a size equal to or less than the number of edges imported from the database. This vector represents the physical topology of the network.
- Aggregator vector: Composed of the vertices that will receive optical splitters and the concentrator (main) node. This vector has a size of $M_{\ell=1} x\, m\ell$; that is, the number of final clients (first level clients), multiplied by the maximum number of levels. This vector stores the individual's logical network topology.

The tree vector deals with the physical topology of the network (the physical connection between a pair of nodes). This would be equivalent to the routing of optical cables, the poles used, and the available routes. On the other hand, the aggregator vector maintains the logical topology of the network, which describes the interconnection between their functional elements. The difference between the physical topology of a network and its logical arrangement associated with the tree and aggregator vectors, respectively, is illustrated in Figure 3. Note that the logical topology of the network does not include the passing node.

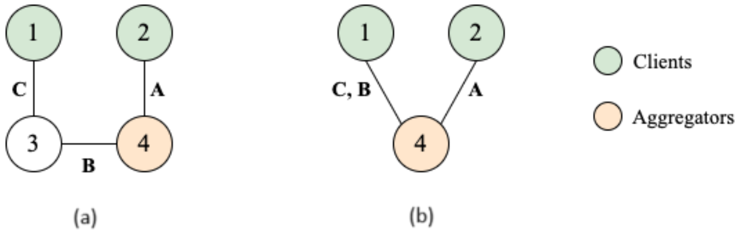


Figure 3. Difference between the codification vectors: (**a**) physical topology of the tree and (**b**) logical topology of the network (aggregators).

The treatment and codification given to the database during the simulations is illustrated in the network depicted in Figure 4.

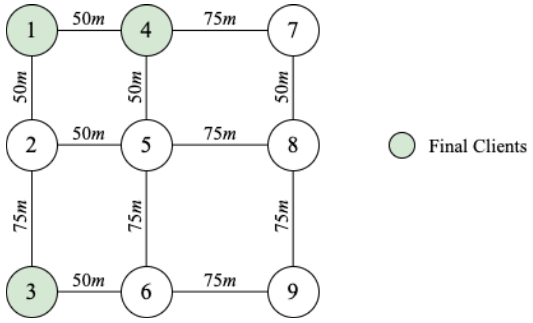


Figure 4. Network with distances (meters) and featured clients.

Nodes 1, 3, and 4 (highlighted in green) are the locations where the final clients, who wish to receive the telecommunications service, are located through the PON network. This generates two tables as follows.

(a) One table with the list of nodes, which contains the identification number ("*ID*") of the node, its geographical coordinates, and its "type" . The nodes of the final clients (users) have the field "type" with value "2" and the other nodes have the value "1" in the same field.

(b) Another table with a list of lines (edges), which presents the identification of the nodes "A" and "B" that make up the line and the distance, in meters, that exists between them.

The data structure described above is given in Tables 1 and 2. Furthermore, Table 2 includes an identifier ("*ID*") for each line between nodes A and B. In this case, the order of nodes (A or B) is not relevant to the execution of the algorithm.

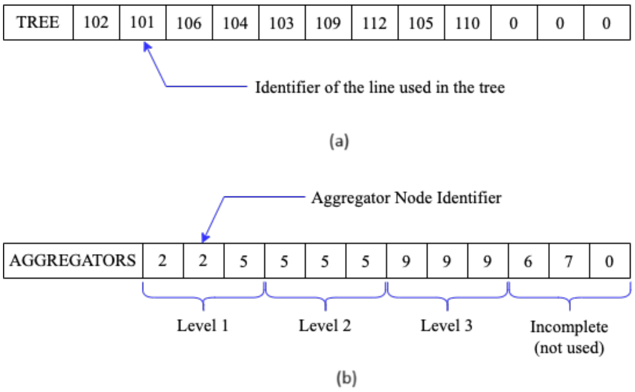
Table 1. Node table.

Node ID	Coord. X	Coord. Y	Node Type
1	0.0	125.0	2
2	0.0	75.0	1
3	0.0	0.0	2
4	50.0	125.0	2
5	50.0	75.0	1
6	50.0	0.0	1
7	125.0	125.0	1
8	125.0	75.0	1
9	125.0	0.0	1

Table 2. Table of lines with identifiers.

Line ID	Node A	Node B	A-B Distance (m)
101	1	2	50.00
102	1	4	50.00
103	2	3	75.00
104	2	5	50.00
105	3	6	50.00
106	4	5	50.00
107	4	7	75.00
108	5	6	75.00
109	5	8	75.00
110	6	9	75.00
111	7	8	50.00
112	8	9	75.00

For the calculation of the OF, both the tree vector and the aggregator vector are required. From the network illustrated in Figure 4 and the information of Table 2, the coding vectors are randomly generated, as shown in Figure 5.



TREE	102	101	106	104	103	109	112	105	110	0	0	0

Identifier of the line used in the tree
(a)
Aggregator Node Identifier

AGGREGATORS	2	2	5	5	5	5	9	9	9	6	7	0

Level 1
Level 2
Level 3
Incomplete (not used)
(b)

Figure 5. Codification vectors: tree (**a**) and aggregator vector (**b**).

Figure 6 represents the resulting network (nodes and lines) from the codification vectors in Figure 5.

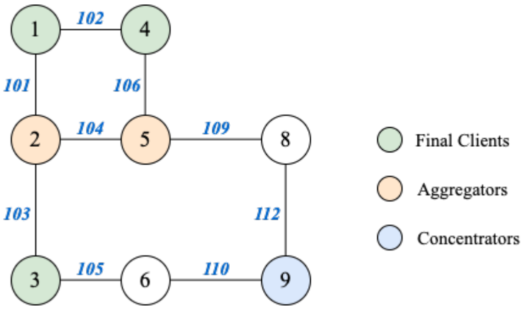


Figure 6. Connections generated from input vectors.

In Figure 6, the green nodes 1, 3, and 4, represent the “final clients” or “initial demands”; node 5 has the function of aggregator of client 4 and node 2, while node 2 has the function of aggregator for clients 1 and 3 at level 1, and becomes a demand of node 5 at level 2. Finally, node 9 is left with the assignment of concentrator node (or third level facility).

An important feature in this coding is that the number of nodes used in the OF calculations will always be a multiple of the number of clients. If there are more nodes in the vector, but an insufficient quantity to complete a level (less than the number of clients), these nodes will, at first, be moved to the end of the vector, to maintain the diversity of the population and will be removed after the reproduction performed by the G.A. By providing, in matrix form, the client nodes and the vector nodes of aggregators from Figure 5b, it is easier to visualize the relationship between clients and the aggregator vector, as seen in Figure 7a.

The research described in [20] was taken as a reference for the development of this work. The main difference lies in the "equivalent tree" adopted in this work and presented in Figure 7b. The proposed codification allows flexibility in the hierarchy of physical levels, illustrated by node 4, which can connect directly to the number 5 aggregator without having to go through an intermediate facility, as happened with clients 1 and 3.

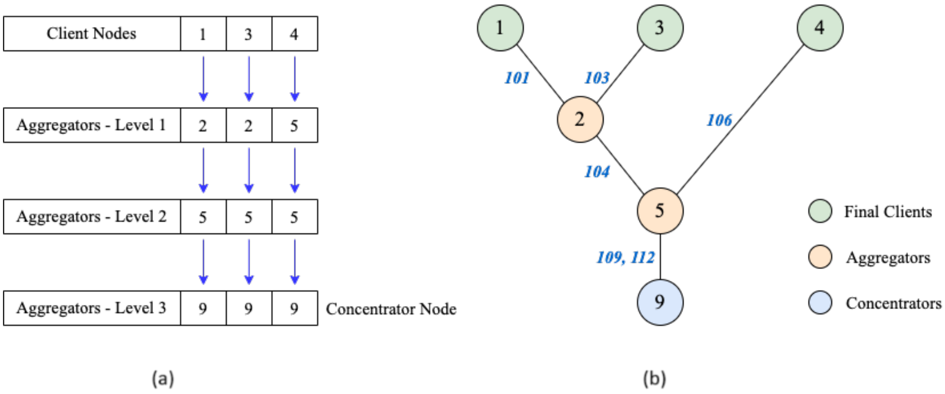


Figure 7. (**a**) Client and aggregator nodes in vertical grouping; (**b**) tree equivalent.

In this model, clients or aggregators can also be connected directly to the concentrator. This feature can only be used because there is no active equipment in the PON networks, which would segment the upper and lower levels, as is a standard in other access technologies for telecommunications services. Although the term "Level" is presented in the OF and throughout the paper, this expression represents only the group of aggregating nodes that meet a certain number of demands, which are being manipulated during the iteration, not characterizing a distinct physical layer in the network, but are instead logical. It should also be noted that, while the levels are processed, the nodes of the previous level become "demands" of the current level under treatment. This change in attribute that the node receives according to the processing level is highlighted in Figure 8.

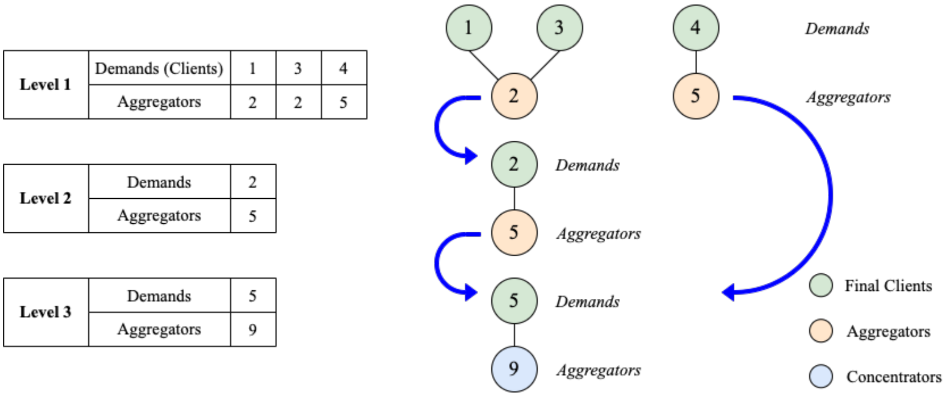


Figure 8. Processing of logical "levels".

3.3. Objective Function Calculation

The premises and variables specific to PON technology were adopted to compute the OF. The variables that are most relevant in the planning of a PON are as follows:

- The optical signal power (power budget);
- The signal attenuation along the path, considering all components: optical fiber, splitters, splices, and connectors;

- The maximum number of ports per splitter;
- The maximum number of users per port of the central OLT equipment.

The values adopted for these and other variables are indicated in Table 3, where the following can be observed:

- The number of concentrator nodes depends on the number of clients;
- The maximum number of clients per OLT port (concentrator) was defined for the highest value supported by the technology, i.e., optimizing the network deployment resources to the maximum, at the expense of greater bandwidth available for each user;
- Although the PON network standards accept optical dividers with 64 or 128 outputs, the maximum adopted, $ms = 32$, corresponds to a typical value which is available and more commercially used. The cost of these devices also varies according to the number of ports, which is defined by the percentage increase in the variable $appa$.
- Besides the cost of access and backbone optical fiber, a distinct value was assigned for unused fiber lines. This variable will be used to penalize a tree that has many unused lines, also focusing on cost reduction.

The variables presented in Table 3 are declared statically in the routine that processes and calculates the OF.

Table 3. Variables and typical values.

Typical Value	Description	Unit
128	Maximum number of clients per concentrator node	Unit
(N° of clients)/*qcp*	Minimum number of main nodes to meet the project	Unit
28	Optical signal power	dB
3	Maximum value of accessory losses (attenuations)	dB
32	Maximum number of ports/outputs per *splitter*	unit
0.4	Backbone fiber attenuation	dB/km
0.5	Drop cable fiber attenuation	dB/km
USD 200	Basic cost per aggregator node (splitter)	USD/unit
10%	Additional cost per port of the aggregator node	USD/port
USD 5000	Basic cost per concentrator node	USD/unit
25	Cost of optical access cable	USD/meter
20	Cost of optical network cable	USD/meter
25	Cost of unused optical cable in the network	USD/meter
pb-atac	Available optical power	dB
apb(dB)/3(dB)	Maximum number of levels	Unit

To illustrate the operation of the algorithm proposed in this work, the input vectors presented in Figure 5 will be used.

3.3.1. Constraint Verifications

This routine validates the quality of the tree and the aggregator nodes that were sent for processing. Since the trees are randomly generated, this test is necessary to discard an unfeasible individual before entering the main loop. At this stage, the following is checked:

(a) Whether the tree vector has at least one line for each client (quantity only).
(b) The total number of levels to be processed, based on the number of aggregator groups, ensuring that the maximum value (ml) is respected.
(c) The aggregators of level ℓ (last level) and potential candidates for concentrator nodes.
(d) If there are enough hub nodes to serve all clients.
(e) Whether the lines connecting the customers are in the tree.

3.3.2. Main Loop for OF Calculation

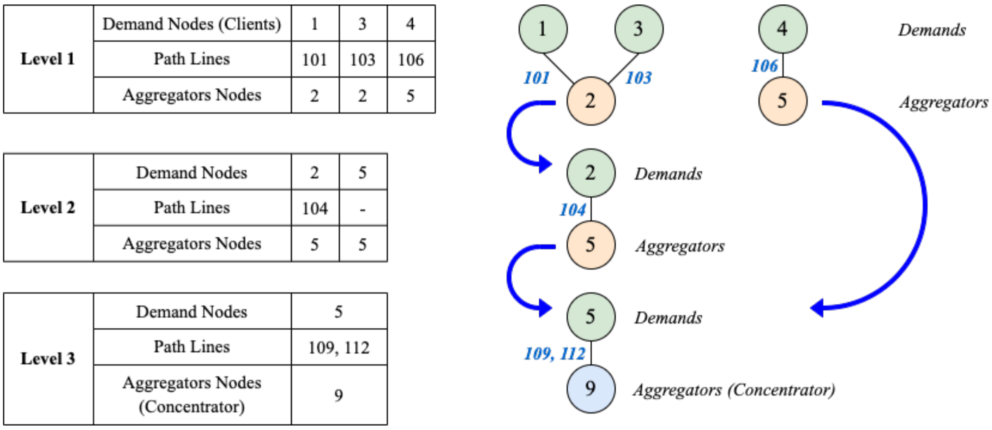
This loop is executed according to the number of possible levels (L), starting with $\ell = 1$ and ending with $\ell = L$, as long as there are no objections to the continuation of the execution.

The processing of level ℓ is interrupted in the following cases:

(a) One or more aggregator nodes of the level are not in the tree, or there are not enough aggregators for the clients in $\ell = 1$ (first level);
(b) There is no path between a customer node and an aggregator node;
(c) The number of clients connected to an aggregator node exceeds the limit defined by ms (32 clients).

In cases where the total attenuation of the path is greater than the limit defined by apb (25 dB), the routine that calculates the OF does not interrupt the processing but inserts an error code into a return variable, which can be used by the GA. This approach aims to provide greater diversity in the population when producing offspring. Before computing distances between client and node aggregators an analysis of repeated pairs and continuity is performed, as shown in Figure 9, so that the cost of the path is computed only once.

The distance is calculated through the path traced by a custom function created for this task, labeled as firstneighbor. This name was given because the process of searching for the path lines that connect node A to node B is always conducted by taking the first adjacent line to node A, and so on, moving back to the previous node and taking the second adjacent line if there is no continuity. After completing the survey of all paths, the number of clients per aggregator is verified. The quantity of ports/outputs of the aggregator is standardized to 2^n, and the price of the aggregator and the attenuation introduced by this equipment in the network are calculated. If this aggregator is at the last level, the value of the concentrator equipment is added to the node's cost.



Level 1	Demand Nodes (Clients)	1	3	4
	Path Lines	101	103	106
	Aggregators Nodes	2	2	5

Level 2	Demand Nodes	2	5
	Path Lines	104	-
	Aggregators Nodes	5	5

Level 3	Demand Nodes	5
	Path Lines	109, 112
	Aggregators Nodes (Concentrator)	9

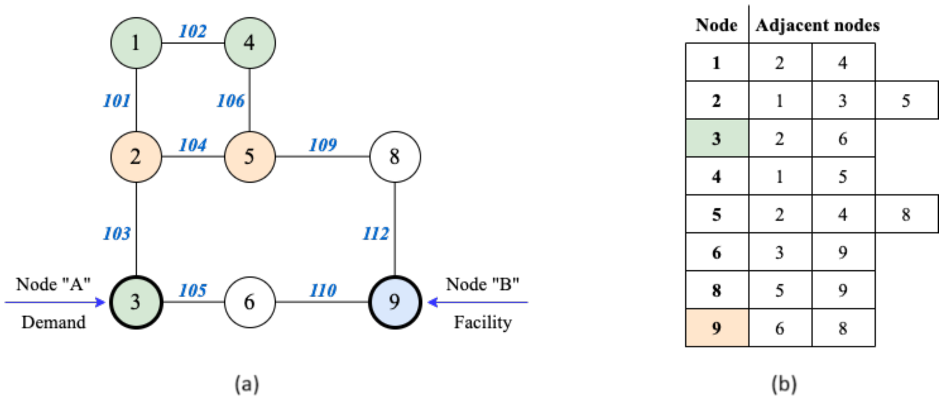
Figure 9. Connectivity and validated paths.

3.3.3. Auxiliary Function Firstneighbor

This subroutine was developed to verify the continuity of the tree and determine the distance between two nodes. Its processing has a simple decision criterion that analyzes only one variable at a time, making the iteration process faster. The inputs of the function are as follows: Node "A" (demand) or origin, Node "B" (facility) or destination, and optionally, forbidden nodes (nodes that should not be used during the path construction). This auxiliary function also has access to some global variables necessary for the routine, which are as follows: tree and aggregator vectors that are sent to the objective function (main function), the adjacency matrix with the identifier numbers of the lines, and the list of nodes that are the final clients. The processing occurs as follows:

A path vector is created with node "A" in the first position. Starting from node "A", the algorithm checks if node "B" is in its adjacency list. If it is not, the first node in that list node, "α", for example, is selected, added to the path vector, and removed from the adjacency list of node "A". The process then repeats for node "α" and so on until node "B" is reached. If there is no connectivity or if at any point during the path construction the first adjacent node is a client node or is already in the path vector, then the auxiliary function firstneighbor eliminates the last node from the path and reverts to the second to last valid node, selecting the first adjacent node again. If there are no more nodes in the adjacency list of a given node, the subroutine returns an error code and an empty path vector.

To illustrate the processing of this auxiliary function, node 3 will be taken as demand (node "A") and node 9 as facility (node "B"), highlighted in Figure 10. The firstneighbor routine will be executed to verify the path between the two nodes.



Node	Adjacent nodes		
1	2	4	
2	1	3	5
3	2	6	
4	1	5	
5	2	4	8
6	3	9	
8	5	9	
9	6	8	

Figure 10. (**a**) Path to be found; (**b**) list of node adjacencies.

The step-by-step process is illustrated in Figure 11, where nodes 1, 3, and 4 are other client nodes (nodes to be avoided).

In this example, the path between node 3 and node 9, obtained by the routine, was as follows: 3, 2, 5, 8, 9. It can also be observed that in the processing steps (c), (d), and (g), other client nodes (to be avoided) were not included in the path, and in step (f), a node that was already in the path was removed. Finally, in step (i), the first adjacent node (node 5, in this case) was not selected because node 9, which is the demand being sought (node "B"), was already in the adjacency list of node 8, thus ending the auxiliary function search.

3.3.4. Results and Outputs Calculation

Once the previous step is completed, the *tree* vector is sorted, placing the used branches at the beginning and the others at the end of the list. Then, the cost of the unused lines is accounted for based on the value defined by the variable cfm. This amount is added to the final value of the objective function. The objective function routine returns the values indicated in Table 4.

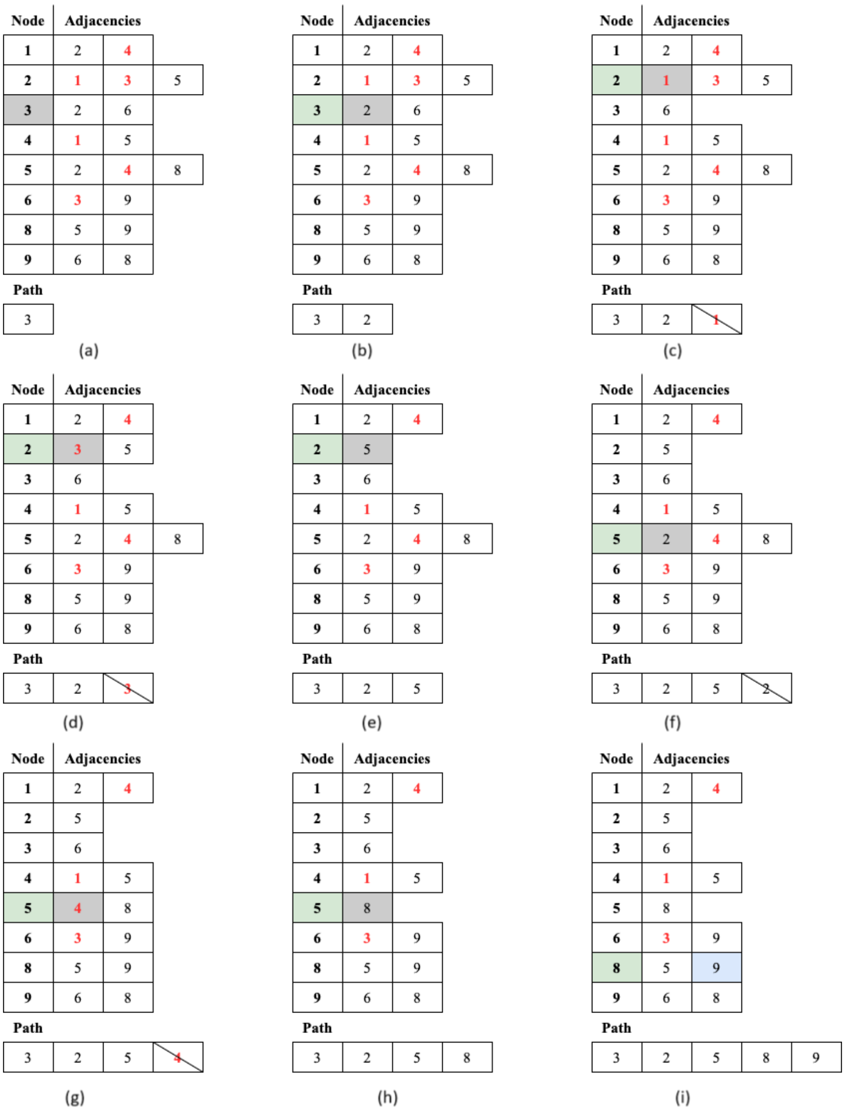


Figure 11. Step-by-step processing of the auxiliary function firstneighbor.

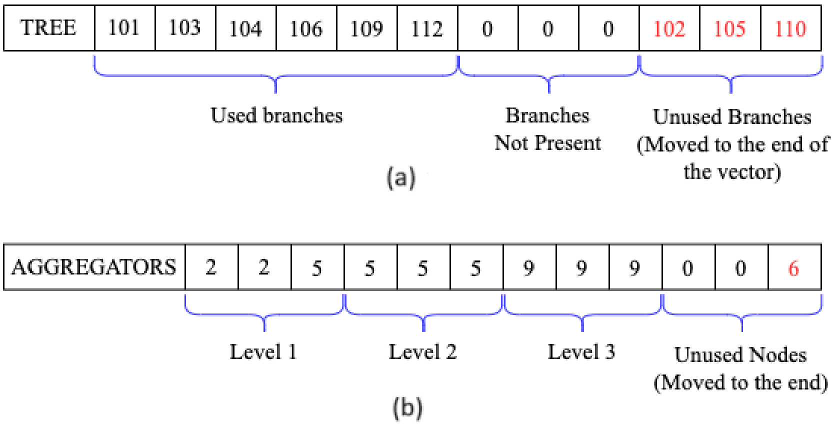
The reorganized tree and aggregator vectors, which return with the objective function, are shown in Figure 12. The resulting network, considering only the lines that were used for constructing the paths during the continuity and connectivity verification process, is represented in Figure 13.

Table 4. Return variables of the objective function calculation.

Variable	Information Transported	Value
fvalor	Value of the objective function, in monetary units. Includes the costs of the fiber-optic path, equipment, and unused branches.	USD 18,230.00
fatt	Value of the highest attenuation found for the path between an end customer and a concentrator node.	6.740 dB
ferro	Vector with error codes, if any.	[∅]

Table 4. *Cont.*

princ	Vector with the concentrator nodes.	[9]
fcontag	Auxiliary matrix, with information on node function (1 = concentrator, 2 = customer, 3 = aggregator, 0 = no specific function) and ports/outputs per aggregator. This information is grouped by concentrator node.	Table 5
fsortre	Organized tree vector, with separation of used and unused lines (if any).	Figure 12a
fsortag	Ordered aggregator vector without nodes that do not form a level.	Figure 12b



TREE	101	103	104	106	109	112	0	0	0	102	105	110

Used branches
Branches Not Present
Unused Branches (Moved to the end of the vector)
(a)

AGGREGATORS	2	2	5	5	5	5	9	9	9	0	0	6

Level 1
Level 2
Level 3
Unused Nodes (Moved to the end)
(b)

Figure 12. Processed codification vectors: (**a**) tree and (**b**) aggregators.

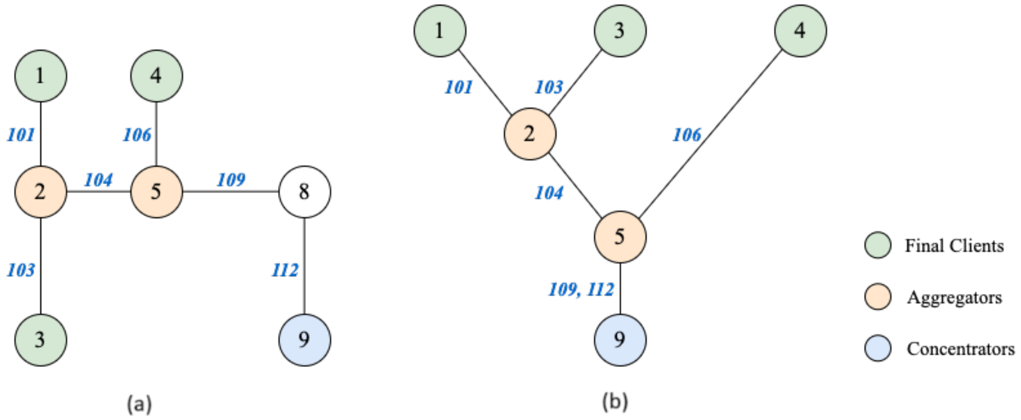


Figure 13. (**a**) Network diagram and (**b**) tree diagram after OF processing.

The *fcontag* table, which is returned by the function, accumulates and stores the attribute of the node type and the number of branches, in the case of aggregators and concentrators. It is part of the function's outputs to validate and monitor the operation of all subprograms involved in the processing. The nodes can be classified into four types, as shown in Table 5.

- 1: Concentrator or main;
- 2: Final clients (level 1);
- 3: Aggregators, regardless of level;
- 0: Transit node or not used in this simulation.

Table 5. Detailed *fcontag* node count table.

Node	1	2	3	4	5	6	7	8	9
Function/Type	2	3	2	2	3	0	0	0	1
Derivations	0	2	0	0	2	0	0	0	1

The structure of Table 5 is replicated proportionally to the number of main nodes; that is, if there were two main nodes, the same table would have five rows: one with the node index and two sets of "Type" and "Derivations" count rows, one set for each main node.

Table 6 shows the length of the lines that are in the tree vector indicated in Figure 12a, while Table 7 presents the values of the OF, item by item.

Table 6. Length of the lines used in the OF calculation.

Line ID	Node A	Node B	A-B Distance (m)
101	1	2	50.00
103	2	3	75.00
104	2	5	50.00
106	4	5	50.00
109	5	8	75.00
112	8	9	75.00
102	1	4	50.00
105	3	6	50.00
110	6	9	75.00

Table 7. Length of the lines used in the OF calculation.

Item	Calculation Reference	Unit	Calculated Value	Unit Cost	Total Cost
Fibers for Access (last-mile)	Lines: 101, 103, 106	Meter	50 + 75 + 50 = 175 m	25	4375
Level 1 Aggregator	Node 2—With two ports	Unit	200 + (20 × 2) = USD 240	240	240
Level 2 Aggregator	Node 5—With two ports	Unit	200 + (20 × 2) = USD 240	240	240
Backbone Fibers	Lines: 104, 109, and 112	Meter	50 + 75 + 75 = 200 m	20	4000
Concentrator Node	Node 9	Unit	USD 5000	5000	5000
Unused Lines	Lines: 102, 105, and 110	meter	50 + 50 + 75 = 175 m	25	4375
				Value of the OF:	18.230

The path attenuation is measured from the client node to its respective concentrator node for all clients. The function returns the worst-case scenario found, which in this example is the attenuation of the following segment: client from node 3 to the concentrator at node 9, as highlighted in Figure 14.

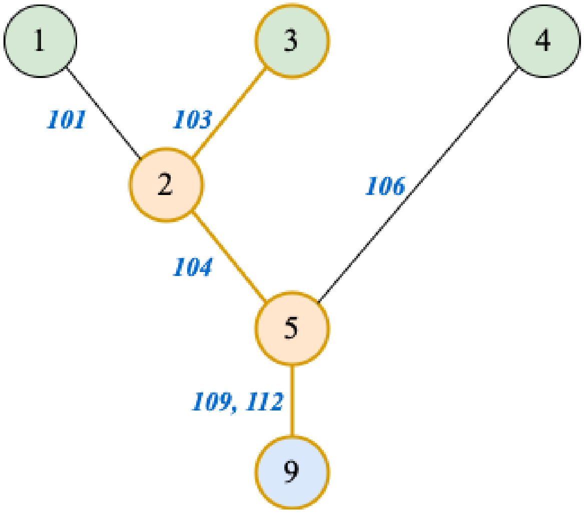


Figure 14. Highlight for the path with the highest calculated attenuation.

Table 8 describes all the items that are computed to calculate the attenuation on the complete path between nodes 3 and 9, highlighted in Figure 14; it also presents a summary of the attenuation for the path between the other clients and their concentrator node.

Table 8. Detailed attenuation calculation.

Path: Client ↔ Concentrator	Item	Attenuation Components	Calculated Attenuation
Node 3 ↔ Node 9	Line 103	0.5 dB/km × 0.075 km=	0.038 dB
	Node 2—With two ports	10 × log(1/2) + 10% =	3.311 dB
	Node 5—With two ports	10 × log(1/2) + 10% =	3.311 dB
	Lines: 104, 109, and 112	0.4 dB/km × 0.2 km =	0.080 dB
	Total Path Attenuation: 3–9		6.740 dB
Node 1 ↔ Node 9	Total Path Attenuation: 1–9		6.728 dB
Node 4 ↔ Node 9	Total Path Attenuation: 4–9		3.410 dB

3.4. Implemented Genetic Algorithm

A Chu–Beasley GA was developed for optimizing PON network designs. Since the focus of this optimization is on reducing total cost, the primary criterion for evaluating the solution was the value returned by the OF calculation. The control variables are declared at the beginning of the algorithm’s execution, such as the mutation rate, initial population size, and criteria for crossover and recombination. The complete list of these parameters is presented in Table 9.

Another step that proved necessary to reduce the time in the search for the optimal solution was the prior creation of a population at least ten times larger than the initial population, called the “general population”. The larger the “general population”, the greater the diversity in the initial population, which favors the convergence of the GA. The use of this cycle was necessary for two reasons: the first reason was because generating tree and aggregator vectors randomly, without applying any specific heuristic, is very inefficient, and produces few valid individuals. The other reason is that this approach allows for the implementation of elitism, which is the practice of preserving the best individuals (solutions) from one generation to the next, ensuring that the top-performing solutions are not lost during the selection and crossover processes.

Before the start of the main loop execution, a group of individuals is selected randomly, but is distinct from the elite population, in a number equal to *npop-npel*. This group is combined with the elite population, giving rise to the "initial population", which will be used during the simulation. The information of the best individual in the initial population, as well as the average values of the objective functions in this set, is stored to verify the quality of the offspring and to compare the evolution of generations.

Table 9. Control variables of the GA.

Variable	Definition
niter	Maximum number of iterations
npop	Total initial population size
npel	Size of the elite population
tm	Parameter for mutation. Mutation will occur if a randomly generated value is equal to or less than *tm*
pm	Three-cell vector storing the mutation incidence proportion as follows (in order): mutation in the tree vector, mutation in the aggregators vector, or insertion of a random individual into the initial population
pena	Penalty multiplier for the OF if attenuation exceeds the available power limit
reco	Controls how crossover between parents will be performed for offspring generation: 0 = mix tree vector; 1 = mix aggregators vector; 2 = choose one of the three types according to a proportion defined by *pc*; 3 = direct (or simple) generation: use the tree vector from one parent and the aggregators vector from the other
pc	Three-cell vector storing the incidence proportion of each crossover type in the following order: tree, aggregators, simple generation. Used only when the *reco* variable is set to 2
propfoat	Selection proportion for elite population composition: the closer to 1, the more weight the OF has in elite selection; the closer to 0, the more weight attenuation (lower) has in elite selection

3.4.1. Selection of the Elite Population

From the "general population," the best individuals are selected up to the quantity defined by *npel*. The selection of the elite population also takes into account another control variable, *propfoat*, which allows the adjustment of the proportion of individuals in this population based on OF values or attenuation values. The closer *propfoat* is to 1, the more weight the OF will have in ranking individuals; the closer it is to 0, the more individuals with lower attenuation will be in the elite group.

This operator was implemented because attenuation is the second most important decision variable in this type of project. Once the elite group is formed, the individuals are inserted into the initial population in random positions. It is also checked whether any individual is repeated throughout the initial population. In such cases, the repeated individual is replaced until the initial population reaches the size defined by *npop* with the following composition: $npop = npel + (npop - npel)$, without repetition.

3.4.2. Main Loop of the GA

The number of iterations of the GA within the main loop is controlled by the variable *niter*, which serves as the stopping criterion in this work. Within this loop, the following routines are executed.

Parent Selection: This is carried out through the roulette wheel method, where the individual with the highest fitness (lowest OF value) has a higher weight, while the individual with the lowest fitness has a lower weight. The roulette selection is performed until two distinct parents are obtained.

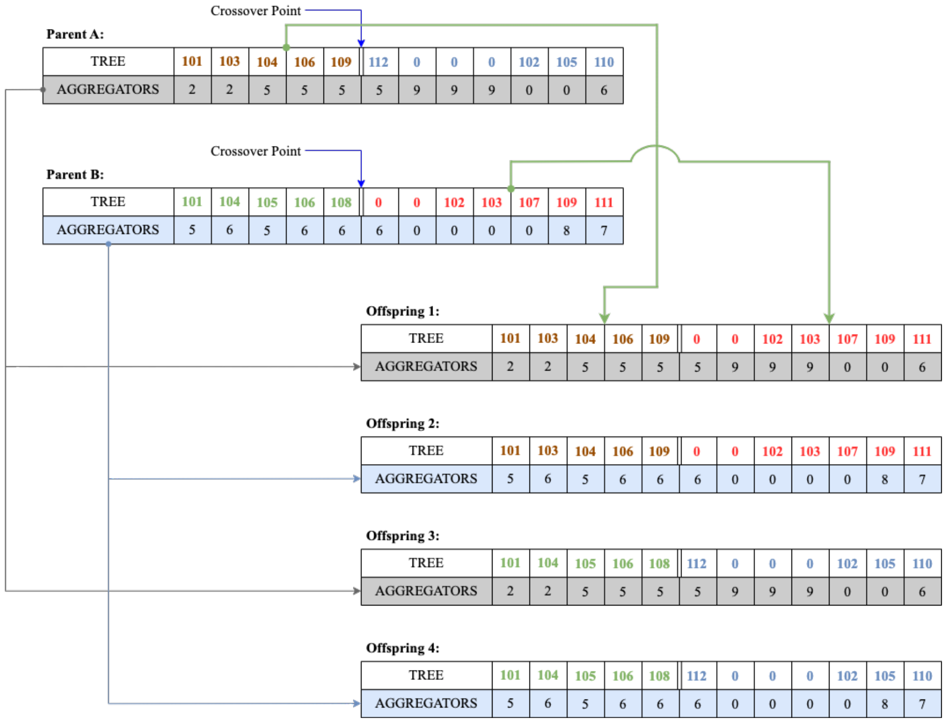
Crossover or generation of new solutions: The crossover operator is one of the contributions of this paper, as it was specifically designed and adapted for PONs. In this case, depending on the value defined by the variable *reco*, the new offspring can be generated through crossover in the tree vector, crossover in the aggregator vector, or via direct crossover or generation. The possible values for this variable are as follows:

- *reco* = 0: Performs the crossover in the tree vector, combining part of the tree vector from "Parent A" with the complementary part of the tree vector from "Parent B". The inverse process is also performed. The crossover point is chosen randomly between the second and the penultimate position of the tree. The aggregator node vectors are not altered, and this operation produces four offspring, with the "genetic load" presented in Table 10.

Table 10. Structure of offspring generated by tree crossover *reco* = 0.

Offspring 1	Combined tree: initial segment from Parent A + final segment from Parent B Aggregator from Parent A
Offspring 2	Combined tree: initial segment from Parent A + final segment from Parent B Aggregator from Parent B
Offspring 3	Combined tree: initial segment from Parent B + final segment from Parent A Aggregator from Parent A
Offspring 4	Combined tree: initial segment from Parent B + final segment from Parent A Aggregator from Parent B

The crossover of the tree vector can be visualized in Figure 15, where the green lines indicate the segments of information from the parents that are combined.



Parent A:

TREE	101	103	104	106	109	112	0	0	0	102	105	110
AGGREGATORS	2	2	5	5	5	5	9	9	9	0	0	6

Parent B:

TREE	101	104	105	106	108	0	0	102	103	107	109	111
AGGREGATORS	5	6	5	6	6	6	0	0	0	0	8	7

Offspring 1:

TREE	101	103	104	106	109	0	0	102	103	107	109	111
AGGREGATORS	2	2	5	5	5	5	9	9	9	0	0	6

Offspring 2:

TREE	101	103	104	106	109	0	0	102	103	107	109	111
AGGREGATORS	5	6	5	6	6	6	0	0	0	0	8	7

Offspring 3:

TREE	101	104	105	106	108	112	0	0	0	102	105	110
AGGREGATORS	2	2	5	5	5	5	9	9	9	0	0	6

Offspring 4:

TREE	101	104	105	106	108	112	0	0	0	102	105	110
AGGREGATORS	5	6	5	6	6	6	0	0	0	0	8	7

Figure 15. Example of tree crossover.

- *reco* = 1: Performs crossover on the aggregator nodes similar to the previous method, where part of the aggregator vector from "Parent A" is replaced by the complementary part of the aggregator vector from "Parent B" and vice versa. The crossover point is also chosen randomly between the second and the penultimate position. In this case, the tree vectors remain intact, and the generated offspring have the configuration indicated in Table 11.

Table 11. Structure of offspring generated by tree crossover with *reco* = 1.

Offspring 1	Tree from Parent A Aggregators combined: start of Parent A + end of Parent B
Offspring 2	Tree from Parent B Aggregators combined: start of Parent A + end of Parent B
Offspring 3	Tree from Parent A Aggregators combined: start of Parent B + end of Parent A
Offspring 4	Tree from Parent B Aggregators combined: start of Parent B + end of Parent A

The crossover of the aggregator vector is illustrated in Figure 16, where the green lines indicate the gene blocks from the parents that are combined, while the tree vector remains unchanged.

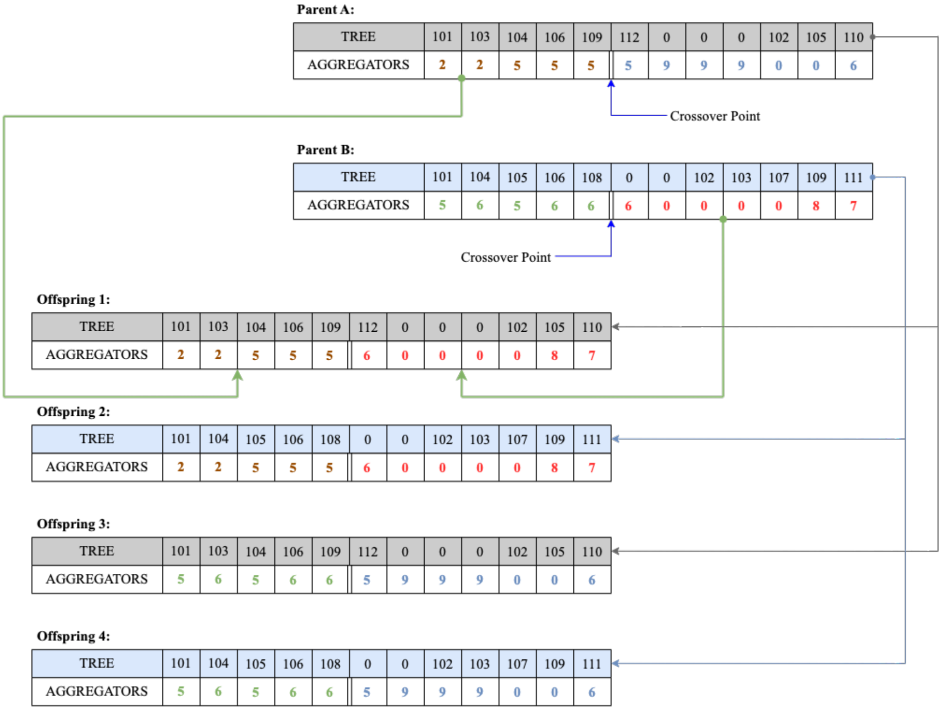

Parent A:
TREE 101 103 104 106 109 112 0 0 0 102 105 110
AGGREGATORS 2 2 5 5 5 5 9 9 9 0 0 6
Crossover Point
Parent B:
TREE 101 104 105 106 108 0 0 102 103 107 109 111
AGGREGATORS 5 6 5 6 6 6 0 0 0 0 8 7
Crossover Point
Offspring 1:
TREE 101 103 104 106 109 112 0 0 0 102 105 110
AGGREGATORS 2 2 5 5 5 6 0 0 0 0 8 7
Offspring 2:
TREE 101 104 105 106 108 0 0 102 103 107 109 111
AGGREGATORS 2 2 5 5 5 6 0 0 0 0 8 7
Offspring 3:
TREE 101 103 104 106 109 112 0 0 0 102 105 110
AGGREGATORS 5 6 5 6 6 5 9 9 9 0 0 6
Offspring 4:
TREE 101 104 105 106 108 0 0 102 103 107 109 111
AGGREGATORS 5 6 5 6 6 5 9 9 9 0 0 6

Figure 16. Example of aggregator crossover.

- *reco* = 2: This value acts as an auxiliary control variable, as it allows more than one form of parent combination to be executed during processing, following the incidence proportion defined by the vector *pc*, which has the following format: $p = [Tree\_crossover\_weight, Aggregator\_crossover\_weight, Direct\_generation\_weight]$, where the sum of the weights equals 1. This functionality is useful during the GA tuning process and can also be used for dynamic adjustments. For example, if the average OF values stagnate or show very little variation after a certain number of iterations, this vector can be modified, also changing the incidence of how the offspring are generated.

An example of crossover weight vector is as follows: $pc = [0.50, 0.85, 1]$. This example shows that crossover based on the tree vector ($reco = 0$) may occur in up to 50% of iterations, while crossover in the aggregator vector ($reco = 1$) may occur in up to 35% (85% − 50% = 35%) of iterations, and simple generation or crossover ($reco = 3$) in up to 15% of iterations (100% − 85% = 15%).

- *reco* = 3: Executes "simple generation" or "direct generation". This is a unique genetic operator in this work, which consists of generating offspring by combining the tree from Parent A with the aggregators from Parent B, and vice versa, without altering the sequence of the vectors and code. This option generates two offspring with the vector combinations shown in Table 12.

Table 12. Structure of the offspring combined for direct generation.

Offspring 1	Tree from Parent A Aggregators from Parent B
Offspring 2	Tree from Parent B Aggregators from Parent A

- For any other value of *reco*, no operation is executed, and no offspring are generated. The process moves to the mutation stage.
  Various types of parent crossover were used: crossover of the tree chromosome, crossover of the aggregator chromosome, and direct generation, with the aim of testing which genetic operator has the greatest influence on convergence to the best solution.

Generation evaluation: After producing the offspring, the OF is calculated to assess the validity and quality of that generation. Offspring that exceed the optimal power limit are penalized according to the multiplier defined in the variable *pena*. On the other hand, if no errors are found, the branches of the tree vector that were not used in the offspring during the connectivity check are removed, and the cost of these lines is deducted from their OF values.

If both offspring are valid, the one with the lowest OF value is selected first and compared with the worst individual in the population, leading to two possible situations:

1. The "best offspring" yields a worse result than the existing "worst individual". In this case, the other offspring are not evaluated, and the process moves on to the next iteration.
2. The "best offspring" yields a better result than the existing "worst individual", so it is checked whether the offspring already exists in the population:
   (a) If it does not exist, the offspring is included in the population in the position of the "worst individual".
   (b) If the offspring already exists in the population, the following ratio is calculated: (number of tree lines + number of aggregator nodes) divided by the number of clients. Preference is given to the individual that presents the lowest value in this ratio, meaning the one with fewer lines and aggregator nodes for the same cost and attenuation. Although this option reduces population diversity, it is coherent from an operational and maintenance perspective, as it results in a more "streamlined" network, less "spread out", and one that serves more clients in a concentrated (optimized) manner.

The above process (1 and 2) is repeated with the next valid offspring until all have been analyzed.

Mutation: In addition to crossover or generation, a mutation may also occur in all iterations if a randomly generated value between 0 and 1 is less than or equal to the mutation rate (tm). This operation is performed only once but can occur in three different ways: in the tree vector, in the aggregator node vector, or in the population, according to the incidence proportions defined by the control variable *pm*, which has the following structure:

$$pm = [Tree\_mutation\_weight, Aggregator\_mutation\_weight, New\_individual\_weight] \quad (16)$$

After the injection of the new "gene" into the encoding vectors or the creation of the new element, the OF is calculated. If valid, the individual that underwent the mutation is reinserted into the population with its modification and the new OF value. If the mutated individual is invalid, it is discarded, and no changes are made to the population.

Best solution verification: The first criterion for evaluating the fitness of a solution is the total network cost. However, it is possible for an offspring to present a total network cost equal to the OF of the (*incumbent*). When this occurs, the following mathematical relation is checked:

$$R = \frac{Number of non - zero tree branches + Number of non - zero aggregator nodes}{Number of clients} \quad (17)$$

Preference is given to the individual with the lowest value in this relation, assuming (in the case of the offspring) that the position of the best solution is found. After all genetic operations are performed, the population is scanned to evaluate if a better solution than the current (incumbent) exists. If such a solution exists, it takes the position of the new (incumbent). Following these assessments, the process moves to the next iteration.

3.4.3. Local Improvement Operators

Some local improvement operators were implemented to accelerate the algorithm's convergence. Most of them have already been described and will be highlighted in this section.

- The control variable *reco*, which allows for adjusting how the offspring will be generated. As suggested, this parameter can be used for dynamic adjustments when a control criterion is triggered (e.g., stabilization of the average OF values after "n" iterations).
- The removal of unused branches in the tree vectors of the generated offspring.
- The calculation of the ratio (number of tree lines plus number of aggregator nodes divided by the number of clients) as a decision criterion for selecting individuals that present the same OF value.
- The control variable *propfoat*, allowing another parameter besides the OF to be used in the selection and creation of the Elite Population.
- The inclusion of the elite population mixed into the initial population, which favored the selection of these elements in the roulette.

3.4.4. Block Diagram of the Implemented Genetic Algorithm

The processing flowchart of the GA described in the previous sections, is illustrated in Figure 17.

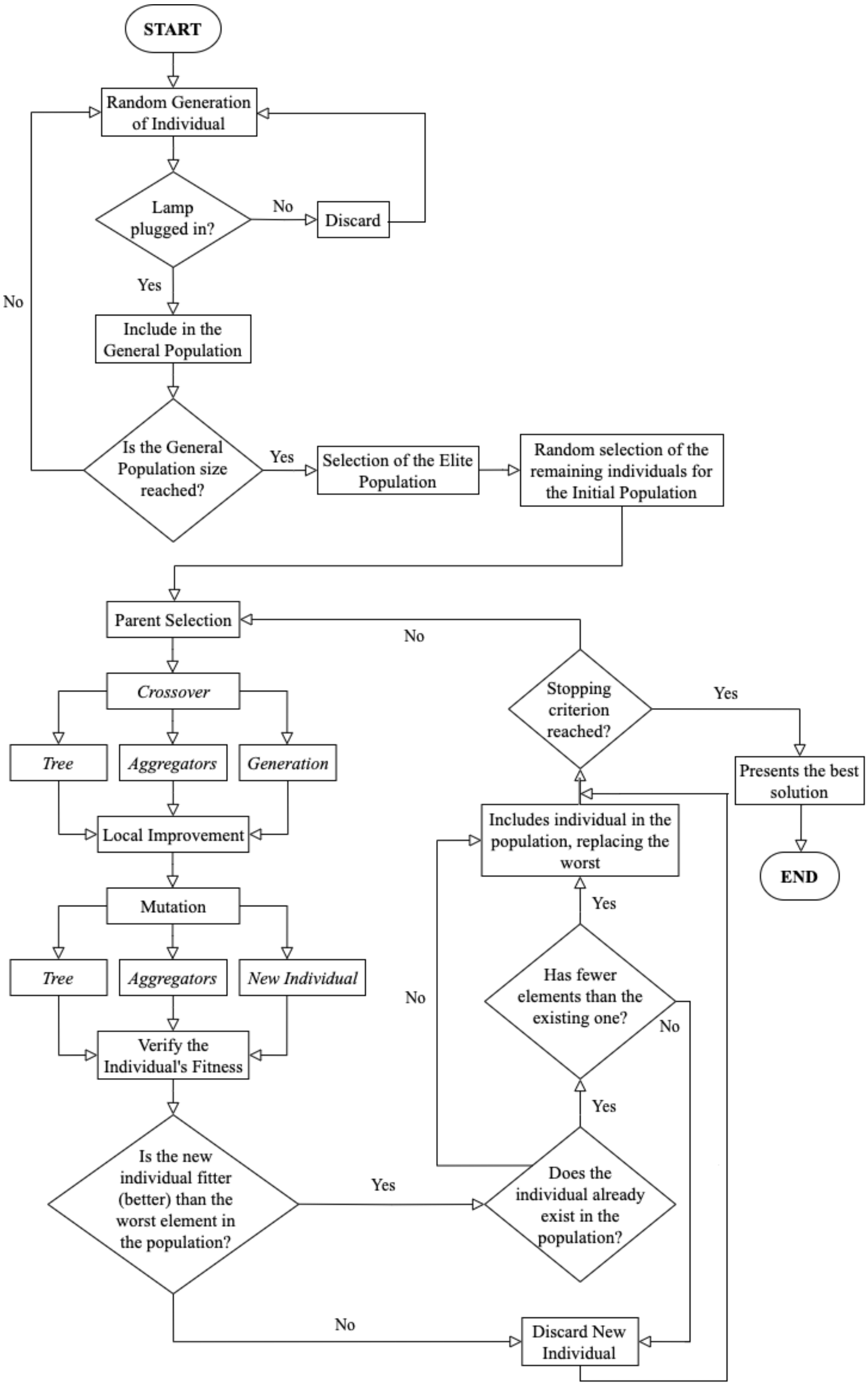

START
Random Generation of Individual
Lamp plugged in?
No
Discard
Yes
Include in the General Population
No
Is the General Population size reached?
Yes
Selection of the Elite Population
Random selection of the remaining individuals for the Initial Population
Parent Selection
No
Stopping criterion reached?
Yes
Presents the best solution
END
Crossover
Tree
Aggregators
Generation
Local Improvement
Includes individual in the population, replacing the worst
Yes
Mutation
Has fewer elements than the existing one?
No
No
Tree
Aggregators
New Individual
Verify the Individual's Fitness
Yes
Is the new individual fitter (better) than the worst element in the population?
Yes
Does the individual already exist in the population?
No
Discard New Individual

Figure 17. Flowchart of the implemented GA.

4. Tests and Results

4.1. Test System

The database used in the simulations is available in [20] and contains georeferenced information from a region of the city of Curitiba, Paraná, Brazil. The test system is composed of 51 nodes, 55 edges, and 12 clients. Figure 18 represents the geographical arrangement of the nodes and lines of this system.

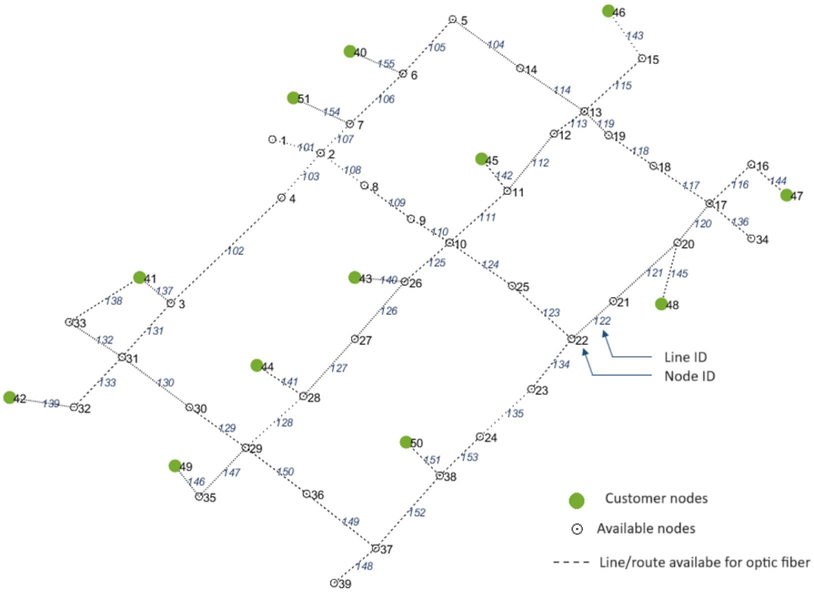


Figure 18. Illustration of the network test system.

4.2. Results

Several simulations were previously conducted to determine the best parameters of the proposed GA. The parameters used are illustrated in Table 13. Table 14 contains the results obtained for the test system and the network generated from the iterations of the GA is illustrated in Figure 19.

Table 13. Parameters of the GA.

Parameter	Value
Initial Population Size	200
Elite Population Size (10% of Initial Population)	20
Number of Iterations	2500
Mutation Probability	≤20%
Weight for Mutation in Tree Vector	33%
Weight for Mutation in Aggregator Vector	33%
Weight for Insertion of New Individual	34%
Type of Crossover or Generation	Proportional
Weight for Crossover in Tree Vector	33%
Weight for Crossover in Aggregator Vector	33%
Weight for Simple Generation of Individual	34%

Table 14. Results obtained for the test system.

Value of the OF	USD 50,490.38
Incumbent Iteration	1071
Highest Attenuation (dB)	21.212
Number of Lines in Tree	42
Number of Aggregators	72
Maximum of the Mean of Population OF	USD 494,305.24
Minimum of the Mean of Population OF	USD 52,117.54
OF Reduction Proportion	9.48

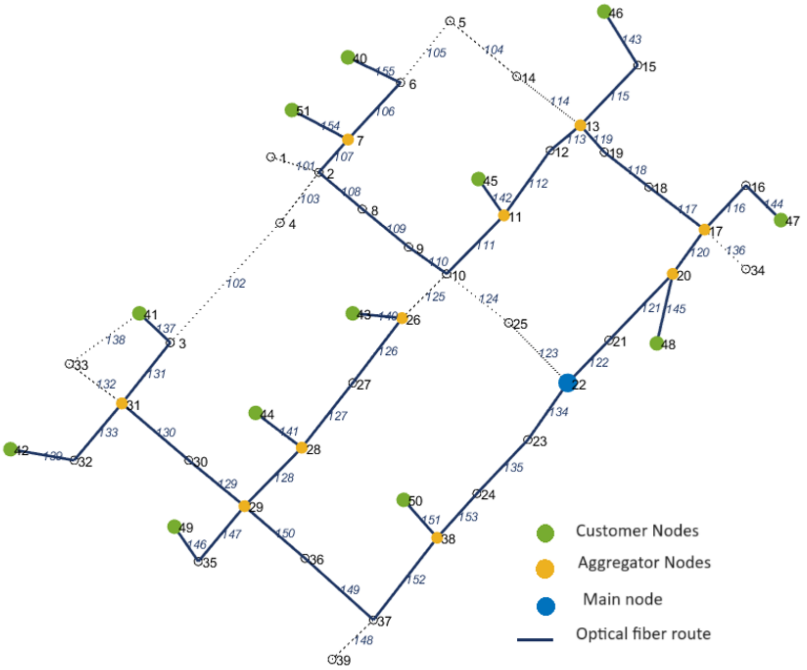


Figure 19. Best topology found by the GA.

4.3. Comparison of Results

To compare the results obtained in this work, two other solutions for the same test system were used: a manually designed PON network and the previous solution obtained by [20], which implements a Multiple Attribute Decision-Making (MADM) algorithm.

The manually designed PON network for the test system resulted in the physical topology shown in Figure 20. This solution has a total cost of USD 51,155.46 and a maximum attenuation value of 21.21 dB.

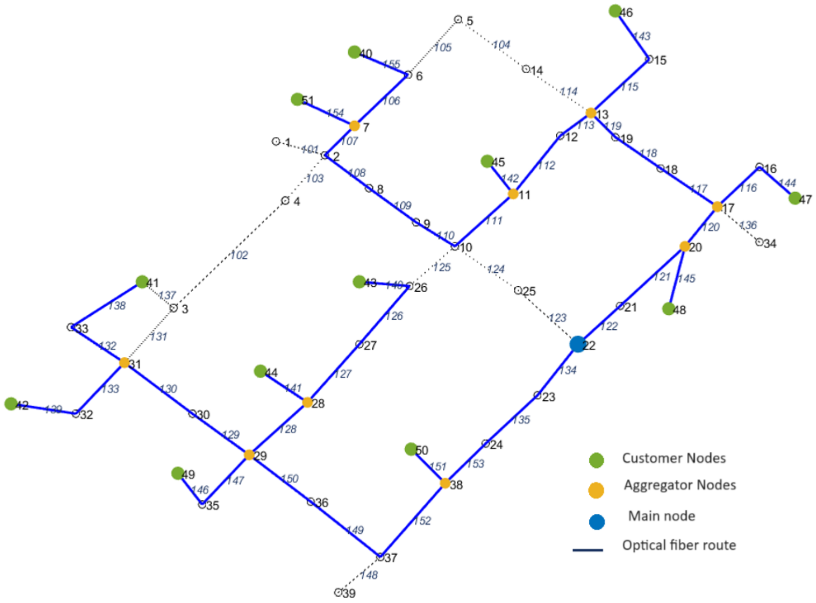


Figure 20. Network topology manually computed.

The "MAND" algorithm proposed by [20] has a multi-objective objective function, where it aims to minimize the total cost and maximize the number of aggregator nodes. However, the author himself tested his algorithm with a single objective, that is, only seeking to minimize the total cost of the network, which is identified as "Individual

Process". In Table 15, the main results obtained for the same test system are presented concisely for the validation of this work.

Table 15. Consolidated Results.

Process for Obtaining the Result	Source	Objective Function Value
Proposed Method	This Work	USD 50,490.38
Manual Calculation	This Work	USD 51,155.46
"Individual Process"	da Silva 2012	USD 71,940.00

Note that the manual calculation and the proposed method reached similar values of the objective function. The savings in this particular case are not high, but this might not be the case for larger networks. Furthermore, a metaheuristic approach will always be faster than a typical manual calculation and the automation of procedures within a company can save valuable time for engineers, allowing them to dedicate their efforts to other tasks.

5. Conclusions

This work presented a specialized genetic algorithm applied to the optimization of Passive Optical Network (PON) planning, aimed at reducing the total deployment costs of this technology. One of the main contributions of this work is the codification of potential solutions using two different arrays or vectors within a genetic algorithm, which is uncommon in the literature. In this research, a three vector was designed to account for the physical connections of the network, while an aggregator vector was designed for coding its logical topology. This strategy allowed the separation of the network's physical topology, which is typically hierarchical, from the network's logical topology, which has its layers defined according to the technology used. This dual-vector approach provided a richer and more flexible method for the entire network design and optimization process.

The adopted model abstracted not only the physical path but also the node characteristics. Unlike traditional approaches where an aggregator node has a fixed number of output ports (or other resources) and a corresponding fixed attenuation, this work allowed the number of ports and attenuation to vary according to the optimal solution found. This strategy proved efficient through the obtained results, showing its effectiveness in optimizing network design.

Another pertinent implementation was the creation of the "firstneighbor" function for connectivity testing and measuring the physical distance between two nodes, which demonstrated good computational performance during simulations due to its simple heuristic compared to pre-built functions in the simulation software. Additionally, this work highlighted the interdependence between genetic operators and their influence on the convergence of the objective function. It concluded that it is impossible to single out a parameter that would make the objective function converge more rapidly, emphasizing the need to consider the interplay of multiple parameters in genetic algorithms.

Author Contributions: Conceptualization, O.I.P. and E.M.C.-F.; data curation, O.I.P. and E.M.C.-F.; formal analysis, O.I.P., E.M.C.-F., N.M.-G., and J.M.L.-L.; funding acquisition, N.M.-G. and J.M.L.-L.; investigation, O.I.P., E.M.C.-F., N.M.-G., and J.M.L.-L.; methodology, O.I.P., E.M.C.-F., N.M.-G., and J.M.L.-L.; project administration, E.M.C.-F., N.M.-G., and J.M.L.-L.; resources, E.M.C.-F. and J.M.L.-L.; software, O.I.P., E.M.C.-F., N.M.-G., and J.M.L.-L.; supervision, E.M.C.-F., N.M.-G., and J.M.L.-L.; validation, O.I.P., E.M.C.-F., N.M.-G., and J.M.L.-L.; visualization, O.I.P., E.M.C.-F., N.M.-G., and J.M.L.-L.; writing—original draft, O.I.P. and E.M.C.-F.; writing—review and editing, E.M.C.-F., N.M.-G., and J.M.L.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the UdeA-UNIOESTE Cooperation Agreement N° 21.893.839-6/2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available through authors via e-mail.

Acknowledgments: The authors gratefully acknowledge the Colombian Ministry of Science, Technology, and Innovation "MinCiencias" through "Patrimonio Autónomo Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación, Francisco José de Caldas" (Perseo Alliance, Contract No. 112721-392-2023)

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, J.; Wosinska, L.; Chughtai, M.; Forzati, M. Scalable Passive Optical Network Architecture for Reliable Service Delivery. *J. Opt. Commun. Netw.* **2011**, *3*, 667–673. [CrossRef]
2. Asaka, K. Consideration of Tunable Components for Next-Generation Passive Optical Network Stage 2. *J. Light. Technol.* **2015**, *33*, 1072–1076. [CrossRef]
3. Miyatake, R.; Katsurai, H.; Fukada, Y.; Sekiguchi, M.; Yoshida, T. Ultra-Low-Power Optical Network Unit Driven by Optical Power Supply Using Single-Mode Fiber. *IEEE Photonics Technol. Lett.* **2023**, *35*, 874–877. [CrossRef]
4. Liu, X. Enabling Optical Network Technologies for 5G and Beyond. *J. Light. Technol.* **2022**, *40*, 358–367. [CrossRef]
5. Orphanoudakis, T.G.; Matrakidis, C.; Politi, C.; Stavdas, A. Passive Optical Network design optimization for wireless backhauling. In Proceedings of the 2011 13th International Conference on Transparent Optical Networks, Stockholm, Sweden, 26–30 June 2011; pp. 1–4. [CrossRef]
6. Yu, Y.; Wang, Y.; Li, H.; Liu, G.; Gu, R.; Ji, Y. Automatic multi-layer planning for PON design and failure simulation. In Proceedings of the 2014 13th International Conference on Optical Communications and Networks (ICOCN), Suzhou, China, 9–10 November 2014; pp. 1–4. [CrossRef]
7. Mas Machuca, C.; Kellerer, W. Planning methodology towards next generation optical access networks. In Proceedings of the 2014 16th International Telecommunications Network Strategy and Planning Symposium (Networks), Funchal, Portugal, 17–19 September 2014; pp. 1–6. [CrossRef]
8. Aguiar, A.L.S.; Sousa, F.B.C.; de Melo, Y.V.L. Optical Distribution Network Design Using PSO. *IEEE Commun. Lett.* **2023**, *27*, 239–242. [CrossRef]
9. Ciurana, A.; Martin, V.; Martinez-Mateo, J.; Schrenk, B.; Peev, M.; Poppe, A. Entanglement Distribution in Optical Networks. *IEEE J. Sel. Top. Quantum Electron.* **2015**, *21*, 37–48. [CrossRef]
10. Brestas, G.; Kanakis, G.; Spyropoulou, M.; Avramopoulos, H. Beyond 100G: All-Optical Processor for High-Capacity Access Networks. *Photonics* **2024**, *11*, 640. [CrossRef]
11. Guo, H.; Yang, C.; Chen, Z.; Li, H. Enhanced PON and AMCC Joint Transmission with GMM-Based Probability Shaping Techniques. *Photonics* **2024**, *11*, 227. [CrossRef]
12. Zukowski, C.; Mukhopadhyay, A.; Payne, D.B.; Ruffini, M. Cost analysis of rural roll-out using a long-reach passive optical network: trading off the upfront cost under uncertainty of the user take-up rate. *J. Opt. Commun. Netw.* **2021**, *13*, 69–84. [CrossRef]
13. Sanchez, R.; Hernandez, J.A.; Larrabeiti, D. Network planning for dual residential- business exploitation of next-generation passive optical networks to provide symmetrical 1 Gb/s services. *J. Opt. Commun. Netw.* **2016**, *8*, 249–262. [CrossRef]
14. Ullah, R.; Ullah, S.; Imtiaz, W.A.; Khan, J.; Shah, P.M.A.; Kamran, M.; Ren, J.; Chen, S. High-Capacity Free Space Optics-Based Passive Optical Network for 5G Front-Haul Deployment. *Photonics* **2023**, *10*, 1073. [CrossRef]
15. Calvo-Salcedo, A.F.; González, N.G.; Jaramillo-Villegas, J.A. Dynamic Spectrum Assignment in Passive Optical Networks Based on Optical Integrated Microring Resonators Using Machine Learning and a Routing, Modulation Level, and Spectrum Assignment Method. *Appl. Sci.* **2023**, *13*, 13294. [CrossRef]
16. Wang, X.; Gan, C.; Tong, L. Adaptive Scheduling Algorithm for the Coexistence of ONUs with Different Tuning Time in Virtual Passive Optical Network. *IEEE Photonics J.* **2019**, *11*, 1–8. [CrossRef]
17. Montalvo, J.; Tapetado, A.; Montero, D.S.; Vázquez, C. New Fiber Supervision Technique for Passive Optical Networks Supporting Mobile Services. *IEEE Photonics Technol. Lett.* **2016**, *28*, 501–504. [CrossRef]
18. Hajduczenia, M.; Lakic, B.; da Silva, H.J.; Monteiro, P.P. Optimized passive optical network deployment. *J. Opt. Netw.* **2007**, *6*, 1079–1104. [CrossRef]
19. Żotkiewicz, M.; Mycek, M. Reducing the costs of FTTH networks by optimized splitter and OLT card deployment. *J. Opt. Commun. Netw.* **2017**, *9*, 412–422. [CrossRef]
20. da Silva, H.A.; de Souza Britto, A.; de Oliveira, L.E.S.; Koerich, A.L. Network infrastructure design with a multilevel algorithm. *Expert Syst. Appl.* **2013**, *40*, 3471–3480. [CrossRef]
21. Agudelo, L.; López-Lezama, J.M.; Muñoz-Galeano, N. Vulnerability assessment of power systems to intentional attacks using a specialized genetic algorithm. *Dyna* **2015**, *82*, 78–84. [CrossRef]
22. Gomes, E.; Pereira, L.; Esteves, A.; Morais, H. Metaheuristic Optimization Methods in Energy Community Scheduling: A Benchmark Study. *Energies* **2024**, *17*, 2968. [CrossRef]

23. Pérez Posada, A.F.; Villegas, J.G.; López-Lezama, J.M. A Scatter Search Heuristic for the Optimal Location, Sizing and Contract Pricing of Distributed Generation in Electric Distribution Systems. *Energies* **2017**, *10*, 1449. [CrossRef]
24. Carreno, E.M.; Romero, R.; Padilha-Feltrin, A. An Efficient Codification to Solve Distribution Network Reconfiguration for Loss Reduction Problem. *IEEE Trans. Power Syst.* **2008**, *23*, 1542–1551. [CrossRef]
25. Vinasco, G.; Rider, M.J.; Romero, R. A Strategy to Solve the Multistage Transmission Expansion Planning Problem. *IEEE Trans. Power Syst.* **2011**, *26*, 2574–2576. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.