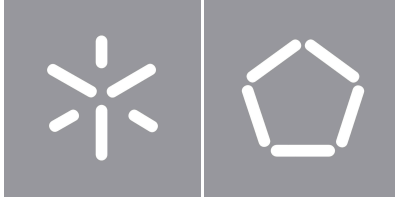


University of Minho
School of Engineering

João Silva Torres

Automatic Network Design



University of Minho
School of Engineering

João Silva Torres

Automatic Network Design

Master's Dissertation in Informatics Engineering

Dissertation supervised by

César Analide de Freitas e Silva da Costa Rodrigues

Copyright and Terms of Use for Third Party Work

This dissertation reports on academic work that can be used by third parties as long as the internationally accepted standards and good practices are respected concerning copyright and related rights.

This work can thereafter be used under the terms established in the license below.

Readers needing authorization conditions not provided for in the indicated licensing should contact the author through the RepositóriUM of the University of Minho.

License granted to users of this work:



CC BY-NC-SA

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Acknowledgements

I would like to thank my academic supervisor, Professor César Analide, for his availability in guiding the writing of this dissertation, as well as for his support and encouragement.

A special thanks to InovaRia for the opportunity to carry out this internship, and to my supervisors at Altice Labs, particularly Antero Carvalhido and João Guerrinha. My gratitude also goes to Xavier Silva for his continuous support throughout the project.

To my grandparents, who celebrate all my achievements and constantly inspire me to be a better person. Especially to my grandfather Xavier and my grandmother Virgínia, who always dreamed of this moment with me but didn't live long enough to witness its realization. I hope they continue to watch over me and feel proud.

To my parents and my sister, for their unconditional love, care, and support in all moments of my life. Without them, none of this would be possible.

To my friends, for all the moments we've shared over these past five years. You have made this journey special and unforgettable.

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, Braga, october 2024

João Silva Torres

Abstract

With the increasing demand for high-quality internet services, deploying Gigabit Passive Optical Network/Fiber-to-the-Home networks is one of the biggest challenges internet providers have due to the significant investments involved. Automated network design usage becomes more critical to aid with planning the network by minimizing the costs of planning and deployment.

In today's inventory environment, a designer typically takes around 1.5 days to create a network design for a map with approximately 128 homes (a distribution cell). The challenges include: positioning the installation equipment such as cabinets, poles, or ducts; determining the number of apartments and houses; planning cable pathways; defining the primary and secondary networks; establishing the influence area (which houses are served by each Optical Distribution Point); routing the cables; and setting up the connections.

The design must be validated at the end to ensure it complies with the optical budget, coverage areas, splitting, and necessary adjustments. The algorithms need to address several criteria to develop a feasible solution. Specific rules and local-related variations, such as the maximum distance between the optical line terminal and the optical network unit or the permissible number of splits in a cable, must be considered, as well as preferences and constraints related to the network design.

This dissertation presents an AI-based solution that relies on a genetic algorithm using a two-level representation to design the networks automatically. The business rationale behind implementing the supported features (focused on the distribution cell) and the associated flexibility will be detailed, allowing for easy adaptation and configuration according to specific rules. The validation of the approach will be presented by comparing the quality of the generated solutions with the handmade designs deployed in real-world scenarios. The results will reassure that this method can save costs and time in finding suitable and better solutions than existing ones, indicating its potential as a support design tool of solutions for Gigabit Passive Optical Network/Fiber-to-the-Home networks.

On average, across the five provided networks, it was possible to achieve a cost reduction of approximately 50% and reduce the creation time from several hours to less than 30 seconds for all generated

networks. Future work has already been outlined, including extending the automatic design to the primary network and integrating the system into the inventory system.

Keywords Fiber-to-the-Home, Gigabit Passive Optical Networks, Graph Theory, Optimization, Meta-heuristic Algorithms, Genetic Algorithms, Steinner Tree.

Resumo

Com o aumento da procura por serviços de internet de alta qualidade, a implementação de redes *Gigabit Passive Optical Network/Fiber-To-The-Home* torna-se um dos maiores desafios para os provedores de internet, devido aos significativos investimentos envolvidos. A automatização do *design* de redes é cada vez mais crítica para auxiliar no planeamento da rede, minimizando os custos e o tempo de criação.

Atualmente, um *designer* leva cerca de 1,5 dias para criar um projeto de rede para uma área com aproximadamente 128 residências (uma célula de distribuição). Os desafios incluem: posicionamento dos equipamentos de instalação, como armários, postes ou dutos; determinação do número de apartamentos e casas; planeamento das rotas de cabos; definição das redes primária e secundária; estabelecimento da área de influência (quais residências são atendidas por cada Ponto de Distribuição Óptica); roteamento dos cabos; e configuração das conexões.

No final, o *design* deve ser validado para garantir que está em conformidade com o orçamento ótico, áreas de cobertura, divisões e ajustes necessários. Os algoritmos precisam de respeitar diversos critérios para desenvolver uma solução viável, considerando regras específicas e variações locais, como a distância máxima entre a *optical line terminal* e a *optical network unit*, ou o número permitido de divisões num cabo, além de preferências e restrições relacionadas ao *design* da rede.

Esta dissertação apresenta uma solução baseada em inteligência artificial que utiliza um algoritmo genético com uma representação em dois níveis para automatizar o *design* das redes. Serão detalhadas as regras de negócios utilizada para implementar as funcionalidades suportadas (focadas na célula de distribuição) e a flexibilidade associada, permitindo fácil adaptação e configuração às regras específicas. A validação da abordagem será apresentada através da comparação da qualidade das soluções geradas com os *designs* manuais implementados no mundo real. Os resultados irão demonstrar que este método é capaz de economizar custos e tempo na procura e geração de soluções adequadas e mais eficientes do que as existentes, o que indica o seu potencial como uma ferramenta de suporte ao *design* de soluções para *Gigabit Passive Optical Networks/Fiber-To-The-Home*.

Em média, nas cinco redes fornecidas, foi possível obter uma redução de custo de aproximadamente

50% e reduzir o tempo de criação de várias horas para menos de 30 segundos em todas as redes geradas. O trabalho futuro já foi delineado, incluindo a extensão do *design* automático para a rede primária e a integração do sistema no sistema de inventário.

Palavras-chave Fiber-to-the-Home, Gigabit Passive Optical Networks, Teoria dos Grafos, Otimização, Algoritmos Metaheurísticos, Algoritmo Genético, Árvore de Steiner.

Contents

1	Introduction	1
1.1	Context	1
1.1.1	Altice Labs	2
1.1.2	Operational Support Systems	2
1.1.3	Business area – Unified Inventory System	3
1.2	Motivation	4
1.3	Objectives	4
1.4	Document Structure	5
2	Problem Statement	7
2.1	Evolution of Network Planning	7
2.1.1	Historical context	7
2.1.2	Current Scenario and Respective Limitations	8
2.1.3	Proposed Solution	8
2.2	Gigabit Passive Optical Network	9
2.2.1	Background	9
2.2.2	GPON Overview	10
2.2.3	Advantages	11
3	State of the Art	13
3.1	Graph Theory	13
3.1.1	Definitions	13
3.1.2	Tree and Minimum Spanning Trees	14
3.1.3	Classic Graph Theory Problems	15
3.2	Machine Learning	16
3.2.1	Machine Learning vs Traditional Programming	16

3.2.2	Types of ML Algorithms	17
3.2.3	Key elements and phases of a ML algorithm	18
3.3	Metaheuristic algorithms	19
3.3.1	Population-based metaheuristic algorithms	19
3.3.2	Single-based metaheuristic algorithms	25
3.4	Math-oriented solutions	28
3.5	Related work	28
3.6	Commercial Solutions	29
4	Decisions for the Development	31
4.1	Solution Design	31
4.2	Tools Analysis	32
5	Developments	34
5.1	Methodology	34
5.1.1	Sprints Management	34
5.1.2	Benefits of the Methodology	35
5.2	Data Exploration	36
5.2.1	Description of the Data	36
5.2.2	Assumptions and pre-processing of the Data	39
5.3	Initial State of the System	39
5.4	Genetic Algorithm	41
5.4.1	Overview	41
5.4.2	Network Design	44
5.4.3	Fitness Function	46
5.4.4	Variation Operators	49
5.4.5	Allocating termination points to homes	52
5.5	HTTP Request	54
5.6	Configurations	59
5.7	Optical Budget	60
6	Expirements And Results	61
6.1	Experimental Setup	61
6.1.1	Computer Specifications	61

6.1.2	Network and Algorithm Specifications	61
6.1.3	Networks Details	63
6.2	Design Results	64
6.2.1	Network A	64
6.2.2	Network B	67
6.2.3	Network C	70
6.2.4	Network D	73
6.2.5	Network E	76
6.3	Summary of the Experiments	79
7	Conclusions and future work	82
7.1	Conclusions	82
7.2	Prospect for future work	84
	Bibliography	85

List of Figures

1	Global Internet Size Market, from [1]	1
2	GPON Architecture, from [2]	11
3	Types of Machine Learning, from [3]	18
4	Phases of Machine Learning, from [4]	19
5	SCRUM Process Diagram, from [5]	35
6	Example of the initial state of a network	38
7	Result of the application of the Steiner Tree to the initial state of a network	40
8	Automatic Network Design Flow Chart	43
9	GPON network considered by Altice Labs (Source: Internal documents, Altice Labs) . . .	44
10	Representation of a chromosome in the implemented solution (Source: Internal documents, Altice Labs)	45
11	Representation of the home-termination point association in the implemented solution (Source: Internal documents, Altice Labs)	46
12	Fitness evaluation and selection over a set of individuals (Source: Internal documents, Altice Labs)	47
13	Representation of the uniform crossover over the parent chromosomes (Source: Internal documents, Altice Labs)	51
14	Representation of the bit-flip mutation over a parent chromosome (Source: Internal documents, Altice Labs)	52
15	Initial network structure from the raw data (Source: Internal documents, Altice Labs) . .	53
16	Optimized network with all the relevant connections (Source: Internal documents, Altice Labs)	53
17	Network Design Flow	55
18	Handmade Solution Evaluation Flow	56

19	Optical Budget per PDO used in the solution	60
20	Handmade design of Network A	65
21	Generated design of Network A	66
22	Handmade design of Network B	68
23	Generated design of Network B	69
24	Handmade design of Network C	71
25	Generated design of Network C	72
26	Handmade design of Network D	74
27	Generated design of Network D	75
28	Handmade design of Network E	77
29	Generated design of Network E	78

List of Tables

1	Start Equipment Data Dictionary	36
2	Locations Data Dictionary	37
3	Routes Data Dictionary	37
4	Fitness value formulas	48
5	Details of the different handmade networks	63
6	Network A value comparison between handmade and generated designs	67
7	Network B value comparison between handmade and generated designs	70
8	Network C value comparison between handmade and generated designs	73
9	Network D value comparison between handmade and generated designs	76
10	Network E value comparison between handmade and generated designs	79
11	Fitness value comparisons of various network designs	80
12	Run time comparisons of various network designs	81

Acronyms

ACO Ant Colony Optimisation.

AI Artificial Intelligence.

CAGR Compound Annual Growth Rate.

CO Central Office.

CSP Communications Service Provider.

DSP Digital Service Provider.

FTTB Fiber-To-The-Building.

FTTH Fiber-To-The-Home.

FTTX Fiber-To-The-X.

GA Genetic Algorithm.

GPON Gigabyte Passive Optical Network.

ILP Integer Linear Programming.

MDU Multi-Dwelling Unit.

MILP Mixed Integer Linear Programming.

ML Machine learning.

MST Minimum Spanning Tree.

OLT Optical Line Terminal.

ONU Optical Network Unit.

OSS Operational Support Systems.

PON Passive Optical Network.

PSO Particle Swarm Optimisation.

SA Simulated Annealing.

SDU Single-Dwelling Unit.

SJ Splitter Joint.

TP Termination Point.

TS Tabu Search.

Chapter 1

Introduction

1.1 Context

In the past few years, the urge for Internet services has grown exponentially due to the growing dependence on digital technologies and the widespread adoption of online platforms. The COVID-19 pandemic has been a primary driver of this increase because much of the global population was locked at home. High-bandwidth applications, such as video streaming, cloud gaming, and online multiplayer games, were also a significant contributor to this increase.

The demand for higher data speeds, lower latency and seamless connectivity will become even more pronounced, requiring advances in the underlying infrastructure to support this evolution. According to the source [1], in 2022, the Internet service market was valued at 498620 million dollars. It is expected to grow to 644870 million by 2028, thus registering a Compound Annual Growth Rate (CAGR) of 4.4%. The figure 1 represents the expected global internet service market increase between 2020 and 2031.

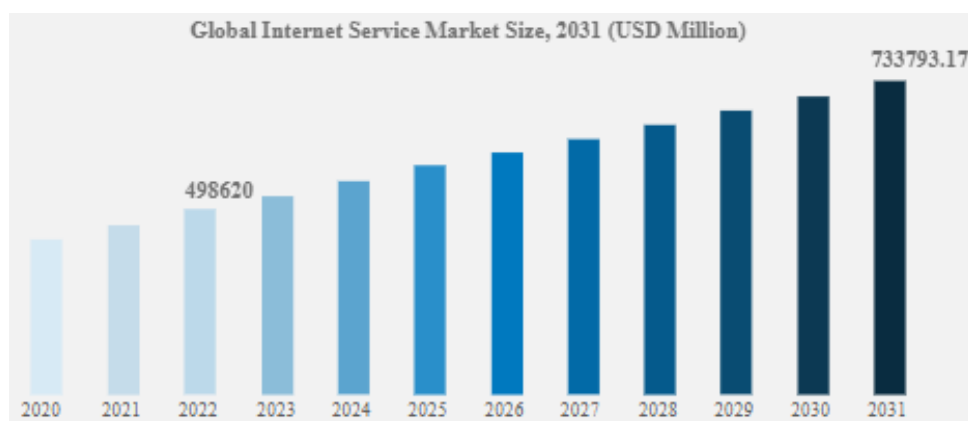


Figure 1: Global Internet Size Market, from [1]

For these reasons, Communications Service Provider (CSP) and Digital Service Provider (DSP) always face challenges in providing the best customer service, reducing costs, evolving technology, and building

networks faster than competitors; this has led them to explore advanced technologies such as Gigabyte Passive Optical Network (GPON) and Fiber-To-The-Home (FTTH) to enhance their infrastructure.

However, the rapid growth of the Internet service market necessitates a strategic approach to network planning. Automating the design of fiber optic networks becomes imperative for CSP to keep pace with evolving technologies and efficiently meet customer demands while reducing costs. Thus, integrating automatic planning tools is essential to ensure that fiber optic networks can be deployed and optimized efficiently to meet the dynamic needs of the digital connectivity market.

1.1.1 Altice Labs

Altice Labs is a company that is part of the Altice Group, a multinational telecommunications corporation operating in several countries. Its origin dates back to 1950, when an engineering team was created to develop the most advanced telecommunications technology internally.

This institution focuses on the innovation, development, and implementation of technologies in the telecom sector, particularly communication networks and digital services, which play key roles in creating advanced technological solutions.

The company's dedication to innovation is evident in its partnerships and collaborations with leading industry players, academic institutions, and research organizations. By fostering an ecosystem of collaboration, Altice Labs ensure that it remains at the vanguard of technological advancements, bringing together diverse perspectives and expertise to drive the industry forward.

With its rich history of commitment to technological advancement, Altice Labs is poised to continue providing solutions that empower individuals, businesses, and societies to thrive in an increasingly connected world.

1.1.2 Operational Support Systems

Operational Support Systems (OSS) are a software or computer systems dedicated to telecommunication service providers to control, analyze and manage an Internet network system. They are responsible for support management functions such as network inventory, service provisioning and assurance, network configuration, and fault management.

1.1.3 Business area – Unified Inventory System

The CSP market is highly competitive, with suppliers offering a various evolving services supported by a diverse variety of technologies, platforms, and networks. To meet these challenges, companies are investing in solutions that allow unified resource management and near real-time service delivery.

Any company offering products or services to its customers must compete daily to increase its revenue stream. It is essential that they make the most of their own resources to maximize their operational returns. This is particularly true for telecommunications operators for the following reasons:

- It is one of the sectors in which technological evolution has the greatest impact;
- The diversity of technologies, suppliers, and services creates a complex operational map;
- The pace at which services are created and changed requires a rapid and adaptable response from the support infrastructure;
- The quantity and variety of equipment on the network means that the resources installed require careful control, in location and availability.

To respond to these and other specificities, operators must invest in network resource management systems integrated with the operational processes of provisioning, maintenance, and performance monitoring.

Therefore, Altice Labs developed Netwin, a unified inventory system from Altice Labs NOSSIS One Suite, responsible for network development and inventory in OSS operations. An innovative inventory that enables rapid deployment of new services and technologies. One of the main features and advantages is the enhanced operational efficiency, rapid service delivery, competitive advantage, and increased revenue. Another key feature is the network design, which allows operators to inventory and plan more quickly and accurately.

Specifically, it is intended for a wide range of users including engineering teams, operational teams, network project managers and planners (for project construction), field teams (for field surveys and construction), and system integrators.

In conclusion, Netwin offers a single highly configurable system, full inventory management, to support all CSP operational and business processes from project and construction to service provisioning, reducing operational costs while enabling rapid service provisioning.

1.2 Motivation

Artificial Intelligence (AI) has been integrated to optimize network planning continuously in recent years. As mentioned in the source [6], "Relying on humans to manage such complex and dynamic networks is becoming unsustainable. Network automation promises to reduce the cost of administration and maintenance of network infrastructure [...]".

Current tools and platforms, which are primarily human-centric, make it difficult for operations to meet their objectives at a high level of quality, speed to market, and satisfaction while simultaneously reducing costs. On the other hand, computational processing is cheaper, more powerful, and faster than human resources.

The tight deadlines imposed on planners limit their ability to thoroughly evaluate different project scenarios, causing them to rush through the planning process and overlook the potential cost savings. This lack of careful consideration leads to higher project expenses. Moreover, individual planners' unique experiences and perspectives, coupled with the varying complexities of a specific network can lead to diverse design approaches [7]. Consequently, CSP and DSP are looking for automation to address these problems.

Therefore, Altice Labs has the evolution of Netwin in its portfolio, an innovative system responsible for the development and inventory of networks in the OSS. Thus, the company intends to integrate and further enhance its inventory system by introducing AI algorithms for automatic network design.

It has been consistently reported that good planning translates into significant savings in the building and operating an FTTH network, sometimes within the range of 30% [8].

The two main advantages of automating/optimizing network design are minimizing the capex and reducing time-to-create from days to hours. Therefore, using a cognitive AI-based approach allows the operator to automate network planning and designing .

The ability to automate network design using AI algorithms represents a significant advancement in the industry. As a result, the company can provide agile and efficient solutions for customers. This ultimately promotes competitiveness in the market and the constant evolution of telecommunications.

1.3 Objectives

The main objective of this dissertation is to integrate and enhance Netwin's automatic network design based on AI algorithms to minimize the capex and reduce time-to-create from days to hours. Accordingly,

using a cognitive AI-based approach allows the operator to automate network planning and design process from an Optical Line Terminal (OLT) to multiple Optical Network Unit (ONU).

To attain the objectives, the work plan includes the following goals:

- Study of the business area;
- Acquisition of general knowledge regarding a GPON network;
- Algorithms research;
- Algorithms development;
- Testing;
- Product integration in Netwin.

1.4 Document Structure

The dissertation is organized into eight chapters, each of which plays a fundamental role in presenting and developing the addressed theme. It aligns with the internship's chronological structure, incorporating the essential theoretical concepts corresponding to each topic.

The first chapter, introduction, introduces the central theme of the dissertation, establishing the context of the internship and the reasons that motivated the realization of this work. It discusses the objectives that are intended to be achieved, aligning the project expectations with the challenges identified in the field of study. Additionally, this chapter provides an overview of the document's structure.

In the second chapter, problem statement, the evolution of network planning is addressed. It starts with a historical context that underpins current practices. The limitations faced in the contemporary scenario are discussed, justifying the need for innovations, specifically the creation of this proposed solution.

The third chapter contains the state of the art. The graph theory and machine learning concepts are explored. Metaheuristic algorithms and mathematically oriented solutions are studied. This chapter also analyzes existing commercial solutions related to the addressed problem.

The fourth chapter, solution design, focuses on the analysis of all metaheuristic algorithms, comparing them to reach a conclusion on which is the most appropriate for solving the problem.

In the fifth chapter, tools analysis, the tools used throughout this project are analyzed.

The sixth chapter is dedicated to the development of the project. It discusses sprint management, data exploration, and the description of the initial state of the system. Additionally, the chapter details the

implementation of the genetic algorithm and the entire program.

In the seventh chapter, the experiments conducted and results obtained are explained in order to evaluate the algorithm.

Finally, the eighth chapter, conclusions and future work, synthesizes the main conclusions of the work, reflecting on the contributions of the internship and the knowledge gained throughout the process. Furthermore, future work perspectives are discussed, proposing directions and areas that deserve further investigation to deepen the understanding of the addressed theme.

Chapter 2

Problem Statement

This section discusses the evolution of network planning. First, a historical context is provided, then the current scenario and its limitations are discussed, followed by a proposal to overcome these drawbacks. Finally, an exploration of GPON networks is provided.

2.1 Evolution of Network Planning

Throughout history, network planning has evolved significantly along with the development of technologies and societal needs. In this dynamic evolution, strategic adaptations have been crucial to meet the changes in connectivity demands.

2.1.1 Historical context

Initially, the design was performed manually, especially in networks using cables suspended between poles, and intuitively, based on engineers' experience. Typically, they were constructed from a central point, branching out towards more distant areas.

The first materials used for network creation were copper and coaxial because of their high conductivity and ease of installation. In the initial stages, they were primarily employed to transmit analogue signals such as voice and video. Subsequently, the transmission of digital signals, including data and images, was introduced. However, as technology has advanced, some disadvantages have become apparent, such as limited range, susceptibility to interference, and the need for regular maintenance.

Beyond the limitations mentioned in the previous paragraph, the documentation of the entire process often involved physical paper records, which were frequently prone to human errors and challenges in updating as the infrastructure expanded.

Over time, migration to underground cables has become a common practice, offering numerous benefits, such as enhanced protection against adverse weather conditions and external interferences,

as well as a cleaner and more organized urban aesthetic. However, manual management remained a challenge because underground infrastructure expansion and maintenance were complex and required detailed documentation. Simultaneously, the demand for bandwidth increased and the complexity of networks grew, necessitating the development of new methods to streamline this entire process.

2.1.2 Current Scenario and Respective Limitations

Owing to technological advancements, new techniques, tools, and equipment have been developed to address these issues.

FTTH technology has emerged, utilizing fiber optic cables to transmit data from the OLT to customers' homes. It offers several advantages over traditional networks, including higher transmission speed, lower latency, increased immunity to interference, and improved service quality. Optical fibers replaced the copper and coaxial cables. This new material features a greater transmission capacity, increased resistance to interference, reduced signal loss, greater bandwidth, faster speeds, longer distances, and enhanced reliability.

The transition to unified inventory systems was one of the improvements introduced, providing greater agility in management and better overall visibility of projects, which led to more efficient decision-making. It was also essential to introduce network design and planning experts to handle the growing complexity of fiber-optic infrastructure in FTTH networks.

However, over the years, this technique has required innovation. The implementation of these networks is both complex and expensive. Relying solely on humans for network design is not feasible, as planning depends on each individual's experience, and errors can occur. As a result, the output varies from specialist to specialist. Moreover, the execution time of a plan can take several days, and the associated costs of this process are high, owing to both the accuracy of the planning and the specialists' salaries.

Therefore, meticulous planning is necessary to ensure efficiency and profitability.

2.1.3 Proposed Solution

The proposed solution involves implementing advanced automation systems in network design through AI. The primary objective is optimizing the network creation process, significantly reducing the time required from days to just a few hours while decreasing the associated implementation costs.

The introduction of AI into the network design offers several advantages. First, it can analyze large datasets simultaneously, ensuring optimal resource allocation. In addition, it allows for consistent standardization of the planning process, eliminating variations that arise when relying exclusively on human

experience. In summary, it avoids common mistakes and continually refines its capabilities.

This new approach will also reduce dependence on human experts, thereby avoiding inconsistencies in planning. This contributes to a more sustainable approach, as automation can be scaled according to demand without proportionally increasing the number of specialists.

By adopting this innovative approach, the expectation is to achieve a significant reduction in costs and implementation time and an improvement in the operational efficiency and reliability of FTTH networks. In turn, propels the ongoing advancement of communication infrastructure. These networks can be implemented through various technologies, but one of the best is GPON.

2.2 Gigabit Passive Optical Network

As the demand for higher data speeds and seamless connectivity continues to surge, GPON started to consider using faster technologies to meet the demands of fiber optics such as Fiber-To-The-X (FTTX). There are several types, but the most common are FTTH, which connects directly to the customer's home, and Fiber-To-The-Building (FTTB), which connects directly to the buildings [9]. There are various ways to provide these services, but the most common is the GPON, one of the possible standards for a Passive Optical Network (PON) network topology.

A GPON is a high-bandwidth communication system that uses fiber-optic cables to deliver high-speed Internet access to subscribers with a point-to-multipoint architecture, in which each user has a direct connection to the network provider or distributor. A central OLT sends signals to multiple ONU at the client's residence. The splitters divide the signal from the OLT, allowing a single fiber-optic cable to serve multiple subscribers. A cost-effective and efficient way to deliver high-speed internet connectivity to large areas [10].

2.2.1 Background

The design of a GPON involves three key components: OLT representing the shared ISP network equipment, ONU on the client side, and splitters. The splitter divides the signal from the OLT into multiple ONU with a 1:N ratio, where N represents the number of output ports.

To optimize GPON/FTTH design, the splitter ratio and level must also be considered. Centralized approaches may use a 1:32 ratio, serving up to 32 homes, whereas cascading approaches involve multiple splitters with varying ratios. The choice depends on the region, with centralization suited for urban areas and cascading for extended coverage.

The strategic placement of network equipment is crucial for minimizing costs. Factors such as the splitter and ONU locations, the headend position, and the optical distribution point placement must be considered. For FTTB scenarios, demand variations, such as 50 ports for a building versus one for a house influences cabinet placement.

The optical power budget is vital regardless of the topology. Therefore, this aspect must be considered and respected. The budget accounts for the light required to transmit signals through a fiber connection over a certain distance. The sources of budget loss include two main factors: splitter loss and cable length, which are considered significant contributors. The first occurs when the splitter divides the signal into different fibers. The second can be reduced by minimizing the cable distance, which enhances network efficiency by reducing loss. Other components such as connectors, can also lead to losses.

In summary, the automated design of a GPON network involves determining the optimal equipment locations, splitter ratios and respecting the optical power budget. The goal is to achieve effectiveness while minimizing financial costs.

2.2.2 GPON Overview

A GPON can adopt various architectures, but a tree-based topology is the most common. The components that constitute the network are:

- Central Office (CO): functions as the main hub for network management, coordination and control;
- Feeder Network: carries the aggregated data from the CO to the distribution points;
- Splitter Joint (SJ): combines signals from various feeder fibers, dividing them into N outputs (1:N ratio);
- Distribution Network: Delivers aggregated data from the SJ to the Termination Point (TP) through drop cables;
- Splitter: divides the incoming optical signal from the OLT into multiple paths;
- TP: physical location that serves as an access point to connect to the ONU;
- ONU: The device is located in the client's house and receives and processes optical signals, facilitating high-speed internet and telecommunication services;
- Drop Cable: fiber-optic cable directly connects the TP to various ONU, ensuring the final link for high-speed broadband services.

According to the source [11], a GPON network can reach a maximum distance of 20 km between the service distributor and the customer. This network has two data transfer directions: downstream and upstream. The downstream transmission is from the CO to the users and generally has a speed of 2.4 Gbit/s, whereas the upstream transmission is from the users to the CO and generally has a speed of 1.2 Gbit/s.

Owing to the inherent asymmetry of Internet usage patterns, there is a difference in upstream and downstream transmission speeds. This imbalance stems from the fact that users perform more downloads than uploads. Additionally, streaming videos and downloading files are much more bandwidth intensive than uploading photos and videos to the cloud. This variance also represents a good balance between cost and performance. It provides ample downstream bandwidth for most users while being affordable for network operators. The figure 2 is a visual representation of a GPON architecture.

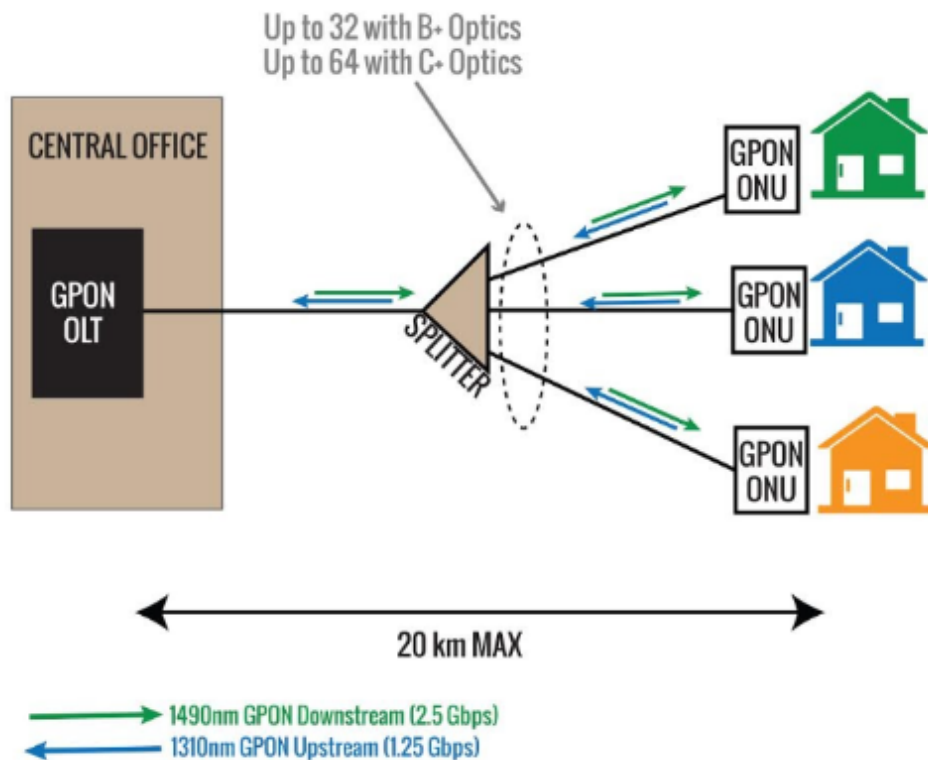


Figure 2: GPON Architecture, from [2]

2.2.3 Advantages

A study conducted by [12] compared the energy consumption of GPON and copper networks. The study found that GPON networks consumed 95% less energy than traditional copper networks. Thus, they proved to be the best option for the CSP.

Another study from July 2021 [\[13\]](#) found that GPON has decreased cost, size and weight compared to copper cables networks because “fiber optic cables are easier to install, require less duct space, weigh 10 to 15 times less and cost less than copper.” Furthermore, the authors discovered fiber optics have lower losses, no sparks or shorts, better security and grounding, and better electrical isolation than copper wires.

Chapter 3

State of the Art

The primary objective of this section is to thoroughly review the literature on automatic network design. First, there is a comprehensive exploration of AI and its diverse applications, followed by examining and discussing various algorithms to identify the most suitable for this dissertation. Subsequently, approaches proposed by other researchers in this field are investigated. Finally, commercial solutions are studied, providing a practical lens to assess real-world applications of AI in network design.

3.1 Graph Theory

Throughout this dissertation, networks are consistently explored. As previously mentioned, the costs and distances between different points in a topology are essential for creating an optimized infrastructure, as observed in certain graph scenarios. Therefore, this section delves into these structures, presenting definitions, then examining some common problems that partially resemble the one being explored.

3.1.1 Definitions

In Graph Theory, graphs are abstract representations of the relationships between entities. These graphs have two main components: nodes (or vertices) and edges. Nodes represent distinct entities such as devices or locations, whereas edges represent the connections or interactions between these entities. Graphs can be directed, where edges have a specific direction, or undirected, where the connections are bidirectional.

Additionally, graphs can be classified as cyclic if there is a closed path through the edges that starts and ends at the same vertex, or acyclic if there are no cycles. They can also be considered complete when every pair of distinct vertices is connected by a unique edge. Weighted graphs introduce another dimension, where each edge is assigned a numerical value or weight that represents a measure of distance, cost, time, or any other relevant quantity associated with the represented relationship [14].

3.1.2 Tree and Minimum Spanning Trees

Trees are fundamental data structures in Graph Theory. They are hierarchical representations of connected objects, where each node represents an object, and each edge represents a connection between two objects. These structures exhibit two essential properties: connectivity and no cycles. Each node in the tree must be connected to another node or a subset of nodes through a unique path, and there is no circular path that returns to the same node, which guarantees the hierarchical structure of the tree.

Spanning trees are subgraphs of trees that contain and connect all vertices of the original graph. The network planning problem can often be represented as a weighted undirected graph, where each edge is associated with a weight that represents a cost measure. Therefore, it can be compared with the Minimum Spanning Tree (MST) problem, which is a spanning tree that minimizes the sum of the weights of its edges. This structure represents the path with the lowest cost to connect all nodes of the graph. They are particularly useful in representing communication networks, where nodes represent devices or network points and edges represent physical or logical connections [15].

Prism vs Kruskal

The problem of finding the MST can be solved using various algorithms, including Prim's and Kruskal's algorithms [16].

Prim's algorithm is an iterative algorithm that maintains two sets of nodes: a set of nodes that are already in the MST and a set of nodes that are not yet in. The algorithm starts with an arbitrary vertex and then grows, always choosing the edge with the lowest weight that connects the tree to the external set, adding it to the internal set. This process continues until all vertices are included in the MST. This characterizes it as a local greedy algorithm.

Kruskal's algorithm is another iterative approach capable of solving the same problem. However, it starts by sorting all the edges in ascending order of weight. Subsequently, it adds the lightest edges with those already selected, ensuring that the addition of an edge does not create cycles. This characterizes it as a global greedy algorithm.

Thus, the algorithms considered exhibit some disparities. While Prim employs a vertex selection strategy and operates locally, Kruskal selects the edges and operates globally. However, the major difference between them lies in the density of the graph, where Prim performs more efficiently in dense graphs (i.e., with many edges), whereas Kruskal excels in sparse graphs (i.e., the number of edges is relatively small compared to the number of vertices) [17].

3.1.3 Classic Graph Theory Problems

In the context of GPON network planning, some classical problems from graph theory can be applied to automate network design: the shortest path problem and the Steiner tree problem.

Shortest paths

As we have seen previously, the Prism algorithm is useful for connecting all points in a graph with the minimum total weight, thereby creating the MST. However, if the goal is to determine the shortest path between two nodes, this approach is not the most suitable. In a weighted graph, the most efficient way to calculate this path is by using Dijkstra's algorithm.

Dijkstra's algorithm starts with the source vertex and, at each iteration, selects the nearest node and updates the distances between its neighbors. Therefore, it is a greedy algorithm, and its time complexity is more efficient in dense graphs.

In the context of GPON network planning, this approach can be used to determine the most efficient path for cable routing between two network nodes, for example, between the OLT and a particular client [18].

Steiner Tree Problem

The Steiner Tree Problem is another classic challenge in graph theory, where the objective is to find the minimum tree that connects a specific set of vertices, even if these vertices do not form a subset of all vertices in the graph.

Considering a weighted undirected graph, along with a specific set of vertices known as terminals, the goal is to find a tree that connects all terminals with the lowest possible cost. The overall cost of a tree is determined by the sum of the weights of its edges.

This problem is recognized as NP-complete, indicating that there is no known algorithm capable of solving it in polynomial time for all the cases. The complexity associated with this type of approach means that as the number of vertices in the graph increases, the time required to find the exact solution grows exponentially [19].

Various strategies have been explored to tackle the Steiner Tree Problem. However, techniques such as metaheuristic algorithms or combinatorial optimization methods based on mathematical techniques, are commonly used to find acceptable solutions within a reasonable amount of time.

3.2 Machine Learning

Machine learning (ML) is a branch of AI that utilizes data and algorithms to reproduce the human learning process, thereby progressively enhancing its accuracy.

This data analysis technique is centered on the development of algorithms capable of learning and improving through experience, with the aim of improving performance in a specific task. Instead of being explicitly programmed to perform a task, these algorithms learn from data by progressively employing statistical methods and with minimal human intervention.

These algorithms operate by identifying patterns in pre-processed input data, with the goal of making predictions and autonomous decisions. They also possess the ability to learn from their own outcomes through an iterative training process, adjusting their internal parameters based on feedback obtained by comparing their predictions with the actual labels in the training data. This enables the model to continuously improve its performance adaptively, learning from previous samples as humans learn from experience.

3.2.1 Machine Learning vs Traditional Programming

ML and traditional programming are two distinct approaches that address problems in different ways.

In traditional programming, developers explicitly write instructions for computers to follow. Programs are rigid and follow a fixed set of instructions; they take input data, process it through predefined logic, and produce output. If the input or requirements change, the program must be modified manually. Considering these factors, the output quality depends solely on the accuracy of the programmed rules.

However, although traditional programming is suitable for tasks with clearly defined rules and logic, it may struggle with complex tasks involving large amounts of data or uncertain patterns.

To address these limitations, ML has emerged, where the computer learns from data by identifying patterns and relationships to make predictions or decisions, rather than being explicitly programmed for the task. Moreover, it has the ability to adapt and improve its performance over time as it is exposed to more data. Consequently, it can be generalized from the training data to make predictions on new, unseen data [20].

Thus, it proves to be particularly effective for complex tasks where it is difficult to define rules explicitly. ML models can automatically discover patterns in large datasets, which can be challenging in traditional programming.

Unlike traditional programming, ML algorithms are highly dependent on the quality and quantity of

training data. The models learn from it, and the performance is influenced by the diversity of this set. Therefore, they are particularly effective for tasks involving pattern recognition.

To conclude, ML was not created to replace traditional programming but rather to work together with it.

3.2.2 Types of ML Algorithms

ML is categorized into various types, with the three primary ones being:

- **Supervised Learning** - The algorithm is trained on a labelled dataset, where the input data are associated with the corresponding correct output. The algorithm learns to map the input to the output, becoming capable of performing regressions or classification tasks on new unseen data
- **Unsupervised Learning** - The algorithm is trained on an unlabeled dataset. It discovers patterns and relationships in the data without explicit guidance from the output. Clustering and association are common tasks for learning more about given data.
- **Reinforcement Learning** - The algorithm learns by interacting with the environment. This paradigm enables an agent to learn how to make decisions in an environment without being explicitly programmed for it. The agent learns to maximize the accumulated rewards through trial and error using feedback (rewards or penalties) from its actions.

These three fundamental types of ML algorithms have numerous practical applications that have contributed to the evolution of technology. The following image 3 illustrates these in real-world scenarios.

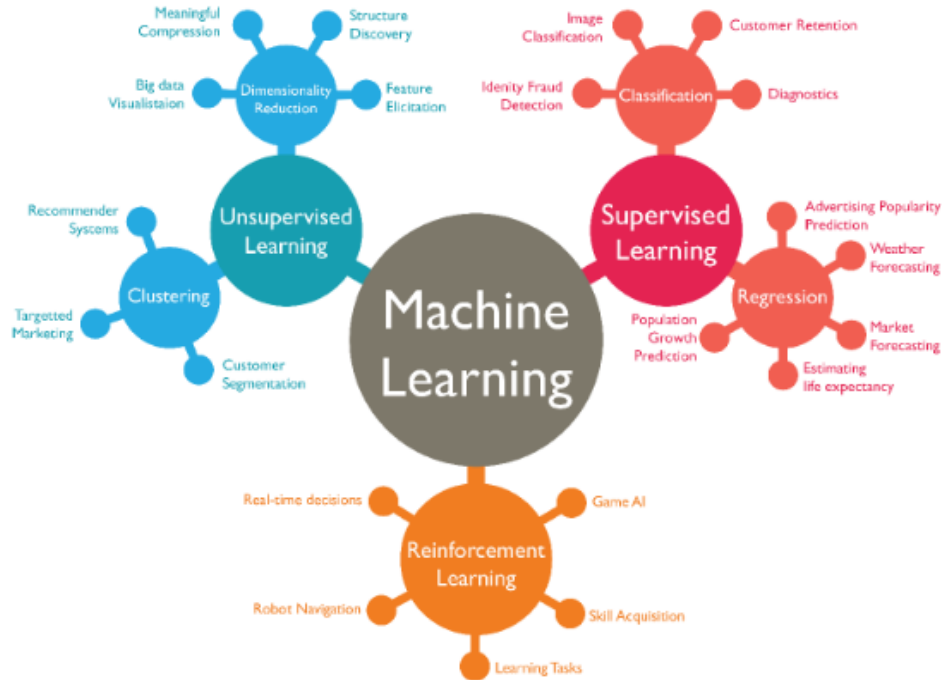


Figure 3: Types of Machine Learning, from [3]

3.2.3 Key elements and phases of a ML algorithm

Three factors must be considered in its implementation to ensure the effective performance of a ML algorithm: representation, evaluation, and optimization.

Representation refers to how the data is presented and structured for the algorithm to understand. This element is crucial for enabling the model to learn patterns efficiently. **Evaluation** involves analysing the model's performance using appropriate metrics to determine how well it generalizes to new data. It is essential for distinguishing between good and bad models. **Optimization** is related to enhancing the model based on evaluation results, adjusting parameters and strategies to improve its accuracy and effectiveness. The figure 3 provides a visual representation of the different types of ML.

The fundamental phase of this implementation begins with **data management**, which establishes an informational foundation. The **training model** phase follows, allowing the algorithm to assimilate patterns and relationships from the data prepared in the previous stage. The **evaluation** phase is responsible for measuring the effectiveness of the model and comparing its results. Finally, the **deployment** phase introduces the trained model into a real-world scenario where it can make predictions or decisions based on new data [21].

These four phases represent a logical sequence, outlining the algorithm's journey from initial data organisation to its practical application in the real world. The figure 4 shows the cyclical structure of ML

projects and the different processes within each phase.

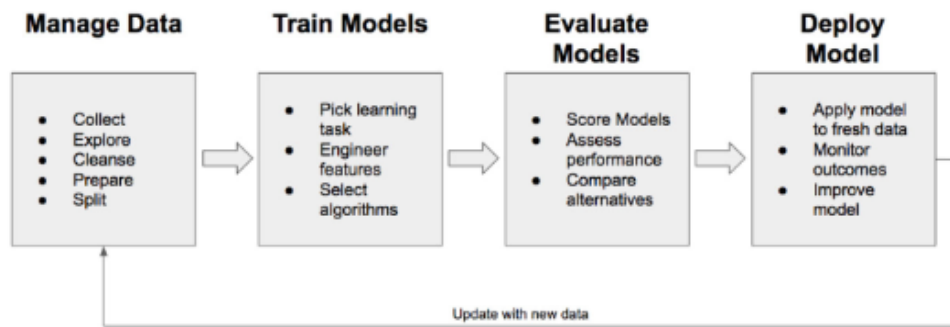


Figure 4: Phases of Machine Learning, from [4]

3.3 Metaheuristic algorithms

Metaheuristic algorithms have been discovered to be the most effective approach for tackling challenging optimization problems. Some of these are based on processes that occur in nature. Because they proved to have good results based on the execution times and are less likely to get stuck in optimal locations (points in the solution space where the algorithm gets stuck in a sub-optimal solution, unable to find a better one nearby, which are solutions that are better than their neighbours but not the globally best solution) [22], they have been helping planners achieve their daily objectives [23]. Generally, these methods provide the best solution by determining a high-potential area from the solution space and exploiting it by identifying promising regions and thoroughly exploring them [24]. Therefore, metaheuristic algorithms are useful for solving the problem under study: automatic network design [25].

The following sections delve into two primary approaches: population-based and single-based metaheuristic methods.

3.3.1 Population-based metaheuristic algorithms

Population-based metaheuristic algorithms are a type of optimization that utilizes a population of candidate solutions to iteratively search and explore the solution space to find an optimal or near-optimal solution to a given problem. Unlike traditional optimization methods that operate on a single solution, this class of approaches simultaneously maintains and evolves a diverse set of solutions, making it suitable for global searches because of its ability for global exploration and local exploitation [26]. In this section, three of these are explored: Particle Swarm Optimisation (PSO), Genetic Algorithm (GA), and Ant Colony Optimisation (ACO).

Particle Swarm Optimization

PSO was developed by Kennedy and Eberhart (1995). This algorithm mimics the collective intelligence of biological populations, such as bird flocks or fish schools, where individuals often share information and coordinate their actions to achieve a common goal, such as finding rich sources of food and avoiding predators, thereby developing an evolutionary advantage [27].

This metaheuristic search algorithm is used to determine the optimal solution for a given problem. The algorithm begins by initializing a population of particles, each of which represents a potential solution to the studied problem. They navigate the search space influenced by both their individual experiences and the collective knowledge of the entire swarm in an attempt to find the optimal solution.

The particles within the collective navigate the search space by adjusting both their position and velocity. Each particle's velocity is updated based on its current location, the optimal position it has previously explored, and the best position discovered by the entire swarm. This updated velocity is calculated as a weighted sum of three factors: the particle's current velocity, the best position it has encountered, and the most favorable position reached by the entire swarm. Consequently, this updated velocity is applied to modify the position of the particle.

The particles move through the search space iteratively, and each iteration is called a generation or an iteration. Within each iteration, the fitness of the individual particles is evaluated, representing a metric of the effectiveness of the particle's current position in solving the given problem. The determined fitness value is subsequently employed to modify both the particle's and the collective swarm's optimal position.

The PSO algorithm persistently iterates through the search space, until a specified stopping criterion is satisfied. This criterion may involve reaching a predetermined fitness value or completing a set maximum number of iterations [28].

In summary, the PSO algorithm serves as a heuristic search method that continuously refines the positions and velocities of a particle population to identify the optimal solution for a given problem. The algorithm relies on the collective experience of both the individual particles and the entire swarm. Owing to its simplicity, efficiency, and efficacy, the PSO algorithm has been widely adopted to optimize diverse problem sets.

Principle stages in PSO

The PSO algorithm typically comprises the following procedural stages:

- Initialization: Define the problem domain and randomly initialize a population of particles (repre-

senting candidate solutions) within the specified domain;

- Evaluation: Assess the fitness of each particle based on its position in the problem space;
- Update the personal best: Revise the personal best position of each particle according to its evaluated fitness value;
- Update the global best: The global best position is adjusted by considering the best positions attained by all the particles in the swarm;
- Update velocity and position: The velocity and position of each particle are modified using its current position and velocity as well as the personal and global best positions;
- Termination: Steps 2–5 are repeated for a predetermined number of iterations or until a defined stopping criterion is satisfied (e.g., achieving a target fitness value).

PSO Advantages

PSO has emerged as a powerful optimization technique that has gained popularity owing to its versatility and effectiveness. Here, we delve into the key advantages that make it a preferred choice for solving complex optimization problems.

The first advantage is its ability to tackle optimization problems with high-dimensional search spaces, which traditional algorithms often struggle with. This ability stems from its swarm-based nature, in which particles interact and learn from each other's experiences, allowing the swarm to efficiently search the large solution space.

The second advantage lies in the algorithm's computing architecture. The algorithm can be easily parallelized because of its population-based nature, which allows multiple particles to explore the search space simultaneously.

This algorithm is notable for its simplicity in terms of implementation and understanding, because it can be implemented in a few lines of code, and from its straightforward structure, which makes it readily accessible to developers and practitioners. In addition, PSO incorporates a short-term memory mechanism, enabling particles to explore the surrounding area while maintaining a connection to the local and global best positions [29].

PSO often demonstrates better computational efficiency when applied to unconstrained nonlinear problems with continuous variables [30].

PSO Disadvantages

The PSO algorithms have two main disadvantages. The first is the premature convergence of a swarm, which means that the particles tend to converge to a single solution too early before finding the global optimal solution. This can happen because the particles move towards the average of the best positions, which may not be the global optimum [31]. In addition, rapid information sharing between particles can lead to a loss of diversity, making it more difficult to escape local optima [32].

The second disadvantage is the dependence on parameter settings due to its poor exploration. This means that the performance of the PSO algorithm can vary significantly depending on the values of the parameters, such as the inertia weight, learning factors, and social and cognitive acceleration coefficients. No single set of parameters works best for all problems, and the optimal values may need to be tuned for each specific problem. This can be a time consuming and challenging task [32].

Genetic Algorithm

GA are a type of evolutionary optimization technique that mimics natural selection to iteratively improve a population of solutions over generations using a given characteristic. Their procedure is based on the Darwinian theory of biological evolution by natural selection, in which the fittest survives and passes on their characteristics to their descendants, ensuring that the best characteristics are present in a given population [33], [34].

An initial population is created with a predefined number of individuals, each represented by a genetic string. Each individual has an associated fitness measure. The concept that the fittest individuals in a population produce the fittest offspring is then implemented to reproduce the next population. The selected individuals are chosen for reproduction in each generation, with an appropriate mutation factor to randomly modify an individual's genes in order to develop a new population. The result is another set of individuals based on the original individuals, which leads to subsequent populations with better fitness. Individuals with lower fitness are naturally eliminated from the population.

The four phases of GA are evaluation, selection, crossover, and mutation. The **evaluation** measures the fitness of each individual solution in the population by assigning a score. The **selection** randomly chooses individuals from the current population to develop the next generation. The **crossover** takes two selected individuals and combines them around a crossover point, creating two new individuals. The **mutation** randomly modifies the genes of an individual subject to a small mutation factor, thereby introducing more randomness into the population. Thus, the GA is not restricted to staying in optimal locations. These characteristics offer an effective methodology for global searches in the context of the network de-

sign problem. This iterative procedure continues until (i) a known optimum or acceptable solution level is reached, (ii) a maximum number of generations has been made, or (iii) a certain number of generations has occurred with no improvement in fitness [35].

GA advantages

GA have emerged as powerful and versatile tools for solving optimization problems because they offer a unique combination of advantages. In the following paragraphs, these aspects are explored and explained.

The GA has the ability to avoid getting trapped in a local optimum [36] because of the mutation process, which is responsible for the introduction of randomness in the population, allowing the algorithm to explore a wider range of solutions. Consequently, they can identify multiple good solutions even in cases where the optimal solution is not unique. Therefore, another remarkable advantage is its efficiency when exploring large solution spaces, making it well-suited for tackling complex optimization problems. Both these advantages are valuable in network design, where multiple objectives, such as minimizing costs and transmission efficiency, are considered simultaneously.

Additionally, they offer flexibility and adaptability because, unlike other optimization algorithms that rely on explicit mathematical formulations, GA can handle complex constraints without requiring detailed mathematical modelling.

Furthermore, the algorithm features parallel exploration and diversification, because it simultaneously evaluates multiple potential solutions, leading to a diverse set of candidates. This characteristic accelerates the search process and increases the likelihood of identifying high-quality solutions.

The GA can also be hybridized. Its flexibility allows it to be combined with other optimization techniques, to improve their performance. As conclude in the study [37] the hybridization of GA has improved the quality of iterations obtained. Additionally, it can be adapted to imperfect or noisy data, as shown in the study by [38] where GA outperformed other optimization techniques when finding the best solution to problems involving high-dimensional search space and noisy data.

GA has a better computational efficiency when applied to linear constrained problems with discrete variables [30].

GA disadvantages

Despite their numerous strengths, GA have certain limitations that should be considered when applying them to network design.

The first is the high computational complexity involved in dealing with large search spaces. They

require many resources to run, making them slow. They also maintain a large population of solutions, which is computationally expensive.

The second disadvantage is the dependence on randomness during the search process. This arises from the inherent randomness of the selection, crossover, and mutation operators. On one hand this characteristic allows it to explore a broader search space, but on the other hand it also introduces the possibility of converging to suboptimal solutions due to chance factors [39].

The third drawback is the sensitivity and difficulty of tuning the parameters. The algorithm based on the evolutionary theory relies on a set of parameters, such as the mutation rate, crossover rate, and population size, which significantly influence their performance. Selecting suboptimal or incorrect values for these parameters can result in premature convergence, where the algorithm prematurely settles on a solution without thoroughly exploring the search space, leading to poor performance. Conversely, excessively high parameter values can lead to inefficient exploration, hindering the discovery of high-quality solutions [36].

Ant Colony Optimization

ACO is an algorithm inspired by the foraging behaviour of ants, where communication among them relies on the use of pheromones, which are chemical substances that they deposit on the ground to convey information. When an ant finds a food source, it leaves a trail of pheromones on its way back to the nest. The better the trail, the more chemical substances it leaves behind. Other ants are more likely to follow the trail with more pheromones, which helps the colony find the best food sources. The ACO works in a similar way, except that the ants are not looking for food sources, but for solutions to problems [40].

In order to construct a solution, it is necessary to build a graph, where the selection of edges is typically done randomly, with the probability of each edge being chosen being proportional to the pheromone value associated with it.

At each iteration, each ant must travel a path that is evaluated by considering the fitness function. Based on the results, all paths were updated with more or less pheromones. More pheromones are added to the paths that represent a good solution to the problem, fewer are added to the others. Thus, pathways with more chemicals, which are the best solutions to date, attract more ants.

Finally, at the end of each iteration, evaporation occurs, in which the value of the pheromones associated with each edge is reduced by a predetermined percentage. Thus, the least favorable edges have fewer pheromones with each iteration, which contributes to making them less active; consequently, these routes are not chosen [7].

ACO advantages

ACO has parallel exploration and good stability and performance in solving discrete problems [41].

ACO disadvantages

However, this method has several limitations. According to the study [42], this metaheuristic algorithm is very sensitive to parameter settings, making it prone to becoming trapped in local optima. It also has slow convergence speed, poor solution accuracy and slow computational speed when dealing with a large amount of data [41]. Finally, the pheromone trail-laying method used by ants to communicate is very time-consuming [43].

3.3.2 Single-based metaheuristic algorithms

Single-based metaheuristic algorithms are a type of optimization that operates with a single solution candidate at a time, iteratively exploring its solution space to improve its quality. Unlike population-based metaheuristics, which simultaneously maintain and evolve a set of solutions, this approach focuses on refining a single candidate through a series of iterations [44]. In this section, two of these are explored: Simulated Annealing (SA) and Tabu Search (TS).

Simulated Annealing

SA is an algorithmic approach and stochastic gradient method inspired by the physical process of annealing which refers to an analogy with thermodynamics, specifically with the way that metals cool and anneal [44].

In this process, the metal is heated to a high temperature and then slowly cooled. The atoms have more time to organize themselves into a more stable configuration. When implemented, it works in a similar manner; however, instead of cooling a metal, it cools a solution to a problem. It starts with a random solution and then gradually modifies it to accept inferior alternatives with a certain probability. As the algorithm cools, it becomes less likely to accept worse ones and eventually converges to a good one.

One of the main advantages of SA is that it can avoid being trapped in poor local optima because it adopts suboptimal solutions. However, the likelihood of adopting them decreases as the temperature decreases [45], [46]. It is also suitable for optimization problems with a large solution space [47]. Overall, it is a flexible method that can handle a wide range of optimization problems.

However, this method has several limitations. It has a very high computational complexity [47] and,

as pointed out in a previous study [48], a long running time.

Tabu Search

TS is a metaheuristic introduced by Fred W. Glover (1986) that uses local search methods to find solutions to complex optimization problems [49]. The principal idea is to avoid getting stuck in local optima by allowing non-improving moves. To achieve this, it maintains a memory called the tabu list, which is a set of recently visited solutions temporarily forbidden from being revisited. It also records the actions that are taking place [50]. Thus, it has been applied to many optimization problems because it achieves good results and delivers near-optimal solutions.

The two main advantages of TS include ease of implementation and high computational efficiency [47]. However, it also has limitations. It is a single-based metaheuristic algorithm that operates with a single solution. Therefore, it only iterates per solution at a time, which is very time-consuming.

Algorithm	Resumed Description	Advantages	Disadvantages
PSO	PSO is inspired by the collective intelligence observed in biological swarms. A population of particles is initialized, each representing a potential solution. They navigate the search space, influenced by both individual experiences and collective knowledge. The goal is to find the optimal solution for the given problem, leveraging coordination similar to biological populations for evolutionary advantages.	<ul style="list-style-type: none"> - Efficient exploration of large solution spaces; - Parallel exploration; - Easy to implement and understand; - Short-term memory mechanism; - Computational efficiency for unconstrained non-linear problems with continuous variables. 	<ul style="list-style-type: none"> - Premature convergence; - Dependence on parameter settings.
GA	GA uses evolutionary principles to iteratively improve a population of solutions over generations. It is inspired by Darwinian theory, where the fittest individuals are selected for reproduction, passing on their characteristics. The algorithm involves four phases: evaluation, selection, crossover, and mutation.	<ul style="list-style-type: none"> - Ability to avoid getting trapped in local optima; - Efficient exploration of large solution spaces; - Parallel exploration; - Flexibility and Adaptability; - Possibility of hybridisation; - Computational efficiency for linear constrained problems with discrete variables. 	<ul style="list-style-type: none"> - High computational complexity; - Dependence on randomness; - Sensitivity and difficulty of parameter tuning.
ACO	ACO is inspired by the foraging behavior of ants. It uses pheromones trail-laying method, with better paths receiving more pheromones. Their nestmates are more likely to follow the ones with stronger concentrations. The algorithm applies a similar concept, where artificial ants search for solutions. Those with better results are reinforced, guiding subsequent iterations to converge towards local optima.	<ul style="list-style-type: none"> - Parallel exploration; - Good stability; - Good performance in solving discrete problems. 	<ul style="list-style-type: none"> - Sensitive to parameter settings; - Slow convergence speed; - Poor solution accuracy; - Slow computational speed; - Pheromone trail-laying method is very time-consuming.

3.4 Math-oriented solutions

Mathematical optimization is the process of maximizing or minimizing an objective function by determining the best available values across a set of inputs that can be used in AI, particularly in ML and operations research. Traditional mathematical techniques, such as Integer Linear Programming (ILP) and Mixed Integer Linear Programming (MILP), have been widely used to solve optimization problems because they consider the interactions between variables, making them more rigorous [51].

In an ILP, the fitness function is described by a linear expression of costs. However, in most cases of network design, this is not sufficient to precisely describe the true dynamics of the network's value because of the large number of variables and the high complexity of the constraints. Hence, the overall cost is the sum of a series of nonlinear components [47].

To adapt the problem to the specifications of linear programming, it is necessary to incorporate certain simplifications that produce approximate costs, leading to nonoptimal solutions in certain scenarios. However, it is important to note that this approach does not produce a cost-effective network. Instead, it provides solutions that depend on the previously introduced approximations [52].

ILP proves to be effective in problems with a small solution space and therefore manages to obtain an optimal solution. However, as the complexity of the problem increases, the solution space also increases. Thus, when applied to network planning, it lags behind other algorithms, such as metaheuristics, because the execution time required to find an acceptable solution is much longer [53].

The main advantage of math-oriented algorithms is that it is possible to hybridize them with metaheuristic algorithms to overcome previous limitations. This technique combines the accuracy of math-oriented methods with the flexibility and efficiency of alternative heuristics [54], [55].

3.5 Related work

A recent work by Aguiar [56] presented an approach combining PSO with georeferenced data and a pricing scheme to tackle the optical distribution network design problem in the FTTH network. The proposed algorithm achieved a 12% cost reduction compared with the conventional method and a 6% mean computational time reduction for 270 ONU compared to the random strategy.

Another work [35] combined GA with graph theory to solve PON projects, considering three different types of basic topologies. The proposed approach was responsible for topology searching, cable routing, and automatic determination of the splitter placement in the network, aiming for both the lowest cost and

the best performance network. To achieve this goal, the authors' solution for the GA relies on the paths between the OLT and each ONU, which are determined using graph theory algorithms.

In this paper [7] ACO, an algorithm is proposed to automate the FTTH planning process. In fact, it searches for the optimal or near-optimal path between the OLT and ONU, identifying the optimal splitter locations and the most efficient paths to the routing cable. This method proved to be promising for tackling this problem because it was able to simultaneously improve the running time and reduce overall costs.

A study combined MILP with SA [57] for distribution network planning. This approach involves decomposing the original planning problem into sequences of sub-problems using a MILP model to reduce complexity. SA is then used to guide the decomposition and solution processes to obtain the minimum total cost solution. The results showed that the model was able to minimize investment costs, losses, and network signal attenuation.

A procedure using TS was also developed for the optimal design of networks. In this approach [58] the authors proposed a TS technique to solve this problem and the results showed that this algorithm was able to minimize the investment and operation cost of the network.

Another study integrated ILP with metaheuristics into a network design problem. The mathematical technique was first used to obtain the initial network configuration, and then the metaheuristic was applied to modify the previous solution to obtain a more cost-effective solution. In fact, this approach allowed researchers to obtain a solution within 7% of the optimum solution and to achieve capex savings of up to 15% [59].

3.6 Commercial Solutions

To develop an innovative and competitive tool, it is essential to acquire an in-depth understanding of the existing commercial solutions. This involves conducting an analysis to be aware of the key features and functionalities of different approaches as well as understanding the specific needs of customers, which is crucial for developing a product that meets their expectations.

Fast@Home is a tool for automated and cost-optimized planning of FTTB networks that allows users to increase their network efficiency, automatically plan development areas quickly, and determine the costs associated with various planning scenarios. This solution uses mathematical algorithms to optimize routes and minimize construction costs. It serves as the basis for studies and subsequent decisions regarding the investment and allocation of resources [60].

The X-Brain Fibre is a FTTH network planning and design solution that employs a unique algorithm

that considers various factors, including terrain, existing infrastructure, and customer demand, to generate optimized network designs. The platform can handle various network topologies, process real-time data from various sources, and provide interactive visualizations of network designs, thereby allowing for easy analysis and modification. This is useful for reducing the planning time and costs [61].

The Ksavi Network Inventory is a platform that combines geographic information systems and data management capabilities to support the planning and design of fiber-optic networks. It provides a centralized repository for storing and managing network assets such as fiber cables, poles, ducts, and other infrastructure elements [62].

Although it is not possible to obtain detailed information regarding the architecture of each approach, as this is part of the trade secret, it is possible to conclude that all of them use AI to achieve the same objectives:

- Reduce planning time;
- Reduce overall project costs;
- Optimize network design;
- Improve network performance.

To conclude, the introduction of AI in network planning is essential in order to maintain competitiveness in the telecommunications sector.

Chapter 4

Decisions for the Development

In this chapter, a comparative analysis of the various studied algorithms is conducted to draw conclusions that enable the identification of the most suitable method for the specific needs of automated network planning. After that, a detailed analysis of the tools used in the development of the project will be conducted.

4.1 Solution Design

The ILP proves to be limited for automatic network planning because of the problem's vast solution space. Although the fitness function is expressed linearly with costs, the high complexity of the constraints and the large number of variables in the network dynamics require simplifications that lead to approximate costs and non-optimal solutions in certain scenarios. Furthermore, this approach does not guarantee a cost-effective network, depending on previously introduced approximations [52], [47]. It is only effective when dealing with small solution spaces and requires significantly more runtime to find acceptable solutions for more complex problems [53]. As a result, using ILP alone is not the best option for this study.

As pointed out in a previous study [48], when compared with the GA, the SA is more likely to discard potentially good solutions more often because of its single-solution focus. In contrast, the GA maintains a population of solutions, which leads to a tighter distribution of solutions, with a standard deviation of almost half that of SA. Consequently, because of its long running time, SA may not be the best option for the automation of network design [47].

The same problem occurs with TS owing to its single-solution focus. Compared with other population-based methods, TS only iterates per solution at a time, whereas the others maintain a population of solutions. It is suited for well-structured problems. However, because network design is a complex problem with a vast solution space, this approach might not be the best option.

In summary, the studied single-based metaheuristic algorithms (TS and SA) proved to be less effective

than population-based approaches, because the latter can operate in multiple populations simultaneously, enabling a more comprehensive and efficient exploration of the solution space. For instance, while GA uses genetic operators to generate new populations and explore multiple possibilities concurrently, TS and SA operate sequentially, iterating from one to another. This results in slower convergence, leading to less effective exploration of the solution space.

Taking into account the conclusions of the previous paragraphs, only population-based algorithms remain, namely PSO, GA, and ACO.

The first approach explored will be ACO. As analysed earlier, one of the major disadvantages of this approach is that the trail-laying method is very time-consuming. Consequently, when exposed to problems with large amounts of data, there are some issues in the convergence speed and solution accuracy. However, network planning is characterized by a vast solution space. Therefore, the ACO algorithm may not be the best option for automatic network design.

The choice is between PSO and GA. According to the source [63], there is no significant difference between the approaches when used for cases with relatively small data. However, the design of networks is a highly complex problem because of the vast solution space.

A study conducted in Texas [30] concluded that the computational effort of these approaches depends on the problem under study:

- PSO outperforms GA when applied to unconstrained nonlinear problems involving continuous design variables;
- GA outperforms PSO when applied to nonlinear problems with constraints involving continuous or discrete design variables.

Network design involves constraints, such as the maximum signal attenuation and splitting limit. Thus, it is concluded that GA is better suited for the automation of network design.

4.2 Tools Analysis

Choosing the right tools is crucial to ensure the efficiency and effectiveness of the proposed solutions.

In the development of this project, the programming language used was Python. This choice was made due to its flexibility, extensive library of tools, and ease of handling data in formats like JSON, which are essential to the project's workflow.

For submitting design and evaluation requests for the network, Postman was used. This tool allowed for efficient execution and automation of HTTP requests, ensuring smooth communication between the system and the backend components involved.

Additionally, Flask was chosen as the web framework due to its simplicity and flexibility in developing REST APIs.

Visualizing the results played a crucial role in the interpretation and analysis of the data generated during the network design process. For this, Python's matplotlib library was utilized, enabling the creation of graphs and visual representations of the data. These visualizations were essential for understanding and optimizing the generated designs.

Additionally, JSON was the primary format used for analyzing the data received and sent, and it was also used in the contracts and communication between services. Essentially, it is the language of the service.

All decisions made throughout the development process were supported and aligned with the company's guidelines and objectives, ensuring that the chosen tools and methods met the business needs effectively.

Chapter 5

Developments

This chapter presents the key developments made throughout the project, highlighting the methodology employed, data exploration efforts, and the initial state of the system. The focus is on the implementation of a genetic algorithm for network design, detailing the structure of the algorithm, its fitness function, and the variation operators utilized. Additionally, the chapter covers the management of termination points, emphasizing the strategic approach to optimizing network connections. Finally, it discusses the integration of HTTP requests in the system and the optical budget, providing insights into how the network design process is facilitated through this interface.

5.1 Methodology

In this project, an agile approach was adopted, specifically using the Scrum framework, to manage and organize task development. The agile methodology promotes flexibility and continuous delivery of value, while Scrum, as one of the most recognized agile frameworks, implements these principles through iterative cycles known as sprints. Scrum is widely noted for its effectiveness in software development environments, providing flexibility and continuous improvement [64].

5.1.1 Sprints Management

The project was conducted in development cycles called sprints, each lasting three weeks. At the beginning of each sprint, a planning meeting was held where the objectives were defined. During this phase, tasks, known as issues, were identified and detailed, and later received an effort estimate, measured in story points, reflecting the complexity and time required for their completion. The tasks were then distributed among the team members.

During the sprint, daily meetings were held to update the team on the progress of work completed the previous day. These brief meetings were essential for keeping everyone informed about the current status

of tasks and for quickly resolving any issues that arose.

To track activities and task progress, we used the Jira platform. This tool facilitated the visualization of the status of each issue, from its creation to completion. It also allowed constant monitoring of progress, highlighting tasks in different stages of development: "TO DO", "IN PROGRESS", "UNDER REVIEW", "RESOLVED (READY FOR TESTING)", and "DONE".

In addition to planning and review meetings, we held daily stand-ups to update the team on the progress of work done the previous day. These brief meetings helped keep everyone informed about the current status of tasks and quickly address any impediments.

At the end of each sprint, two meetings were held: the review meeting and the retrospective meeting. During the review meeting, the team presented and discussed the results achieved, evaluating the objectives met and the issues completed. This was followed by the retrospective meeting, where the team analyzed the successes and challenges faced throughout the sprint. In this session, the difficulties encountered were identified and points for improvement for the future were discussed. The figure 5 provides a detailed diagram illustrating the key components and processes of the SCRUM methodology.

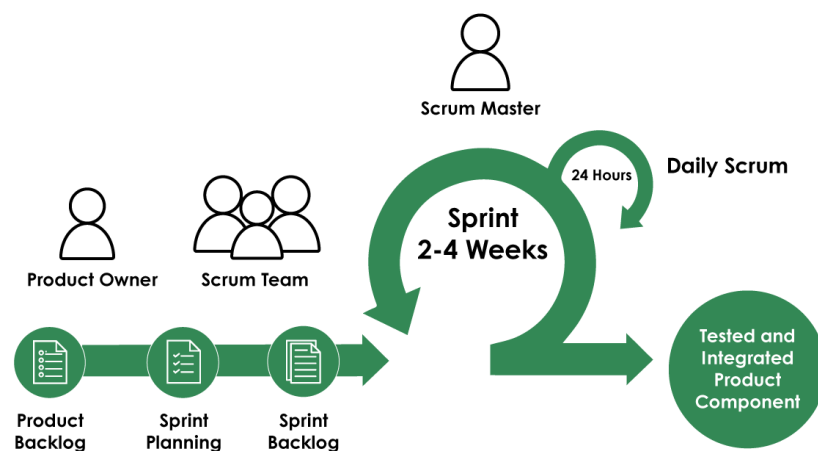


Figure 5: SCRUM Process Diagram, from [5]

5.1.2 Benefits of the Methodology

Scrum emphasizes collaboration, flexibility, and the ability to respond quickly to changes, allowing teams to adapt rapidly to new requirements and deliver high-quality products in less time [65].

- **Flexibility:** Allowed quick adjustments to changes in requirements and priorities, effectively re-

sponding to the project's needs;

- **Collaboration:** Promoted constant and efficient communication among team members;
- **Transparency:** The use of Jira ensured complete visibility into the status and progress of tasks;
- **Continuous Improvement:** The review meetings enabled reflection on sprint performance and identification of areas for improvement, increasing the efficiency and effectiveness of processes over time.

5.2 Data Exploration

In this section, an analysis and exploration of the provided dataset is conducted, and the assumptions considered are presented.

5.2.1 Description of the Data

The provided datasets are JSON files that represent the elements that constitute the network:

- **Start Equipment:** The SJ used to split the signal into several different directions within the network;
- **Locations:** The geographical coordinates (latitude, longitude) of the different network equipments and the clients homes;
- **Routes:** possible paths to connect the different elements of the network (existing structures where the distribution cable is placed).

Start Equipment			
Column	Description	Type	Nullable
name	name of the equipment	string	false
locationUID	location id	string	false
x	longitude	float	false
y	latitude	float	false

Table 1: Start Equipment Data Dictionary

Locations				
Column	Description	Type	Nullable	Comments
type	catalog type description	string	false	either installation point or homes
typeKey	key that represents the catalog type	string	false	
clientUnits	number of clients per home	number	true	installation points or homes
sduAvailableTakeOff	boolean that indicates if that location is suitable for a SDU	boolean	false	
name	location name	string	false	
x	longitude	float	false	
y	latitude	float	false	
uid	location unique id	string	false	

Table 2: Locations Data Dictionary

Routes				
Column	Description	Type	Nullable	Comments
type	catalog type description	string	false	Buried, No Support, Supported
typeKey	key that represents the catalog type	string	false	BRRD, NSUP, SUPP
name	route name	string	false	
userLength	route length	float	false	
locationA	location of the start of the route	string	false	
locationZ	location of the end of the route	string	false	
uid	route unique id	string	false	

Table 3: Routes Data Dictionary

As we can see in Table 2, the locations contain the number of client units, which represents the demand of each customer. If it is a building, it can be considered an Multi-Dwelling Unit (MDU) and, therefore, there is usually a higher demand for fiber. If it is a house, it can be considered an Single-Dwelling Unit (SDU) and, normally, the demand is often one.

In the Table 3 the routes are specified. There are three types: Buried, No Support, Supported. A cost

per meter is assigned to each one. The Buried is the most expensive, followed by the Supported, and finally, the cheapest is the No Support. It is important to consider the type of route chosen to develop a network optimized for efficiency and cost.

The Figure 6 is a visual representation of the initial state of the network and a complement to what has been presented in this subsection. Here is its caption:

- Starting Point - **Red node**;
- Homes SDU - **Orange nodes**;
- Buildings MDU - **Purple nodes**;
- Network equipment - **Blue nodes**;
- Possible routes - **Red edges**.

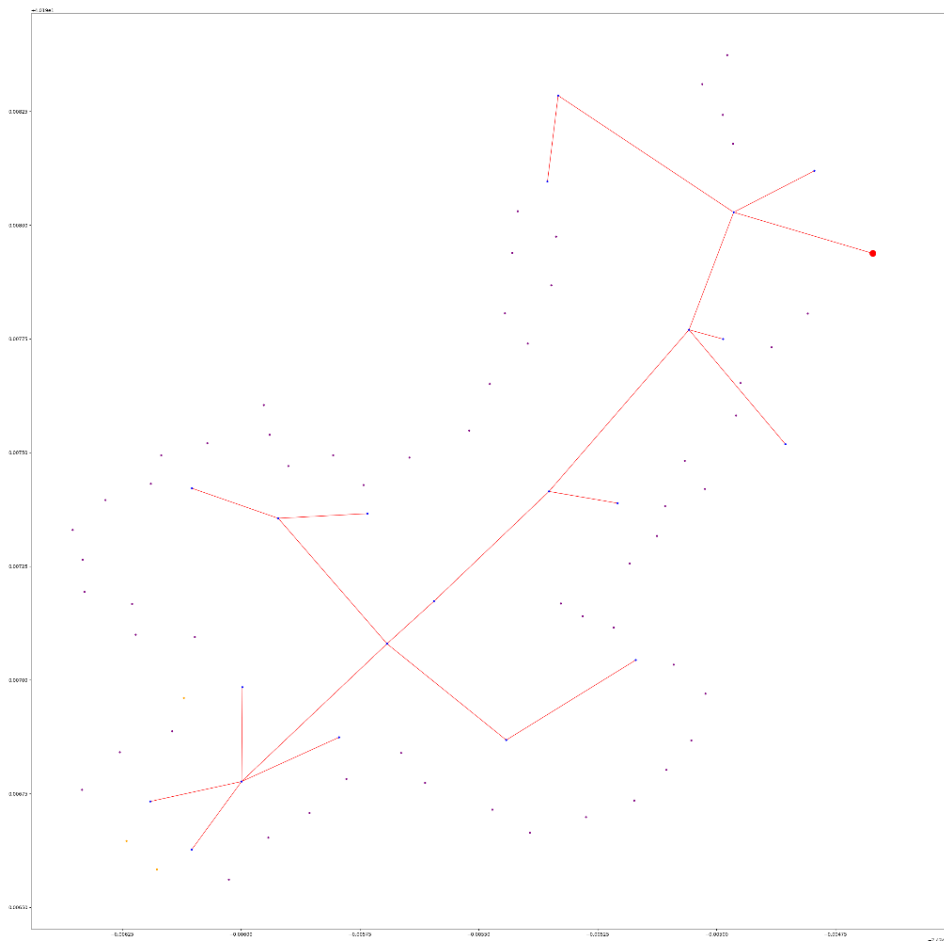


Figure 6: Example of the initial state of a network

5.2.2 Assumptions and pre-processing of the Data

These datasets represent actual networks or parts of them, and they may contain inconsistencies. For instance, there might be network equipment nodes (like poles or pedestals) that are not connected to the main network. To handle this, certain assumptions are made:

- Isolated network equipment is discarded, focusing only on the connected components of the network, corresponding to the sub-graph containing the starting point;
- The only infrastructure considered for addition is the drop cables connecting homes to the network equipment;
- Geographical coordinates are used, and the Haversine metric calculates the angular distance between points on a sphere.

If needed, the starting point (splitter joint) can be computed by considering the influence of homes. Dijkstra's algorithm is used to find the center of the graph based on the shortest paths between network equipment nodes and homes, choosing the point that minimizes the overall distance.

5.3 Initial State of the System

The process begins by loading essential base data, including routes, homes, and installation points, which are represented as a graph. This graph forms the basis of the entire design process. To enhance efficiency, the original graph undergoes a pre-optimization phase using the Steiner tree algorithm. This algorithm can obtain a minimum cost tree that covers only a specific set of terminal nodes (in this case, the clients' houses). This step reduces the graph to its most critical connections, minimizing unnecessary paths and simplifying the subsequent design stages. The Figure 7 is a visual representation of the application of the Steiner Tree to the initial state network example. Here is its caption:

- Starting Point - Red node;
- Homes SDU - Orange nodes;
- Buildings MDU - Purple nodes;
- Network equipment - Blue nodes;
- Covered routes with distribution cable - Red edges.

- Drop cable used to connect the ONU to the TP - Green edges.
- Unused routes - Black edges.

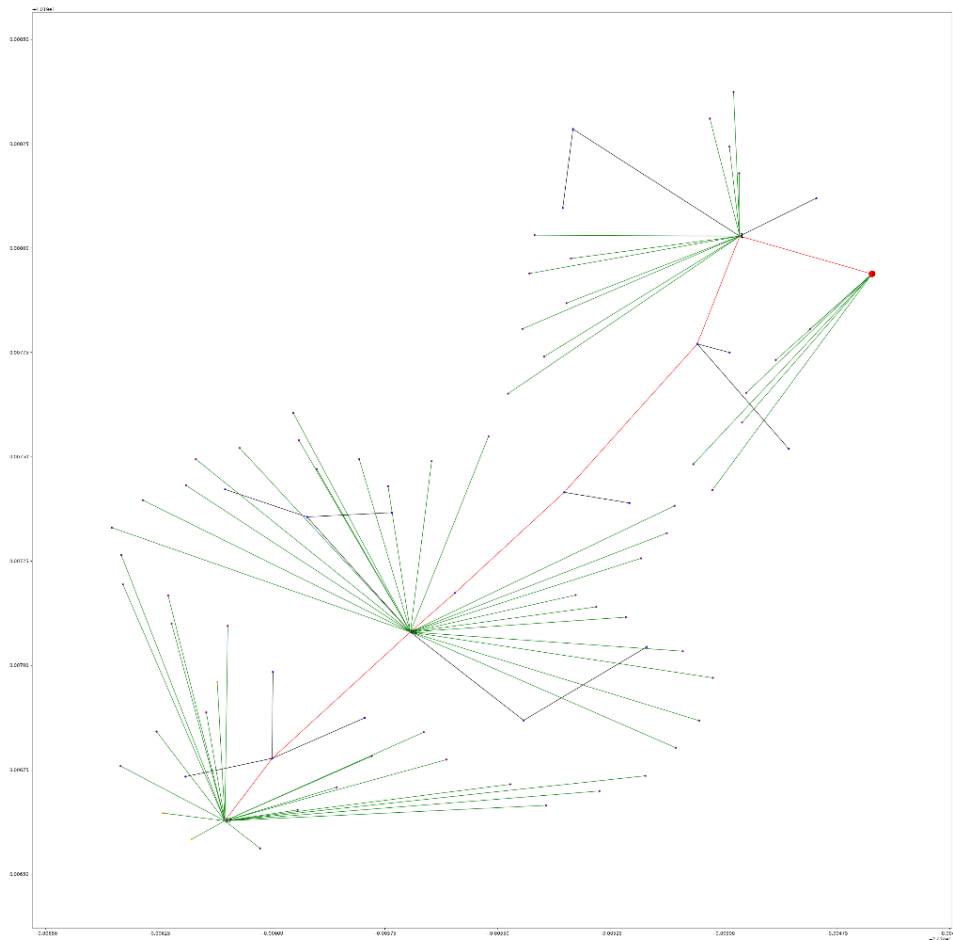


Figure 7: Result of the application of the Steiner Tree to the initial state of a network

To recap, as Steiner Trees are crucial for minimizing the total cost of connecting a specific subset of nodes in the network, an essential aspect of efficient network design, since it is not necessary to take into account every node in the network. As highlighted in the subsection 3.1.3, this problem is NP-complete, and exact solutions are computationally intensive for large graphs. To address this complexity, the initial solution employed GA, which are heuristic methods well suited to finding near-optimal solutions within a reasonable timeframe.

GA are inspired by the process of natural selection and iteratively evolve a population of solutions toward better fitness. In the context of network planning, this involves generating potential designs, evaluating their costs, and iteratively improving them. To perform this iterative process, techniques such as selection, crossover, mutation, and selection are used. This method effectively navigates the vast solution space

and is well-suited for this problem, as it is the most efficient when applied to nonlinear problems with constraints involving continuous or discrete design variables.

Thus, this approach aligns with the theoretical conclusions discussed in the previous sections. In the section 3.1.3, it was concluded that metaheuristic algorithms is effective in finding optimal or near-optimal solutions for complex optimization problems, such as the Steiner Tree, because it determines the optimal route from all possible ones [66]. Furthermore, in the section 4, it was concluded that GA is the most suitable for this problem for the reasons mentioned in the previous paragraph.

5.4 Genetic Algorithm

With the refined graph, a genetic algorithm is employed to generate the most suitable network configuration based on predefined objectives such as cost, efficiency, and coverage. The algorithm iteratively evolves potential designs, selecting and refining solutions that best meet the criteria. Once the optimal design is identified, the resulting graph is post-processed, during which additional information—such as details regarding the optical budget and network component quantities—is incorporated into the nodes. This section will offer an overview of the developed system, highlighting the key algorithmic design decisions that were taken into account.

5.4.1 Overview

The design of networks involves numerous constraints, requiring a flexible and robust approach. Given the need to explore a vast solution space and gradually evolve optimal designs, a GA is considered ideal. This optimization technique excels at iteratively refining solutions over successive generations, ultimately producing high-performance results.

The Genetic Algorithm (GA) implementation follows some aspects, in the network design process, namely:

- **TP Placement:** Identify optimal positions for network equipment, specifically termination points, to connect homes. Termination points have limited ports;
- **Route Considerations:** Evaluate routes or edges to minimize resource usage, avoiding expensive buried routes whenever possible;
- **Client Demand:** Account for each client's demand (1 unit corresponds to 1 allocated port). If buildings are present, place termination points on-site for practicality and reduced drop cable usage;

- **Optimal Drop Cable Links:** Determine the best drop cable links considering the capacity of termination point ports;
- **Distance Constraints:** Impose maximum distances for drop cables (restricting cable usage) and distribution cables (limiting network size and adhering to optical budget);
- **Material Costs Impact:** Consider weights (material costs) influencing multiple configurations in the design;
- **Intersection Constraints:** Address constraints related to intersections, ensuring drop cables do not intersect with distribution cables. Optional but relevant, especially for aerial cables;

These features and constraints guide the GA in creating an efficient and cost-effective network design, considering both practical and technical considerations. The workflow of the evolutionary process is summarized in the Figure 8.

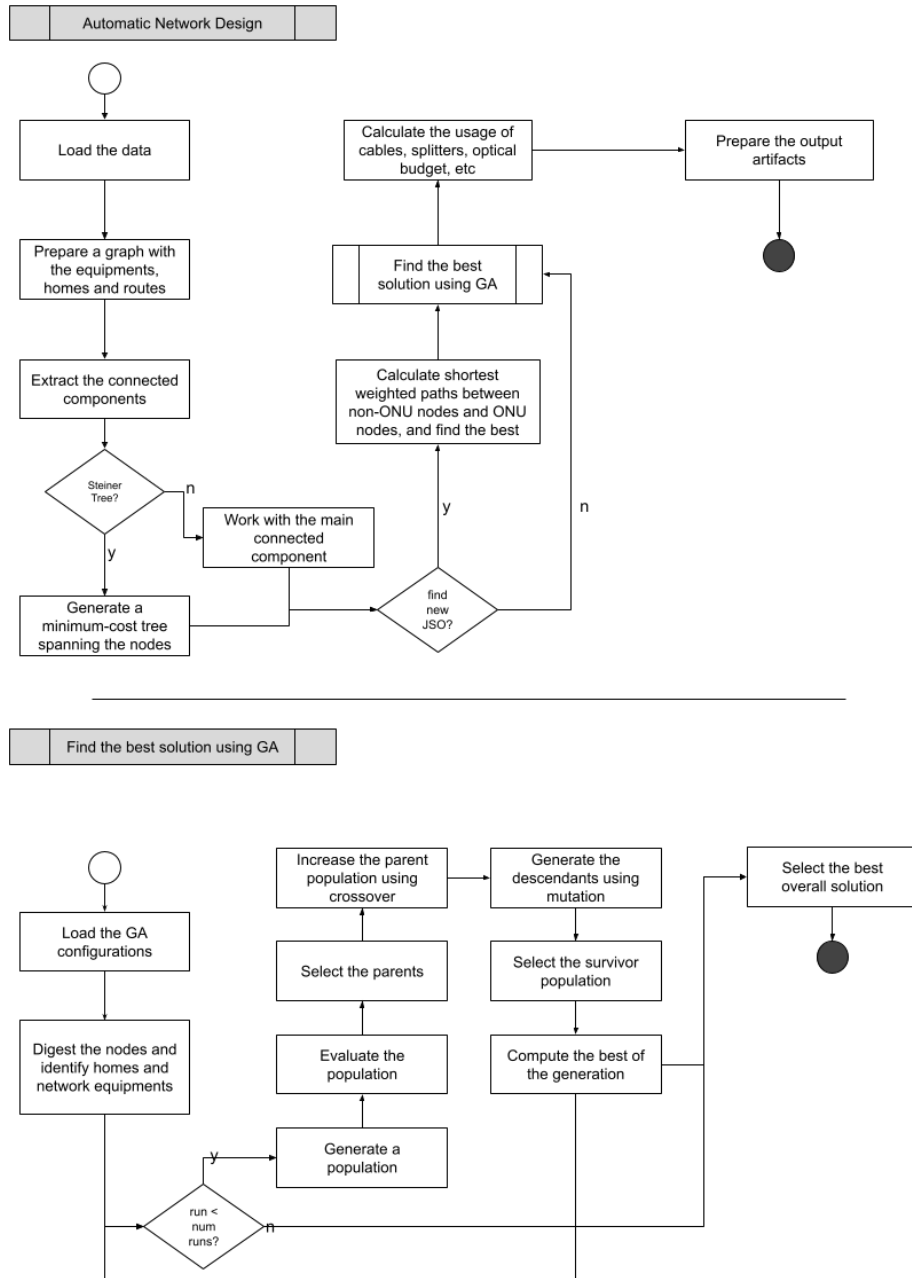


Figure 8: Automatic Network Design Flow Chart

5.4.2 Network Design

To effectively apply the GA in network design, it is essential to first understand the fundamental components of the GPON architecture considered by Altice Labs, which will serve as the basis for developing realistic and practical designs. These components are:

- **SJ**: Starting node of the secondary network;
- **Distribution cable**: Cable that carries optical signals from the SJ to distribution points near end users (TP);
- **TP**: Components that connect the drop cable infrastructure at the client end to the main distribution network;
- **Drop cable**: Cable that connects the TP to the client's home;
- **SDU**: Single residential unit;
- **MDU**: Building with multiple residential units;
- **Installation points**: Locations where network equipment, such as poles, can be installed.

These components are carefully integrated into the GA framework to ensure the final network design is practical and matches GPON standards. Figure 9 illustrates the GPON network configuration considered by Altice Labs, providing a visual representation of the key elements discussed.

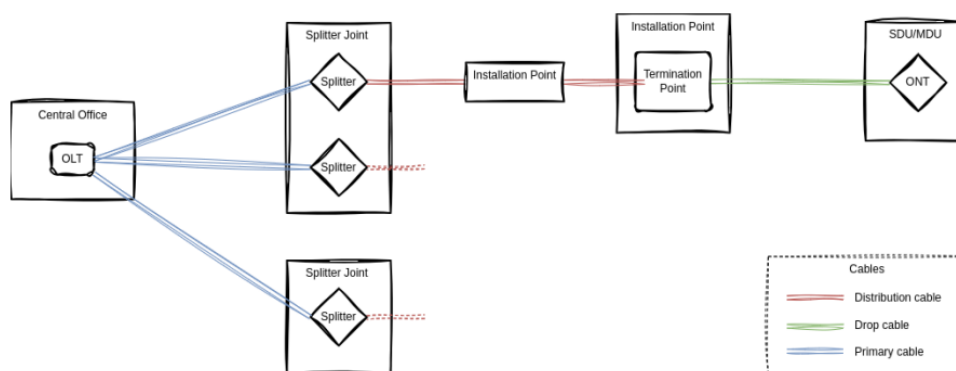


Figure 9: GPON network considered by Altice Labs (Source: Internal documents, Altice Labs)

To apply a GA in network design, the problem first needs to be translated into a form that it can be processed. This involves transforming each potential network configuration into a representation known as a chromosome. In this context, a chromosome is a sequence of information that describes specific design

options for the network. Each part of this sequence, called a "gene," corresponds to a design decision, such as selecting a route or a connection point.

The core idea behind this approach is that, by using operations like mutation (randomly modifying parts of the solution), selection (choosing the best solutions to move forward), and crossover (combining two different solutions to generate new ones), the algorithm can explore a wide range of possible solutions. Over time, this helps refine the network configurations and find the one that best meets the given criteria.

In the specific case of a distribution network, the algorithm works with a dataset that contains all the possible cable distribution paths, as explain in the chapter 5.2. The GA's ultimate goal is to determine which connections between homes and the main network are the most efficient. This is achieved by encoding the possible locations for termination points (where the cables connect) into a chromosome. Each home is then assigned to one of these termination points, and the process of mutation, selection, and crossover is repeatedly applied until an optimized solution is found. Figure 10 provides a visual representation of this chromosome.

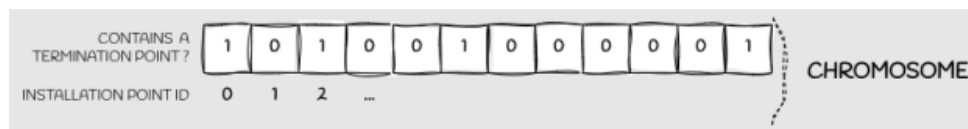


Figure 10: Representation of a chromosome in the implemented solution (Source: Internal documents, Altice Labs)

The network design process begins with the creation of a chromosome that defines the locations of termination points. This chromosome is refined through genetic operations to ensure efficient positioning. After that, an auxiliary map is generated to allocate homes to these termination points. This allocation follows strict criteria, such as appropriately sizing the termination points in relation to the number of homes they will serve and adhering to the capacity and length constraints of the drop cables. By combining genetic and heuristic methods, an optimized and efficient network is achieved.

Once the termination points are determined, an algorithm seeks to identify the types and quantities of cables required to complete the network design. This step involves calculating the number of fibers in each segment, ensuring that all homes are connected to the main network. The algorithm also calculates the amount of drop and distribution cables needed, considering the specific network requirements and configurations. This process ensures that the network is viable, both in terms of implementation and cost, while optimizing resource usage. The fitness function ensures that the best individuals (i.e, best solutions obtained in this step) evolve throughout the process. Figure 11 provides a visual representation of this association map.

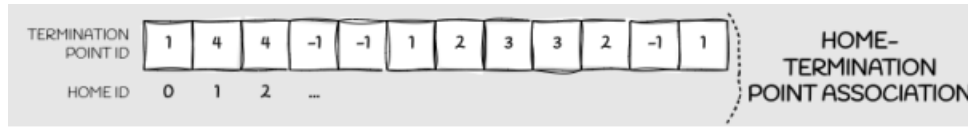


Figure 11: Representation of the home-termination point association in the implemented solution (Source: Internal documents, Altice Labs)

In summary, each individual of the Network Design GA implementation contains 2 chromosomes, a map of the specific termination points used, a map associating homes with cables and the fitness value:

- **chromosome_1**: binary array, representing the placement of termination points in the network equipments; each equipment is defined by an array index, and a value of 1 means that it contains a termination point;
- **chromosome_2**: integer array, mapping the association between a customer's home and a termination point; each index represents a customer's home, and the value represents the termination point linked to it; -1 is used when no termination point can be linked to the home;
- **termination_point_details_map**: a dictionary mapping the termination point ID with the specific details needed;
- **home_cable_details_map**: a dictionary mapping the home IDs with the specific cable details needed;
- **fitness value**: a numerical score representing how well the current network design meets the pre-defined objectives and constraints, guiding the optimization process.

5.4.3 Fitness Function

The fitness function is a fundamental component in GA, used to evaluate the quality of each proper solution obtained during the population generation process. It assigns a numerical value to each individual in the population, allowing the solutions to be ranked based on how well they meet the defined criteria and constraints for the problem.

In practical terms, the fitness function measures the performance of a network design by evaluating the total amounts of drop and distribution cables used, the number of clients allocated and not allocated to a termination point, the types and quantities of termination points deployed, and the splitters adopted. It also represents a measure of the network's efficiency and effectiveness, with the GA aiming to minimize this value over the generations an runs, as illustrated in figure 12.

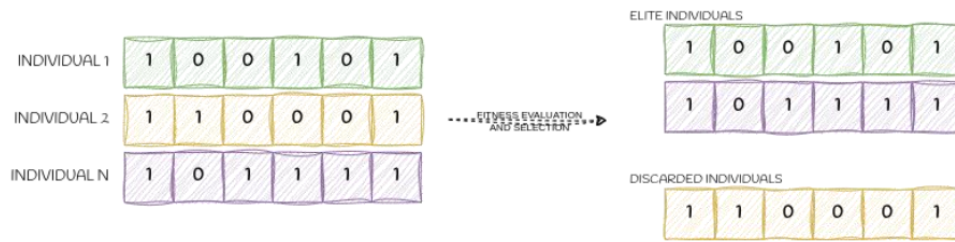


Figure 12: Fitness evaluation and selection over a set of individuals (Source: Internal documents, Altice Labs)

The solution with the lowest fitness value is considered the optimal design and is returned as the final output of the algorithm. This approach ensures that the network design meets the technical requirements and minimizes costs.

How it was obtained

To develop the fitness function, a review of different research papers was conducted, analyzing all the formulas presented. Subsequently, the key elements from each of the fitness functions were combined to identify the most commonly considered factors across all formulas. The results of this review are shown in Table 4.

Each fitness function presented in the table corresponds to formulas found during the research that were compared. Here is the table caption:

- Formula 1 - Parameters considered by Altice Labs to evaluate handmade solutions;
- Formula 2 - Parameters considered according to [53];
- Formula 3 - Parameters considered according to [56]
- Formula 4 - Parameters considered according to [7]

Costs	Formula 1	Formula 2	Formula 3	Formula 4
Drop Cable	Yes	Yes	Yes	Yes
Distribution Cable	Yes	Yes	Yes	Yes
Labour costs	No	Yes	Yes	No
Fusion splicing cost	No	Yes	Yes	No
Total cost of splitting	No	Yes	Yes	Yes
Splitting attenuation	No	Yes	Yes	No
OLT cost	No	No	Yes	No
ODF cost	No	No	Yes	No
ONU cost	No	No	Yes	No
PDOs Cost & N^o	Yes	No	Yes	No
Optical Splitter Cost	No	Yes	Yes	Yes

Table 4: Fitness value formulas

After this step, this research was presented to the product manager. Since he has full knowledge of the business, he needed to decide what should or should not be considered by Altice Labs in the design and implementation of fiber optic networks. As a result, the fitness function is calculated with this formula:

$$\text{FITNESS_VALUE} = \text{drop_cable_cost} * \text{drop_cable_length} + \text{distribution_cable_cost} * \text{distribution_cable_length} + \text{termination_points_total_cost} + \text{splitters_total_cost}$$

How it was used

The fitness function's value reflects the efficiency and cost-effectiveness of the network design, with lower values indicating superior solutions. In essence, the goal is to minimize the fitness value, as this implies a more cost-effective network design that still satisfies all necessary requirements.

The fitness function in the Netwin Automatic Network Design system considers five critical factors, each contributing to the overall cost and effectiveness of the network:

1. **Drop Cables:** These are the cables that connect individual homes to the larger network. The fitness function considers both the length and cost per meter of these cables. The longer and more expensive the drop cables, the higher the fitness value, which indicates a less optimal solution. Therefore, minimizing the length and cost of drop cables is crucial for reducing the overall network cost.

2. **Distribution Cables:** These cables are used to distribute the signal from the central junction,

often referred to as the Joint Splitter Optic (JSO), to various points in the network. Like the drop cables, the fitness function evaluates the length and cost per meter of the distribution cables. Efficient use of distribution cables, by optimizing their routing and minimizing their length, can significantly lower the fitness value.

3. **Termination Points:** Termination points are critical components responsible for distributing fibers from the network to individual homes. The fitness function takes into account both the quantity and cost of these termination points. Since termination points are expensive, their efficient placement and minimal usage are essential for achieving a lower fitness value.

4. **Splitters:** Splitters are devices used to divide the optical signal into multiple branches, allowing it to reach more homes. The fitness function evaluates the number and cost of splitters used in the network. An optimal design strategically places splitters to minimize their number and cost while ensuring adequate signal distribution, thereby reducing the fitness value.

5. **Unallocated Homes Penalty:** The fitness function imposes a significant penalty for each home that the algorithm fails to allocate to the network. This penalty ensures that the algorithm prioritizes connecting as many homes as possible, as failing to do so results in a dramatic increase in the fitness value. This factor is crucial in ensuring that the network design not only remains cost-effective but also meets coverage requirements.

5.4.4 Variation Operators

Variation operators are mechanisms used in GAs and other evolutionary computation techniques to modify the population of solutions (individuals) across generations. They are essential for introducing diversity and guiding the search toward optimal or improved solutions. The three main types of variation operators are crossover, mutation and selection (two types of selection were used).

Parents selection

Selection is a crucial step in genetic algorithms that determines which individuals from a population will be used for crossover and mutation. After a population has been generated, a subset of individuals is selected based on their fitness, with the goal of passing on the best traits to the offspring. The selection process ensures that the most promising individuals have a higher chance of contributing their genetic material, improving the overall quality of the population over time. In this step the parents are selected. Several strategies exist for performing selection, each with distinct advantages and biases.

- **Roulette wheel selection:** Individuals are selected with a probability proportional to their fitness,

where fitter individuals have a higher chance of being chosen;

- **Rank selection:** Individuals are sorted based on their fitness, and their selection probability is assigned according to their rank, avoiding biases from large differences in fitness;
- **Tournament selection:** A small group of individuals is randomly chosen, and the fittest among them is selected to be a parent.

Tournament selection was chosen for this algorithm due to its simplicity and efficiency. It iterates through the population, randomly selecting individuals for each tournament, and chooses the individual with the best fitness value as the winner.

Roulette wheel selection can be biased towards individuals with high fitness, as it relies on fitness-proportionate selection. This makes it sensitive to large fitness differences, potentially leading to overly dominant individuals and reducing diversity in the population. In contrast, tournament selection is less sensitive to such imbalances and allows for better control over selection pressure [67].

Tournament selection allows for easy adjustment of selection pressure by changing the tournament size. Larger tournament sizes increase pressure, while smaller sizes decrease it, offering a flexible balance between exploration and exploitation [68], which is ideal for the given problem as different tournament sizes can be used depending on the network and goals.

Rank selection requires sorting the population based on fitness, which introduces an additional computational overhead. Tournament selection avoids this by selecting a subset of individuals and only comparing them, making it simpler and more efficient to implement [69].

Crossover - increase of the parent population

Crossover combines two or more parent solutions to produce one or more offspring solutions. The idea is to mix the features of the parents in such a way that the offspring inherit useful traits from each parent, in order to increase the parent population. This is crucial for creating new combinations of genes and exploring new areas of the solution space [70]. There are several types of crossover:

- **Single-point crossover:** One crossover point is selected, and the parents exchange parts of their solution after this point;
- **Multi-point crossover:** Multiple points are selected, leading to more complex recombination;
- **Uniform crossover:** Each gene (or part of the solution) is randomly chosen from one of the parents.

The crossover operator is implemented through the uniform crossover function, illustrated in figure 13. In this approach, the algorithm takes two parent chromosomes and creates offspring by combining genes from both parents. Unlike other crossover methods that might involve splitting the chromosome at a specific point, uniform crossover evaluates each gene position independently, deciding with a certain probability whether the offspring will inherit the gene from the first parent or the second parent. This method ensures that the offspring has a more diverse combination of traits from both parents, thereby maintaining a broad exploration of the solution space. In the context of this work, uniform crossover allows for the generation of new network configurations by mixing and matching termination point allocations from different parents. This operation enhances the genetic diversity of the population and helps the algorithm to more effectively search for better solutions by exploring a wider range of potential configurations.



Figure 13: Representation of the uniform crossover over the parent chromosomes (Source: Internal documents, Altice Labs)

Mutation - descendants generation

Mutation introduces random changes to individuals by altering one or more of their genes. It helps maintain genetic diversity within the population and can introduce entirely new solutions that might not be achievable through crossover alone. It is also key to avoiding premature convergence and ensuring exploration of the solution space [71], [72]. There are several types of mutation:

- **Bit-flip mutation:** In binary representations, a bit's value is flipped (0 to 1 or vice versa);
- **Swap mutation:** Involves swapping two genes in the chromosome;
- **Gaussian mutation** (in real-valued individuals): Alters a gene value by a small amount, often according to a Gaussian distribution.

The mutation operator is implemented using a bit-flip mutation technique, as shown in figure 14. This method involves altering the value of specific genes within the chromosome, flipping them from zero to one or vice versa. In this work, each bit in the chromosome represents the allocation of a termination point at a particular installation point. A value of one indicates that a termination point is allocated to that location, while zero suggests that it is not. During the mutation process, the algorithm randomly selects

bits in the chromosome and flips their values. The mutation is crucial for maintaining genetic diversity within the population and helps to prevent premature convergence on suboptimal solutions.



Figure 14: Representation of the bit-flip mutation over a parent chromosome (Source: Internal documents, Altice Labs)

Selection of the survivor population

After the mutation process, a selection of the best individuals generated must be carried out to form the next generation's population. This process is fundamental to ensure that the most promising solutions from the current generation, including both parents and offspring, are preserved and continue to evolve. The purpose of selection at this stage is to ensure that individuals with greater fitness, or those possessing advantageous traits, move on to the next generation, promoting continuous improvement in the quality of solutions.

In this implementation, elitism selection was employed. This selection type preserves the best individuals from both the current population and the offsprings, allowing them to be passed directly to the next generations. The algorithm user can choose the percentage they consider most appropriate to filter the best individuals from the populations.

The key idea behind elitism is to maintain a certain number of elite individuals (often a small percentage of the total population) to ensure that the overall quality of the population doesn't degrade over generations. These elite individuals act as a safeguard against losing the best solutions and help maintain a certain level of fitness within the population.

5.4.5 Allocating termination points to homes

The input data allows the representation of the initial network with the different network equipment in-place, but with no connection between the final nodes, as seen in figure 15. A two-step approach was developed to allocate homes to termination points: i) connecting each home to the closest free termination point and ii) optimizing the allocation using Hill Climbing.

The solution starts with the first option, and if any homes remain unallocated to their nearest termination point, they are flagged for further optimization using the complementary approach. The optimization then attempts to swap drop cable connections between homes and termination points to reduce costs

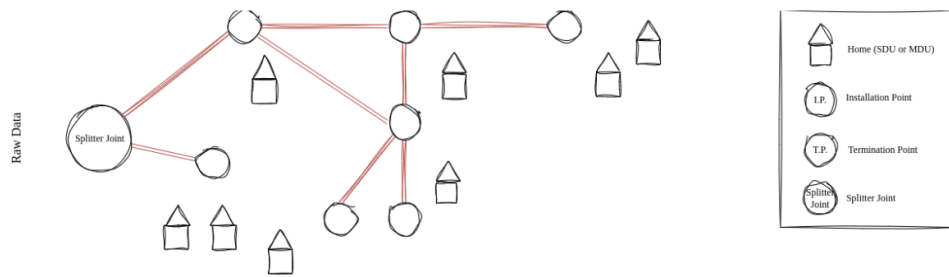


Figure 15: Initial network structure from the raw data (Source: Internal documents, Altice Labs)

while ensuring that all constraints—such as drop cable range and termination point port limits—are respected. The second option can be turned off in the configurations. This resulting network is illustrated in figure 16.

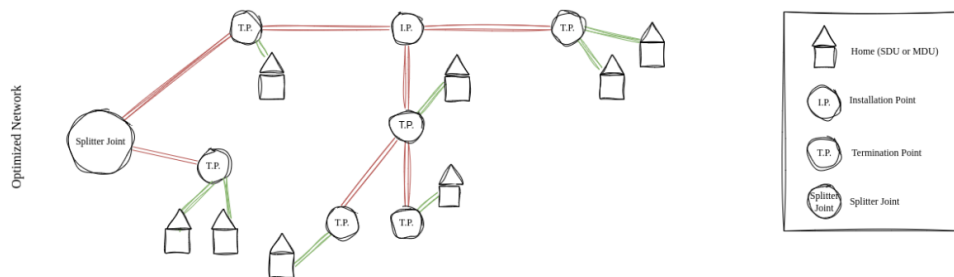


Figure 16: Optimized network with all the relevant connections (Source: Internal documents, Altice Labs)

An additional configurable feature introduced in the network design is the ability to place termination points directly inside homes, with an exclusivity option. This enhancement allows for nesting termination points within MDU and SDU homes, connected to the nearest installation point via distribution cables. The Genetic Algorithm can consider these new in-home locations by enabling this feature when optimizing network connections. The exclusivity setting ensures that termination points within specified homes are dedicated solely to those units, preventing sharing with others and thereby maintaining higher service quality.

Figures 10 and 11 illustrates the initial raw scenario, showing the possible routes, all installation points on the map, and the homes disconnected from the network. It is followed by the optimized network, where unnecessary paths have been removed, and homes are connected to appropriate termination points placed at selected installation points.

5.5 HTTP Request

In this section, the development of a web service focused on creating and evaluating network design solutions will be presented. The project was developed with a focus on performance and scalability, utilizing technologies and frameworks widely adopted in the industry. The service architecture was designed to allow easy integration with external systems, ensure data consistency, and meet specific validation requirements. The development emphasized simplicity and robustness, with a clear code structure and proper handling of HTTP requests.

HTTP Service

The HTTP service was implemented using the Flask framework, which provides a simple and efficient interface for creating web servers. This server exposes two main resources via REST endpoints:

1. **/design**: This endpoint handles POST requests for network design, where the request body is validated, processed, and an optimized solution is generated. Figure 17 is a sequence diagram that represents the design flow.

2. **/evaluate**: This endpoint also handles POST requests, where manually created solutions are evaluated and compared to the automated solutions. Figure 18 is a sequence diagram that represents the solution evaluation flow.

Both endpoints follow RESTful practices, returning appropriate HTTP status codes, such as 200 for success, 400 for validation errors, and 500 for unexpected errors.

Development in Python

The development was carried out in Python, with a class-oriented architecture, where each part of the code was organized to fulfill specific functions. For the business logic, two main classes were implemented: `NetworkDesignProcessor`, responsible for network design, and `HandmadeSolutionEvaluator`, for evaluating manual solutions. The separation of responsibilities also makes the code more maintainable and scalable.

Additionally, Flask was chosen as the web framework due to its simplicity and flexibility in developing REST APIs.

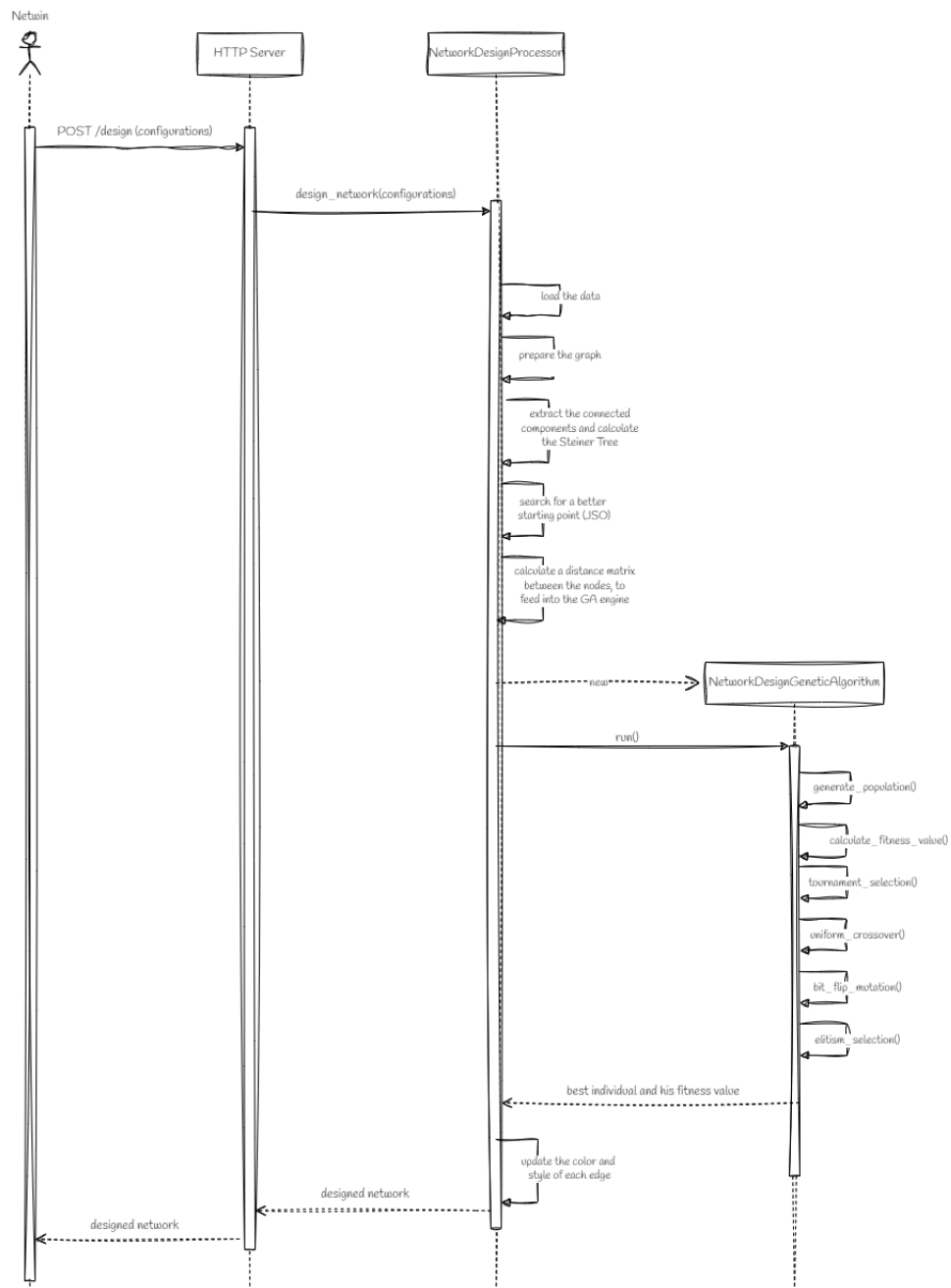


Figure 17: Network Design Flow

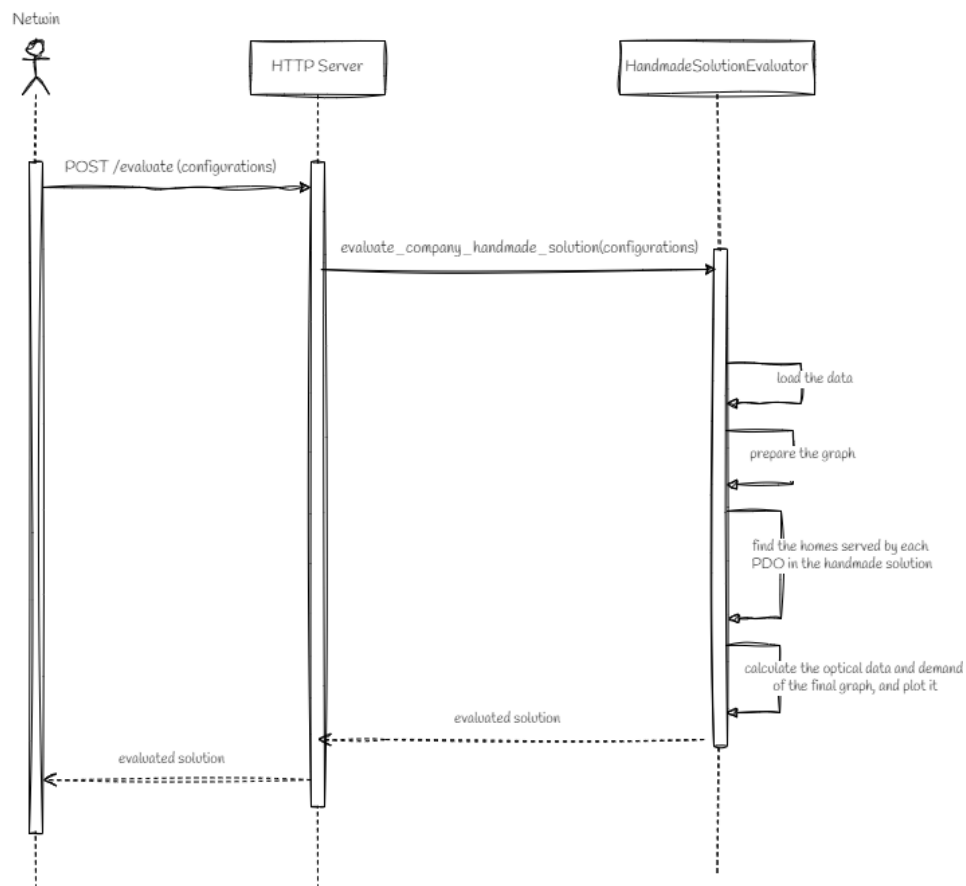


Figure 18: Handmade Solution Evaluation Flow

Response Data Structure

Requests are processed in compliance with a specific data structure defined in the code. Each endpoint validates the request body through JSON schemas using the jsonschema library, and in case of validation errors, a clear and descriptive message is returned. This ensures that the received data is always in the correct format, avoiding errors during processing.

Design Parameters - HTTP request body used to design a network

The HTTP request is responsible for providing the existing data regarding the network components, penalties, routes, and optical budget, and allows the ability to tune the design in different ways. A brief description is presented below, with an example value.

Network

- maxLength: Maximum length of the entire network, in meters (20000 m).
- dataset: Dataset identifier.
- centralToSplitterJointLength: Distance from the central office to the Splitter Joint (0 m).

NetworkEquipments

- cables
 - available:
 - * networkType: Type of network (e.g., "drop", "distribution").
 - * numFibers: Number of fibers in the cables (e.g., 1, 2, 32, 64, 144, 288).
 - * cost: Cost per meter of the cable (e.g., 2, 4, 25, 50, 100, 200 monetary units).
 - constraints
 - * length: Maximum length constraints for different network types.
 - * dropCableLimitExtension: Conditions for drop cable limits.
- terminationPoints
 - available:
 - * capacity: Amount of client units the termination points can handle (e.g., 12, 24).
 - * cost: Cost of each termination point (e.g., 300 monetary units).

- keepFreeMargin: Percentage of capacity to be kept free as a margin (e.g., 0.075).
 - exclusiveTypeKeys: Array of exclusive type keys (e.g., ["key_1", "key_n"]).
- splitterJoint
 - splitterRatio: Ratio of the optical splitter (e.g., 64).
 - cost: Cost associated with the splitter joint (e.g., 1 monetary unit).
- connectors
 - includeCost: Indicates whether connectors are included in the cost calculation (e.g., FALSE).
 - cost: Cost per connector (e.g., 1 monetary unit).

Routes

- type: Type of route (e.g., "buried", "supported", "no support").
- cost: Cost associated with each type of route (e.g., 10, 1.5, 2 monetary units).

Penalties

- type: Type of penalty (e.g., "mdu", "unallocated").
- cost: Cost associated with each type of penalty (e.g., 1, 300000 monetary units).

OpticalBudget

- cableAttenuation: Attenuation per kilometer of optical cable (e.g., 0.35 dB/km).
- splitterAttenuation: Attenuation introduced by the optical splitter (e.g., 13.8 dB).
- connectorAttenuation: Attenuation per connector (e.g., 0.4 dB).
- attenuationMargin: Additional margin for attenuation (e.g., 7 dB).
- maxValue: Maximum permissible total attenuation (e.g., 28 dB).

5.6 Configurations

The configuration file contains various parameters and settings for the network design process. The different sections of the file are explained below.

HTTP Server

- HTTP_PORT: Specifies the port number for the HTTP server (8080).

Data

- DATASETS_DIR: Directory path for dataset storage ('../res/data/alice_json_datasets/').
- OUTPUT_BASE_DIR: Base directory path for network design outputs ('../res/outputs/').
- HANDMADE_SOLUTIONS_FILE_PATH: Path to the file containing handmade solutions data ('../res/data/alice_json_datasets/results_handmade.xlsx').

Network Design

- USE_STEINER_TREE: Boolean indicating whether to use the Steiner tree algorithm (True).
- SEARCH_OPTIMAL_SPLITTER_JOINT_LOCATION: Boolean indicating whether to search for an optimal splitter joint location (False).

Genetic Algorithm Parameters

- RANDOM_SEED: Seed for random number generation (2023).
- NUMBER_OF_RUNS: Number of runs for the genetic algorithm (1).
- POPULATION_SIZE: Size of the population in each generation (100).
- NUMBER_OF_GENERATIONS: Number of generations (100).
- CROSSOVER_PROBABILITY: Probability of crossover (0.85).
- TOURNAMENT_SIZE: Size of the tournament for selection (5).
- ELITISM_PERCENTAGE: Percentage of the population treated as elites (0.1).
- MUTATION_PROBABILITY: Probability of mutation (-2).

Plotting

- `DISPLAY_OPTICAL_BUDGET`: Boolean indicating whether to display the optical budget (True).
- `SHOW_NODE_LABELS`: Boolean indicating whether to show node labels (True).
- Node sizes, colors, and edge styles for distinct types of nodes and cables.

5.7 Optical Budget

The optical budget is calculated for the existing termination point nodes in the network, considering factors such as the cumulative distance along the network branch, the length of the drop cable associated with the termination point, cable attenuation, splitter attenuation, connector attenuation, and an attenuation margin.

The calculated optical budget represents the remaining capacity for optical signals at the current node and is updated in the network graph's node attributes. This ensures that the optical budget is appropriately accounted for in the network design, considering signal loss and other relevant factors.

The optical budget is also used to prepare an output artifact (a bar plot), saved with the other network artifacts. The image 19 shows an example of this plot.

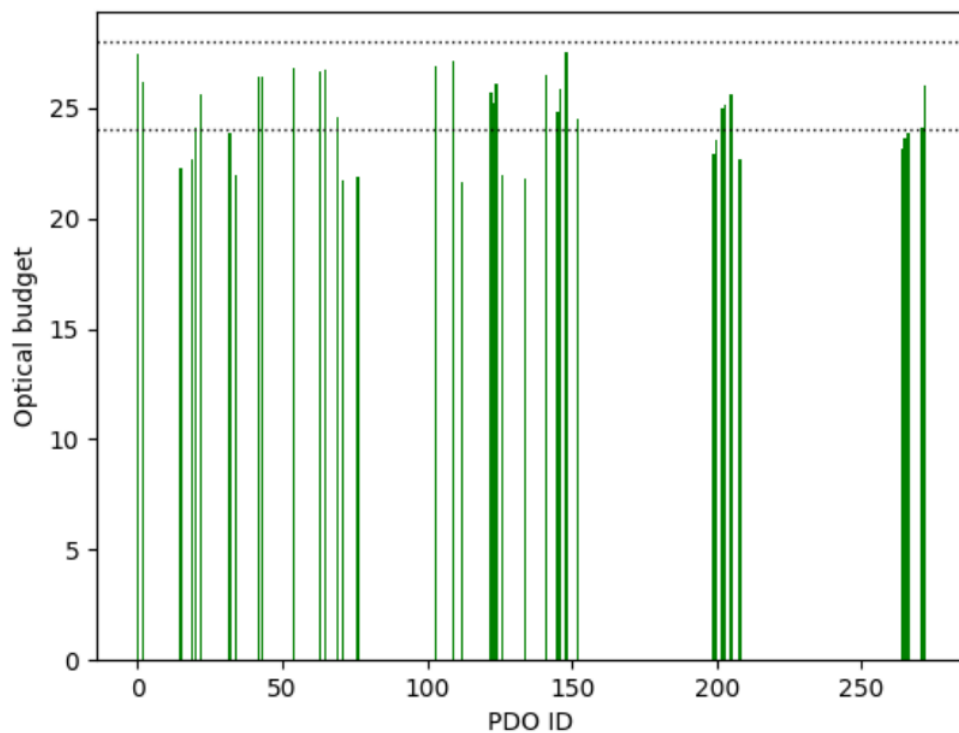


Figure 19: Optical Budget per PDO used in the solution

Chapter 6

Experiments And Results

This chapter involves the observation and analysis phases of the results. Firstly, the experimental setup used to execute the multiple tests with the provided maps will be presented. Then, the results obtained will be discussed.

6.1 Experimental Setup

This section defines the experimental setup to execute the multiple tests. It mentions the the computer specifications and the maps details. It is important to note that the GA parameters and design configurations used are the same as those presented in chapters 5.5 and 5.6.

6.1.1 Computer Specifications

This subsection will present the specifications of the hardware used. All the tests presented were performed on the same machine. The specs are:

- HP HP ProBook 440 14 inch G10 Notebook PC.
- 13th Gen intel® Core i5-1335U x 12.
- AMD Radeon Graphics.
- RAM: 32GiB.

6.1.2 Network and Algorithm Specifications

This subsection presents the network and algorithm specifications used. The network parameter values were provided by the product owner, while the GA parameters were chosen by the entire team using a test-and-try method. The values that produced the best results were ultimately selected.

Network parameters:

- **Drop Cables**

- **1 Fiber:** Cost = 2
- **2 Fibers:** Cost = 4
- **Maximum Length:** = 130

- **Distribution Cables**

- **32 Fibers:** Cost = 25
- **64 Fibers:** Cost = 50
- **144 Fibers:** Cost = 100
- **288 Fibers:** Cost = 200
- **Maximum Length:** = 20000

- **Termination Points**

- **12 capacity:** Cost = 300
- **24 capacity:** Cost = 300

- **Splitter Joint**

- **64 splitter ratio:** Cost = 1

- **Connectors** (flag is false, therefore they are not considered)

- **includeCost:** false
- **Cost:** 1

- **Routes, penalties and optical budget**

- The same values as those defined in subsection 5.5 were considered.

GA parameters:

- **Random Seed:** 2023;
- **Number of Runs:** 1;
- **Population Size:** 100;
- **Number of Generations:** 100;
- **Crossover Probability:** 0.85;
- **Tournament Size:** 5;
- **Elitism Percentage:** 0.1;
- **Mutation Probability:** -2 (if probability ≥ 0 , it is used as is; if < 0 , then calculated as $|\text{value}| / \text{length of } gene1$).

6.1.3 Networks Details

As mentioned in the section 5.2.1 , our product owner provided multiple networks (datasets in the JSON format) to test the system. They correspond to the handmade solutions. Table 5 details the available maps.

Dataset	Number of locations (nodes)	Total Clients: SDU & MDU	Sum of the distance of all routes (meters)
Network A	188	108	14013,89
Network B	82	60	3849,99
Network C	222	87	24911,68
Network D	273	85	32599,44
Network E	288	95	44826,48

Table 5: Details of the different handmade networks

It is important to note that the main objective of the experiments is to generate a solution for each map and compare it to a handmade solution. After that, the GA will be used to generate a new, improved solution for each dataset, with the primary goal of minimizing costs (fitness value).

6.2 Design Results

This section presents the design results for all datasets (networks) described in Table 5. To illustrate the effectiveness of the Netwin's Automatic Network Design system, we compare two network layouts: one created manually by a network designer (handmade solution) and another generated automatically by the genetic algorithm (generated solution).

6.2.1 Network A

The first map analyzed is "Network A". Figure 20 corresponds to the network layout created manually by a network designer and Figure 21 corresponds to the automatic solution generated by the GA.

As shown in Figure 20, Network A represents a manually designed network that, while functional, required many hours of labor to create, and still resulted in a high fitness value of 60091,86 monetary units. This high cost is due to the suboptimal placement of cables and especially termination points and splitters, resulting in a significant increase in the fitness value.

In contrast, Figure 21 showcases the optimized network layout generated by the genetic algorithm in less than a few minutes. The automated design process reduces the fitness value to an impressive 13351,66 monetary units, approximately a 78% decrease in cost. Table 6 establishes a comparison between handmade and generated designs. Network A took 12,93 seconds to be generated.

```

-- (handmade) --
EDGES
> [red] distribution cable (19956 m)
> [green] drop cable (2748.93 m)
> [black] unused route
> [dotted] buried route

NODES
> [grey] termination points (62 un.)
> [orange] SDUs (103 un.)
> [purple] MDUs (5 un.)
> [blue] free installation points (17 un.)
> [red] splitter joint

STATISTICS
> clients: 108
> penetration rate: 1.0
> unallocated homes: 0
> fitness value: 60091.86

```



Figure 20: Handmade design of Network A

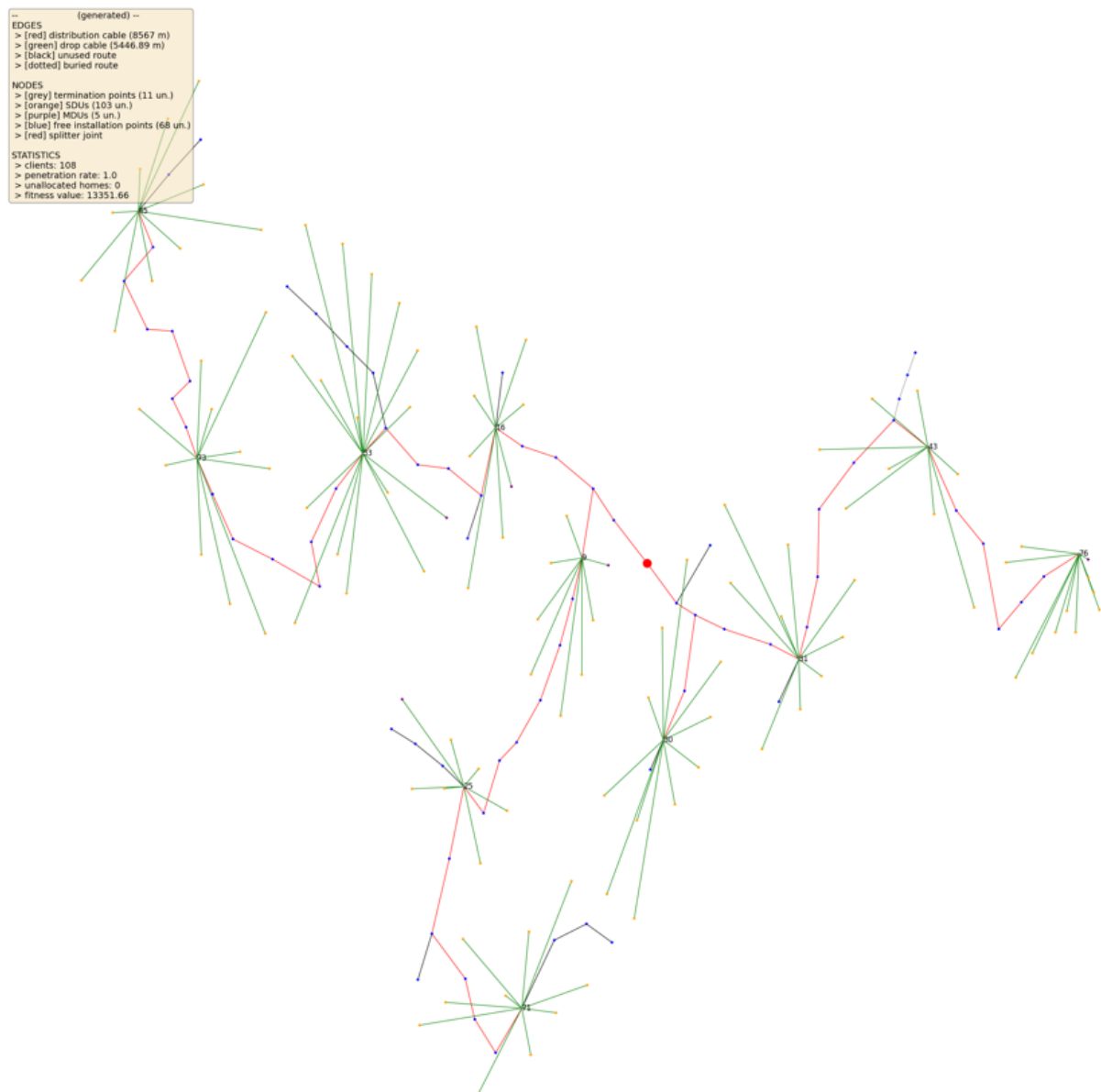


Figure 21: Generated design of Network A

Parameter	Handmade	Generated
Fitness Value	60091,86	13351,66
Pen. rate	1,00	1,00
Distrib. cable	19956,00	8567,00
Drop cable	2748,93	5446,89
# Term. Point	62	11
# SDU	103	103
# MDU	5	5
# Free inst. Point	17	68
# Clients	108	108

Table 6: Network A value comparison between handmade and generated designs

6.2.2 Network B

The second map analyzed is "Network B". Figure 22 corresponds to the network layout created manually by a network designer and Figure 23 corresponds to the automatic solution generated by the GA.

As shown in Figure 22, Network B has a high fitness value of 6110,6 monetary units.

Figure 23 showcases the optimized network layout generated by the genetic algorithm in less than a few minutes. The automated design process reduces the fitness value to an impressive 3752,85 monetary units, approximately a 38,58% decrease in cost. Table 7 establishes a comparison between handmade and generated designs. Network B took 8,01 seconds to be generated.

-- (handmade) --

EDGES

- > [red] distribution cable (2690 m)
- > [green] drop cable (1250.99 m)
- > [black] unused route
- > [dotted] buried route

NODES

- > [grey] termination points (11 un.)
- > [orange] SDUs (3 un.)
- > [purple] MDUs (57 un.)
- > [blue] free installation points (10 un.)
- > [red] splitter joint

STATISTICS

- > clients: 60
- > penetration rate: 1.0
- > unallocated homes: 0
- > fitness value: 6110.6

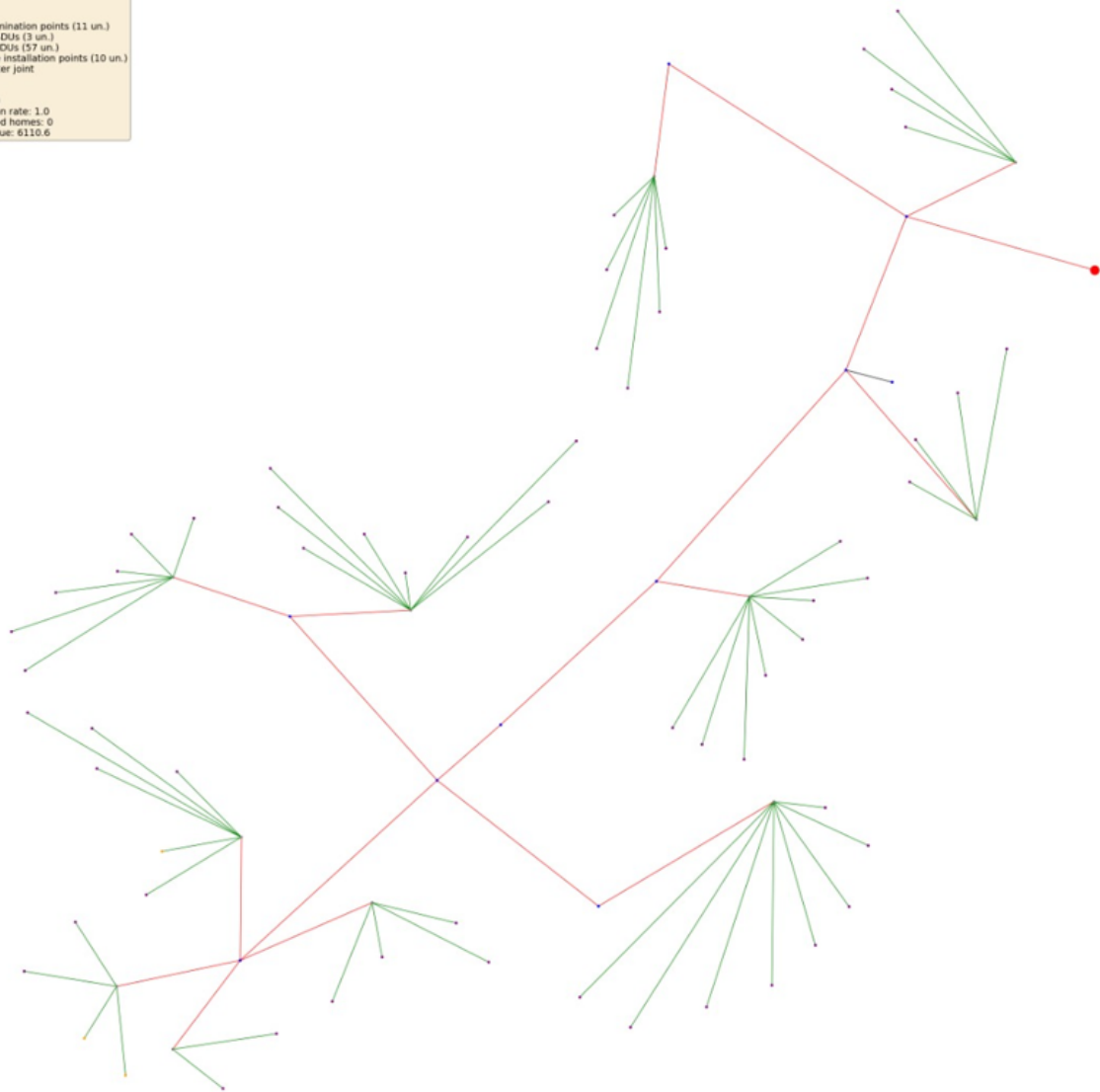


Figure 22: Handmade design of Network B

```
-- (generated) --
EDGES
> [red] distribution cable (1516 m)
> [green] drop cable (2704.44 m)
> [black] unused route
> [dotted] buried route

NODES
> [grey] termination points (4 un.)
> [orange] SDUs (3 un.)
> [purple] MDUs (57 un.)
> [blue] free installation points (18 un.)
> [red] splitter joint

STATISTICS
> clients: 60
> penetration rate: 1.0
> unallocated homes: 0
> fitness value: 3752.85
```

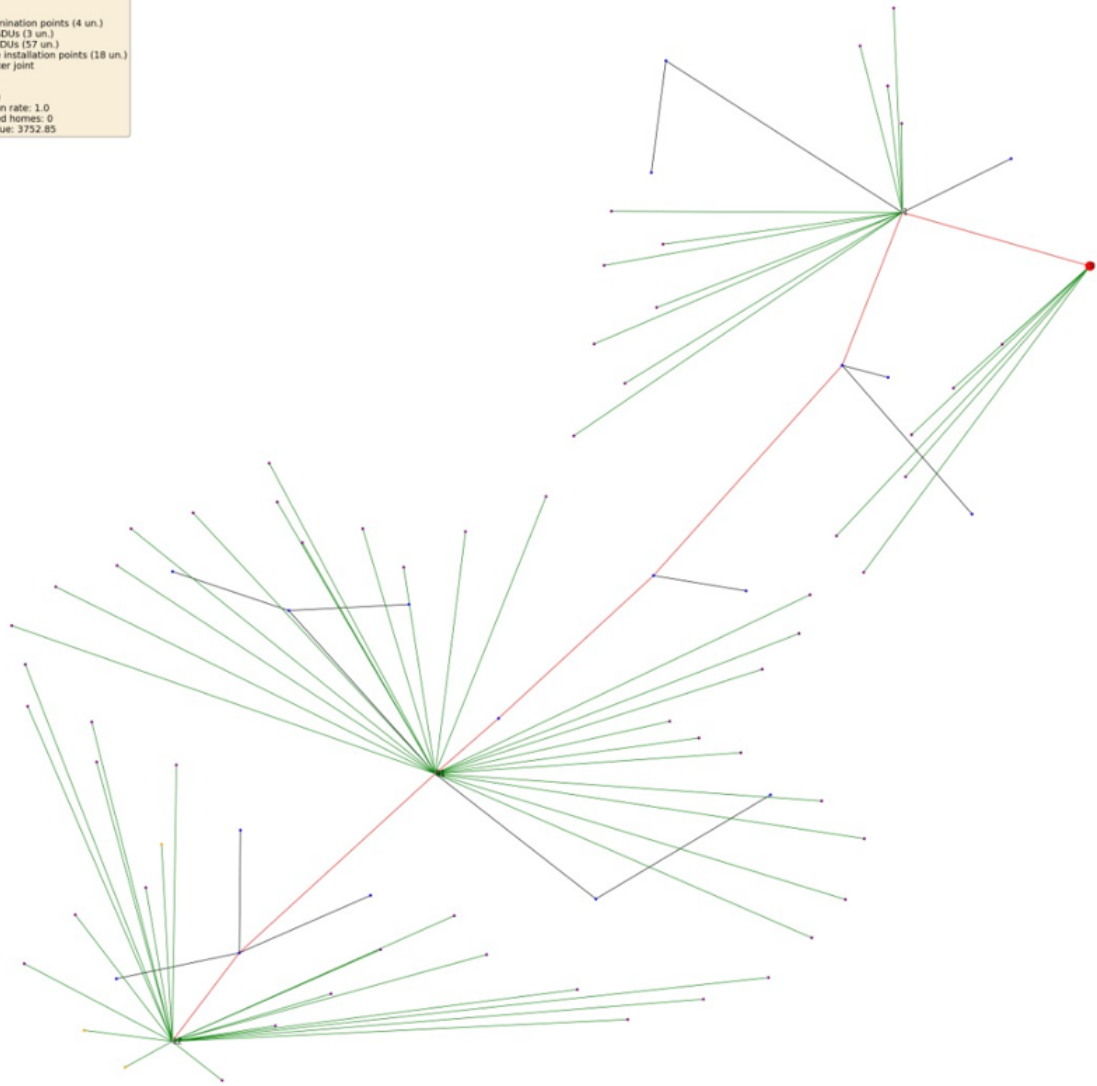


Figure 23: Generated design of Network B

Parameter	Handmade	Generated
Fitness Value	6110,60	3752,85
Pen. rate	1,00	1,00
Distrib. cable	2690,00	1516,00
Drop cable	1250,99	2704,44
# Term. Point	11	4
# SDU	3	3
# MDU	57	57
# Free inst. Point	10	18
# Clients	60	60

Table 7: Network B value comparison between handmade and generated designs

6.2.3 Network C

The third map analyzed is "Network C". Figure 24 corresponds to the network layout created manually by a network designer and Figure 25 corresponds to the automatic solution generated by the GA.

As shown in Figure 24, Network B has a high fitness value of 46819,54 monetary units. As previously mentioned in network A, this high cost is due to the suboptimal placement of cables and especially termination points and splitters, leading in a significant increase in the fitness value.

On the other hand, Figure 25 showcases the optimized network layout generated by the genetic algorithm in less than a few minutes. The automated design process reduces the fitness value to an impressive 11370,60 monetary units, approximately a 75,71% decrease in cost. Table 8 establishes a comparison between handmade and generated designs. Network C took 11,86 seconds to be generated.

(handmade) --	
EDGES	
>	[red] distribution cable (21380 m)
>	[green] drop cable (3531.68 m)
>	[black] unused route
>	[dotted] buried route
NODES	
>	[grey] termination points (27 un.)
>	[orange] SDUs (76 un.)
>	[purple] MDUs (11 un.)
>	[blue] free installation points (107 un.)
>	[red] splitter joint
STATISTICS	
>	clients: 87
>	penetration rate: 1.0
>	unallocated homes: 0
>	fitness value: 46819.54

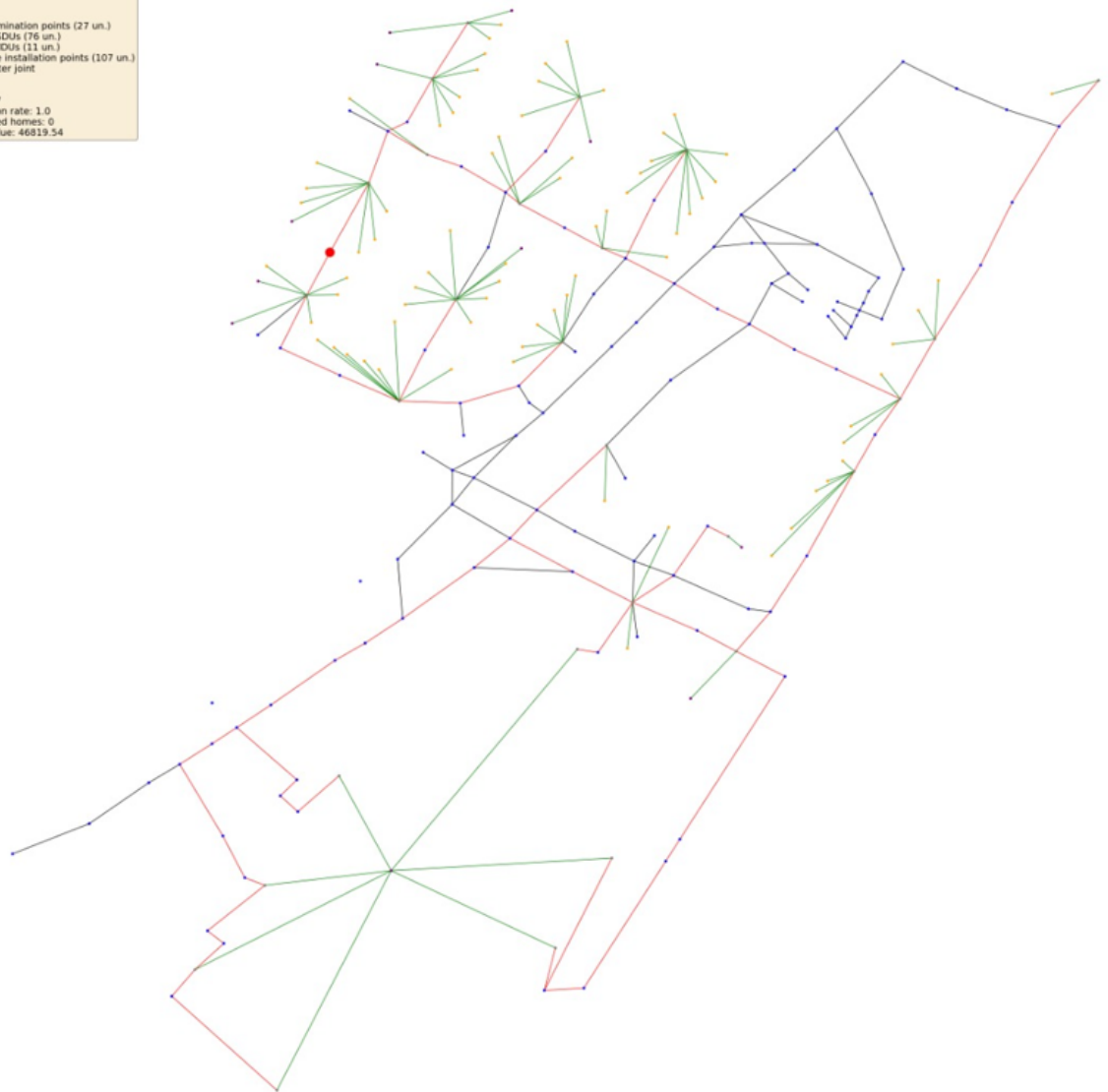


Figure 24: Handmade design of Network C

```
-- (generated) --
```

EDGES

- > [red] distribution cable (7763 m)
- > [green] drop cable (4958.07 m)
- > [black] unused route
- > [dotted] buried route

NODES

- > [grey] termination points (8 un.)
- > [orange] SDUs (76 un.)
- > [purple] MDUs (11 un.)
- > [blue] free installation points (127 un.)
- > [red] splitter joint

STATISTICS

- > clients: 87
- > penetration rate: 1.0
- > unallocated homes: 6
- > fitness value: 11370.6

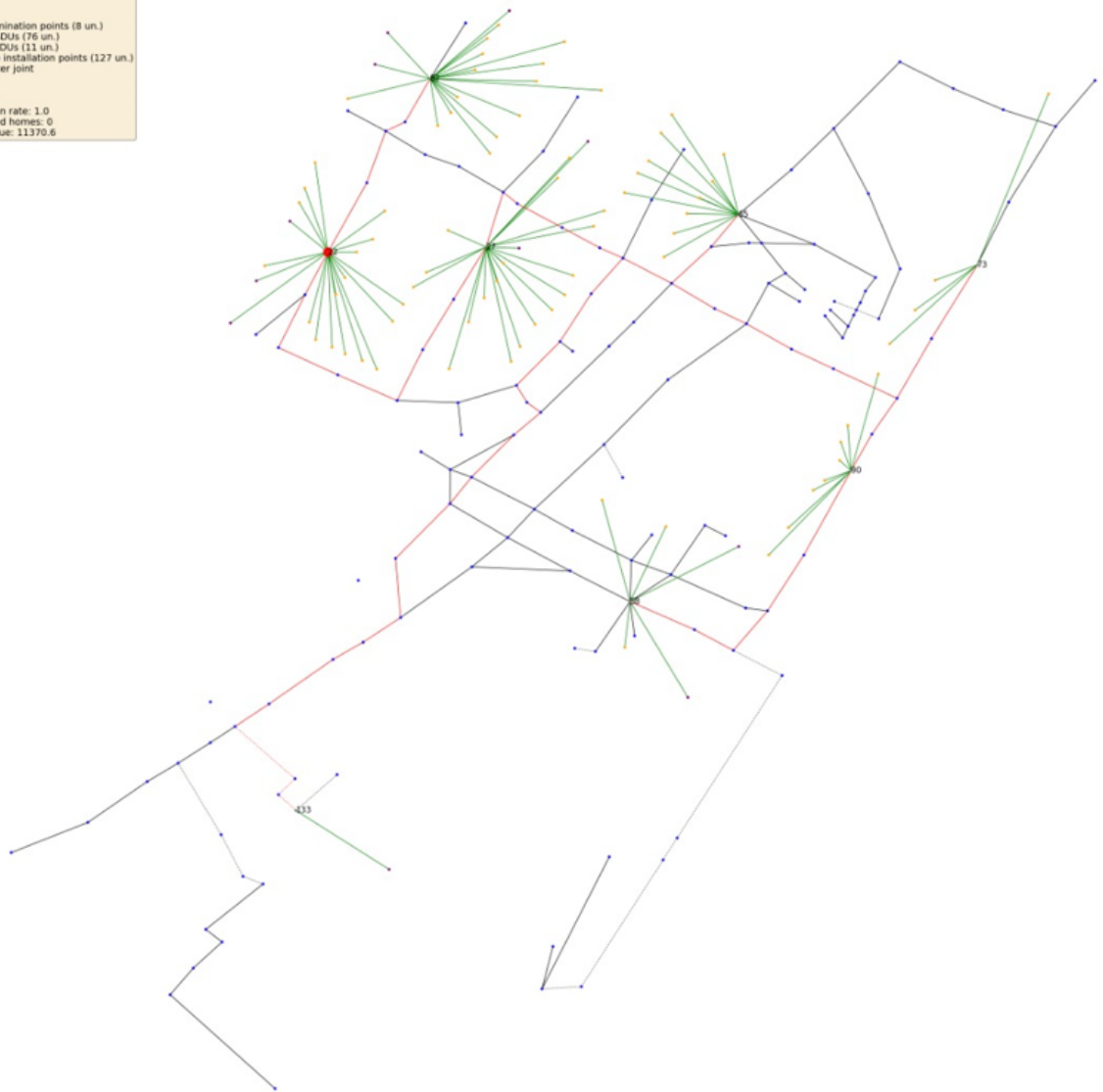


Figure 25: Generated design of Network C

Parameter	Handmade	Generated
Fitness Value	46819,54	11370,60
Pen. rate	1,00	1,00
Distrib. cable	21380,00	7763,00
Drop cable	3531,68	4958,07
# Term. Point	27	8
# SDU	76	76
# MDU	11	11
# Free inst. Point	107	127
# Clients	87	87

Table 8: Network C value comparison between handmade and generated designs

6.2.4 Network D

The fourth map analyzed is "Network D". Figure 26 corresponds to the network layout created manually by a network designer and Figure 27 corresponds to the automatic solution generated by the GA.

As shown in Figure 26, Network B has a high fitness value of 79019,13 monetary units.

On the other hand, Figure 27 showcases the optimized network layout generated by the genetic algorithm in less than a few minutes. The automated design process reduces the fitness value to an impressive 76468,30 monetary units, approximately a 3,23% decrease in cost. In this network, there was a small reduction in the fitness value because the handmade design was already quite efficient. However, while the handmade design takes hours to develop, the generated design is completed in only a few minutes. Therefore, it is still advantageous to use Netwin's automatic network design system. Table 9 establishes a comparison between handmade and generated designs. Network D took 17,64 seconds to be generated.

(handmade) --	
EDGES	
>	[red] distribution cable (26653 m)
>	[green] drop cable (5946.44 m)
>	[black] unused route
>	[dotted] buried route
NODES	
>	[grey] termination points (32 un.)
>	[orange] SDUs (49 un.)
>	[purple] MDUs (36 un.)
>	[blue] free installation points (155 un.)
>	[red] splitter joint
STATISTICS	
>	clients: 85
>	penetration rate: 1.0
>	unallocated homes: 0
>	fitness value: 79019.13

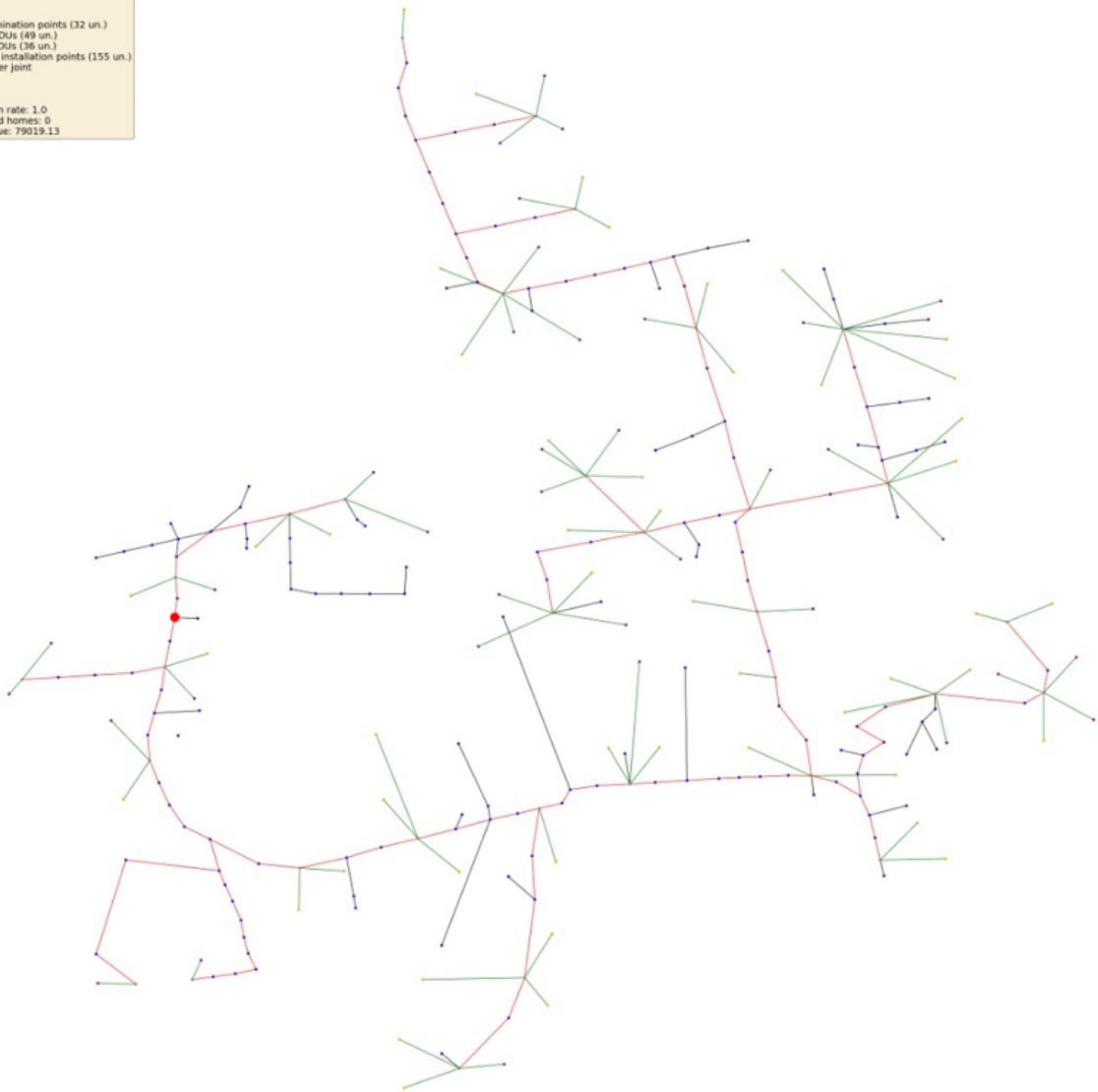


Figure 26: Handmade design of Network D

```

-- (generated) --
EDGES
> [red] distribution cable (42159 m)
> [green] drop cable (6416.98 m)
> [black] unused route
> [dotted] buried route

NODES
> [grey] termination points (29 un.)
> [orange] SDUs (49 un.)
> [purple] MDUs (36 un.)
> [blue] free installation points (158 un.)
> [red] splitter joint

STATISTICS
> clients: 85
> penetration rate: 1.0
> unallocated homes: 0
> fitness value: 76468.3

```

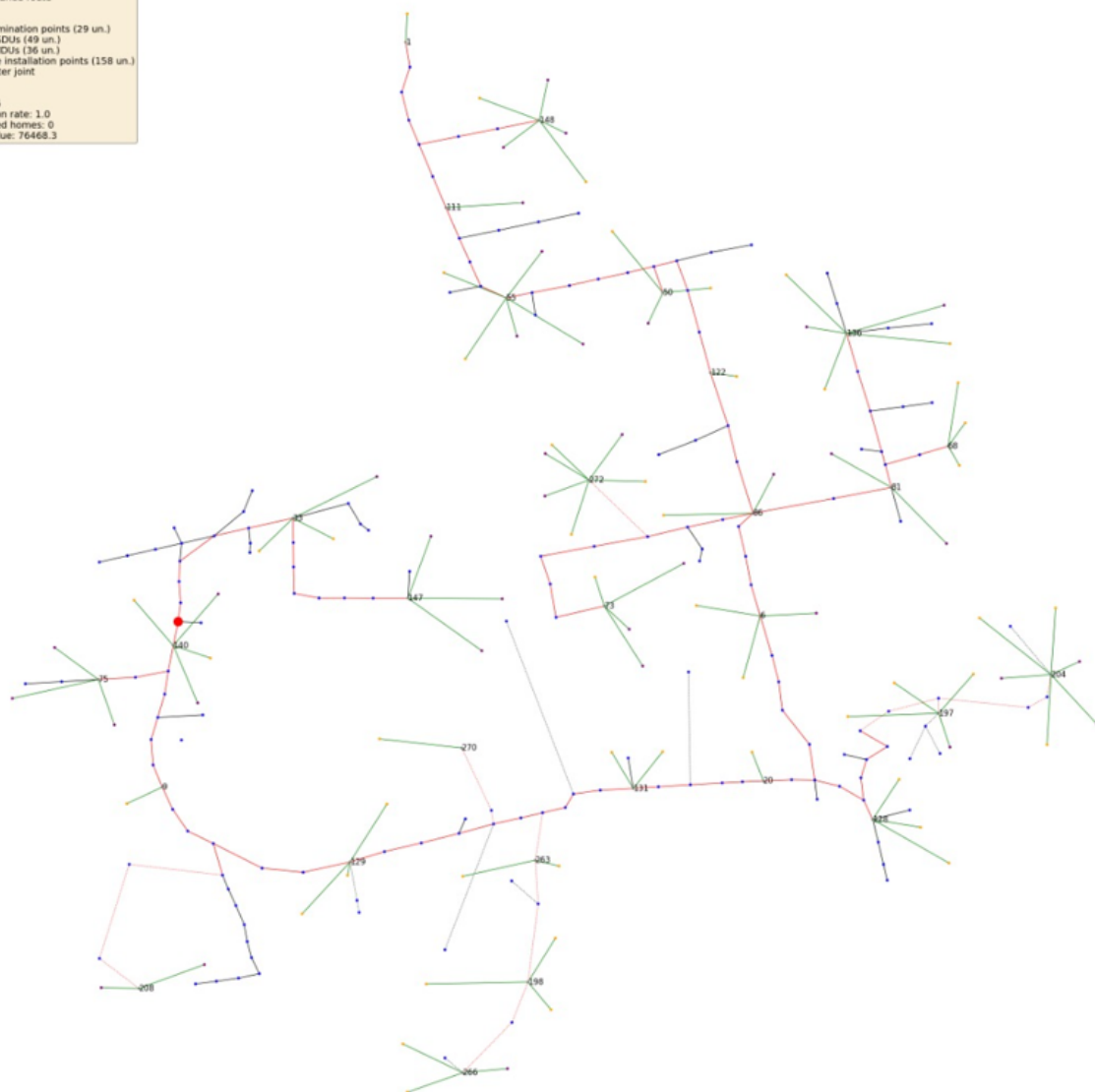


Figure 27: Generated design of Network D

Parameter	Handmade	Generated
Fitness Value	79019,13	76468,30
Pen. rate	1,00	1,00
Distrib. cable	26653,00	42159,00
Drop cable	5946,44	6416,98
# Term. Point	32	29
# SDU	49	49
# MDU	36	36
# Free inst. Point	155	158
# Clients	85	85

Table 9: Network D value comparison between handmade and generated designs

6.2.5 Network E

The fifth map analyzed is "Network E". Figure 28 corresponds to the network layout created manually by a network designer and Figure 29 corresponds to the automatic solution generated by the GA.

As shown in Figure 28, Network B has a high fitness value of 79369,46 monetary units.

Figure 29 showcases the optimized network layout generated by the genetic algorithm in less than a few minutes. The automated design process reduces the fitness value to an impressive 34915,51 monetary units, approximately a 56% decrease in cost. Table 10 establishes a comparison between handmade and generated designs. Network E took 16,70 seconds to be generated.

-- (handmade) --

EDGES

- > [red] distribution cable (40246 m)
- > [green] drop cable (4580,48 m)
- > [black] unused route
- > [dotted] buried route

NODES

- > [grey] termination points (54 un.)
- > [orange] SDUs (83 un.)
- > [purple] MDUs (12 un.)
- > [blue] free installation points (139 un.)
- > [red] splitter joint

STATISTICS

- > clients: 95
- > penetration rate: 1.0
- > unallocated homes: 0
- > fitness value: 79369.46

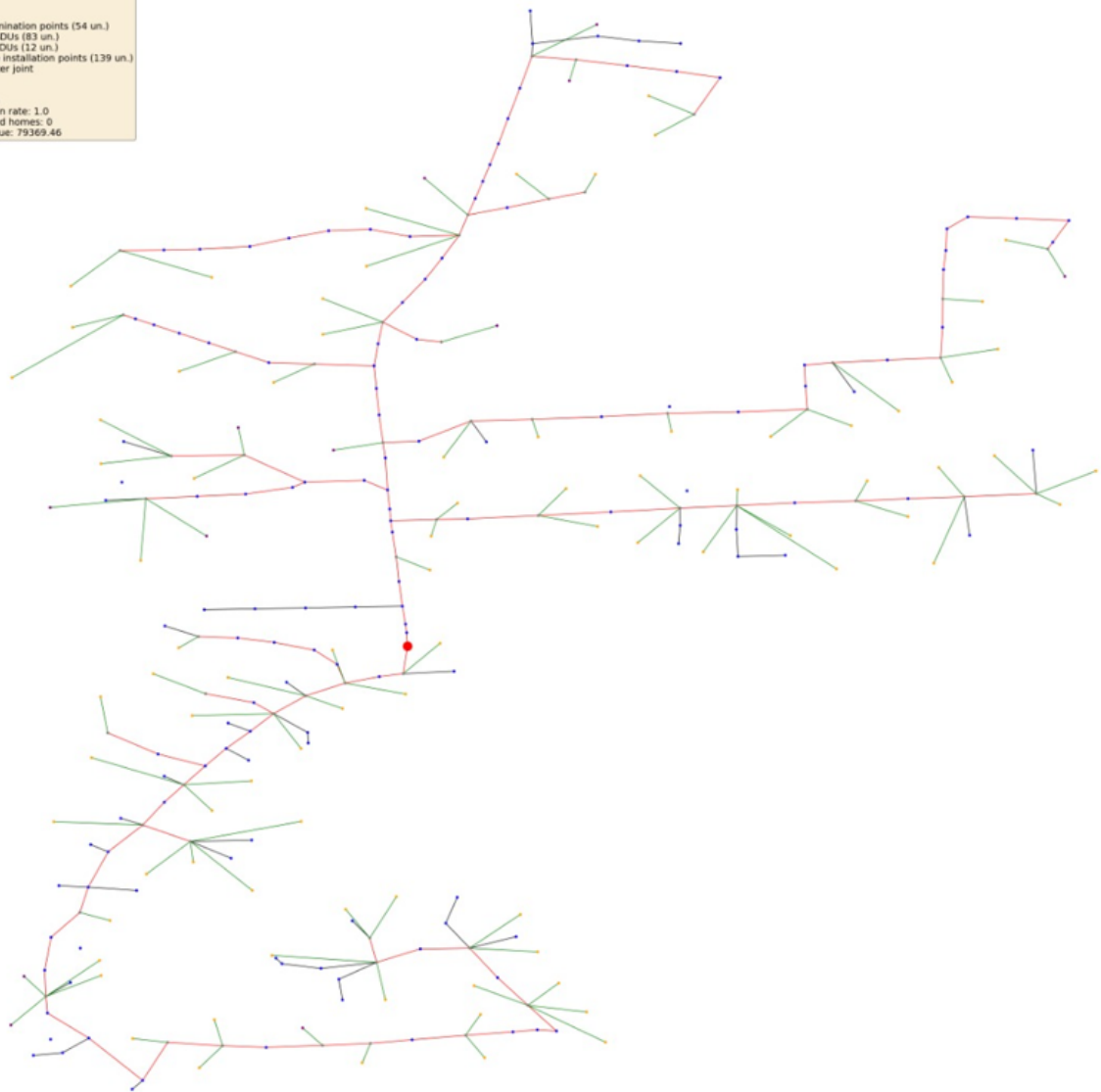


Figure 28: Handmade design of Network E

```
-- (generated) --
```

EDGES

- > [red] distribution cable (23860 m)
- > [green] drop cable (6338.97 m)
- > [black] unused route
- > [dotted] buried route

NODES

- > [grey] termination points (24 un.)
- > [orange] SDUs (83 un.)
- > [purple] MDUs (12 un.)
- > [blue] free installation points (170 un.)
- > [red] splitter joint

STATISTICS

- > clients: 95
- > penetration rate: 1.0
- > unallocated homes: 0
- > fitness value: 34915.51



Figure 29: Generated design of Network E

Parameter	Handmade	Generated
Fitness Value	79369,46	34915,51
Pen. rate	1,00	1,00
Distrib. cable	40246,00	23860,00
Drop cable	4580,48	6338,97
# Term. Point	54	24
# SDU	83	83
# MDU	12	12
# Free inst. Point	139	170
# Clients	95	95

Table 10: Network E value comparison between handmade and generated designs

6.3 Summary of the Experiments

In each network, the comparison between the handmade and the generated designs and their respective table values vividly demonstrates the power of genetic algorithms in network design.

In every network, there was a cost decrease. Networks A and C showed significant cost reductions of more than 70%. Networks B and E also showed good results, though less impressive than the two mentioned previously, with 38% and 56% reductions, respectively. Finally, network D presented a less significant cost reduction (3%). Table 11 establishes a comparison between the handmade and generated fitness values for each network design, as well as the corresponding percentage decrease in cost achieved by the algorithm.

Network Name	Handmade Fitness Value	Generated Fitness Value	Cost Decrease
Network A	60091,68	13351,66	-78%
Network B	6110,60	3752,85	-38%
Network C	46819,54	11370,60	-76%
Network D	79019,13	76468,30	-3%
Network E	79369,46	34915,51	-56%

Table 11: Fitness value comparisons of various network designs

This drastic reduction is achieved through the algorithm's ability to explore a vast solution space and iteratively improve the design by minimizing the length and cost of cables, optimizing the placement of termination points and splitters, and ensuring that all homes are connected. In all the networks tested, when comparing the manually created solution with the automatically generated one, it is evident that there is a decrease in distribution cable and an increase in drop cable usage, as the cost of drop cable is lower. Consequently, since the algorithm prioritized drop cable, the number of TP used also decreased. The algorithm reduces the number of devices used, as they are costly. The increase in drop cable usage compensates for this reduction. Finally, the number of available installation points increases, due to the reduced number of TP utilized.

The design of all these networks took only a few seconds (less than 20 seconds for all five networks), while the manual design of each one takes hours (this information was provided by a network designer at Altice). Table 12 provides a comparison of the time taken for each network to be generated.

Network Name	Generated Network Run Time (seconds)
Network A	12,93
Network B	8,01
Network C	11,86
Network D	17,64
Network E	16,70

Table 12: Run time comparisons of various network designs

The optical budget was respected and maintained in all the generated networks, so all customers continue to have access to their services without any limitations when compared to the handmade designs.

Concluding, by automating the design process, the algorithm not only reduces costs but also quickly produces a network layout that meets all necessary requirements far more efficiently than a manually created solution.

Chapter 7

Conclusions and future work

This chapter provides conclusions about the work presented in this dissertation, highlighting its significance and impact. Additionally, it discusses the limitations encountered during the development and implementation phases. Potential future work is outlined, focusing on possible extensions or improvements that could enhance the overall effectiveness of the system.

7.1 Conclusions

Currently, network design is a manual process carried out by planners. The tight deadlines imposed cause them to rush through the planning process and overlook potential cost savings. Development time is also a limitation, as designing a network can take several hours or even days. To address this issue, Altice Labs decided to evolve its Netwin inventory system, an innovative system responsible for network development and inventory, by incorporating AI in network planning and design. Thus, the main goal of this dissertation is to integrate and enhance Netwin's automatic network design based on AI algorithms to minimize capex and reduce time-to-create from days to hours.

Reflecting on the dissertation, it began with the presentation of the problem, contextualizing it with the increase in dependence on digital technologies and the widespread adoption of online platforms, along with the need for advances in the underlying infrastructure to keep up with growing demand. Within this topic, an introduction to Altice Labs was also made, along with the department responsible for maintaining the inventory system and the motivation to solve this problem. This study aligns with the first objective, which involved the study of the business area.

Next, the problem statement was addressed, where the evolution of network development was discussed, and an analysis of the GPON network was performed. This phase fulfilled the second objective, which involved the acquisition of general knowledge regarding a GPON network.

In order to conceptualize the developed solution, the state of the art regarding useful technologies

and algorithms for the development of the automated system was explored. Topics in graph theory, such as trees and minimum spanning trees, and classic graph theory problems were examined, along with an introduction to machine learning. Special attention was given to the study of metaheuristic algorithms and math-oriented solutions, where various approaches capable of solving this problem were studied. This phase fulfilled the third objective, which involved algorithms research. Within this chapter, commercial solutions were also studied, and a brief review of related works was provided.

The focus of the dissertation was on selecting and subsequently developing the most suitable algorithm for automating network development: the GA. In the development chapter, the methodology used, data exploration, the development of the GA, HTTP requests, system configurations, and the optical budget were detailed. At this stage, the fourth objective was achieved, which involved the design, development, and implementation of the chosen algorithm and the entire system.

To evaluate the system's quality and performance, tests were conducted on the five provided networks. Comparisons were made between handmade solutions (manually developed by a network planner) and the solutions generated by the algorithm. At this stage, the fifth objective was fulfilled, which involved evaluating the performance and quality of the developed solution.

The sixth and final objective involved product integration into the Netwin system; however, this was not accomplished. During the project's development, frequent meetings were held with the product owner, and other feature priorities for the system emerged. As a result, there was not enough time to achieve this last objective. It is important to emphasize that this does not affect the quality or functionality of the project, as it is still possible to visualize and access the status of all the generated networks.

Throughout the development of the project, several challenges were faced. The first major challenge was understanding GPON networks in general, including all the components that integrate them, their characteristics, and their utilities. This was a completely new topic for me, and my knowledge in the area was nearly non-existent. Several days of research and a few meetings with my internship supervisors were necessary to gain knowledge in this field. The second difficulty was the study of existing algorithms for the automation of fiber-optic network development. Since this area is often associated with business, there is limited information available online due to trade secrets.

In conclusion, the main objectives were achieved: reducing both the cost and the time required for network creation. According to the results analysis, network A experienced a cost reduction of 78%, network B 38%, network C 76%, network D 3%, and network E 56%. It was possible to drastically lower network costs in most cases. Furthermore, all the solutions generated by the algorithm were produced in under thirty seconds, demonstrating the system's functionality and efficiency, especially considering that all handmade

solutions took several hours to develop. The remaining objectives were also fulfilled, as explained in the previous paragraphs, excluding product integration. Overall, I consider this project to be a success.

7.2 Prospect for future work

The solution developed in this dissertation, while effective in addressing the problem at hand, has room for optimization to further enhance its performance. Optimization is crucial because the current system, despite its ability to reduce costs and shorten network creation time, operates within a set of initial conditions and assumptions that may limit its potential. By refining certain components, it is possible to improve the quality and precision of the network designs, as well as make the process more adaptable to different scenarios. Additionally, further optimization can ensure that the solution remains competitive and scalable in real-world applications where business requirements or environmental factors may vary.

One of the primary areas for optimization involves fine-tuning the GA hyperparameters. The current configuration strikes a balance between computational time and the quality of the generated solutions, but this balance could be improved through more focused tuning. Adjusting parameters such as population size, mutation rate, and crossover rate could lead to better exploration of the solution space, potentially improving the cost savings and overall network efficiency. Additionally, while the GA has been effective in generating networks in under thirty seconds, it may be possible to further reduce computational time without compromising the quality of the results.

Another avenue for improvement is the dynamic exploration of splitter joint and central office placements, extending the design to include the primary network. This would allow for a more flexible and optimized network layout. Additionally, analyzing field topology to determine optimal routes from the main graph and investigating methods for identifying primary routes from a location could result in more efficient network designs. Finally, a deeper analysis of intersections of existing roads when assigning termination points to homes could yield a more precise and cost-effective solution.

A key next step in this project is the product integration into the Netwin inventory system, which was not completed during the current phase. Despite the system's effectiveness in generating and displaying network designs, full integration into the existing inventory system would ensure seamless access and further enhance the automation of network development. Achieving this integration is essential for fully realizing the benefits of the developed solution in a real-world environment.

In conclusion, further optimization of the solution through hyperparameter tuning, enhanced network design capabilities, and full integration with the Netwin system represent the logical next steps. These

improvements would refine the current approach, making the tool more adaptable and efficient, and positioning it as a robust solution for automating network planning in diverse scenarios.

Bibliography

- [1] BusinessResearchInsights. Internet service market, 2023. URL <https://www.businessresearchinsights.com/market-reports/internet-service-market-104518>. [Accessed November 1, 2023].
- [2] GPON.com. Gigabit passive optical network (gpon), 2023. URL <https://www.gpon.com/>. [Accessed November 11, 2024].
- [3] Rajesh Khadka. Machine learning types #2. Medium; Towards Data Science, September 7 2017. URL <https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1>.
- [4] Maurice Chang. 4 stages of the machine learning (ml) modeling cycle. LinkedIn, 2017. URL <https://www.linkedin.com/pulse/4-stages-machine-learning-ml-modeling-cycle-maurice-chang>.
- [5] One Beyond. Agile software development methodology, 2024. URL <https://www.one-beyond.com/process/agile-software-development-methodology/>. Accessed: March 3, 2024.
- [6] C. Arzo, S. T.and Naiga, F. Granelli, R. Bassoli, M. Devetsikiotis, and F. Fitzek. A theoretical discussion and survey of network automation for iot: Challenges and opportunity. *Internet of Things Journal*, pages 12021–12045, 8 2021. doi: 10.1109/JIOT.2021.3075901.
- [7] Andrej Chu, Anis Ouali, and Kin Fai Poon. An enhanced ant colony optimization for ftth access network design. In *2012 17th European Conference on Networks and Optical Communications*, pages 1–6. Institute of Electrical and Electronics Engineers Inc., 2012. doi: 10.1109/NOC.2012.6249910.
- [8] FTTH Council Europe. *FTTH Handbook*. FTTH Council Europe, 7th edition, 2014. URL https://lafibre.info/images/doc/201602_ftth_handbook_v7.pdf. (n.d.).
- [9] A. Zhu. Fundamentals of fttx network, June 30 2016. URL <https://medium.com/@AriaZhu/fundamentals-of-fttx-network-de2423af294>. [Accessed November 7, 2023].

- [10] R. Gogoni. O que é gpon?, 2021. URL <https://tecnoblog.net/responde/o-que-e-gpon/>. [Accessed November 10, 2023].
- [11] Faruk Selmanovic and Edvin Skaljo. Gpon in telecommunication network. In *International Congress on Ultra Modern Telecommunications and Control Systems*, pages 1012–1016, 2010. doi: 10.1109/ICUMT.2010.5676500.
- [12] TechNavio. Gigabit passive optical network technology market size, trends and forecast 2023-2027, 2023. URL <https://www.technavio.com/report/gpon-technology-market-analysis>. TechNavio Market Report.
- [13] Zouhaira Abdellaoui, Yiyi Dieudonne, and Anoir Aleya. Design, implementation and evaluation of a fiber to the home (ftth) access network based on a giga passive optical network gpon. *Array*, 10: 100058, 2021. ISSN 2590-0056. doi: 10.1016/j.array.2021.100058.
- [14] D. Reinhard. *Graph Theory*. Springer, 5th edition, 2017. doi: 10.1007/978-3-662-53622-3.
- [15] University of Texas at Dallas. Applications of minimum spanning trees, 2012. URL <https://personal.utdallas.edu/~besp/teaching/mst-applications.pdf>. [Accessed November 16, 2023].
- [16] Telecom Trainer. Mst (minimum spanning tree), 2023. URL <https://www.telecomtrainer.com/mst-minimum-spanning-tree/>. [Accessed November 15, 2023].
- [17] A. Youssef, M. Assem, and M. Joudi. Minimum spanning tree: A comparison between prim's & kruskal's algorithms. *ResearchGate*, December 2019. doi: 10.13140/RG.2.2.26624.66563.
- [18] Sadhana H. Barkund, A. Sharma, and Haribhau R. Bhapkar. Survey of shortest path algorithms. *International Journal of Research and Engineering*, 10(11):46–57, November 2022. doi: 10.58443/IJREX.10.11.2022.46-57.
- [19] M. Laha and R. Datta. A steiner tree based efficient network infrastructure design in 5g urban vehicular networks. *Computer Communications*, 201(2):1–9, March 2023. doi: 10.1016/j.comcom.2023.01.016.
- [20] Thomas Rincy N and Roopam Gupta. A survey on machine learning approaches and its techniques:. In *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, pages 1–6, 2020. doi: 10.1109/SCEECS48394.2020.190.

- [21] J. Gandhi and Z. Narmawala. A comprehensive survey on machine learning techniques in opportunistic networks: Advances, challenges and future directions. *Pervasive and Mobile Computing*, 100:101917, 2024. ISSN 1574-1192. doi: 10.1016/j.pmcj.2024.101917.
- [22] A. M. S. Asih, B. M. Sopha, and G. Kriptaniadewa. Comparison study of metaheuristics. *International Journal of Engineering Business Management*, 9:184797901774360, 2017. doi: 10.1177/1847979017743603.
- [23] N. Rincon-Garcia, B. J. Waterson, and T. J. Cherrett. A hybrid metaheuristic for the time-dependent vehicle routing problem with hard time windows. *International Journal of Industrial Engineering Computations*, pages 141–160, 2017. doi: 10.5267/j.ijiec.2016.6.002.
- [24] I. H. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63: 511–623, October 1996. doi: 10.1007/BF02125421.
- [25] J. Mata, I. de Miguel, R. J. Durán, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania. Artificial intelligence (ai) methods in optical networks: A comprehensive survey. *Optical Switching and Networking*, 28:43–57, 2018. doi: 10.1016/j.osn.2017.12.006.
- [26] T. He, H. Wang, and S. W. Yoon. Comparison of four population-based meta-heuristic algorithms on pick-and-place optimization. *Procedia Manufacturing*, 17:944–951, 2018. ISSN 2351-9789. doi: 10.1016/j.promfg.2018.10.112. 28th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2018), June 11-14, 2018, Columbus, OH, USA Global Integration of Intelligent Manufacturing and Smart Industry for Good of Humanity.
- [27] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. doi: 10.1109/icnn.1995.488968.
- [28] Tareq M. Shami, Ayman A. El-Saleh, Mohammed Alswaitti, Qasem Al-Tashi, Mhd Amen Summakieh, and Seyedali Mirjalili. Particle swarm optimization: A comprehensive survey. *IEEE Access*, 10: 10031–10061, 2022. doi: 10.1109/ACCESS.2022.3142859.
- [29] Micael Couceiro and Pedram Ghamisi. *Particle Swarm Optimization*, pages 1–10. Springer International Publishing, Cham, 2016. ISBN 978-3-319-19635-0. doi: 10.1007/978-3-319-19635-0_1.

- [30] R. Hassan, B. Cohanin, O. de Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2005. doi: 10.2514/6.2005-1897.
- [31] F. van den Bergh and A. P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):225–239, 2004. doi: 10.1109/TEVC.2004.826069.
- [32] K. Premalatha and A. M. Natarajan. Hybrid pso and ga for global maximization. *International Journal of Open Problems in Computer Science and Mathematics*, 2(4), 2009. ISSN 1998-6262. URL https://www.researchgate.net/publication/267239415_Hybrid_PSO_and_GA_for_global_maximization. Copyright © ICSRS Publication, 2009 www.i-csrs.org.
- [33] R. Aguilar-Rivera, M. Valenzuela-Rendón, and J. J. Rodríguez-Ortiz. Genetic algorithms and darwinian approaches in financial applications: A survey. *Expert Systems with Applications*, 42(21):7684–7697, 2015. doi: 10.1016/j.eswa.2015.06.001.
- [34] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine Learning*, 3(2/3):95–99, 1988. doi: 10.1023/a:1022602019183.
- [35] S. Villalba, Rossi, M. R. Marcos Perez Mokarzel, Salvador, M. Almeida, A. C. Cesar, M. A. Romero, and M. de. Design of passive optical networks using genetic algorithm. In *2009 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference*, 2009. doi: 10.1109/imoc.2009.5427496.
- [36] Z. Zuhri and I. V. Paputungan. A hybrid optimization algorithm based on genetic algorithm and ant colony optimization. *International Journal of Artificial Intelligence & Applications*, 4(5):63–75, 2013. doi: 10.5121/ijaia.2013.4505.
- [37] A. Sadegheih. Optimization of network planning by the novel hybrid algorithms of intelligent optimization techniques. *Energy*, 34(10):1539–1551, 2009. doi: 10.1016/j.energy.2009.06.047.
- [38] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, Jan 1986. doi: 10.1109/TSMC.1986.289288.
- [39] A. Beg and M. Zahidul. Advantages and limitations of genetic algorithms for clustering records. In *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, page 7604009. IEEE, 2016. doi: 10.1109/ICIEA.2016.7604009.

- [40] M. Dorigo, G. D. Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999. doi: 10.1162/106454699568728.
- [41] J. Yu, X. You, and S. Liu. A heterogeneous guided ant colony algorithm based on space explosion and long–short memory. *Applied Soft Computing*, 113:107991–107991, 2021. doi: 10.1016/j.asoc.2021.107991.
- [42] Z. Xiao and W. Jiang-qing. Hybrid ant algorithm and applications for vehicle routing problem. *Physics Procedia*, 25:1892–1899, 2012. doi: 10.1016/j.phpro.2012.03.327.
- [43] Yucheng Kao, Ming-Hsien Chen, and Yi-Ting Huang. A hybrid algorithm based on aco and pso for capacitated vehicle routing problems. *Mathematical Problems in Engineering*, 2012(1):726564, 2012. doi: <https://doi.org/10.1155/2012/726564>.
- [44] Saman Almufti. Single-based and population-based metaheuristics algorithms performances in solving np-hard problems. *Iraqi Journal of Science*, 2021. doi: 10.24996/ijcs.2021.62.5.34.
- [45] Vincent F. Yu, Ahmad A. N. P. Redi, Yudi A. Hidayat, and O. J. Wibowo. A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, 53:119–132, 2017. doi: 10.1016/j.asoc.2016.12.027.
- [46] Nader Ghaffarinasab, S. Ghazanfar Ahari, and Mehdi Ghazanfari. A hybrid simulated annealing based heuristic for solving the location-routing problem with fuzzy demands. *Scientia Iranica*, 20(3): 919–930, 2013. doi: 10.1016/j.scient.2013.02.006.
- [47] F. Wen and C. S. Chang. Transmission network optimal planning using the tabu search method. *Electric Power Systems Research*, 42(2):153–163, 1997. doi: 10.1016/s0378-7796(96)01200-x.
- [48] D. R. Thompson and G. L. Bilbro. Comparison of a genetic algorithm with a simulated annealing algorithm for the design of an atm network. *IEEE Communications Letters*, 4(8):267–269, 2000. doi: 10.1109/4234.864190.
- [49] Fred Glover. Tabu search—part i. *ORSA Journal on Computing*, 1(3):190–206, 1989. doi: 10.1287/ijoc.1.3.190.
- [50] Michel Gendreau. *An Introduction to Tabu Search*, pages 37–54. Springer US, Boston, MA, 2003. ISBN 978-0-306-48056-0. doi: 10.1007/0-306-48056-5_2.

- [51] S. Russenschuck. Mathematical optimization techniques, 1999. URL https://www.academia.edu/29594304/Mathematical_optimization_techniques. Accessed: January 20, 2024.
- [52] E. Costamagna, A. Fanni, and G. Giacinto. A tabu search algorithm for the optimisation of telecommunication networks. *European Journal of Operational Research*, 106(2-3):357–372, 1998. doi: 10.1016/s0377-2217(97)00279-8.
- [53] Ahmet Kokangul and Abdulhamid Abba. Optimization of passive optical network planning. *Applied Mathematical Modelling*, 35(7):3345–3354, 2011. doi: 10.1016/j.apm.2011.01.017.
- [54] G. C. Oliveira, A. P. C. Costa, and S. Binato. Large scale transmission network planning using optimization and heuristic techniques. *IEEE Transactions on Power Systems*, 10(4):1828–1834, 1995. doi: 10.1109/59.476047.
- [55] S. Voß, Vittorio Maniezzo, and T. Stützle. *MATHEURISTICS: Hybridizing Metaheuristics and Mathematical Programming*. Annals of Information Systems. Springer, 2009.
- [56] A. L. S. Aguiar, F. B. C. Sousa, and Y. V. L. de Melo. Optical distribution network design using pso. *IEEE Communications Letters*, 27(1):239–242, 2023. doi: 10.1109/lcomm.2022.3218530.
- [57] Ž. N. Popović, V. Dj. Kerleta, and D. S. Popović. Hybrid simulated annealing and mixed integer linear programming algorithm for optimal planning of radial distribution networks with distributed generation. *Electric Power Systems Research*, 108:211–222, 2014. doi: 10.1016/j.epsr.2013.11.015.
- [58] N. C. Koutsoukis, P. S. Georgilakis, and N. D. Hatziaargyriou. A tabu search method for distribution network planning considering distributed generation and uncertainties. *CiteSeer X (the Pennsylvania State University)*, 2014. doi: 10.1109/pmaps.2014.6960627.
- [59] António Eira, J. Pedro, and J. Pires. Optimized design of multistage passive optical networks. *Journal of Optical Communications and Networking*, 4(5):402–402, 2012. doi: 10.1364/jocn.4.000402.
- [60] Esri. Fast@home: Automated and optimized fiber network planning with gis, 2024. URL <https://www.esri.de/content/dam/distributor-restricted/esri-de/files/arcgis/produkte/fast-at-home/fast-at-home-en.pdf>. Accessed: January 22, 2024.

- [61] X-Brain. X-brain fibre for ai-based fibre network planning, 2024. URL <https://x-brainfibre.io/solution/:https://x-brainfibre.io/solution/>. Accessed: January 22, 2024.
- [62] KSavi. Ksavi network inventory | fiber optic network planning and design, 2024. URL https://ksavinetworkinventory.com/fiber-optic-network-planning-and-design/#geospatial_network_inventory. Accessed: January 22, 2024.
- [63] F. D. Wihartiko, H. Wijayanti, and F. Virgantari. Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem. *IOP Conference Series: Materials Science and Engineering*, 332:012020, 2018. doi: 10.1088/1757-899x/332/1/012020.
- [64] M. Kumar and R. Dwivedi. Applicability of scrum methods in software development process. *EngRN: Computer-Aided Engineering (Topic)*, 2020. doi: 10.2139/ssrn.3610759.
- [65] H. Cervone. Understanding agile project management methods using scrum. *OCLC Systems & Services: International digital library perspectives*, 27:18–22, 2011. doi: 10.1108/10650751111106528.
- [66] A. Siahaan and M. Furqan. A review of prim and genetic algorithms in finding and determining routes on connected weighted graphs. 2018. doi: 10.31227/osf.io/e6rj7.
- [67] J. Zhong, Xiaomin Hu, Jun Zhang, and Min Gu. Comparison of performance between different selection strategies on simple genetic algorithms. *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 2:1115–1121, 2005. doi: 10.1109/CIMCA.2005.1631619.
- [68] A. Orriols-Puig, K. Sastry, P. Lanzi, D. Goldberg, and Ester Bernadó-Mansilla. Modeling selection pressure in xcs for proportionate and tournament selection. pages 1846–1853, 2007. doi: 10.1145/1276958.1277325.
- [69] Rakesh Kumar and Jyotishree. Blending roulette wheel selection rank selection in genetic algorithms. *International Journal of Machine Learning and Computing*, pages 365–370, 2012. doi: 10.7763/IJMLC.2012.V2.146.

- [70] O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, and R. Kharel. Variation operators for grouping genetic algorithms: A review. *Swarm and Evolutionary Computation*, 60:100796, 2021. doi: 10.1016/j.swevo.2020.100796.
- [71] Y. Yuan, H. Xu, and B. D. Wang. An experimental investigation of variation operators in reference-point based many-objective optimization. *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015. doi: 10.1145/2739480.2754655.
- [72] H. K. Birru. Empirical study of two classes of bit variation operators in evolutionary computation. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 3: 1917–1924 Vol. 3, 1999. doi: 10.1109/CEC.1999.785507.