

MEC-SETEC
INSTITUTO FEDERAL MINAS GERAIS - *Campus* Formiga
Curso de Ciência da Computação

***SOFTWARE PARA AUTOMATIZAÇÃO DO PROJETO DE REDES
ÓPTICAS PASSIVAS (PON)***

Saulo Ricardo Dias Fernandes

Orientador: Prof. Me. Everthon Valadão dos
Santos

Formiga - MG

2019

SAULO RICARDO DIAS FERNANDES

***SOFTWARE PARA AUTOMATIZAÇÃO DO PROJETO DE REDES
ÓPTICAS PASSIVAS (PON)***

Trabalho de Conclusão de Curso apresentado ao Instituto Federal Minas Gerais - *Campus* Formiga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Everthon Valadão dos Santos

Formiga - MG

2019

Fernandes, Saulo Ricardo Dias.
004 Software para Automatização do Projeto de Redes Ópticas Passivas
(PON) / Saulo Ricardo Dias Fernandes. -- Formiga : IFMG, 2019.
56p. : il.

Orientador Prof. Msc. Everthon Valadão dos Santos
Trabalho de Conclusão de Curso – Instituto Federal de Educação,
Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1.Redes Ópticas Passivas. 2. Problema de Localização de Facilidades.
3. Projeto de Rede. 4. PON. I. Título.

CDD 004

SAULO RICARDO DIAS FERNANDES

**SOFTWARE PARA AUTOMATIZAÇÃO DO PROJETO DE REDES
ÓPTICAS PASSIVAS (PON)**

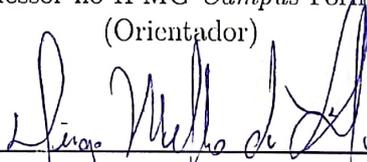
Trabalho de Conclusão de Curso apresentado
ao Instituto Federal Minas Gerais - *Campus*
Formiga, como requisito parcial para a ob-
tenção do título de Bacharel em Ciência da
Computação.

Trabalho aprovado em: 15 de junho de 2019.

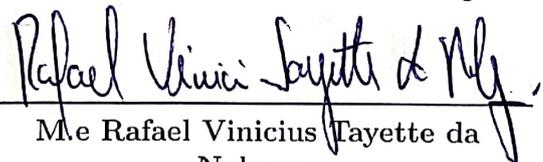
BANCA EXAMINADORA



M.º Everthon Valadão
Professor no IFMG *Campus* Formiga
(Orientador)



M.º Djego Mello da Silva
Professor no IFMG *Campus* Formiga



M.º Rafael Vinicius Layette da
Nobrega
Professor no IFMG *Campus* Formiga

Formiga - MG

2019

Agradecimentos

Agradeço primeiramente à minha família, em especial meus pais, Ricardo Fernandes Acácio da Silva e Naliana Dias Leandro e meu irmão João Vitor Dias Fernandes, que fizeram de mim quem sou hoje e que me deram suporte em minhas decisões e nos momentos de dificuldade ao longo da vida.

Agradeço ao Professor Everthon Valadão por ter sido sempre paciente, mesmo com minhas dificuldades e por ter me auxiliado em todos os problemas que eu tive durante o desenvolvimento deste trabalho.

Agradeço à minha namorada Renata dos Santos Pereira por sempre me acalmar nos momentos em que precisei e por aturar o meu mau humor quando as coisas estavam dando errado. Agradeço aos meus amigos Wesley Henrique, Ana Flávia, Arthur Teodoro, Renato Borges, que tornaram possível passar por estes quatro anos e meio de faculdade sem que eu perdesse completamente a sanidade.

E por último, apesar de que nunca vão saber deste agradecimento, gostaria de agradecer aos meus animais de estimação, tanto os que estão presentes, Mel, Melissa, Max, Kiara e Banguela, quanto os que se foram, Nitro, Theo e Loki, pelo amor incondicional que apenas um animal pode dar.

“Let There Be More Light”
(Pink Floyd, A Saucerful of Secrets)

Resumo

A comunicação é uma das bases da sociedade como a conhecemos e sempre buscamos formas mais eficientes de nos comunicar de uma maneira mais rápida. A nossa necessidade de transmitir informação aumenta a cada dia e precisamos de tecnologias que consigam acompanhar essa necessidade. Atualmente, as comunicações baseadas em DSL não estão mais suportando a grande taxa de dados demandadas pelos usuários, por isso cada vez mais redes de fibra óptica estão sendo amplamente utilizadas. Um dos grandes problemas de implantação das redes ópticas é devido seu custo elevado de instalação, dessa forma, deve-se realizar um projeto de rede óptica passiva (PON) adequado, o que nem sempre é uma tarefa simples de ser feita. Portanto ferramentas computacionais podem ser utilizadas na tomada de decisão em vários quesitos do projeto. Neste sentido, neste trabalho de conclusão de curso é desenvolvido um *software* para projeto de rede óptica passiva, considerando o trajeto de fibras ópticas mais adequado, o posicionamento dos *splitters* primário e secundário, de acordo com a demanda de clientes, com o intuito de reduzir os custos de instalação da rede.

Palavras-chaves: Redes Ópticas Passivas, Problema de Localização de Facilidades, Projeto de Rede PON.

Abstract

Communication is one of the foundations of the society as we know it, people are always looking for more efficient ways of informing ourselves more quickly. Our need to transmit information increases every day and we need technologies that can keep up with this lack. Currently, DSL-based communications are no longer supporting the high rate of data demanded by users, so more and more fiber optic networks are being widely used. One of the major problems of optical network deployment is due to its high cost of installation, so an adequate passive optical network (PON) project must be performed, which is not always a simple task to do. Therefore, computational tools can be used in decision-making on various design issues. In this sense, in this work of course completion a software is developed for passive optical network design, considering the most suitable optical fiber path, the positioning of the primary and secondary splitters depending on the demand customers, in order to reduce network installation costs.

Keywords: Passive Optical Networks, Facility Location Problem, PON Network Project.

Lista de ilustrações

Figura 1 – Arquitetura Básica de uma Rede PON.	18
Figura 2 – Transmissão da luz através da fibra.	20
Figura 3 – Reflexão e Refração.	20
Figura 4 – Exemplo <i>splitter</i> 1:4.	22
Figura 5 – Exemplo de uma Árvore de Steiner.	24
Figura 6 – Mapa de Formiga-MG gerado após execução do <i>script</i> gerador de XML.	32
Figura 7 – Exemplo do XML gerado.	33
Figura 8 – Mapa de Formiga-MG obtido com a biblioteca <i>gmpplot</i> a partir dos dados do Overpass-Turbo	34
Figura 9 – Divisão da área delimitada em células	37
Figura 10 – Rede PON com dois níveis de divisão de potência	38
Figura 11 – Exemplo de como fica a visualização de cada tipo de <i>splitter</i>	40
Figura 12 – Comparação visual da rede PON da solução manual e da automatizada	43
Figura 13 – Três enlaces da rede, gerados a cada iteração do programa, que juntos formam a rede final.	44
Figura 14 – Custo total da rede PON conforme aumento na demanda de clientes – Algoritmo guloso “pela demanda”	46
Figura 15 – Porcentagem de clientes atendidos conforme aumento na demanda – Algoritmo guloso “pela demanda”	47
Figura 16 – Custo por cliente atendido conforme aumento na demanda – Algoritmo guloso “pela demanda”	48
Figura 17 – Custo total da rede PON conforme aumento na demanda de clientes – Algoritmo guloso “pela distância”	49
Figura 18 – Porcentagem de clientes atendidos conforme aumento na demanda – Algoritmo guloso “pela distância”	49
Figura 19 – Custo por cliente atendido conforme aumento na demanda – Algoritmo guloso “pela distância”	50
Figura 20 – Porcentagem de clientes atendidos pelo <i>software</i> proposto	51
Figura 21 – Custo por cliente obtido pelo <i>software</i> proposto	52

Lista de tabelas

Tabela 1 – Tecnologias DSL	17
Tabela 2 – Comparação entre tecnologia EPON e GPON	19
Tabela 3 – Solução proposta no trabalho de Oliveira (2018)	45
Tabela 4 – Solução encontrada neste trabalho	45

Lista de abreviaturas e siglas

API	<i>Application Program Interface</i>
BER	<i>Bit Error Rate</i>
CAD	Desenho Assistido por Computador
CEO	Caixa de Emenda Óptica
CGI	Comitê Gestor da Internet no Brasil
CO	<i>Central Office</i>
CTO	Caixa Terminal Óptica
DSL	<i>Digital Subscriber Line</i>
EPON	<i>Ethernet Passive Optical Network</i>
ET	<i>Element-Tree</i>
FDH	<i>Fiber Distribution Hub</i>
GPL	<i>General Public License</i>
GPON	<i>Gigabit Passive Optical Network</i>
IPEA	Instituto de Pesquisa Econômica Aplicada
JOSM	<i>Java OpenStreetMap Editor</i>
OLT	<i>Optical Line Terminal</i>
ONT	<i>Optical Network Terminal</i>
OSM	<i>OpenStreetMap</i>
PDF-G	<i>Population Density Function Embedded Graphs</i>
PON	<i>Passive Optical Network</i>
PSF	<i>Python Software Foundation</i>
RALA	<i>Recursive Association and Location Algorithm</i>
UDWDM	<i>Ultra-dense Wavelength Division Multiplexing</i>

UTP	Cabo de par Trançado
XML	<i>Extensible Markup Language</i>
DIO	<i>Distribuidor Interno Óptico</i>
TOA	<i>Terminal Óptico do Assinante</i>

Sumário

1	INTRODUÇÃO	14
1.1	Justificativa	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Estrutura do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Tecnologias de Acesso à Internet	17
2.1.1	DSL	17
2.1.2	Tecnologias PON (<i>Passive Optical Network</i>)	17
2.1.2.1	Definição de Rede PON	18
2.1.2.2	EPON e GPON	19
2.2	Componentes Básicos da rede PON	19
2.2.1	Cabo Óptico	19
2.2.2	Propagação da Luz na Fibra Óptica	19
2.2.3	Transmissores e Receptores Ópticos	21
2.2.4	Emendas Ópticas	21
2.2.5	Divisores Ópticos (<i>Splitters</i>)	21
2.2.6	Distribuidor Interno Óptico (DIO)	22
2.3	Projeto Tradicional de Redes de Fibra Óptica	22
2.4	Problema de Localização de Instalações	23
2.5	Algoritmo Guloso	23
2.6	Árvore de Steiner	24
2.7	Trabalhos Relacionados	24
3	MATERIAIS E MÉTODOS	28
3.1	<i>OpenStreetMap</i>	28
3.2	Overpass API e Overpass-turbo	29
3.3	Josm	29
3.4	Python	29
3.5	Bibliotecas para Python	30
3.6	Metodologia	30
3.7	<i>Hardware</i>	31
4	PROJETO E DESENVOLVIMENTO	32

4.1	Mapa da cidade	32
4.2	Manipulação do XML	33
4.3	Estrutura de dados	34
4.4	Caminho mínimo	35
4.5	Balanco de potência	36
4.6	Demanda de Clientes Simulada	36
4.7	Atendimento às ruas	37
4.7.1	Versão 1.0: Atendendo às Esquinas	38
4.7.2	Versão 2.0: Atendendo à demanda gerada via simulação	38
4.8	Abordagem Gulosa	41
5	RESULTADOS E ANÁLISE	42
5.1	Comparação do projeto automatizado com um projeto manual	42
5.2	Abordagem Gulosa (pela Demanda)	46
5.2.1	Comparação sob diferentes distribuições e demandas	46
5.3	Abordagem Gulosa (pela Distância)	48
5.3.1	Comparação sob diferentes distribuições e demandas	48
5.4	Abordagem Gulosa: pela demanda VS. pela distância	51
6	TRABALHOS FUTUROS	53
7	CONSIDERAÇÕES FINAIS	55
	REFERÊNCIAS	56

1 INTRODUÇÃO

A popularidade de conexões *Digital Subscriber Line* (DSL) vem caindo no Brasil, enquanto conexões de fibra óptica crescem em presença e representatividade. Segundo o Comitê Gestor da Internet no Brasil ¹, houve uma queda de 26% para 19% na presença de DSL nos domicílios brasileiros com Internet, entre 2015 e 2016, ao passo que a presença de conexões de fibra óptica cresceu de 24% para 28% no mesmo período (CGI.br, 2017). Apesar deste crescimento, a mesma pesquisa do CGI.br aponta que, em mapeamento apresentado pelo Instituto de Pesquisa Econômica Aplicada (IPEA) ², verificou-se que existem 2.325 municípios sem cabeamento de fibra óptica no Brasil, sendo que 58% deles estão nas regiões Norte e Nordeste.

Um dos grandes problemas das redes ópticas é seu custo elevado de instalação. Por isso o projeto da Rede Óptica Passiva (PON) depende diretamente de quantos clientes podem ser atendidos em determinada localidade, sendo necessário levar em conta a geografia e infraestrutura existentes no local desejado. Pela complexidade do problema, ferramentas computacionais devem ser utilizadas na tomada de decisão sobre o trajeto dos cabos de uma rede óptica, da posição de instalação dos *splitters* e até mesmo na escolha do tipo de *splitter* a ser usado no projeto. Neste sentido, o presente trabalho desenvolve um *software* para projetar uma rede óptica passiva, informando também o trajeto das fibras ópticas, de acordo com a demanda de clientes, visando reduzir os seus custos de instalação.

1.1 Justificativa

Após levantamento bibliográfico, pode-se inferir, que há uma escassez de *softwares* que auxiliem na decisão do projeto da fibra óptica, o que engloba desde os caminhos que a fibra irá percorrer até o posicionamento dos *splitters* que irão atender aos clientes. Conversando com profissionais que trabalham no ramo, tomou-se conhecimento de que até mesmo em provedores de internet de médio e grande porte, o processo de definição do trajeto que os cabos de fibra óptica irão percorrer, utilizam apenas ferramentas tradicionais de desenho assistido por computador (CAD), tais como AutoCAD, FreeCAD ou LibreCAD. Portanto, se faz desejável e relevante o desenvolvimento de um *software* que, utilizando um mapa da localidade desejada, características e restrições da rede óptica do tipo PON e técnicas de otimização computacional, crie soluções para o projeto de uma PON bem como informações acerca da previsão do custo e do desempenho esperados. Por fim, implementar tal *software*, o mesmo permitirá aplicar diversos conceitos aprendidos no curso como,

¹ <https://cgi.br/>

² <http://www.ipea.gov.br/>

por exemplo: projeto e análise de algoritmos, estrutura de dados, redes de computadores, inteligência computacional, dentre outros conhecimentos aprendidos ao longo do curso.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um *software* para elaborar de forma automatizada o projeto de rede óptica passiva (PON) para uma determinada localidade, visando maximizar o alcance da cobertura de clientes, bem como minimizar o custo de sua implantação.

1.2.2 Objetivos Específicos

Dentre os objetivos específicos, estão:

- Obter um mapa da malha viária de algumas cidades, visando transformá-lo em um grafo a ser utilizado pelo algoritmo;
- Modelar as características de uma solução válida para o problema, para que o algoritmo possa determinar o trajeto de fibras ópticas e a posição dos distribuidores de acordo com a demanda de clientes para, assim, fornecer uma solução compatível com as restrições impostas;
- Codificar as características físicas de uma rede óptica passiva, para que o algoritmo possa avaliar o desempenho esperado para a PON e o custo relacionado à sua implantação;
- Gerar um relatório com o resultado final, exibindo o trajeto proposto para os cabos da rede, o posicionamento dos distribuidores bem como as métricas de custo e desempenho esperado.

1.3 Estrutura do Trabalho

Esta monografia de conclusão de curso é organizada em 7 capítulos. No [Capítulo 2](#), após a contextualização feita na Introdução, são apresentados os conceitos teóricos necessários para a compreensão do trabalho, como redes ópticas passivas, conceitos sobre a fibra óptica e também sobre como é feito o projeto tradicional da rede. No [Capítulo 3](#), são explicadas as ferramentas utilizados no desenvolvimento do *software* proposto, como linguagens de programação, bibliotecas, componentes e plataformas de desenvolvimento. Já no [Capítulo 4](#), são tratados os tópicos sobre o desenvolvimento do trabalho, como foram obtidos os dados do problema, sua manipulação e a estrutura de dados utilizada para

armazená-los, também são explicadas algumas funções básicas do programa. O [Capítulo 5](#) apresenta os resultados obtidos a partir dos experimentos realizados, a fim de avaliar o uso do *software* para automação do projeto de fibras ópticas. No [Capítulo 6](#) é feita uma alusão aos possíveis trabalhos futuros e, por fim, no [Capítulo 7](#), são apresentadas as considerações finais, seguidas pelas referências bibliográficas utilizadas neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Tecnologias de Acesso à Internet

2.1.1 DSL

Tecnologias de acesso tradicionais são estruturadas com elementos metálicos, como cabo de par trançado (UTP) ou cabo coaxial. Um exemplo de tecnologia de comunicação digital baseada em infraestrutura de cabos de par trançado é a *Digital Subscriber Line* (DSL), largamente difundida e muito popular como tecnologia de acesso à internet nos lares brasileiros. A [Tabela 1](#) mostra algumas tecnologias DSL e suas respectivas taxas de *downstream*, *upstream* e distância máxima de transmissão.

Tabela 1 – Tecnologias DSL

Tipo de DSL	<i>Downstream</i>	<i>Upstream</i>	Distância Máxima
ASDL	8 Mbps	800 Kbps	5.500 m
HDSL	1,54 Mbps	1,54 Mbps	3.650 m
MSDL	2 Mbps	2 Mbps	10.700 m
RADSL	7 Mbps	1 Mbps	5.500 m
SDSL	2,3 Mbps	2,3 Mbps	6.700 m
VDSL	52 Mbps	16 Mbps	1.200 m

Fonte: ([TYSON, 2001](#))

Popularmente, conexões DSL são associadas com acesso à internet por banda larga e presentes em muitos lares brasileiros. Entretanto, segundo pesquisa do Comitê Gestor da Internet no Brasil (CGI.br):

"A conexão DSL via linha telefônica estava presente em 19% dos domicílios com Internet, proporção que era de 26%, em 2015. Já a conexão via cabo de TV ou fibra ótica foi citada em 28% dos domicílios com Internet, quatro pontos percentuais a mais do que em 2015 (24%). Outros tipos de conexão de banda larga fixa apresentaram estabilidade em relação à edição anterior do estudo, como a via rádio (10%) e a via satélite (7%), enquanto a conexão discada foi utilizada, em 2016, por apenas 1% dos domicílios com Internet." FONTE: ([NIC.BR, 2016](#))

2.1.2 Tecnologias PON

Redes de fibra óptica têm demonstrado uma crescente demanda, por suas vantagens em relação a redes que utilizam cabos metálicos. Vantagens essas que incluem: menor perda de dados e maior banda (até 2,5 Gbps em *downstream* e 1,25 Gbps em *upstream* no padrão G-PON), bem como apresentar menos ruído por não sofrer interferências eletromagnéticas e possuir maior alcance ([FARMER et al., 2016](#)). Comercialmente, redes ópticas do tipo PON tem uma distância máxima de alcance entre 10 e 20 km, porém outras tecnologias

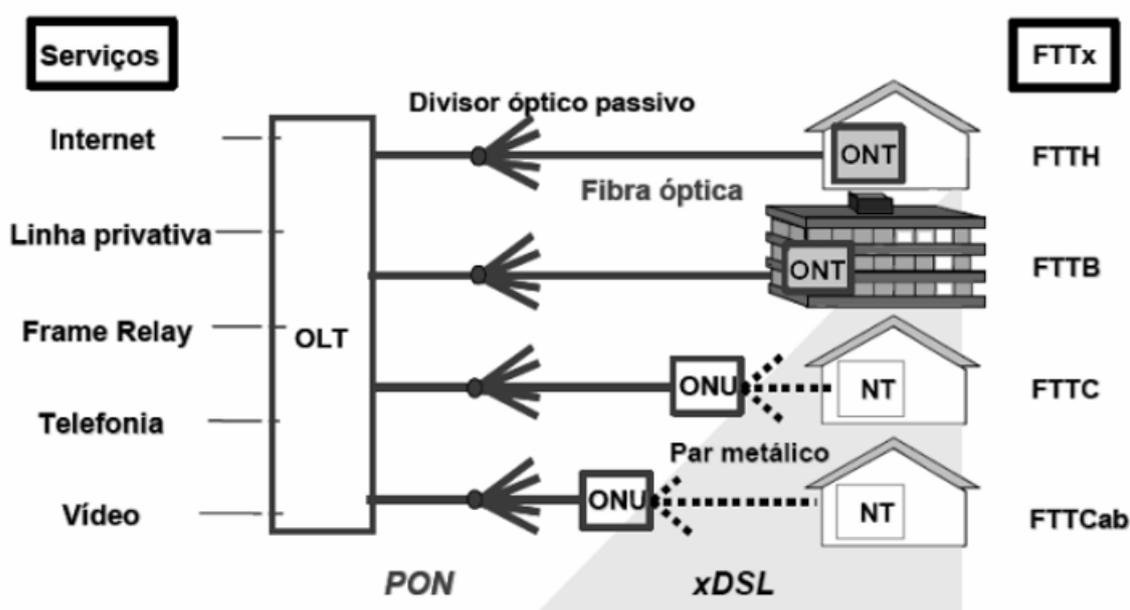
de fibras ópticas permitem até mesmo redes transoceânicas com dezenas de milhares de quilômetros.

2.1.2.1 Definição Rede PON

Segundo [Pinheiro \(2017\)](#), redes PON apresentam elementos passivos, ou seja, não utilizam amplificadores ativos ao longo da rede. Os elementos de uma rede PON são o *Central Office (CO)*, escritório central da operadora de onde a rede parte; *Optical Line Terminal (OLT)*, equipamento geralmente utilizado no CO que concentra e distribui a rede; *splitters* que tem a função de distribuir o sinal óptico de uma fibra para várias outras, aumentando assim a ramificação da rede PON e deixando-a com mais capilaridade e *Optical Network Unit (ONU)*, equipamento que termina a rede e que pode ser instalado na casa do usuário fazendo a interface optico-eletrica.

Segundo [Takeuti \(2005\)](#) uma PON (*Passive Optical Network*) é uma rede de acesso em fibra óptica interligada, em topologia estrela e na configuração ponto-multiponto, que possui somente componentes ópticos passivos entre o OLT e a ONU. O termo passivo se origina da principal característica dessa rede, uma vez que não existem elementos ativos, isto é, elementos que necessitem de energia elétrica para seu funcionamento entre os equipamentos do cliente e do prestador de serviço, conforme pode ser observado na [Figura 1](#).

Figura 1 – Arquitetura Básica de uma Rede PON.



Fonte: ([FERNANDES., 2018](#))

2.1.2.2 EPON e GPON

A rede PON possui duas principais versões, a GPON (*Gigabit Passive Optical Network*) e a EPON (*Ethernet Passive Optical Network*). Uma comparação entre as duas tecnologias pode ser vista na [Tabela 2](#).

Tabela 2 – Comparação entre tecnologia EPON e GPON

Diferenças	EPON	GPON
Padrão e protocolo	IEEE 802.3ah, Ethernet	ITU-T G.984, GEM (variados protocolos)
Tamanho dos pacotes	1518 bytes	de 53 até 1518 bytes
Velocidade de transmissão	Simétrica Down – 1.25 Gbps Up – 1.25 Gbps	Assimétrica Down – 2.5 Gbps Up – 1.25 Gbps
Comprimento de onda	Down – 1490 nm Up – 1310 nm	Down – 1490 nm Up – 1310 nm
Distâncias	Até 20km	Até 20km
Divisões	16, 32 ou 64	32, 64 ou 128
Eficiência	67%	93%
Compatibilidade	Tecnologia Ethernet permite compatibilidade entre vários fabricantes	A ITU-T ¹ nao aconselha compartilhar equipamentos de diferentes fabricantes

2.2 Componentes Básicos da rede PON

2.2.1 Cabo Óptico

Segundo [Pinheiro \(2017\)](#) a estrutura básica de cabos de fibra óptica é :

A fibra óptica é composta por um material dielétrico, normalmente sílica (vidro com impurezas) ou mesmo plástico capaz de confinar a luz em seu interior, possibilitando que pulsos luminosos possam ser codificados, estabelecendo uma comunicação entre suas extremidades.

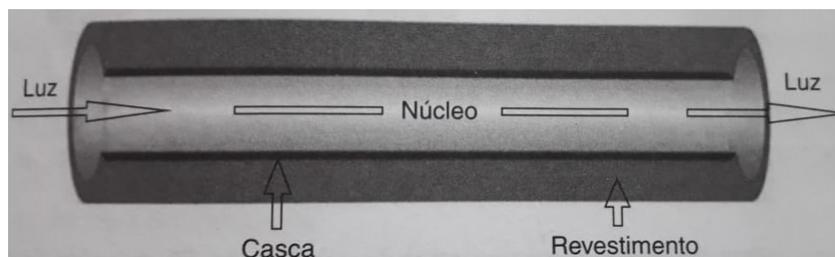
Sua estrutura básica é formada por uma região central, chamada de núcleo, constituída de um material dielétrico (em geral, vidro), envolta por uma camada, também de material dielétrico (vidro ou plástico) com índice de refração ligeiramente inferior ao do núcleo.

A [Figura 2](#) demonstra como a luz é transmitida dentro da fibra óptica e também os componentes básicos da fibra.

2.2.2 Propagação da Luz na Fibra Óptica

Não é possível utilizar um modelo único e preciso devido a dualidade da luz ([CHESMAN; MACEDO; ANDRÉ, 2004](#)), que pode tanto se comportar como partículas eletromagnéticas que se movem em alta velocidade (denominadas Fótons) ou, como uma onda eletromagnética. Com relação à fibra óptica a melhor maneira para explicar como elas funcionam é descrita na Lei de Snell. Segundo [Pinheiro \(2017\)](#):

Figura 2 – Transmissão da luz através da fibra.



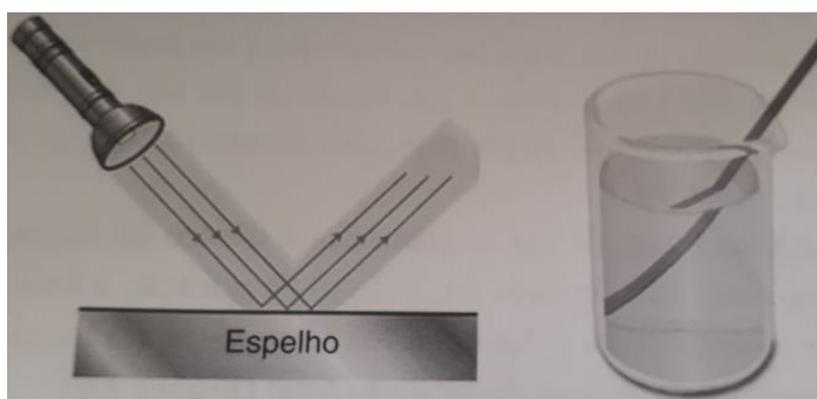
Fonte: [Pinheiro \(2017\)](#)

A lei de Snell afirma que a relação entre o seno do ângulo de incidência e o seno do ângulo de refração é igual à relação entre as velocidades de propagação da onda nos dois meios respectivos. Isso é igual a uma constante, que representa a relação entre o índice de refração do segundo meio e do primeiro. Ou seja, o funcionamento da fibra óptica está baseado em dois fenômenos: a reflexão e a refração.

A reflexão e a refração da luz são fenômenos ópticos relacionados com a forma como a luz se propaga no meio. Quando a luz incide sobre uma superfície, pode ser refletida (feixe de luz de uma lanterna na direção de um espelho) ou refratada (um lápis no interior de um copo com água), gerando diferentes impressões visuais.

A [Figura 3](#) demonstra a diferença entre reflexão e refração. À esquerda a reflexão da luz de uma lanterna em um espelho, à direita, a refração da luz causa um efeito visual no lápis quando observado em dois meios diferentes, água e ar.

Figura 3 – Reflexão e Refração.



Fonte: [Pinheiro \(2017\)](#)

A fibra óptica segue estes dois princípios para transmitir a informação, irradiando a luz por dentro do cabo de uma ponta à outra mesmo com curvaturas pelo percurso desde que tais curvaturas não interfiram de maneira prejudicial no ângulo de incidência para a reflexão da luz no interior da fibra.

2.2.3 Transmissores e Receptores Ópticos

Segundo Pinheiro (2017) os transmissores e receptores ópticos são os componentes responsáveis pela transformação do sinal elétrico em óptico e vice-versa, sendo que o transmissor é responsável por converter o sinal elétrico em óptico e o receptor é responsável por converter o sinal óptico em elétrico. As principais características do receptor óptico são a sua sensibilidade, a taxa de erros de Bit ou BER (*Bit Error Rate*) e o campo de variação dinâmico. A sensibilidade é o nível mínimo e máximo de potência que um sinal precisa apresentar para ser decodificado com um número limitado de erros de bit. A taxa de erros de Bit é o número máximo de erros de bit permitido em relação à taxa de bits transmitidos entre o transmissor e o receptor. O campo de variação dinâmico define a potência média recebida dentro da BER do detector.

2.2.4 Emendas Ópticas

Segundo Pinheiro (2017) uma emenda é uma junção permanente de dois segmentos de fibras ópticas. Elas são necessárias para ampliar ou dar continuidade a um lance de fibra óptica quando o comprimento do sistema é maior que o comprimento de cabo disponível. A emenda também é necessária para inserir novos componentes ópticos no sistema ou na ocorrência de ações corretivas devido à rompimentos no cabo óptico. Há dois métodos para a emenda de fibras ópticas: por fusão e mecânica.

Segundo Pinheiro (2017) a emenda por fusão é o processo que apresenta melhor desempenho quanto à perda por inserção e à perda de retorno. É utilizado um equipamento na fibra óptica nua que, após uma sequência de procedimentos preparatórios, é introduzida na máquina e submetida a um arco voltaico que eleva a temperatura nas faces das fibras, o que provoca seu derretimento e soldagem. A emenda mecânica é baseada no alinhamento das fibras com o uso de estruturas mecânicas. O método é uma alternativa à técnica de emenda por fusão e não requer o uso da máquina por fusão (equipamento de custo mais elevado).

2.2.5 Divisores Ópticos (*Splitters*)

Segundo Pinheiro (2017) os divisores ópticos têm como função dividir o sinal óptico transmitido a partir de uma fonte em comum. O divisor óptico, comumente chamado de *splitter*, é um elemento passivo utilizado em redes ópticas que realiza a divisão do sinal óptico proveniente de uma fibra para N fibras, em razões usuais de 1:2, 1:4, 1:8, 1:16, 1:32 e 1:64. Um exemplo de divisor óptico pode ser visto na Figura 4, que ilustra um *splitter* 1:4.

Figura 4 – Exemplo *splitter* 1:4.

Fonte: [Pinheiro \(2017\)](#)

2.2.6 Distribuidor Interno Óptico (DIO)

Segundo [Oliveira \(2018\)](#):

O DIO (Distribuidor Interno Óptico) é onde podem ser alocados os *splitters* e terminações da fibra. O DIO da central aloca o *splitter* primário e está dentro do *rack*, já o DIO do cliente, comumente chamado de TOA (Terminal Óptico do Assinante), é onde aloca-se a fibra que chega às residências.

2.3 Projeto Tradicional de Redes de Fibra Óptica

No site da Furukawa⁰, fabricante de implementos para redes de computadores, é descrito um passo a passo para o planejamento de Projeto de redes de fibra óptica, conforme este passo a passo é utilizado pela maioria das empresas fornecedoras deste serviço. Os passos são:

- Passo 1) Definir o limite da área do projeto e a demanda potencial;
- Passo 2) Definir a Razão de Divisão do projeto: 1:32 / 1:64 / 1:128;
- Passo 3) Definir a Topologia, quais *splitters* e onde serão instalados;
- Passo 4) Dividir a área delimitada na quantidade de ‘células’ prevista, normalmente utilizando a premissa de que a área tem demanda com densidade uniforme;
- Passo 5) Posicionar a Caixa de Emenda (CEO) e *splitters* de primeiro nível;
- Passo 6) Desenhar as rotas da Rede Primária (cabos ‘*feeder*’) e planejar a quantidade de fibras em cada trecho, tanto de fibras para pronto uso quanto reserva para expansões, considerando o planejamento para o futuro;

⁰ <https://www.efurukawa.com/br/monte-sua-rede/resultado-preconf?solutionId=400002modal-subscribe>

Passo 7) Posicionar as Caixas Terminais (CTO) e desenhar as rotas da rede secundária ou distribuição, esta fase ocorre sob demanda;

2.4 Problema de Localização de Instalações

Segundo [Cornuéjols, Nemhauser e Wolsey \(1983\)](#) um problema econômico de grande importância prática é escolher a localização de instalações, como plantas industriais ou armazéns, a fim de minimizar o custo (ou maximizar o lucro) e satisfazer a demanda por alguma mercadoria. Em geral, existem custos fixos para alocar as instalações e custos de transporte para distribuir as mercadorias entre as instalações e os clientes. Este problema tem sido extensivamente estudado na literatura e é comumente referido como o problema de localização de plantas ou problema de localização de instalações. Quando cada instalação potencial tem uma capacidade que é a demanda máxima que pode fornecer, o problema é chamado de problema de localização de instalações capacitadas. Quando a hipótese de capacidade não é necessária, temos o problema de localização de instalação simples. Então se considerarmos a instalação do *splitter* secundário como a localização de instalação e a mercadoria como a conexão, podemos reduzir o problema do Projeto de PON ao Problema de Localização de Instalações [Cornuéjols, Nemhauser e Wolsey \(1983\)](#).

2.5 Algoritmo Guloso

Segundo [Cormen et al. \(2001\)](#) algoritmos de otimização normalmente seguem uma sequência de passos e em cada passo eles possuem um conjunto de soluções possíveis para serem escolhidas, o algoritmo guloso sempre escolhe a solução que é melhor naquele momento, sem levar em consideração as iterações futuras ou, seja, a sua escolha é sempre o ótimo local levando em consideração a função objetivo utilizada, a qual pode ser para minimizar ou maximizar algo. Algoritmos gulosos não costumam retornar a solução ótima porém é bastante poderoso e funciona bem para uma grande variedade de problemas, além de possuir uma implementação relativamente simples. Segundo ([PARBERRY, 1995](#)):

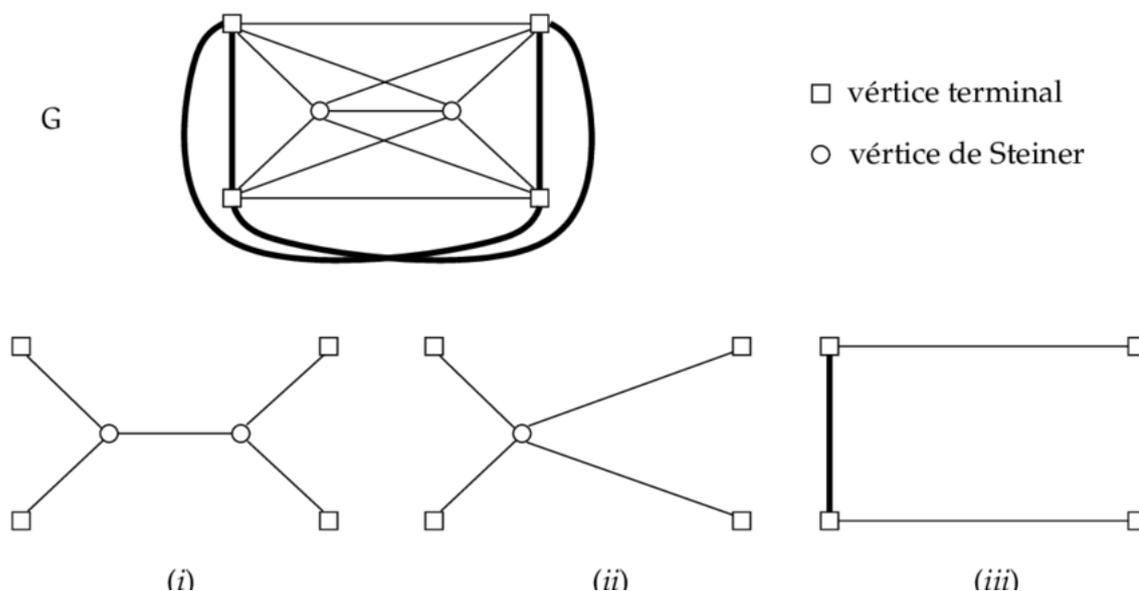
Um algoritmo guloso começa com uma solução para um subproblema muito pequeno e aumenta sucessivamente para uma solução para o grande problema. O aumento é feito de forma "gulosa", isto é, prestando atenção ao ganho local ou de curto prazo, sem levar em conta se isso levará a uma boa solução global ou de longo prazo.

[A maioria dos algoritmos gulosos são enganosamente simples.] Uma coisa que você notará sobre os algoritmos gulosos é que eles geralmente são fáceis de projetar, fácil de implementar, fácil de analisar e eles são muito rápidos, mas são quase sempre difíceis de provar.

2.6 Árvore de Steiner

O problema da Árvore de Steiner (GAREY, 1977) consiste em dado um grafo não direcionado $G = (N, E)$ de arestas com custos positivos e um conjunto T que pertence a N de nós terminais (folhas) busca-se encontrar a árvore de custo mínimo que conecta todos os nós, sendo classificado como problema NP-Difícil. O algoritmo mais comumente utilizado para calcular a árvore de Steiner em tempo polinomial é a heurística Dreyfus-Wagner (DREYFUS, 1971). Segundo (GILBERT; POLLAK, 1968) uma árvore mínima para os pontos A_1, \dots, A_n em um plano é uma árvore que conecta estes pontos utilizando linhas com o tamanho mínimo possível. Árvores mínimas de Steiner são difíceis de se achar pois uma árvore que possui um tamanho mínimo local pode não obter o mínimo absoluto. A Figura 5 mostra um exemplo de Árvore de Steiner.

Figura 5 – Exemplo de uma Árvore de Steiner.



Fonte: [Martinez e Soares \(2004\)](#)

2.7 Trabalhos Relacionados

Durante a fase de levantamento do referencial teórico, foram encontrados vários trabalhos que tentavam reduzir os custos de redes PON com diferentes abordagens e em diferentes instâncias.

Em [Silva et al. \(2001\)](#) é proposta uma solução para o planejamento de expansão de redes de fornecimento de energia elétrica a longo prazo, o qual normalmente tem como problemas a determinação de onde e quando instalar novos equipamentos para encontrar os melhores resultados econômicos e operacionais. A dificuldade principal neste

problema é existência de variáveis de investimento inteiras as quais requerem o uso de algoritmos combinatórios. Outra dificuldade está nas considerações dinâmicas quando se tem mais um ano dentro do tempo de planejamento. A implementação feita foi baseada no algoritmo de Busca Tabu e a sua função objetivo representa os custos de investimento em novas instalações de transmissão (novas linhas de transmissão, novos transformadores e outros) com penalidade por sobrecargas de circuito na rede, onde a penalidade deve ser suficientemente grande para garantir que a sobrecarga seja zero, ou próxima de zero na solução ótima. O modelo também possui restrições para representar o modelo linear da rede de energia segundo as leis de Kirchhoff (KCL e KVL). Para a validação e demonstração do modelo foram utilizados dois estudos de plano de expansão de rede reais que já haviam sido feitos e possuíam suas soluções ótimas já definidas, em ambos os casos foi possível encontrar a solução ótima e também algumas soluções subótimas.

Em Khan (2005) é proposto um algoritmo guloso para a resolução da instalação de redes PON que leva em consideração um PDF-G (*Population Density Function Embedded Graphs*) que é um grafo que deve representar corretamente um terreno com sua população e também os requerimentos de largura de banda dos usuários naquele grafo. O algoritmo funciona em dois passos: primeiro seleciona a posição das ONTs (*optical network terminals*); em seguida seleciona a posição dos *splitters* com o objetivo de diminuir os custos e maximizar o número de clientes atingidos. Os resultados experimentais revelam que os algoritmos propostos economizam em média 45% a 65% do custo de implantação (fibra, equipamento, etc.).

Em Li e Shen (2008) é feito um trabalho para a diminuição de custos na instalação de redes PON ainda em planejamento. Para isso eles propõem um algoritmo chamado RALA (*Recursive Association and Location Algorithm*), cujo o objetivo é achar um ótimo global ou local com foco em minimizar o tamanho do enlace de fibra. Foram feitas simulações para verificar os resultados do algoritmo, que indicaram ser efetivo e com uma redução entre {50% - 70%} no custo de implantação da rede PON em comparação com a referência utilizada.

Em Kokangul e Ari (2011) é proposto um algoritmo genético com o objetivo de maximizar o número de clientes e minimizar os custos de instalação da fibra. No referido trabalho foram considerados uma cidade específica e os atributos físicos da fibra como sua atenuação, a atenuação dos *splitters* e o limite de atenuação do sinal para que o cliente tenha acesso à rede. As variáveis de decisão do algoritmo genético eram a posição dos nós primários e secundários, a posição do *splitter* dos nós primários/secundários e a quantidade de clientes atendidos pelos nós secundários, que são os nós finais. Neste caso para instâncias pequenas foi possível encontrar a solução ótima, já em instâncias grandes nem sempre foi encontrada a solução ótima.

Em Ahmad et al. (2011) é apresentado um algoritmo genético para a solução do

problema porém, o seu objetivo é a diminuição dos gastos energéticos após a instalação da fibra, levando em consideração que os maiores gastos energéticos da rede são nos transceptores ópticos e nos *switches* eletrônicos (que ficam entre os nós intermediários e os finais). Com esta abordagem foi encontrada soluções subótimas próximas ao ótimo.

Em [Agata e Nishimura \(2012\)](#) é proposto um novo algoritmo para encontrar soluções sub-ótimas para o problema de planejamento de Rede de Distribuição Óptica para sistemas PON. Neste algoritmo é levando em conta o custo de construção na rede que normalmente é maior que o custo do cabo de fibra óptica. O algoritmo proposto utiliza dois algoritmos relacionados a grafos como técnica principal para se obter os resultados, um deles é o *Graph Clustering*, o outro é um método de construção de Árvore de Steiner (vide [Seção 2.6](#)) no grafo. A análise de *cluster* (agrupamento) é baseada no agrupamento de um grande número de dados em múltiplos grupos, chamados de *cluster*, de forma que os dados em um mesmo grupo sejam semelhantes de alguma forma. Este é um método de aprendizado não supervisionado e uma técnica de análise estatística e mineração de dados muito utilizada em várias áreas. O algoritmo do *Graph Clustering* é uma extensão do algoritmo básico de *clustering*. Na primeira parte do *software* computa-se com o *Graph Clustering* as melhores posições para a instalação dos *splitters*. Após a escolha do posicionamento dos *splitters* é possível calcular o menor caminho que conecta todos os *splitters* e o escritório central (CO). Para tal um algoritmo comum como Dijkstra poderia ser utilizado. Porém, esta abordagem levaria em conta apenas o tamanho total da fibra e não o custo de sua construção. Para evitar isso usa-se como método de construção um algoritmo de Árvore de Steiner. A validação do *software* foi feita testando-o em um modelo real envolvendo um mapa viário da área urbana do Japão, com uma previsão realista de demanda. O referido autor reporta que o algoritmo proposto foi capaz de gerar uma rede PON subótima.

Em [Pal et al. \(2014\)](#) o trabalho é focado na instalação em áreas rurais, o qual traz uma nova dimensão para o problema de instalação de redes PON. Uma clusterização estratégica das ONUs para formar o *layout* da rede e então a minimização do comprimento da fibra são os objetivos principais deste trabalho. Neste trabalho foi utilizado uma clusterização parecida com a de [Agata e Nishimura \(2012\)](#), porém neste caso ela foi utilizada tanto para definir a posição dos *splitters* quanto para determinar a localização das ONUs e conectá-las aos *splitters*. A partir dessas informações foi possível calcular o tamanho mínimo da fibra óptica necessária com uso de programação linear e método heurístico, tendo sido obtidos valores similares, com o método heurístico se desviando entre 2,5 % e 8,5 % do ótimo, porém reduzindo o tempo computacional. Com tais propostas foi obtido um bom resultado diminuindo o tamanho total de cabos para a instalação da PON.

Em [Arévalo, Hincapié e Gaudino \(2017\)](#) foi feita uma otimização da instalação da rede e também uma comparação entre os custos das redes GPON, XGPON, NGPON2 e UDWDM PON. Os autores consideraram uma região metropolitana com grande número

de usuários e a região da cidade selecionada foi tratada como um grafo ponderado. O algoritmo para a minimização dos custos da rede foi rodado para os protocolos acima mencionados e em todos os casos foi encontrada a topologia ótima.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais e métodos utilizados no desenvolvimento deste projeto, tais como o método para obtenção das informações viárias do *OpenStreetMap* a partir do *Overpass-turbo*; e para o desenvolvimento do *software*, como linguagem de programação escolhida, bibliotecas usadas, além da base de dados escolhida e ferramentas tanto de *hardware* quanto *software* utilizadas para a execução deste projeto.

3.1 *OpenStreetMap*

A representação da topologia da cidade (ruas e avenidas) foi feita através de um grafo, de forma que a instalação do anel principal da rede PON pode ser considerado um ciclo hamiltoniano, já que o anel passa pelos vértices apenas uma vez e retorna ao ponto de partida.

O grafo do local pode ser obtido utilizando-se a base pública do *OpenStreetMap*, um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo, com informações catalogadas pela comunidade (OSM, 2006). Por ser de código livre, qualquer pessoa pode utilizar os dados para qualquer fim, desde que credite a autoria ao projeto OSM e seus contribuidores. Com o *OpenStreetMap* é possível exportar uma parte do mapa, que pode então ser lida pelo *software Java OpenStreetMap Editor*¹. Dentre suas muitas funções, este editor de mapas tem a opção de destacar as ruas e seus cruzamentos, bem como ocultar/remover as diferentes camadas de informações disponibilizadas, facilitando a geração do grafo correspondente à localidade desejada.

OSM é um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo. O OSM é uma ferramenta para criar e compartilhar informação em um mapa. Os mapas foram desenvolvidos e são mantidos pela comunidade voluntária de mapeadores, que inserem e revisam dados de receptores GPS portáteis, fotografias aéreas, imagens de satélite e outras fontes livres. Os mapeadores, com seu conhecimento local, editam os mapas com *softwares* abertos como o JOSM. Todos os mapas, dados descritivos e metadados ofertados pelo OSM são dados abertos, disponíveis sob uma licença Open Database License².

¹ <https://josm.openstreetmap.de/>

² <https://opendatacommons.org/licenses/odbl/index.html>

3.2 Overpass API e Overpass-turbo

O Overpass API é uma API apenas para leitura, que exibe partes selecionadas dos dados do mapa do OSM. Ele age como um banco de dados na *web*: o cliente envia uma consulta à API e recebe o conjunto de dados que corresponde à consulta.

Overpass-turbo é uma ferramenta *web* para obtenção de dados do *OpenStreetMap*. Ele executa todas as consultas do Overpass API e mostra os resultados em um mapa interativo. Com o Overpass-turbo também é possível exportar os resultados em diversos formatos, como XML, JSON entre outros.

3.3 Josm

Segundo JOSM³:

JOSM (*Java OpenStreetMap Editor*) é um aplicativo de desktop originalmente desenvolvido por Immanuel Scholz e atualmente mantido por Dirk Stöcker. Sua homepage está localizada em josm.openstreetmap.de. Embora tenha uma curva de aprendizado relativamente íngreme, o JOSM é popular entre os editores experientes graças aos seus plugins e estabilidade. Existem outros editores para dados do *OpenStreetMap*, como o iD ou o Potlatch 2. O JOSM é um editor rico em recursos com uma interface que pode parecer complexa no início.

3.4 Python

A linguagem utilizada para desenvolver o trabalho foi Python. A linguagem Python começou a ser desenvolvida em 1990 por Guido van Rossum. A linguagem foi baseada em outra linguagem da época, chamada ABC (COELHO, 2007).

Segundo Borges (2010), a linguagem Python possui várias estruturas de alto nível, como listas, dicionários, números complexos e uma grande coleção de módulos prontos, além dos *frameworks* de terceiros que podem ser usados. Python também apresenta características de outras linguagens modernas, como geradores, metaclasses e unidades de testes. Ela também é multiparadigma, suportando programação modular, funcional e também orientação a objetos. Todas as variáveis em Python são objetos.

Python é uma linguagem interpretada através de *bytecode* pela máquina virtual Python, o que acaba tornando o código portátil. Além disso, Python é um *software* de código aberto, utilizando uma licença compatível com a GPL (*General Public License*) porém, menos restritiva, permitindo seu uso em produtos proprietários por exemplo. A especificação da linguagem é mantida pela *Python Software Foundation* (PSF) (BORGES, 2010).

³ <https://wiki.openstreetmap.org/wiki/JOSM>

3.5 Bibliotecas utilizadas

Uma das principais vantagens de Python quando comparada com outras linguagens é a grande gama de bibliotecas e *frameworks* já existentes para ela, tornando assim mais fácil e rápido o desenvolvimento. Para o desenvolvimento do trabalho, várias bibliotecas foram utilizadas, desde bibliotecas embutidas quanto bibliotecas externas.

As bibliotecas utilizadas foram a *gmplot*⁴, para plotar dados no *google maps*, *geopy*⁵, para obter distancias utilizando latitude e longitude de forma precisa, *networkx*⁶, para criar o grafo que representa a cidade, *configparser*⁷, para ser ler o arquivo de configuração, *matplotlib*⁸ para plotar os dados de forma gráfica e quando necessário criar imagens de forma mais ágil que o *gmplot*. Posteriormente também foi utilizada a biblioteca *pyexcel*⁹ para gerar o arquivo de demandas.

3.6 Metodologia

Após o levantamento e estudo do referencial teórico sobre o projeto de PON, passou-se à modelagem e projeto do sistema, na qual foi modelado as principais características do projeto da rede óptica, como o balanço de potência, a demanda da área a ser atendida e a localização do provedor. Também foram modelados os equipamentos relevantes para este trabalho, que são a fibra e os *splitters*, sendo que as características específicas de cada equipamento são definidas em um arquivo de configuração, onde é passado o decaimento e o seu preço.

Posteriormente, desenvolveu-se a fase de implementação e testes do sistema, visando identificar gargalos de desempenho, *bugs* e verificação do sistema para analisar se este estava realmente realizando as tarefas corretamente. Durante o trabalho a implementação foi modificada diversas vezes até que fosse alcançado um resultado factível.

Por fim, foram realizados experimentos para analisar a eficácia do sistema, o qual foi comparado com um projeto de fibra óptica realizado manualmente e posteriormente a verificação dos resultados com o trabalho manual de Oliveira (2018), foram feitos outros testes comparando diferentes entradas para o programa.

⁴ <https://pypi.org/project/gmplot/>

⁵ <https://geopy.readthedocs.io/en/stable/>

⁶ <https://networkx.github.io/>

⁷ <https://docs.python.org/3/library/configparser.html>

⁸ <https://matplotlib.org/>

⁹ <http://docs.pyexcel.org/en/latest/>

3.7 *Hardware*

O computador utilizado para o desenvolvimento do *software* e também para a realização dos testes foi um *nootbook* Acer, modelo Nitro AN515-51, com processador Intel Core i5-7300HQ com *clock* de 2.50GHz, memória RAM de 8 GB e sistema operacional Windows 10.

Também foram realizados testes em outro *nootbook* Dell modelo Inspiron 1510, com processador Intel Core i5-5200U com *clock* de 2.50Ghz, memória RAM de 8,00 GB e sistema operacional Ubuntu 16.04.

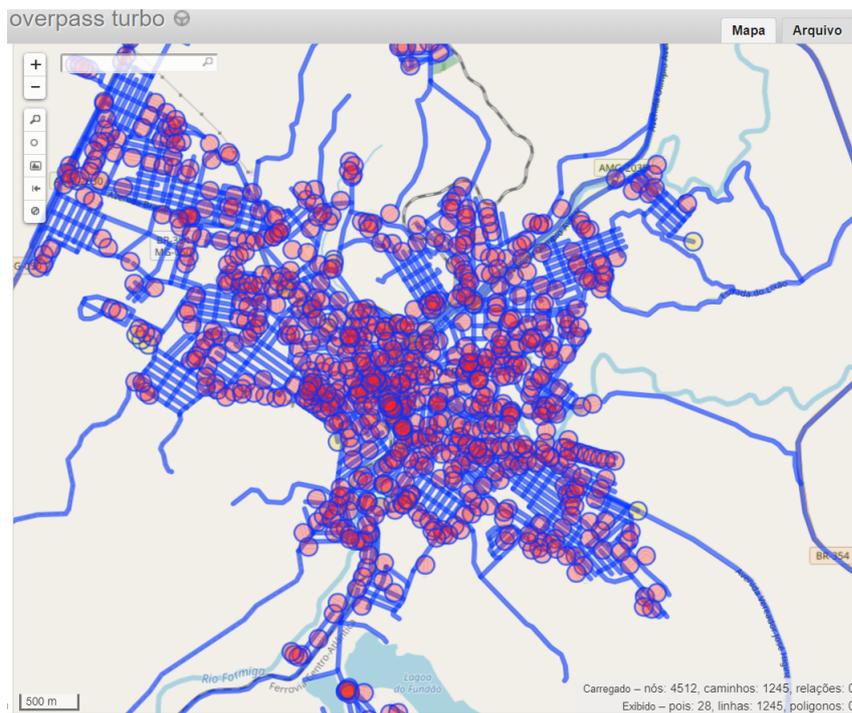
4 PROJETO E DESENVOLVIMENTO

Foram conduzidos estudos bibliográficos para compreensão dos modelos físicos de propagação de sinais sobre a fibra óptica, levando em conta as perdas de sinal e sua atenuação.

4.1 Mapa da cidade

O mapa da cidade foi manipulado a partir do Overpass-Turbo¹. Utilizando o assistente que existe no site foi obtido um *script* utilizado para gerar um XML com os dados necessários para formar as ruas. O resultado também pode ser observado visualmente como na Figura 6.

Figura 6 – Mapa de Formiga-MG gerado após execução do *script* gerador de XML.



Fonte: Próprio Autor

O XML gerado possui três informações: a primeira são componentes das vias como semáforos e rotatórias; a segunda são as ruas com seus nomes e os identificadores dos pontos que as formam; e a terceira são os pontos que formam as ruas onde é informado a latitude e longitude de cada ponto. A Figura 7 demonstra um exemplo do XML.

¹ <http://overpass-turbo.eu/>

Figura 7 – Exemplo do XML gerado.

```

<!-- =====Primeira Parte===== -->
<node id="1528579856" lat="-20.4623109" lon="-45.4304755">
  <tag k="highway" v="traffic_signals"/>
</node>
<node id="1528613513" lat="-20.4903942" lon="-45.4358799">
  <tag k="highway" v="mini_roundabout"/>
</node>
<node id="1528613620" lat="-20.4665558" lon="-45.4331661">
  <tag k="highway" v="mini_roundabout"/>
</node>
<node id="1528613703" lat="-20.4584046" lon="-45.4410950">
  <tag k="highway" v="traffic_signals"/>
</node>
<node id="1528614196" lat="-20.4710270" lon="-45.4311545">
  <tag k="highway" v="traffic_signals"/>
</node>
<node id="1826561915" lat="-20.4627978" lon="-45.4302911">
  <tag k="highway" v="traffic_signals"/>
</node>
<node id="2500052784" lat="-20.4659289" lon="-45.4331284">
  <tag k="highway" v="traffic_signals"/>
</node>
<node id="2500052810" lat="-20.4663044" lon="-45.4332842">
  <tag k="highway" v="mini_roundabout"/>
</node>

<!-- =====Segunda Parte===== -->
<way id="31589583">
  <nd ref="1528579849"/>
  <nd ref="1528579833"/>
  <nd ref="1528579824"/>
  <nd ref="1528614088"/>
  <nd ref="5018467453"/>
  <nd ref="1528579858"/>
  <nd ref="1528613853"/>
  <tag k="highway" v="tertiary"/>
  <tag k="name" v="Rua Sete de Setembro"/>
  <tag k="source" v="IBGE"/>
</way>

```

(a) Primeira parte do XML

(b) Segunda parte do XML

```

<!-- =====Terceira Parte===== -->
<node id="1528617188" lat="-20.5730405" lon="-45.4762313"/>
<node id="4243766030" lat="-20.5742351" lon="-45.4780953"/>
<node id="1528617001" lat="-20.5685405" lon="-45.4709688"/>
<node id="1528617063" lat="-20.5708458" lon="-45.4728142"/>
<node id="1528617216" lat="-20.5710743" lon="-45.4731500"/>
<node id="1528617235" lat="-20.5694496" lon="-45.4716770"/>
<node id="1528617352" lat="-20.5705896" lon="-45.4725567"/>
<node id="1528617066" lat="-20.5681086" lon="-45.4703680"/>
<node id="1528617200" lat="-20.5682693" lon="-45.4706899"/>
<node id="1528617456" lat="-20.5678022" lon="-45.4693112"/>
<node id="1528617003" lat="-20.5675762" lon="-45.4665915"/>

```

(c) Terceira parte do XML

4.2 Manipulação do xml

A manipulação do XML foi feita utilizando a biblioteca *xml.etree.ElementTree* (*ET*), segundo a documentação do ET², o XML é um formato de dados hierárquicos e por isso a maneira mais comum de o representar é por meio de uma árvore, por isso o módulo possui duas classes, a *ElementTree* e a *element*, sendo que a primeira representa todo o documento XML e a segunda um único nó da árvore.

Com o arquivo XML gerado pelo Overpass-turbo, o primeiro passo foi remover as informações sobre os componentes da via que não são necessários neste trabalho, como por exemplo os semáforos. Em seguida foram gravados as ruas e os pontos que as formam. Para isso foram criadas duas classes, a classe Rua, que contém o identificador, o nome e uma lista de pontos que formam a rua e a classe PontosRua que contém o identificador, a latitude e a longitude deste ponto.

Inicialmente estes dados foram guardados em uma lista para que fossem feitos testes simples para verificar se as informações obtidas pelo Overpass-Turbo estavam corretas. Para isso foi utilizada a biblioteca *gmplot*³, que entre outras funções, pode gerar um arquivo *html* que desenha pontos e ligações entre eles no Google Maps. Para desenhar o ponto é necessário passar a sua latitude e longitude.

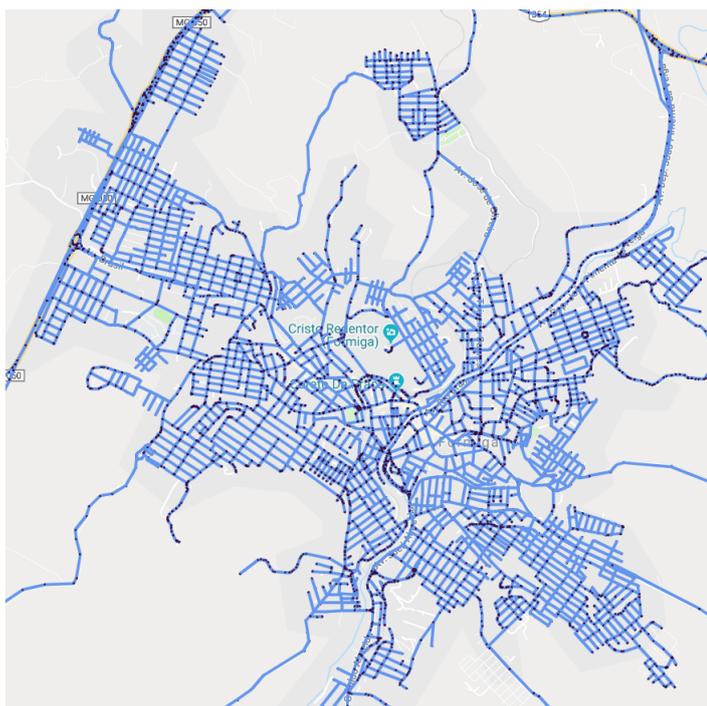
Como pode ser observado na [Figura 8](#), a qual mostra o mapa de Formiga- MG pelo

² <https://docs.python.org/2/library/xml.etree.elementtree.html>

³ <https://pypi.org/project/gmplot/>

Google Maps, os dados obtidos pelo Overpass-Turbo estavam corretos, pois foi possível refazer as ruas e os pontos e linhas são consistentes com o esperado. Também pode ser observado que algumas ruas não foram preenchidas, o que era esperado já que o *OpenStreetMap* é alimentado pela comunidade e não possui a mesma completude do Google Maps.

Figura 8 – Mapa de Formiga-MG obtido com a biblioteca *gmpplot* a partir dos dados do Overpass-Turbo



Fonte: Próprio Autor

Outro teste que também foi realizado para aferir a consistência dos dados obtidos pelo Overpass-Turbo foi pela seleção de ruas aleatórias. Uma vez escolhidas, foi feito o cálculo do comprimento dessa amostra de ruas utilizando a biblioteca *geopy*, que possui uma função para calcular a distância entre dois pontos pela sua latitude e longitude e após o cálculo do tamanho da rua com os dados do Overpass, comparou-se com o tamanho retornado pelo Google Maps.

4.3 Estrutura de dados

A partir da manipulação do XML com o ET, foi gerado um grafo utilizando a biblioteca *NetworkX* que é bem completa e possui várias funções já implementadas, segundo a documentação da *NetworkX*⁴:

⁴ <https://networkx.github.io/>

O NetworkX é um pacote Python para a criação, manipulação e estudo da estrutura, suas dinâmicas e funções de redes complexas.

O NetworkX fornece:

Ferramentas para o estudo da estrutura e dinâmica de redes sociais, biológicas e de infra-estrutura;

Uma interface de programação padrão e implementação gráfica que é adequada para muitas aplicações;

Um rápido ambiente de desenvolvimento para projetos colaborativos e multidisciplinares;

Uma interface para algoritmos numéricos existentes e códigos escritos em C, C++ e FORTRAN;

A capacidade de trabalhar de forma fácil com grandes conjuntos de dados fora do padrão.

Com o *NetworkX* você pode carregar e armazenar redes em formatos de dados padrão e não padrão, gerar vários tipos de redes aleatórias e clássicas, analisar a estrutura da rede, construir modelos de rede, projetar novos algoritmos de rede, desenhar redes e muito mais.

4.4 Caminho mínimo

Uma das funções que já existem na biblioteca *NetworkX* é a de caminho mínimo utilizando o algoritmo de Dijkstra que é descrito em (DIJKSTRA, 1959) e cujo pseudocódigo é dado pelo Algoritmo 1, descrito a seguir:

Algoritmo 1: Algoritmo de Dijkstra para cálculo do Caminho de Custo Mínimo em um Grafo

```

1  Atribua valor zero à estimativa do custo mínimo do vértice  $s$  (a raiz da busca) e
   infinito às demais estimativas;
2  Atribua um valor qualquer aos precedentes (o precedente de um vértice  $t$  é o vértice
   que precede  $t$  no caminho de custo mínimo de  $s$  para  $t$ );
3  repeat
4      Seja  $k$  um vértice ainda aberto cuja estimativa seja a menor dentre todos os
       vértices abertos;
5      Feche o vértice  $k$ ;
6      for Para todo vértice  $j$  ainda aberto que seja sucessor de  $k$  do
7          some a estimativa do vértice  $k$  com o custo do arco que une  $k$  a  $j$ ;
8          if caso esta soma seja melhor que a estimativa anterior para o vértice  $j$  then
9              substitua-a e anote  $k$  como precedente de  $j$ ;
10 until Enquanto houver vértice aberto.;
```

O algoritmo considera um conjunto c de menores caminhos, iniciado com um vértice inicial k . A cada passo do algoritmo busca-se nas adjacências dos vértices pertencentes a c aquele vértice com menor distância relativa a k e adiciona-o a c e, então, repetindo os passos até que todos os vértices alcançáveis por k estejam em c . Arestas que ligam vértices já pertencentes a c são desconsideradas.

4.5 Balanço de potência

O perfeito funcionamento de um enlace óptico requer a seleção adequada dos componentes a serem utilizados na rede (transmissor, receptor, fibra, conector, etc.). São, portanto, relevantes para garantir que o *link* óptico projetado atenda as exigências de potência do transmissor e sensibilidade do receptor, obedecendo a critérios de qualidade e confiabilidade como probabilidade de erro de bit e disponibilidade do sistema. A estimativa que é feita para se calcular se o receptor vai conseguir interpretar as informações do transmissor é chamada de Orçamento de Potência Óptica, que segundo (PINHEIRO, 2017) é a diferença entre a potência óptica (em decibéis) entregue pelo transmissor e a sensibilidade do receptor, incluindo-se as perdas dos componentes passivos e também uma margem de erro devido ao envelhecimento dos componentes, o que acarreta a um maior decaimento ao longo do tempo.

Segundo (PINHEIRO, 2017) a equação que descreve o Orçamento de Potência Óptica é dado na Equação 4.1:

$$O_p = P_{tx} - A_{fo} - P_c - P_e - P_p - M_s - S_{rx} > 0 \quad (4.1)$$

onde :

- O_P = Orçamento de Potência, em dB
- P_{Tx} = Potência de saída (Tx), em dBm
- A_{Fo} = Atenuação total da fibra óptica, em dB
- P_C = Perda total nos conectores, em dB
- P_E = Perda total nas emendas, em dB
- P_P = Perda total nos elementos passivos, em dB
- M_S = Margem de segurança em dB
- S_{Rx} = Sensibilidade (Rx), em dBm

4.6 Demanda de Clientes Simulada

Para que o programa fosse mais realista foi necessário definir a demanda da área a ser atendida no cenário usado nos experimentos. Primeiramente foi pensado em utilizar o *OpenStreetMap* para obter automaticamente a informação da quantidade de casas em cada área (ex.: numeração das casas e prédios em cada rua). Porém, atualmente são

poucas cidades que possuem tal informação cadastrada no *OpenStreetMap*, razão pela qual decidiu-se por simular uma demanda de clientes.

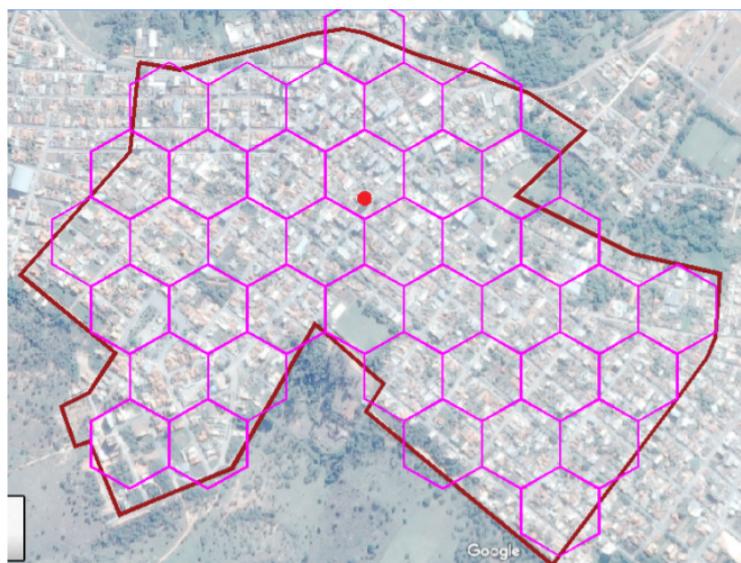
No caso deste trabalho, tal simulação foi feita considerando-se que a demanda de clientes é por rua e não por células de raio definido, como tipicamente é feito em projetos manuais. Consideramos gerar a demanda de clientes de duas formas: uniforme com a mesma demanda para todas as ruas ou com uma distribuição de cauda pesada (distribuição Weibull).

Com a demanda uniforme, ruas pequenas e grandes recebem a mesma quantidade de clientes. A distribuição de Weibull foi escolhida por ser uma distribuição de cauda pesada e que possibilitou calibrá-la para gerar uma demanda de clientes por rua proporcional à provável quantidade de casas de acordo com o comprimento de cada rua. Para esta calibragem foi considerado que, em média, um lote nas cidades analisadas possui 10 metros de largura (frente).

4.7 Atendimento às ruas

Outro passo que foi modificado à forma como é feito o projeto tradicional foi a divisão da área delimitada em células. Já que esta divisão foi considerada desnecessária pois a demanda foi distribuída em relação às ruas (Seção 4.6). A divisão tradicional é feita considerando que a densidade de clientes da área é uniforme e estas células também tem área igual. Um exemplo dessa divisão pode ser visualizado no trabalho de Oliveira (2018) mostrado na Figura 9.

Figura 9 – Divisão da área delimitada em células

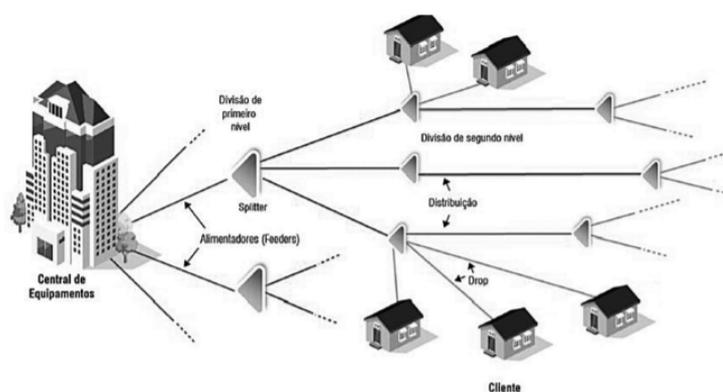


Fonte: (OLIVEIRA, 2018)

4.7.1 Versão 1.0: Atendendo às Esquinas

Na primeira versão do *software* o objetivo foi simplesmente atender todas as esquinas, admitindo-se uma densidade uniforme e também que os *splitters* utilizados são sempre os mesmos definidos previamente pelo usuário nos dois níveis de divisão. Admitindo-se que a rede possuirá sempre dois níveis de divisão de potência (como na [Figura 10](#)), o usuário define o tipo de *splitter* N1 e N2, que podem ser usados, considerando que o máximo de clientes atendidos são 128, 1/2 e 1/64, 1/4 e 1/32 e assim por diante.

Figura 10 – Rede PON com dois níveis de divisão de potência



Fonte: ([MONFRÉ, 2014](#))

Para diminuir o custo do algoritmo o posicionamento dos *splitters* é feito nas esquinas, pois toda esquina possui um poste ou pelo menos algum próximo à ela e por isso pode ser instalado um *splitter*. Na primeira iteração é selecionada a esquina mais próxima ao escritório central (CO) e depois a esquina mais próxima à anterior e assim por diante até que sejam selecionadas um número de esquinas relativa ao tipo de *splitter* que foi instalado na OLT. Por exemplo se este for tipo 1/8 serão selecionadas quatro esquinas e considera-se que cada uma destas esquinas possui um *splitter* 1/16, portanto atendendo 16 clientes cada uma, então estas esquinas são excluídas e começa a próxima iteração até que tenham sido atendidas todas as esquinas na área desejada. Um ponto muito importante é que no momento da seleção da esquina sempre é verificado se a potência que chega a ela é suficiente para atender os clientes, utilizando-se a [Equação 4.1](#) já descrita anteriormente.

4.7.2 Versão 2.0: Atendendo à demanda gerada via simulação

O programa agora roda em duas partes. Na primeira ele faz a distribuição dos clientes ([Seção 4.6](#)) para todas as ruas da área que será atendida. Após a distribuição ter sido gerada, os dados são gravados em um arquivo *Excel* para que então possa ser carregado pelo algoritmo. O intuito desse passo inicial é que, posteriormente o próprio provedor forneça tal arquivo de entrada já com a demanda real de clientes para a cidade que ele pretende atender.

A partir disso então na segunda parte do programa todos os pontos que são esquinas, ou seja, que pertencem a mais de uma rua, são percorridos e é definido a sua demanda, que é a soma das demandas das ruas que aquele ponto faz parte. Estes pontos são então ordenados em relação à maior demanda para a menor, porém também é possível ordená-los de acordo com a distancia do provedor. Foi decidido que seriam escolhidos os pontos que são esquinas para diminuir a complexidade do programa, por diminuir a quantidade de pontos que podem ser atendidos mas também por se saber que neste ponto é possível a instalação do *splitter* já que haverá com grande possibilidade um poste ali, ou em até 20 metros, pois o vão médio entre os postes deve ser de entre 35 e 40 metros no pior caso. Também admitiu-se que o *splitter* primário será sempre alocado dentro do provedor de fibra óptica, de forma que a fibra que sai do provedor o faz a partir de um *splitter* primário.

Após os pontos serem ordenados eles começam a ser atendidos do de maior demanda ao menor, ou do mais próximo do provedor ao mais distante. Por isso a abordagem pode ser considerada Gulosa, seja usando o critério de demanda, seja de distância.

Quando o primeiro ponto é atendido é verificado a sua demanda e com ela é definido o *splitter* primário e secundário da rede. É importante lembrar que apesar do *splitter* primário ser definido ao se selecionar o primeiro ponto, ele fica alocado no provedor e um *splitter* secundário é automaticamente posicionado no primeiro ponto atendido. Se a demanda for maior que 64, o *splitter* primário é 1:2 e os secundários 1:64, é definido que um *splitters* 1:64 será utilizado neste ponto para atender esta rua, porém neste caso a rua que este ponto pertence ainda não foi totalmente atendida, então é subtraído 64 da demanda dos outros pontos que pertencem à esta rua e a rua não é removida da lista de ruas à serem atendidas e o ponto é removido da lista de pontos que podem ser atendidos, desta forma o *splitter* restante é designado para o próximo ponto de maior demanda atendendo-o. Se a demanda for maior que 32 e menor ou igual a 64 então os *splitters* primário e secundário também são respectivamente 1:2 e 1:64 porém, neste caso a demanda da rua foi atendida com um dos *splitters* 1:64 então ela é removida da lista de ruas à serem atendidas e o outro *splitter* vai para o próximo ponto de maior demanda. Isto ocorre para todas as possibilidades de *splitter* à serem utilizados, respeitando o limite de 128 clientes em uma fibra. Isso é feito em um *loop* até que todos os pontos sejam atendidos ou tenham demanda igual à 0.

À cada iteração do programa é gerada uma figura com a fibra óptica gerada para atender até 128 clientes, por exemplo se for definido que o *splitter* primário é 1:8, então o secundário será um *splitter* do tipo 1:16. Neste trabalho não foi considerado o cabo *drop*, o que vai até a casa do cliente. As imagens contidas na [Figura 11](#), demonstram exemplos de quando o *splitter* primário é 1:2, 1:4, 1:8, 1:16, 1:32 e 1:64, respectivamente e também o caminho percorrido por cada fibra até chegar ao local de posicionamento do *splitter* secundário, que é demonstrado por um círculo maior. Para melhorar a visibilidade cada

caminho para o *splitter* secundário foi colorido com uma cor diferente. Cada uma dessas imagens é uma solução parcial e não todo o projeto que atender à demanda, já que cada uma destas soluções atende até no máximo 128 clientes.

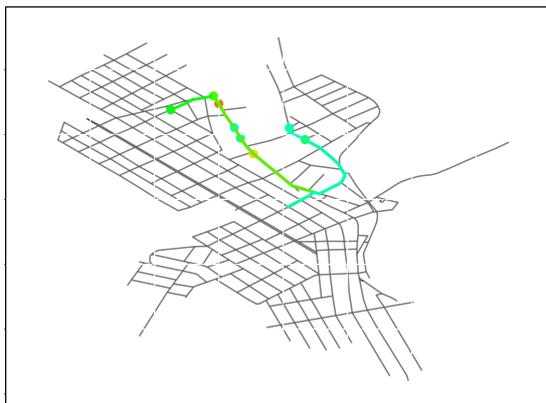
Figura 11 – Exemplo de como fica a visualização de cada tipo de *splitter*



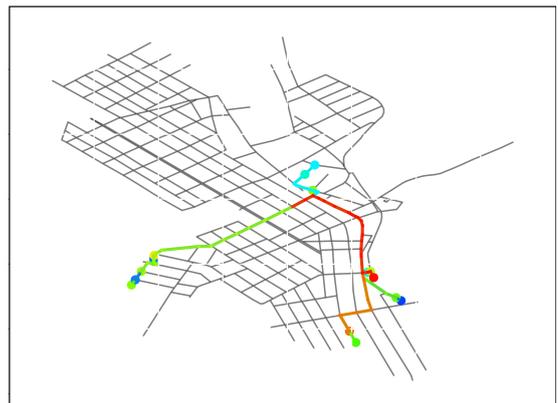
(a) Exemplo de um enlace quando o *splitter* é 1:2



(b) Exemplo de um enlace quando o *splitter* é 1:4



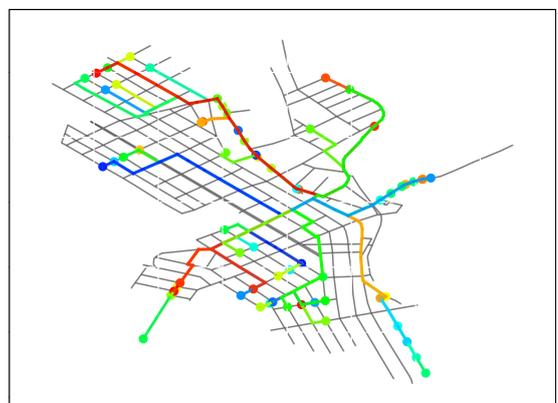
(c) Exemplo de um enlace quando o *splitter* é 1:8



(d) Exemplo de um enlace quando o *splitter* é 1:16



(e) Exemplo de um enlace quando o *splitter* é 1:32



(f) Exemplo de um enlace quando o *splitter* é 1:64

4.8 Abordagem Gulosa

Como pode ser visto na [Seção 4.7](#), este trabalho utiliza uma abordagem Gulosa para gerar as soluções, podendo ser escolhido um dos dois tipos de ótimo local: ou escolhe-se o ponto mais próximo do provedor de fibra óptica, ou escolhe-se o ponto de maior demanda. Essa escolha é feita dentro do próprio código fonte do programa.

A abordagem Gulosa foi escolhida por ser uma heurística bem conhecida e que comumente devolve bons resultados com um tempo de execução polinomial em relação ao tamanho da entrada. Possui uma implementação relativamente simples e sua metodologia é de rápida compreensão.

Por estes motivos um algoritmo guloso foi visto como uma boa abordagem inicial; porém durante o processo de codificação as principais funções foram modularizadas prevendo uma possível utilização em outras heurísticas ou meta-heurísticas para avaliação do resultado parcial, (como a função de atenuação e de cálculo do preço), para facilitar futuras implementações de novos métodos que possam gerar soluções melhores que a abordagem atual.

5 RESULTADOS E ANÁLISE

Os primeiros experimentos realizados consideraram como pré-definidos o tipo de *splitter* (divisor de potência) e a demanda de clientes (conforme primeira versão do *software*, vide Seção 4.7). Tais resultados preliminares foram utilizados na validação da modelagem do problema (*e.g.*, atenuação do sinal) e das estruturas de dados escolhidas e, por fim, para conferir se todos os pontos de demanda estariam sendo atendidos. A partir de tal validação da modelagem do problema, foram conduzidos experimentos subsequentes contemplando diferentes cenários para demanda de clientes e habilitada no algoritmo a escolha dos tipos de *splitters* mais apropriados para cada cenário (conforme segunda versão do *software*, vide Seção 4.7).

5.1 Comparação do projeto automatizado com um projeto manual

Visando realizar um comparativo de desempenho com um projeto manual de rede PON, foi utilizado o trabalho de Oliveira (2018) que documentou e realizou todo o planejamento de uma rede óptica, do início ao fim (OLIVEIRA, 2018). No referido trabalho, também foram providas informações sobre equipamentos, seus custos e características, além de demonstrar o passo a passo de como pode ser conduzido um projeto da rede PON e, portanto, tendo sido utilizado como base deste trabalho.

Para realizar a tal análise, as entradas para o *software* desenvolvido foram as mais próximas possíveis do trabalho de Oliveira (2018), tanto em relação à localização (cidade de Carmo do Cajuru-MG) e área atendida (um raio de 1500m) quanto em relação à demanda a ser atendida (312 clientes). Portanto, para fins de comparabilidade entre os trabalhos a distribuição de clientes foi realizada de forma uniforme buscando manter a similaridade. Vale ressaltar que a distribuição da demanda no trabalho de Oliveira (2018) foi dividida em quadrantes (hexágonos), enquanto que a distribuição da demanda de clientes neste trabalho foi dividida por ruas (proporcional ao seus respectivos comprimentos). Naturalmente, alguns aspectos de ambas as soluções não puderam ser diretamente comparados, pois a quantidade de equipamentos utilizados diferiu já que no trabalho de Oliveira (2018) foi adotada uma topologia em barramento, enquanto que no *software* desenvolvido neste trabalho foi utilizada uma topologia em árvore.

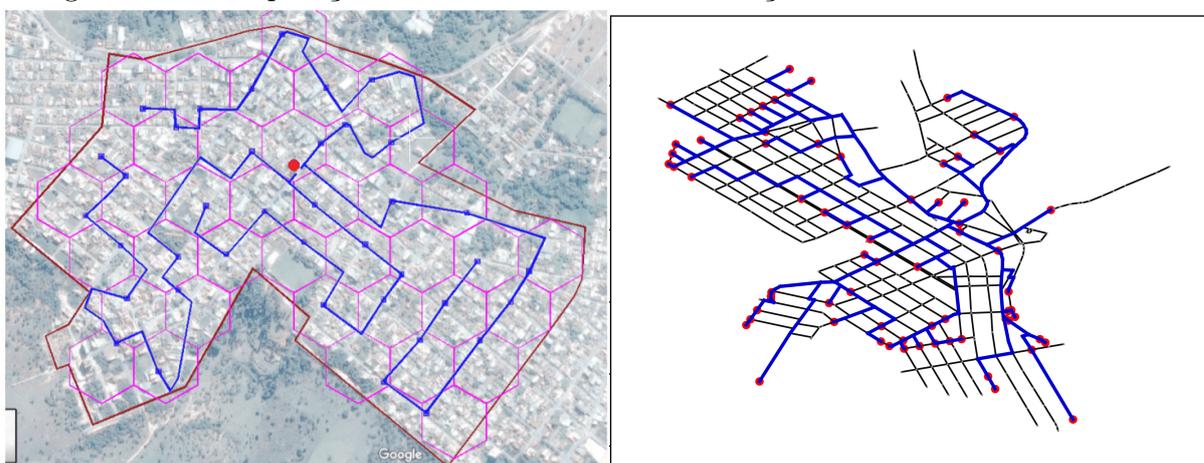
Exemplificando para uma melhor compreensão, no trabalho de Oliveira (2018), os tipos de *splitters* utilizados no projeto da rede PON foram definidos previamente e, por isso, o referido trabalho apresenta as quantidades e levantamento de preços apenas para *splitters* do tipo 1:4 e 1:8. Diferentemente, no *software* desenvolvido neste trabalho de conclusão de

curso, o tipo de *splitter* mais apropriado é definido dinamicamente pelo algoritmo, segundo a demanda em determinado local e, portanto, haverá divergência entre as soluções quanto à quantidade e tipo de tais equipamentos. Portanto, não deve-se comparar diretamente a quantidade absoluta de equipamentos entre as soluções mas sim o custo total dos mesmos, uma vez que o custo individual do *splitter* varia de acordo com seu tipo.

Ainda, há mais uma diferença entre abordagens quanto ao tipo de cabo utilizado: para a otimização de custo o projeto de Oliveira (2018) utiliza um cabo 12FO (que possui doze fibras ópticas no mesmo cabo) porém, neste trabalho de conclusão de curso ainda não foi implementada tal otimização de custo, tendo sido adotado um cabo 2FO (que possui duas fibras ópticas por cabo). Entretanto, como trabalho futuro, tal agregação de pares de fibras que sejam conduzidas por uma mesma rua poderia ser projetada e integrada ao *software*.

Considerando o acima exposto, uma comparação visual das soluções encontradas por cada abordagem (para a distribuição das fibras ópticas pela área demandada) que podem ser observada nas Figuras 12a e 12b, respectivamente a solução manualmente projetada por Oliveira (2018) e a solução gerada de forma automatizada pelo *software* desenvolvido neste trabalho de conclusão de curso.

Figura 12 – Comparação visual da rede PON da solução manual e da automatizada



(a) Solução encontrada por (OLIVEIRA, 2018)

(b) Solução encontrada neste trabalho

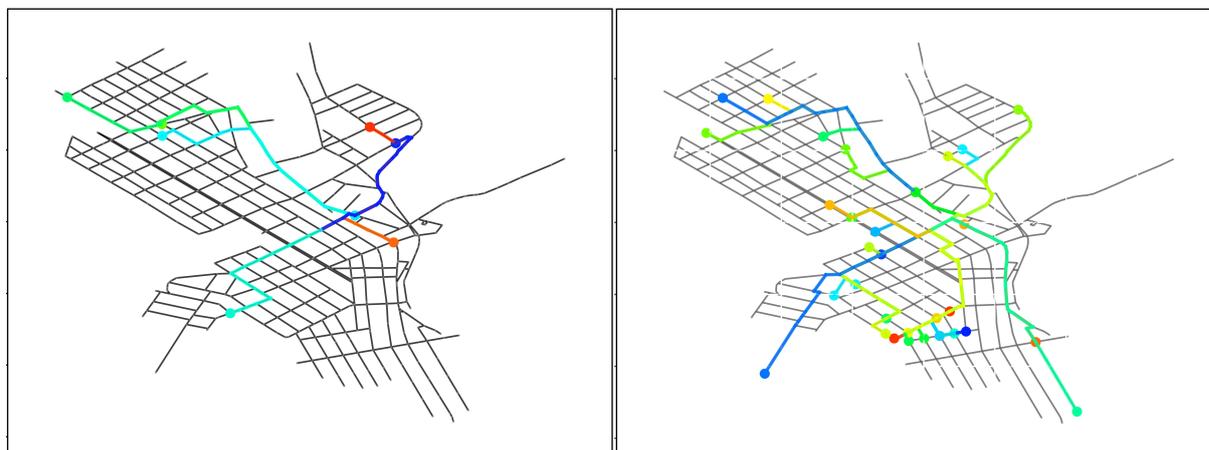
Em ambas as Figuras 12a e 12b, as linhas em azul apresentam o trajeto das fibras ópticas que irão atender a demanda prevista de clientes. O ponto vermelho maior na 12a representa a localização do escritório central (CO) da operadora responsável pela PON. Já os pontos vermelhos menores representam a localização de cada TOA ¹, a partir da qual posteriormente os cabos *drop* sairão para conectar a PON à ONU do cliente, sob demanda.

Nas Figura 13a, Figura 13b e Figura 13c, são apresentadas os três enlaces gerados pelo algoritmo que, combinados, compõem a solução final exibida na Figura 12b. Cada

¹ Local onde o *splitter* secundário da PON fica acondicionado

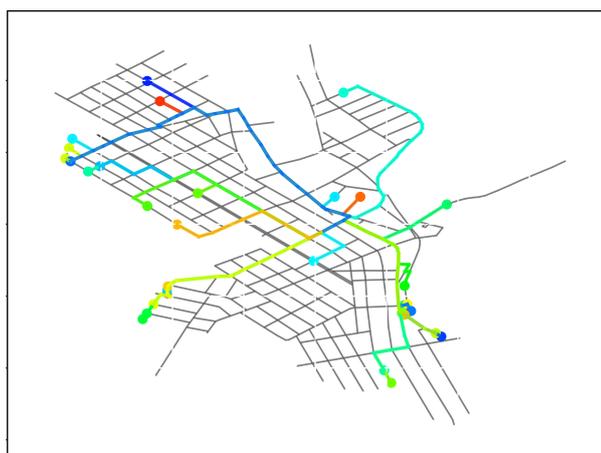
enlace consiste em uma fibra óptica atendendo até o limite de 128 clientes. No cenário em questão (cidade de Carmo do Cajuru-MG), havia uma demanda total prevista de 312 clientes; portanto são necessárias ao menos três redes PON.

Figura 13 – Três enlaces da rede, gerados a cada iteração do programa, que juntos formam a rede final.



(a) Enlace 1 do programa

(b) Enlace 2 do programa



(c) Enlace 3 do programa

Cada imagem da [Figura 13](#) apresenta uma única PON, de maneira que mudança de cores visa apenas indicar a segmentação da fibra óptica ao sair do *splitter* primário, que está alocado no DIO que fica no provedor de fibra óptica, sendo que cada segmento de fibra óptica será ligado ao (*splitter*) de segundo nível armazenado na TOA.

A [Tabela 3](#) apresenta o custo e insumos da solução encontrada por [Oliveira \(2018\)](#), enquanto que a [Tabela 4](#) apresenta o custo e insumos da solução gerada de forma automatizada pelo *software* desenvolvido. Comparando as referidas tabelas, é possível observar que a solução gerada neste trabalho apresentou um custo compatível da rede PON projetada para a mesma localidade. Entretanto, vale ressaltar que o algoritmo desenvolvido não foi capaz de atender totalmente à demanda prevista de 312 clientes, tendo gerado uma solução que atendeu diretamente 288 clientes, ou seja, 92 % da demanda prevista.

Portanto, não deve-se comparar simplesmente o **custo total** de implantação da rede PON, mas alguma outra métrica que considere a proporção da demanda atendida.

Tabela 3 – Solução proposta no trabalho de [Oliveira \(2018\)](#)

312 Clientes Atendidos	Quantidade utilizada	Valor unitário	Valor total
Fibra 12 FO Autossustentada 2 Km	5	R\$ 5.942,29	R\$ 29.711,45
<i>Splitter</i> 1:4	4	R\$ 72,19	R\$ 288,76
<i>Splitter</i> 1:8	39	R\$ 75,18	R\$ 2932,02
TOTAL			R\$32.932,23
Custo Por Cliente			R\$ 105,55

Tabela 4 – Solução encontrada neste trabalho

288 Clientes Atendidos	Quantidade utilizada	Valor unitário	Valor total
Fibra 2 FO Autossustentada 2 Km	21	R\$ 1.199,00	R\$ 25.179,00
<i>Splitter</i> 1:4	64	R\$ 72,19	R\$ 4.620,16
<i>Splitter</i> 1:8	1	R\$ 75,18	R\$ 75,18
<i>Splitter</i> 1:16	8	R\$ 90,07	R\$ 720,56
<i>Splitter</i> 1:32	2	R\$ 150,00	R\$ 300,00
TOTAL			R\$ 30.894,90
Custo Por Cliente			R\$ 107,27

Considere então o **custo por cliente** como uma métrica mais justa para comparação entre os trabalhos. Na solução gerada de forma automatizada pelo *software* proposto, o custo por cliente atendido foi de R\$ 107,27 enquanto que, na solução manualmente projetada por [Oliveira \(2018\)](#) o custo por cliente foi de R\$ 105,55. Portanto, o custo por cliente da solução gerada foi 1,63 % mais caro em relação à solução manualmente gerada por [Oliveira \(2018\)](#). De outra perspectiva, considere ainda que o projeto manual de uma rede PON tipicamente consome algumas semanas a um mês de um projetista, enquanto que o algoritmo desenvolvido apresenta a solução com menos de 2 minutos de processamento para a maior instância testada (18.000 usuários) e em torno de 15 segundos para a menor instância (312 usuários).

Um outro ponto importante a ser considerado é a **capacidade de expansão** da rede. Com a abordagem gulosa utilizada o algoritmo sempre tenta atender ao máximo de clientes com o mínimo de recursos, isso potencialmente poderia diminuir a flexibilização para futuras expansões. De fato, no cenário aqui analisado, o número máximo de clientes que poderiam ser atendidos pela solução encontrada pelo algoritmo seria 384 (25 % de expansibilidade), enquanto que na solução proposta por [Oliveira \(2018\)](#) seria até 512 clientes (39 % de expansibilidade). Porém, uma modificação no algoritmo desenvolvido

poderia garantir uma reserva pré-definida para expansão futura no atendimento de novos clientes não previstos na demanda inicial.

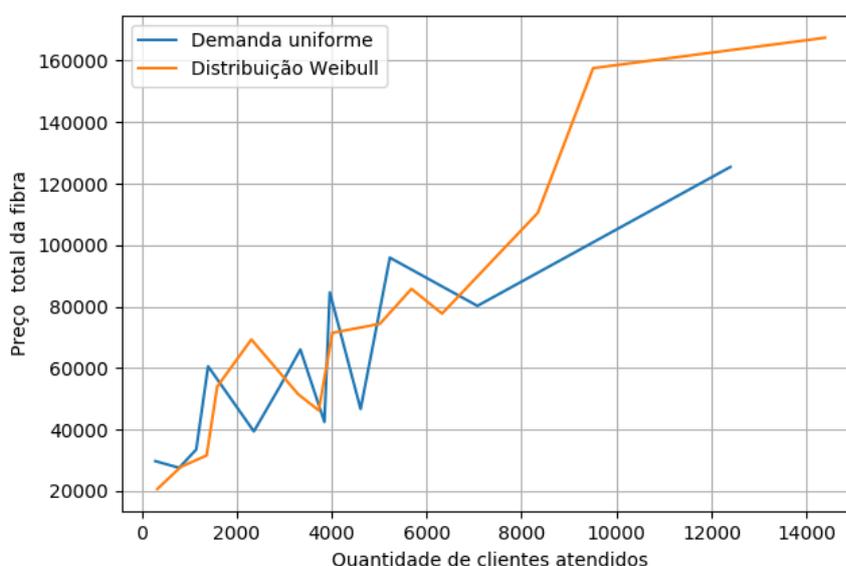
5.2 Abordagem Gulosa (pela Demanda)

5.2.1 Comparação sob diferentes distribuições e demandas

Para analisar as soluções geradas pelo *software* desenvolvido neste trabalho, foram conduzidos na cidade do Carmo do Cajuru-MG experimentos em cenários com gradual aumento na demanda de clientes, em incrementos de 500 clientes a cada novo cenário. Ademais, foram utilizadas duas formas diferentes para gerar a distribuição da demanda de clientes: demanda uniforme e distribuição Weibull (de cauda pesada), esta última visando gerar regiões da cidade com maior demanda do que outras.

A [Figura 14](#) ilustra o aumento de custo do projeto para a rede PON de acordo com maior demanda de clientes. Considerando-se que diferentes ruas, bairros e regiões de uma cidade desenvolvem-se de forma distinta, vale ponderar o risco à expansibilidade da rede ao assumir-se uma demanda uniforme de clientes e ignorar características de diferentes partes da cidade. Não tendo tais informações sócio-econômicas disponíveis, pode-se observar na [Figura 14](#) como o custo de implantação da rede PON é maior quando pressupõe-se uma distribuição de cauda pesada para o posicionamento dos clientes, apresentando locais de maior e menor concentração.

Figura 14 – Custo total da rede PON conforme aumento na demanda de clientes – Algoritmo guloso “pela demanda”

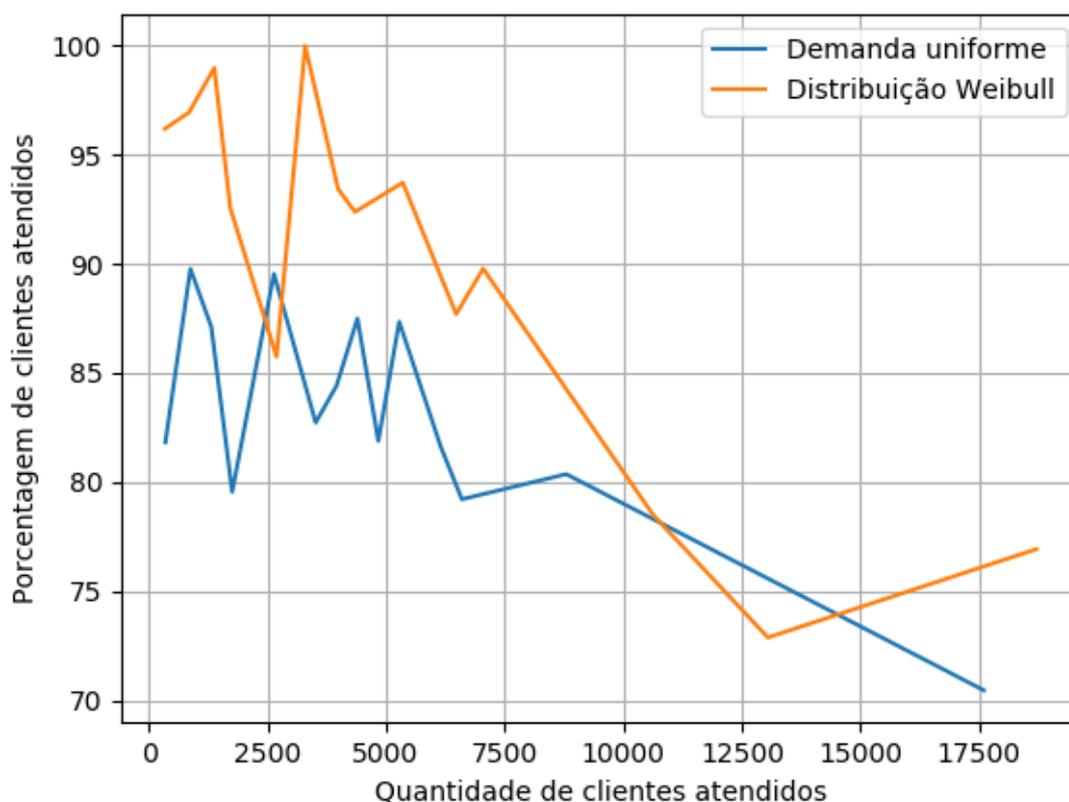


Fonte: Próprio Autor

Já a [Figura 15](#) apresenta uma comparação da porcentagem da demanda de clientes

atendida conforme a distribuição utilizada (Demanda Uniforme \times Distribuição Weibull). Nela, é possível observar que para demandas de até 10.000 clientes, o uso de uma distribuição Weibull apresentou soluções com maior taxa de atendimento da demanda e, a partir desse valor, as soluções com ambas as distribuições de clientes se tornaram semelhantes.

Figura 15 – Porcentagem de clientes atendidos conforme aumento na demanda – Algoritmo guloso “pela demanda”

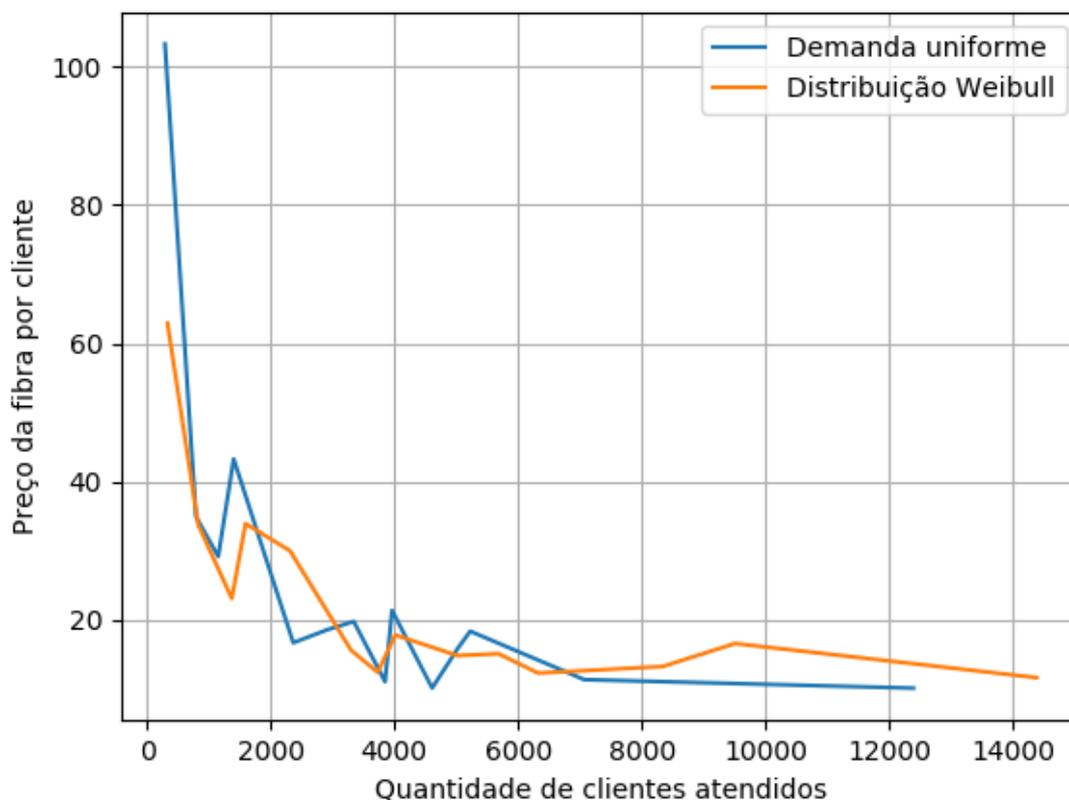


Fonte: Próprio Autor

Foi realizada uma análise do custo total da rede (fibra mais *splitters primários e secundários*) por cliente atendido, conforme gráfico apresentado na [Figura 16](#). Observe que quando a demanda é maior que 3.000 clientes, o custo por cliente “começa a estabilizar” abaixo de R\$ 20,00 por cliente. Para demandas menores que 750 clientes, o custo por cliente é superior a R\$40,00

Curiosamente, observa-se que para a instância com aproximadamente 340 clientes – cenário semelhante ao de [Oliveira \(2018\)](#) – o custo por cliente (assumindo demanda uniforme) é de aproximadamente R\$ 103,00 enquanto que, na distribuição de cauda pesada, o custo foi de R\$ 62,00. Isso sugere que, para instâncias com menor demanda de clientes, o custo de implantação da rede PON seria menor caso houvesse alguma concentração de clientes em determinadas regiões da cidade.

Figura 16 – Custo por cliente atendido conforme aumento na demanda – Algoritmo guloso “pela demanda”



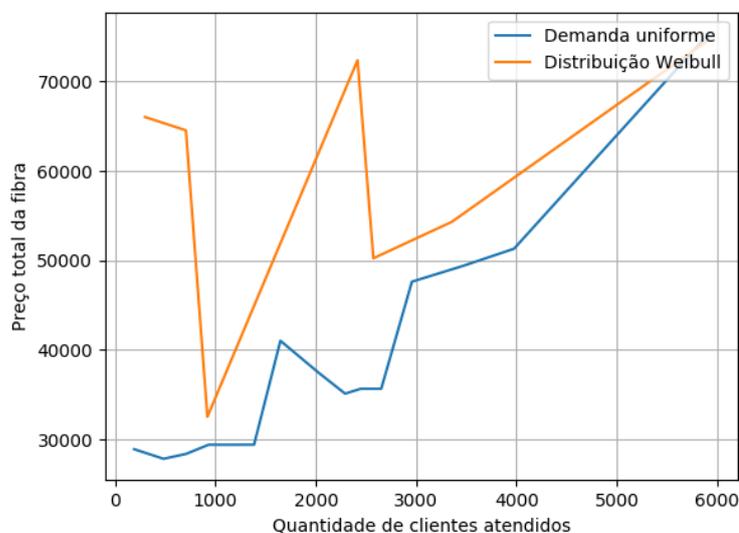
Fonte: Próprio Autor

5.3 Abordagem Gulosa (pela Distância)

5.3.1 Comparação sob diferentes distribuições e demandas

A análise da abordagem gulosa “pela distância” foi conduzida de maneira similar à gulosa “pela demanda” (vide [Subseção 5.2.1](#)). A primeira comparação que pode-se realizar é o aumento de custo do projeto para a rede PON de acordo com maior demanda de clientes, o qual pode ser observado na [Figura 17](#). De forma consistente, observou-se que o custo de implantação da rede PON é maior quando pressupõe-se uma distribuição de cauda pesada para o posicionamento dos clientes, apresentando locais de maior e menor concentração deles.

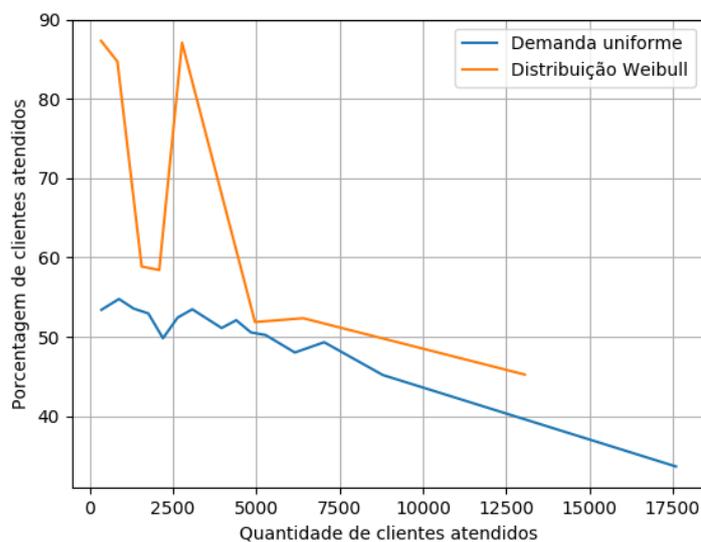
Figura 17 – Custo total da rede PON conforme aumento na demanda de clientes – Algoritmo guloso “pela distância”



Fonte: Próprio Autor

Também foi feita uma análise comparando a porcentagem de clientes atendidos em relação a demanda total existente, distribuída tanto de forma uniforme quanto de cauda pesada. Na [Figura 18](#) é possível observar que utilizando o *software* desenvolvido, ao pressupor uma demanda uniforme de clientes entre as ruas foi possível atender uma porcentagem maior deles, se comparado aos resultados obtidos para uma demanda de clientes gerada com uma distribuição de cauda pesada (Weibull).

Figura 18 – Porcentagem de clientes atendidos conforme aumento na demanda – Algoritmo guloso “pela distância”



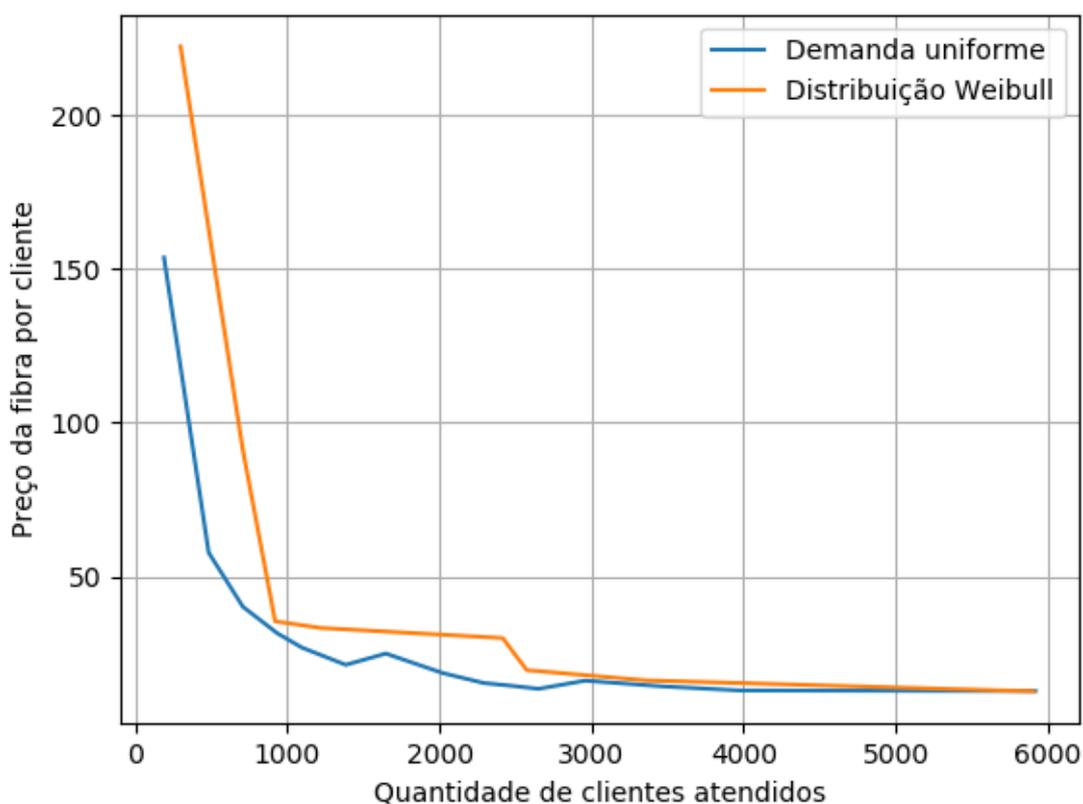
Fonte: Próprio Autor

Esta variação do algoritmo, analisada na presente seção, é gulosa no sentido que visa atender primeiro os clientes que estejam mais próximos do escritório central (CO) da operadora da rede (vide Seção 4.8). Portanto, aqueles clientes localizados em uma região mais periférica da cidade podem não chegar a ser atendidos pela solução proposta com tal algoritmo guloso.

Parcialmente, tal fato ajuda a explicar a menor porcentagem da demanda de clientes atendida com o algoritmo proposto, se compararmos as Figuras 15 (guloso “pela demanda”) e 18 (guloso “pela distância”).

Por fim, foi realizada uma análise sobre o custo por cliente, de acordo com o aumento na demanda de clientes. Conforme pode ser observado na Figura 19. Novamente, como visto na variação gulosa “por demanda” do algoritmo, na versão gulosa “por distância” quando a demanda é maior que 3.000 clientes, o custo por cliente também “começa a estabilizar” abaixo de R\$ 20,00 por cliente.

Figura 19 – Custo por cliente atendido conforme aumento na demanda – Algoritmo guloso “pela distância”



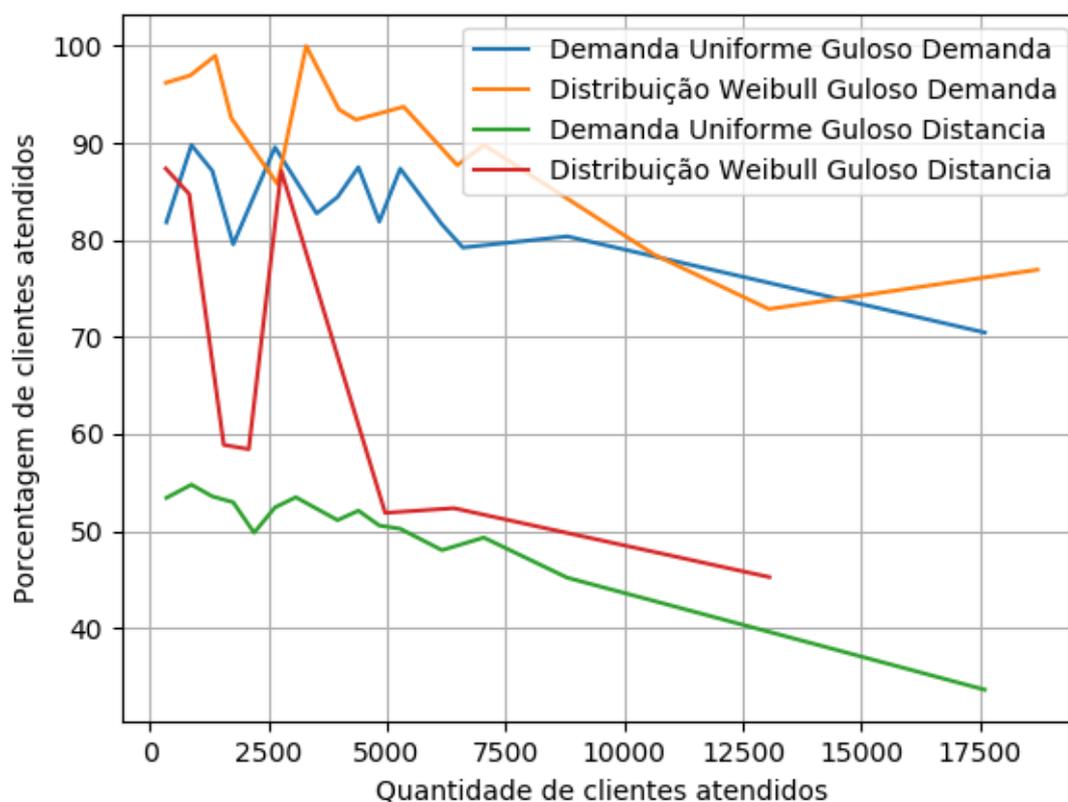
Fonte: Próprio Autor

5.4 Abordagem Gulosa: pela demanda VS. pela distância

Considerando os resultados analisados nas seções anteriores, é interessante comparar diretamente de forma gráfica o desempenho de ambas as abordagens gulosas implementadas no *software* desenvolvido neste trabalho de conclusão de curso.

No caso da porcentagem de clientes atendidos (vide Figura 20), constatou-se que a abordagem gulosa “pela demanda” apresentou uma taxa de atendimento de clientes superior à abordagem gulosa “pela distância”. Caso o objetivo primário de uma operadora de rede PON ao implantá-la seja maximizar a quantidade de clientes atendida dentro a demanda inicialmente levantada, recomenda-se portanto a utilização da abordagem gulosa “pela demanda” implementada no *software*.

Figura 20 – Porcentagem de clientes atendidos pelo *software* proposto

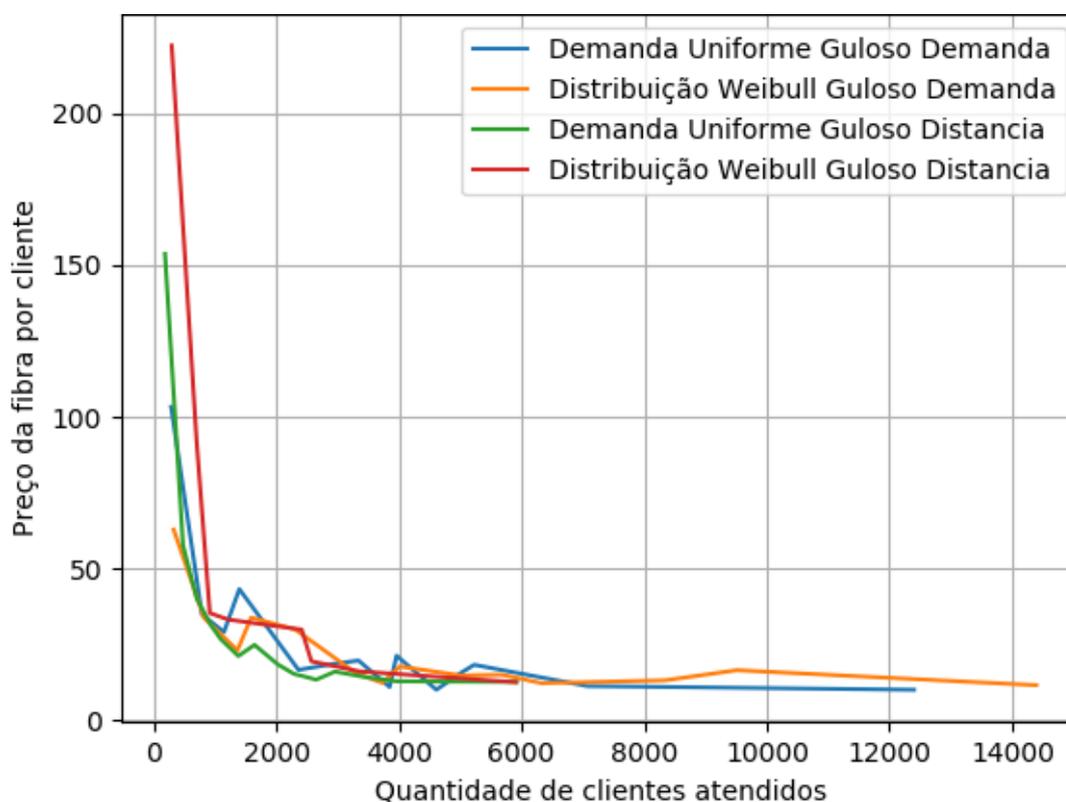


Fonte: Próprio Autor

Em relação ao custo por cliente (vide Figura 21), observamos que para demandas abaixo de 3.000 clientes e com demanda uniforme de clientes entre as ruas, a abordagem gulosa “pela distância” apresentou um custo por cliente inferior do que o obtido pelas soluções geradas de forma gulosa “pela demanda”. Porém isto se inverte no caso de demandas abaixo de 1.500 clientes e com distribuição Weibull de clientes entre as ruas: neste caso, a abordagem gulosa “pela demanda” apresentou um custo por cliente inferior do que o obtido pelas soluções geradas de forma gulosa “pela distância”.

A partir de 3.000 clientes ou mais, ambas as abordagens gulosas e formas de distribuição de clientes geram soluções que convergem para um custo por cliente inferior a R\$ 20,00. Apenas, vale ressaltar que a abordagem gulosa “pela distância” não foi capaz de gerar soluções para instâncias com demanda de clientes superior a 6.000 clientes para o cenário analisado (cidade de Formiga/MG). A abordagem gulosa “pela demanda”, por buscar atender primeiro as maiores concentrações de clientes na cidade (independente da sua distância ao CO), conseguiu gerar soluções para previsões de demanda superior a 12.000 clientes no cenário acima referido.

Figura 21 – Custo por cliente obtido pelo *software* proposto



Fonte: Próprio Autor

6 TRABALHOS FUTUROS

Neste trabalho um dos cuidados foi o de prover uma modelagem do problema que servisse de base para futuras implementações de heurísticas, que talvez possam obter melhores resultados. Portanto, esta é a principal proposta para trabalhos futuros: experimentar com (meta)heurísticas computacionais para melhorar a solução gerada, fugindo de prováveis ótimos locais que estejam limitando a qualidade da solução gerada pela abordagem gulosa utilizada neste trabalho.

Devido a limitações intrínsecas ao algoritmo guloso e a dificuldade em codificar análises para as diversas possibilidades de topologia do sistema, foi necessário definirmos previamente a topologia a ser seguida pelo algoritmo. Assim, uma outra possibilidade para trabalhos futuros seria a implementação de procedimentos que explorem diferentes topologias de rede PON, talvez encontrando soluções com melhor custo-benefício de acordo com as peculiaridades de cada cenário de entrada.

Uma característica relevante, que poderia reduzir o custo total da solução gerada, mas que não chegou a ser implementada neste trabalho é a possibilidade de se utilizar cabos que suportam mais de um par de fibras ópticas em seu interior. Observamos que, por vezes, o trajeto das fibras ópticas que atendem a grupos diferentes de clientes percorrem parcialmente um mesmo caminho. Portanto, seria possível conseguir uma redução drástica no custo do projeto da rede PON gerada pelo *software* ao permitir que o mesmo otimize tais trechos ao adotar um cabo que agregue uma quantidade maior de fibras ópticas. Segundo Oliveira (2018), as fibras são alocadas em tubos em grupos de 2, 6 ou 12 fibras, onde cada cabo pode conter um número máximo de 6 tubos e, desta forma, o mercado fornece de forma mais acessível cabos que contenham de 2 até 288 fibras ópticas.

Uma questão relevante, do ponto de vista do usuário final, é prover uma interface entre o usuário e o programa, já que atualmente ele executa sem interação alguma com o usuário, recebendo as configurações de entrada (cidade e sua demanda de clientes, configuração e custo dos implementos de rede) ao se alterar os arquivos de configuração. Consideremos que, para pessoas leigas, poderia ser pouco prático utilizar o programa invocando-o por um console (terminal) de linha de comando. Talvez, com o auxílio de uma interface (*user-friendly*) seria possível a modificação destes arquivos e execução do programa de forma mais intuitiva.

Ademais, juntamente com a interface para facilitar a utilização do programa, também seria desejável fazer as requisições ao *OpenStreetMap* de forma automática. Atualmente, é necessário acessar manualmente o *website Overpass-Turbo*, pesquisar pela cidade desejada, ir para a área dela que se deseja atender e rodar um *script* para, só então obter

o arquivo com as informações geográficas a ser utilizado como entrada pelo *software* desenvolvido. Portanto, seria interessante tornar os passos acima descritos (semi)automatizados para agilizar e facilitar a utilização do programa.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento do presente estudo possibilitou uma análise de como um *software* para automatização do projeto de redes ópticas passivas (PON) pode facilitar e agilizar tal processo. As etapas que antecederam a modelagem do problema possibilitaram obter dados consistentes sobre as etapas de um projeto de PON, informações sobre os equipamentos nele utilizados e as formas de transformar o projeto manual de redes PON em um modelo computacional.

Durante a implementação do *software* foram experimentadas diferentes abordagens, de forma cuidadosa para que a modelagem final do problema chegasse a uma solução satisfatória e que fosse realmente aplicável no mundo real. As entradas de dados do programa foram variadas e demandaram diversas correções para evitar ao máximo que o programa não esbarrasse em problemas ao analisar diferentes instâncias (demanda de clientes) de diferentes localizações (cidades).

Uma preocupação neste trabalho foi a de facilitar sua expansão em trabalhos futuros, caso queiram otimizar outros aspectos do projeto de PON que não foram abordadas neste trabalho, ou mesmo tentar gerar soluções melhores para as instâncias utilizadas neste trabalho ao utilizar outras abordagens.

A comparação com um projeto de PON realizado manualmente foi de suma importância para se aferir a consistência e validade do programa e, também, para avaliar os resultados obtidos, já que sem entradas de um cenário real seria complicado validar a eficácia e eficiência da solução gerada pelo programa.

Por fim, dado o crescente avanço na utilização de redes ópticas passivas (PON) e a aparente escassez de programas que auxiliem no projeto de construção destas, torna-se desejável o desenvolvimento e aprimoramento de ferramentas que auxiliem operadoras de PON (e provedores de acesso à internet) agilizar o projeto de implantação de novas redes e reduzir os gastos para tal.

Referências

- AGATA, A.; NISHIMURA, K. Suboptimal pon network designing algorithm for minimizing deployment cost of optical fiber cables. *2012 16th International Conference on Optical Network Design and Modelling (ONDM)*, 2012. Citado na página 26.
- AHMAD, A. et al. Power-aware logical topology design heuristics in wavelength-routing networks. In: . [S.l.: s.n.], 2011. p. 1 – 6. Citado na página 25.
- ARÉVALO, G. V.; HINCAPIÉ, R. C.; GAUDINO, R. Optimization of multiple pon deployment costs and comparison between gpon, xgpon, ngpon2 and udwdm pon. *Optical Switching and Networking*, v. 25, p. 80 – 90, 2017. ISSN 1573-4277. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1573427716300522>>. Citado na página 26.
- BORGES, L. E. *Python para Desenvolvedores*. 2. ed. Rio de janeiro: Edição do Autor, 2010. ISBN 978-85-909451-1-6. Citado na página 29.
- CHESMAN, C.; MACEDO, A.; ANDRÉ, C. *Física Moderna Experimental e Aplicada*. LIVRARIA DA FISICA, 2004. ISBN 9788588325180. Disponível em: <<https://books.google.com.br/books?id=WEykihZUkzwC>>. Citado na página 19.
- COELHO, F. C. *Computação Científica com Python: Uma introdução à programação para cientistas*. Petrópolis: Edição do Autor, 2007. ISBN 978-85-907346-0-4. Citado na página 29.
- CORMEN, T. et al. *Introduction To Algorithms*. MIT Press, 2001. (Introduction to Algorithms). ISBN 9780262032933. Disponível em: <https://books.google.com.br/books?id=NLngYyWFl_YC>. Citado na página 23.
- CORNUÉJOLS, G.; NEMHAUSER, G.; WOLSEY, L. *The Uncapacitated Facility Location Problem*. [S.l.], 1983. Citado na página 23.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, v. 1, n. 1, p. 269–271, 1959. Citado na página 35.
- FARMER, J. et al. *FTTx networks: technology implementation and operation*. [S.l.]: Morgan Kaufmann, 2016. Citado na página 17.
- FERNANDES., V. de Araujo Barbosa; Marcus Murilo Metz; Guilherme Thyago de S. Redes Ópticas passivas (pon) o futuro das redes: E suas tendências mercadológicas. *Revista Científica Núcleo do Conhecimento*, p. 158–185, 2018. ISSN 2448-0959. Citado na página 18.
- GILBERT, E. N.; POLLAK, H. O. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, v. 16, n. 1, p. 1–29, 1968. Citado na página 24.
- KHAN, S. Heuristics-based pon deployment. *IEEE Communications Letters*, v. 9, n. 9, p. 847–849, 2005. Citado na página 25.

KOKANGUL, A.; ARI, A. Optimization of passive optical network planning. *Applied Mathematical Modelling*, v. 35, n. 7, p. 3345 – 3354, 2011. ISSN 0307-904X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0307904X11000308>>. Citado na página 25.

LI, J.; SHEN, G. Cost minimization planning for passive optical networks. *OFC/NFOEC 2008 - 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, 2008. Citado na página 25.

MARTINEZ, F.; SOARES, J. A. *Aproximações para restrições do problema de Steiner em grafos*. Tese (Doutorado), 08 2004. Citado na página 24.

MONFRÉ, J. M. G. A. Análise comparativa das formas de conexão de fibras ópticas em redes fttx: fusão óptica versus conectorização. *Tecnologias Infraestrutura Software*, p. 222–234, 2014. ISSN 2316-2872. Citado na página 38.

NIC.BR. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros - TIC Domicílios 2016*. 2016. Disponível em: <<http://cgi.br/publicacao/pesquisa-sobre-o-uso-das-tecnologias-de-informacao-e-comunicacao-nos-domicilios-brasileiros-tic-domi>>. Citado na página 17.

OLIVEIRA, L. R. de. *Projeto e dimensionamento de rede de acesso passiva para pequenas localidades com distribuição uniforme de clientes*. 2018. Citado 10 vezes nas páginas 9, 22, 30, 37, 42, 43, 44, 45, 47 e 53.

PAL, S. et al. Cable length minimisation in long-reach-pon planning for sparsely populated areas. In: *2014 International Conference on Optical Network Design and Modeling*. [S.l.: s.n.], 2014. p. 234–239. Citado na página 26.

PARBERRY, I. *Problems on Algorithms*. Prentice Hall, 1995. ISBN 9780134335582. Disponível em: <<https://books.google.com.br/books?id=tdB8QgAACAAJ>>. Citado na página 23.

PINHEIRO, J. *Redes Ópticas de Acesso em Telecomunicações*. Elsevier Editora Ltda., 2017. ISBN 9788535286137. Disponível em: <<https://books.google.com.br/books?id=AbE4DwAAQBAJ>>. Citado 6 vezes nas páginas 18, 19, 20, 21, 22 e 36.

SILVA, E. L. D. et al. Transmission network expansion planning under a tabu search approach. *IEEE Transactions on Power Systems*, v. 16, n. 1, p. 62–68, Feb 2001. ISSN 0885-8950. Citado na página 24.

TAKEUTI, P. Projeto e dimensionamento de redes ópticas passivas (pons). 2005. Citado na página 18.

TYSON, J. *How VDSL Works*. HowStuffWorks, 2001. Disponível em: <<https://computer.howstuffworks.com/vdsl3.htm>>. Citado na página 17.