



MINISTÉRIO DA EDUCAÇÃO  
Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Colegiado do curso de Engenharia de Produção



# **Roteamento e Alocação de Espectro em Redes Ópticas Flexíveis: interface de usuário e proposta de melhoria do modelo matemático**

**Bruna Cristina Freitas do Carmo**

João Monlevade, MG  
2023

Bruna Cristina Freitas do Carmo

**Roteamento e Alocação de Espectro em Redes Ópticas  
Flexíveis: interface de usuário e proposta de melhoria do  
modelo matemático**

Trabalho de conclusão de curso apresentado ao curso de Engenharia de Produção do Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto, como parte dos requisitos necessários para a obtenção do título de Bacharel em Engenharia de Produção.

Orientadores: Dr. Hervé Kerivin e Dra. Annegret Wagler

João Monlevade, MG

2023

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C287r Carmo, Bruna Cristina Freitas do.  
Roteamento e alocação de espectro em redes ópticas flexíveis  
[manuscrito]: interface de usuário e proposta de melhoria do modelo  
matemático. / Bruna Cristina Freitas do Carmo. - 2023.  
47 f.: il.: color..

Orientadores: Profa. Dra. Annegret Wagler, Prof. Dr. Hervé Kerivin.  
Coorientador:  
Monografia (Bacharelado). Universidade Federal de Ouro Preto.  
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de  
Produção .

1. Comunicações ópticas. 2. Desigualdades (Matemática). 3.  
Interfaces de usuário (Sistemas de computação). 4. Modelos  
matemáticos. 5. Pesquisa operacional. I. Kerivin, Hervé. II. Wagler,  
Annegret. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 519.8

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



## FOLHA DE APROVAÇÃO

**Bruna Cristina Freitas do Carmo**

### **Roteamento e alocação de espectro em redes ópticas flexíveis: interface de usuário e proposta de melhoria do modelo matemático**

Monografia apresentada ao Curso de Graduação em Engenharia de Produção da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Produção

Aprovada em 23 de março de 2023

#### Membros da banca

[Prof. Dr. ] - Thiago Augusto de Oliveira Silva - Presidente da Banca (Universidade Federal de Ouro Preto)

[Prof. Dr.] - Paganini Barcellos de Oliveira - (Universidade Federal de Ouro Preto)

[Prof.Dr.] - Alexandre Xavier Martins - (Universidade Federal de Ouro Preto)

Thiago Augusto de Oliveira Silva, orientador local do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 28/03/2023



Documento assinado eletronicamente por **Thiago Augusto de Oliveira Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 28/03/2023, às 12:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0499419** e o código CRC **E0F961AF**.

## **Dedicatória**

*Dedico este trabalho aos meus pais, que sempre me apoiaram e incentivaram em todas as etapas da minha vida. Obrigado por me ensinarem a perseverar, a nunca desistir dos meus sonhos e por serem o meu porto seguro nos momentos difíceis. Dedico também aos meus amigos e familiares, que sempre estiveram presentes, me encorajando e me dando força para seguir em frente.*

# Agradecimentos

Em primeiro lugar, expresso minha gratidão a Deus por conceder-me sabedoria, paciência e determinação para realizar este trabalho acadêmico. Sem dúvida, sua orientação e força foram fundamentais para concluir esta etapa em minha vida.

Gostaria de aproveitar a oportunidade para agradecer minha família, em especial meus queridos pais, Nilton César do Carmo e Maria das Graças Miranda Freitas, pelo amor incondicional, apoio incansável e compreensão sem limites durante toda minha jornada acadêmica. Sem eles, nada disso teria sido possível, e por isso sou imensamente grata.

Não posso deixar de mencionar meus orientadores, Dr. Hervé Kerivin, Dra. Annegret Wagler e Dr. Thiago Augusto de Oliveira Silva, por sua orientação, conselhos sábios, paciência e tempo investido me ajudando na condução deste trabalho. A dedicação e suporte de vocês foram cruciais para o sucesso desta pesquisa, e sou grata por ter tido a oportunidade de trabalhar com profissionais tão competentes e inspiradores. Também gostaria de agradecer Rafael Colares, autor de uma das principais bases utilizadas nesta pesquisa, pela disposição em colaborar e compartilhar seu conhecimento.

Devo ainda agradecer meus amigos, cujo apoio e motivação me sustentaram durante todo o processo de pesquisa. A presença de vocês foi extremamente importante para manter minha motivação e aliviar a tensão.

Gostaria de expressar minha gratidão à [Coordenação de Aperfeiçoamento de Pessoal de Nível Superior \(CAPES\)](#) pelo auxílio financeiro e pela oportunidade de realizar um intercâmbio acadêmico, o que agregou muito em minha trajetória pessoal e profissional. Também sou grata à [Universidade Federal de Ouro Preto \(UFOP\)](#) e à *Université Clermont Auvergne (UCA)* por proporcionarem experiências incríveis durante minha graduação.

Por fim, gostaria de agradecer a todas as pessoas que, de alguma forma, contribuíram para o sucesso deste trabalho. Agradeço aos revisores que dedicaram seu tempo e expertise para melhorar este trabalho, e a todos os profissionais da área que compartilharam seus conhecimentos e experiências. Sem dúvida, este trabalho foi construído sobre a base do trabalho de muitos outros pesquisadores e acadêmicos, e sou grata por poder contribuir para o avanço da ciência e do conhecimento em minha área de atuação.

*“Não há progresso sem mudanças.”*  
*Confúcio*

# Resumo

Este trabalho é o resultado da elaboração de um projeto realizado no Instituto Superior de Informática, Modelagem e suas Aplicações (ISIMA), no qual foi feito um estudo do problema de Roteamento e Alocação de Espectro (RSA), este é considerado um problema NP-difícil e tem recebido muita atenção nos últimos anos. O objetivo principal deste trabalho é elaborar uma interface de usuário e mostrar sua utilidade propondo novas desigualdades para fortalecer a formulação padrão do RSA, proposta por [Colares, Kerivin e Wagler \(2021\)](#). Estas desigualdades poderão então, serem adicionadas ao modelo já existente, visando reduzir o tempo computacional gasto para encontrar boas soluções. O desenvolvimento deste trabalho envolve, além disso, uma revisão de documentos que tratam do problema RSA, algoritmos *branch-and-cut*, estudos relacionados a biblioteca *Tkinter*, do *Python*, e a compilação de um modelo proposto por [Colares, Kerivin e Wagler \(2021\)](#).

**Palavras-chaves:** Problema de Roteamento e Alocação de Espectro, Rede Óptica, Novas Desigualdades, Programação Linear Inteira Mista.

# Abstract

This work is the result of a project carried out at the Institute of Computer Science, Modeling and its Applications (ISIMA), in which a study of the Routing and Spectrum Assignment (RSA) problem was conducted. This problem is considered to be NP-hard and has received significant attention in recent years. The main objective of this work is to develop a user interface and demonstrate its usefulness by proposing new inequalities to strengthen the standard formulation of RSA, as proposed by [Colares, Kerivin e Wagler \(2021\)](#). These inequalities can then be added to the existing model, with the aim of reducing the computational time required to find good solutions. The development of this work involves the review of documents related to the RSA problem, branch-and-cut algorithms, studies related to the Python library Tkinter, and the compilation of a model proposed by [Colares, Kerivin e Wagler \(2021\)](#).

**Keywords:** Routing and Spectrum Allocation Problem, Optical Network, New Inequalities, Mixed Integer Linear Programming.

# Lista de ilustrações

Figura 1 – Representação de uma rede de fibra óptica . . . . .	5
Figura 2 – Representação das regras a serem satisfeitas no espectro . . . . .	5
Figura 3 – Grafo que representa uma rede óptica $G$ . . . . .	6
Figura 4 – Como os arquivos e pastas são organizados no <i>FlexOptim</i> . . . . .	16
Figura 5 – Ferramenta <i>Tkinter</i> "hierarquia" . . . . .	23
Figura 6 – Interface de usuário para o <i>RSA Problem (FlexOptim)</i> . . . . .	24
Figura 7 – "Escolha de parâmetros": interface de usuário para o <i>RSA Problem (FlexOptim)</i> . . . . .	25
Figura 8 – "Escolha de parâmetros": parâmetros de formulação . . . . .	26
Figura 9 – Executando uma instância, selecionando a função objetivo 1 na interface . . . . .	27
Figura 10 – Executando uma instância, selecionando a função objetivo 2 na interface . . . . .	28
Figura 11 – Executando uma instância, selecionando a função objetivo 2p na interface . . . . .	29
Figura 12 – Executando uma instância, selecionando a função objetivo 4 na interface . . . . .	30
Figura 13 – Executando uma instância, selecionando a função objetivo 8 na interface . . . . .	31

# Lista de abreviaturas e siglas

**ANR** Agência Nacional de Pesquisa Francesa

**CAPES** Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

**ILP** Programação Linear Inteira

**ISIMA** Instituto Superior de Informática, Modelagem e suas Aplicações

**LIMOS** Laboratório de Informática, Modelagem e Otimização de Sistemas

**MIP** Programação Inteira Mista

**RSA** *Routing and Spectrum Assignment*

**UCA** *Université Clermont Auvergne*

**UFOP** Universidade Federal de Ouro Preto

**WSL** *Windows Subsystem for Linux*

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
<b>1.1</b>	<b>Objetivo geral</b>	<b>2</b>
1.1.1	Objetivos específicos	2
<b>1.2</b>	<b>Organização do Trabalho</b>	<b>3</b>
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>4</b>
<b>2.1</b>	<b>Roteamento e Alocação de Espectro - RSA</b>	<b>4</b>
2.1.1	Modelo matemático	6
2.1.1.1	RSA - BASE	6
2.1.1.2	RSA - EXT	10
<b>2.2</b>	<b><i>Branch and Cut</i></b>	<b>11</b>
<b>2.3</b>	<b>Biblioteca <i>Tkinter</i></b>	<b>12</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>14</b>
<b>4</b>	<b>RESULTADOS</b>	<b>16</b>
<b>4.1</b>	<b>Compilação do código do <i>RSA-EXT</i></b>	<b>16</b>
4.1.1	Projeto <i>FlexOptim</i>	16
4.1.2	Instalação de pacotes	17
4.1.3	<i>Windows Subsystem for Linux (WSL)</i>	18
4.1.4	<i>Makefile</i>	19
4.1.5	Parâmetros do problema	19
<b>4.2</b>	<b>Interface de usuário para o <i>RSA Problem (FlexOptim)</i></b>	<b>22</b>
4.2.1	Escolha da linguagem <i>Python</i> e da biblioteca <i>Tkinter</i>	22
4.2.2	Funcionamento da Interface de Usuário	23
4.2.3	Introdução de uma nova desigualdade e proposta de melhoria do modelo	27
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>34</b>
	<b>REFERÊNCIAS</b>	<b>35</b>

# 1 Introdução

Este trabalho refere-se a um projeto realizado desde o início de abril de 2022, no [Instituto Superior de Informática, Modelagem e suas Aplicações \(ISIMA\)](#), localizado em Clermont Ferrand, na França, durante um processo de intercâmbio acadêmico. Tal estudo foi desenvolvido sob a responsabilidade do professor Dr. Hervé Kerivin e da professora Dra. Annegret Wagler, ambos constituintes do corpo docente da [Université Clermont Auvergne \(UCA\)](#). O presente documento tem como objetivo descrever o referido projeto, que foi desenvolvido baseando-se em uma das dez subáreas da engenharia de produção, a área de pesquisa operacional, matemática e ciência da computação, a fim de elaborar uma interface de usuário que possibilite e facilite alterações dos parâmetros do modelo e também auxilie na busca de novas desigualdades que fortaleçam a formulação do problema [Routing and Spectrum Assignment \(RSA\)](#), que será detalhada ao longo deste estudo.

Este projeto foi baseado nos estudos de [Colares, Kerivin e Wagler \(2021\)](#) e [Hadhbi, Kerivin e Wagler \(2019\)](#), que foram realizados no [Laboratório de Informática, Modelagem e Otimização de Sistemas \(LIMOS\)](#) e financiados pela [Agência Nacional de Pesquisa Francesa \(ANR\)](#) e pelo [IMT Atlantique](#), em Brest, na França. O presente documento descreverá as atividades realizadas, os entregáveis e as dificuldades e limitações encontradas durante o desenvolvimento.

Considerando o contexto em que o trabalho se insere é importante ressaltar que, o problema [RSA](#) é um problema crucial para as operadoras de rede resolverem, devido ao crescimento do tráfego de dados com a chegada da tecnologia 5G. Assim, este estudo tem como objetivo somente apresentar uma interface de usuário para o problema, porém, durante o seu desenvolvimento, através de análises das soluções visualizadas com o auxílio dela, notou-se a possibilidade de se aprofundar na parte matemática, visando otimizar o tempo gasto para que fossem encontradas soluções para o problema em questão.

Desta forma, a fim de fortalecer a formulação existente do problema [RSA](#), procurou-se encontrar novas desigualdades, com base em estudos presentes na literatura, sobretudo verificando se haviam semelhanças entre os modelos. Isso se justifica pelo fato de que com a introdução de novas desigualdades válidas pressupõe-se que seria possível encontrar soluções de forma mais rápida e eficiente, reforçando o método *Branch and Cut*. A formulação atual leva muito tempo para encontrar boas soluções, o que foi possível observar com auxílio da interface criada.

No que diz respeito à interface do usuário, também é preciso levar em consideração o contexto atual, em que o desenvolvimento acelerado e constante das tecnologias de informação e comunicação envolve cada vez mais o ser humano com os serviços que oferece. Hoje é possível identificar facilmente o uso constante da tecnologia no dia a dia, quase que inevitavelmente, na vida pessoal e profissional de todos. É possível constatar essa realidade através da evolução que tem-se presenciado em celulares, computadores e demais eletrônicos. São poucas as pessoas que não tiveram contato com nenhum desses materiais hoje em dia.

Assim, surgem possibilidades e necessidades de facilitar e aprimorar a interação do usuário com ela e conseqüentemente melhorar a sua experiência, isso pode ser feito, por exemplo, por meio de uma interface, na qual o usuário interage com um computador, um site ou um aplicativo, exigindo um esforço mínimo de sua parte para receber o resultado máximo desejado.

Em suma, dentre as contribuições trazidas por este estudo ressalta-se que a interface permitirá a manipulação de parâmetros de interesse sem exigir que o usuário tenha conhecimento especializado. Espera-se que a ela facilite a análise de resultados e aumente a eficiência da tomada de decisão. Além disso, pretende-se introduzir uma nova classe de restrições, pouco explorada até o momento, a fim de alcançar soluções ótimas de maneira mais rápida e eficiente, resultando em economia de tempo computacional. Sendo a metodologia desenvolvida estendida a outros modelos de otimização.

Os elementos-chave deste projeto são, portanto, projetar e implementar um aplicativo de interface de usuário e encontrar novas desigualdades que possam reforçar a formulação do problema [RSA](#) já existente.

## 1.1 Objetivo geral

Este projeto visa elaborar uma interface de usuário e justificar a sua utilidade através da proposta de introdução de novas desigualdades que reforcem a atual formulação proposta por [Colares, Kerivin e Wagler \(2021\)](#).

### 1.1.1 Objetivos específicos

Para cumprimento do objetivo geral é necessário atender aos seguintes objetivos específicos:

- Revisar estudos relacionados ao assunto;
- Instalar pacotes e bibliotecas para compilar um projeto já existente;
- Executar o projeto existente;
- Implementar e melhorar as funções da interface de usuário;

- Analisar as soluções obtidas por relaxação linear, com auxílio da interface;
- Revisar as obras de [Agarwal e Aneja \(2017\)](#), [Bienstock e Muratore \(2000\)](#) e [Brunetta, Conforti e Fischetti \(2000\)](#).
- Elaborar uma proposta de melhoria para a extensão do modelo.

## 1.2 Organização do Trabalho

A estrutura deste trabalho está organizada de forma a seguir uma ordem lógica e coerente. O Capítulo 1 se inicia com uma contextualização do problema abordado, apresentando a importância e a relevância do tema em questão. Em seguida, os objetivos da pesquisa são detalhados, descrevendo os resultados esperados. No Capítulo 2, é realizada uma revisão de literatura sobre os temas relacionados ao trabalho, são eles, [RSA](#), *Branch and Cut* e a biblioteca *Tkinter*, que foi utilizada para programar a interface. Através da revisão bibliográfica, é possível verificar quais os avanços, desafios e oportunidades na área, bem como conhecer os estudos mais recentes e relevantes. Em seguida, no Capítulo 3, são descritos os métodos de pesquisa usados no trabalho. Logo após, no Capítulo 4 são apresentados os procedimentos utilizados para instalar os diferentes pacotes e bibliotecas que permitirão que o compilador do código existente resolva o problema, e são apresentados os principais resultados obtidos referentes à interface de usuário e à proposta de introdução de uma nova classe de desigualdades na formulação de [Colares, Kerivin e Wagler \(2021\)](#). Por fim, no Capítulo 5 são apresentadas as considerações finais e perspectivas, em que buscou-se sintetizar todo o trabalho realizado e retomar os objetivos apresentados no início.

## 2 Revisão da Literatura

### 2.1 Roteamento e Alocação de Espectro - RSA

De acordo com [Hadhbi, Kerivin e Wagler \(2019\)](#), o problema de roteamento e alocação de espectro é um problema NP-difícil, ou seja, do ponto de vista prático, nenhum algoritmo eficiente é conhecido para resolvê-lo e vem recebendo atenção crescente nos últimos anos. O problema visa a alocação econômica de espectro, dada uma rede óptica elástica e tráfego a ser atendido. Para que cada requisição de tráfego seja satisfeita, o algoritmo [RSA](#) deve tentar atribuir uma rota e uma configuração de espectro, ou seja, uma modulação definida por um conjunto de subportadoras, também chamadas de *slots*, para alocação do sinal óptico.

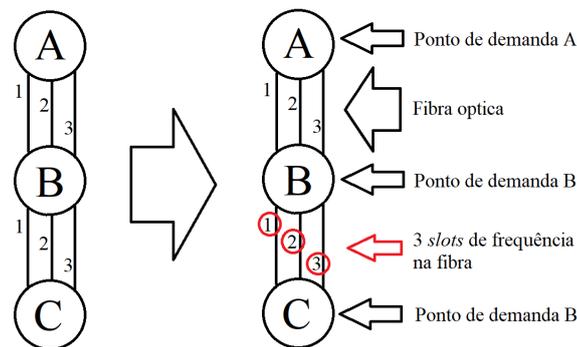
Existem alguns trabalhos relacionados a esta temática na literatura. A citar, [Christodoulopoulos, Tomkos e Varvarigos \(2011\)](#), que propõem um modelo de [Programação Linear Inteira \(ILP\)](#) para o problema de alocação conjunta do [RSA](#), que utiliza pré-computação de rota, e uma formulação [ILP](#) para a escolha da rota que é utilizada em conjunto com algoritmos de alocação de espectro compondo um algoritmo de etapas independentes (*R+SA*). O trabalho aborda a versão *offline* do problema para minimizar a utilização total do espectro sem bloquear as solicitações.

Já em [Jinno et al. \(2010\)](#), foi proposto um algoritmo [RSA](#) com uma estratégia adaptativa de escolha de alocação de espectro baseada no número de saltos necessários ao longo da rota escolhida. Esse algoritmo utilizou diferentes configurações e métodos de modulação, dependendo da distância percorrida pela rota selecionada. Para rotas mais longas, foram usadas modulações mais robustas e eficientes, como a 64 QAM, que é um esquema de modulação digital que permite transportar uma grande quantidade de dados em um mesmo intervalo de tempo e frequência, utilizando 64 símbolos para representar informações.

A maioria dos modelos revisados usa as formulações denominadas *edge-path*, em que as variáveis são mapeadas para todos os caminhos de roteamento possíveis, de forma que o número de variáveis cresce exponencialmente com o tamanho da instância. O modelo proposto por [Hadhbi, Kerivin e Wagler \(2019\)](#) busca fornecer a primeira formulação completa de nó de borda para o problema de roteamento e alocação de espectro que leva a um modelo de programação linear tratável em números inteiros.

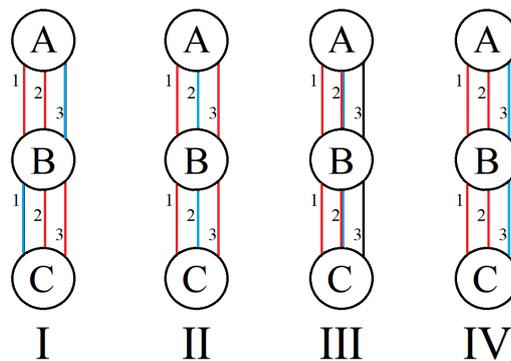
Em seu artigo [Hadhbi, Kerivin e Wagler \(2019\)](#) afirmam que, as soluções para o problema **RSA** são descritas com base em três regras fundamentais: a continuidade, a contiguidade e a não sobreposição do espectro. Essas restrições são cruciais para garantir que o espectro seja utilizado de forma eficiente e sem interferências. Para entender melhor essas regras, é possível recorrer às Figuras 1 e 2, nas quais se considera duas requisições distintas: uma representada pela cor vermelha (de A a C, com 2 slots) e outra pela cor azul (de A a C, com 1 slot), que precisam passar por *links* que disponibilizam 3 slots cada.

Figura 1 – Representação de uma rede de fibra óptica



Elaborada pela autora

Figura 2 – Representação das regras a serem satisfeitas no espectro



Elaborada pela autora

A Figura 2 refere-se à representação das três restrições que devem ser satisfeitas em uma configuração espectral em uma rede óptica, sendo possível observar que:

- No diagrama representado pelo esquema I, a restrição de continuidade espectral determina que, se um comando for aceito, ele usará os mesmos slots de frequência em todos os enlaces do caminho óptico.
- Em II, observa-se a restrição de contiguidade do espectro, ela define que slots de frequência alocados devem ser contíguos, o que significa que os slots devem ser adjacentes na fibra.

- Para a restrição de espectro não sobreposto, representada pelo Esquema III, as unidades de intervalo de frequência só podem ser atribuídas a uma única solicitação.
- Finalmente, no esquema IV, é possível observar uma configuração em que todas as regras são satisfeitas.

Considerando o exposto, o modelo originalmente descrito por [Hadhbi, Kerivin e Wagler \(2019\)](#), juntamente com a formulação melhorada descrita por [Colares, Kerivin e Wagler \(2021\)](#) para o problema **RSA** formam a base deste estudo. A seguir, eles são detalhados.

## 2.1.1 Modelo matemático

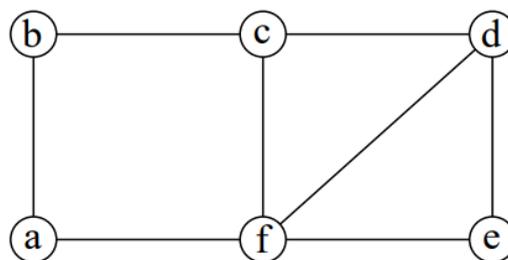
### 2.1.1.1 RSA - BASE

Visando compreender o modelo matemático do problema **RSA** proposto por [Hadhbi, Kerivin e Wagler \(2019\)](#) e posteriormente melhorado por [Colares, Kerivin e Wagler \(2021\)](#), será apresentado a seguir, o detalhamento das entradas, das saídas desejadas e das funções objetivo propostas, obtido através da leitura e de um estudo minucioso dos artigos.

Primeiramente, para melhor entendimento das entradas do problema **RSA**, deve-se considerar uma rede óptica, não direcionada, sem *loop*, representada por um grafo  $G = (V, E)$  como ilustrado pela Figura 3, com seis nós paralelos (fibras ópticas paralelas são instaladas entre dois nós), e para cada aresta  $e \in E$ , um comprimento  $\iota_e \in R_+$ , um espectro óptico  $S = \{1, \dots, \bar{s}\}$  de frequência disponível e um multiconjunto  $K$  de requisições, onde cada requisição  $k \in K$  é especificada por:

- um nó de origem  $o_k \in V$  e um nó de destino  $d_k \in V/o_k$ ,
- um número solicitado  $w_k \in N_+$  de *slots*, e
- um escopo de transmissão  $\bar{\iota}_e \in R_+$ .

Figura 3 – Grafo que representa uma rede óptica  $G$



Fonte: Elaborada pela autora

A tarefa consiste em encontrar, para cada requisição  $k \in K$ , um caminho ótimo  $P_k$  em  $G$  que respeite a faixa de transmissão  $\bar{l}$  e um canal  $S_k \subset S$  de  $w_k$  intervalos de frequência consecutivos (espectro de contiguidade) disponíveis em todas as bordas de  $P_k$ , garantindo a continuidade espectral e a disjunção dos canais  $S'_k$  com todas as outras solicitações  $k' \in K$  roteadas ao longo de uma borda de  $P_k$  (espectro não sobreposto). O objetivo é minimizar uma função objetivo. Vale ressaltar que o modelo considera o caso em que todas as solicitações devem ser atendidas. Sendo assim, a solução desejada para o problema RSA é um caminho para cada requisição  $k \in K$ , que consiste em:

- Um caminho  $(o_k, d_k) P_k$  em  $G$  com  $\sum_{e \in E(P_k)} l_e \leq \bar{l}_k$ ,
- Um subconjunto  $S_k \subset \{1, \dots, \bar{s}\}$  de  $w_k$  slots consecutivos com  $S_k \cap S_{k'} = \emptyset$  para cada solicitação  $k' \in K$  roteada ao longo de uma aresta  $e \in E(P_k)$ .

É fundamental destacar que a otimização de redes de comunicação é um desafio complexo que envolve várias variáveis interconectadas. No contexto específico da alocação de espectro em redes ópticas, a eficiência da rede pode ser medida por meio de diferentes funções objetivo, estas podem ser:

$O_1$ :  $\sum_{e \in E} \sum_{k \in K} x_e^k$ : minimizar a soma de saltos em caminhos (onde saltos se refere ao número de arestas em um caminho  $P_k$ ),

$O_2$ :  $\sum_{e \in E, k \in K} l_e x_e^k$ : minimizar a soma do comprimento total dos caminhos (considerando o tamanho individual das arestas  $l_e$ ),

$O_3$ :  $\sum_{s \in S} \sum_{k \in K} sz_s^k$ : minimizar a posição máxima do slot usado (e portanto a largura do o subespectro de  $S$  usado para alocação de espectro),

$O_4$ :  $p$ : minimizar a soma das posições máximas de slots usadas em todas as requisições.

Entre essas funções, destacam-se  $O_1$  e  $O_2$ , que têm como objetivo encontrar a melhor rota para transmitir dados em uma rede óptica, considerando a alocação de espectro existente. Essas funções são importantes para garantir um fluxo de dados adequado entre diferentes pontos da rede, sem congestionamento e perda de informações.

No entanto, a alocação de espectro em uma rede óptica pode ser otimizada ainda mais, buscando soluções mais eficientes em termos de uso do espectro disponível. Nesse sentido, as funções  $O_3$  e  $O_4$  buscam identificar a melhor alocação de espectro para todas as rotas possíveis na rede, visando maximizar o uso do espectro disponível e reduzir custos de operação.

Dessa forma, é importante destacar que as funções  $O_1$  e  $O_2$  são necessárias para garantir um roteamento eficiente, mas as funções  $O_3$  e  $O_4$  podem levar a uma alocação de espectro mais eficiente e, conseqüentemente, a uma rede óptica mais eficaz e rentável.

Assim, a otimização de redes de comunicação é um processo complexo que envolve uma série de decisões interdependentes. Ao considerar diferentes funções objetivo, é possível identificar soluções mais eficientes e econômicas para a alocação de espectro em uma rede óptica.

Analisando agora as variáveis do problema, tem-se:

- $x_e^k = \{1, 0\}$ ; se a requisição  $k$  for roteada pela aresta  $e$ .
- $z_s^k = \{1, 0\}$ ; se o *slot*  $s$  for o último alocado pela requisição  $k$ .
- $t_{e,s}^k = \{1, 0\}$ ; se o *slot*  $s$  for atribuído à requisição  $k$  pela aresta  $e$ .

Além disso, o problema em questão envolve um conjunto de dez famílias de restrições. Cada uma delas é representada a seguir de forma distinta, a fim de identificar e compreender melhor suas particularidades.

$$\sum_{e \in \delta(v)} x_e^k = 1, \quad \forall k \in K, v \in o_k, d_k, \quad (2.1)$$

As equações (2.1) estabelecem que a soma das requisições  $k$  roteadas pela aresta  $e$  deve ser igual a 1, o que significa que a quantidade total de tráfego que passa por essa aresta deve ser distribuída de forma equilibrada entre as diferentes solicitações que a utilizam como rota.

$$\sum_{e \in \delta(v)} x_e^k \leq 2, \quad \forall k \in K, v \in V / (o_k, d_k), \quad (2.2)$$

As restrições (2.2) tem como objetivo evitar que um nó seja conectado a mais de duas arestas em  $E(k)$ , garantindo que a topologia da rede seja coerente e eficiente. Ao impor essa restrição, limita-se a quantidade de conexões de cada nó e são evitados possíveis gargalos ou sobrecarga em alguns pontos da rede.

$$\sum_{e \in \delta(X)} x_e^k \geq 1, \quad \forall k \in K, X \subseteq V / d_k, o_k \in X, \quad (2.3)$$

As restrições (2.3) garantem que para cada conjunto  $X$  com origem  $o_k \in X$  e destino  $d_k \in V/X$ , exista pelo menos uma aresta  $e \in \delta(X) \cap E(k)$  que cruze o corte  $(X, V/X)$  ao longo do caminho  $P_k$ . Essa condição é importante para garantir que todos os fluxos de tráfego possam ser roteados de forma adequada e eficiente pela rede. Ao impor essa restrição, assegura-se que os caminhos  $P_k$  sigam uma topologia coerente e que haja um equilíbrio na distribuição do tráfego ao longo da rede.

$$\sum_{e \in \delta(X)} x_e^k \geq \begin{cases} 2x_{uv}^k & \text{si } |X \cap \{o_k, d_k\}| = 0 \\ x_{uv}^k & \text{si } |X \cap \{o_k, d_k\}| = 1 \end{cases}, \quad (2.4)$$

$$\forall k \in K, X \subseteq V/d_k : o_k \in X, \quad (2.5)$$

A classe de restrições (2.4) tem como finalidade a eliminação de ciclos. Ciclos são caminhos fechados que não contribuem para a rotação de tráfego e podem causar desperdício de recursos da rede. Essa restrição é fundamental para garantir a eficiência e o bom funcionamento do sistema, já que evita que recursos sejam utilizados de forma desnecessária e maximiza o uso da rede. Ao eliminar ciclos, o problema *RSA* se torna mais simplificado e efetivo, permitindo a resolução de forma mais rápida e precisa.

$$\sum_{e \in E} l_e x_e^k \leq \bar{l}_k, \quad \forall k \in K, \quad (2.6)$$

As restrições (2.6) tem como objetivo garantir que o comprimento do caminho  $P_k$  não exceda o alcance de transmissão, caso a solicitação  $k$  seja aceita. Caso contrário, ou seja, se a solicitação  $k$  não puder ser atendida sem violar o limite de alcance de transmissão, todas as variáveis  $x_e^k$  são forçadas a serem iguais a 0. Isso é importante para evitar que a rede fique sobrecarregada ou inoperante, uma vez que as solicitações precisam ser atendidas de forma compatível com as limitações da tecnologia de transmissão disponível.

$$\sum_{w_k \leq s \leq \bar{s}} z_s^k = 1, \quad \forall k \in K, \quad (2.7)$$

As equações (2.7) tem como objetivo forçar a seleção de um último *slot*  $s$  no canal sempre que um caminho sai. Essa medida é importante para garantir que os recursos da rede sejam utilizados de forma eficiente e que não ocorram interrupções ou falhas na transmissão de dados.

$$\sum_{1 \leq s \leq w_k} z_s^k = 0, \quad \forall k \in K, \quad (2.8)$$

As equações (2.8) especificam quais são os últimos *slots*  $s$  disponíveis para o canal da solicitação  $k$ . Ela é importante para evitar a interferência entre as solicitações que compartilham o mesmo canal de comunicação. Ao especificar quais são os últimos *slots* disponíveis, garante-se que a transmissão de dados ocorra sem colisões.

$$\sum_{s \in S} t_e^{s,k} = w_k x_e^k, \quad \forall k \in K, e \in E, \quad (2.9)$$

As equações (2.9) asseguram que  $w_k$  slots sejam alocados pela aresta  $e$  para cada solicitação  $k$ , somente se a solicitação  $k$  for roteada pela aresta  $e$ . Ao alocar slots somente para as solicitações que utilizam determinada aresta, evita-se o desperdício de recursos.

$$x_e^k + \sum_{s' \in S} z_{s'}^k \leq 1 + t_e^{s,k}, \quad \forall k \in K, e \in E, s \in S, \quad (2.10)$$

As restrições (2.10) forçam a alocação de um slot  $s$  para a solicitação  $k$ , somente se a solicitação  $k$  passar pela aresta  $e$  e o slot  $s$  pertencer ao canal atribuído para a demanda  $k$ .

$$\sum_{k \in K} t_e^{s,k} \leq 1, \quad \forall e \in E, s \in S, \quad (2.11)$$

As restrições (2.11) possuem a função de garantir que o slot  $s$  seja alocado para no máximo uma solicitação  $k$ .

$$x_e^k, z_s^k, t_e^{s,k} \in \{0, 1\}, \quad \forall k \in K, e \in E, s \in S, \quad (2.12)$$

Por fim, é apresentado o domínio das variáveis.

#### 2.1.1.2 RSA - EXT

Em um estudo posterior, Colares, Kerivin e Wagler (2021) refinaram ainda mais essa abordagem. Eles propuseram algumas mudanças na formulação a fim de obter melhores resultados, o novo modelo foi denominado RSA - EXT, por caracterizar uma extensão do RSA - BASE. As melhorias baseiam-se em uma mudança na interpretação do grafo, representado pela Figura 3 e a adição de um conjunto completamente novo de variáveis. Assim, a formulação extendida proposta utiliza um grafo direcionado  $G'$  da rede  $G$ , construído criando um arco interno e um arco externo para cada aresta de  $G$ .

A partir dessa nova abordagem surgiu a necessidade de criação de uma variável de fluxo, como representado a seguir:

- $f_s^k$ ; variável binária, que indica se a requisição  $k$  é roteada pelo arco  $a$  e se o slot  $s$  é o último slot do canal;

Com essa variável em mãos, os autores sugerem que as variáveis  $x_e^k$ ,  $z_s^k$  e  $t_{uv}^{sk}$  do modelo anterior, classificado como RSA-BASE, sejam escritas em função da nova variável  $f_s^k$ .

$$x_e^k = \sum_{s=1}^{\bar{s}} (f_{u,v}^{sk} + f_{v,u}^{sk}), \quad \forall uv \in E, k \in K, \quad (2.13)$$

$$z_s^k = \sum_{a \in \delta^+(o_k)} f_a^{sk}, \quad \forall k \in K, s \in S \quad (2.14)$$

$$t_{uv}^{sk} = \sum_{i=s}^{\min\{\bar{s}, s+w_k-1\}} (f_{u,v}^{ik} + f_{v,u}^{ik}), \quad \forall uv \in E, k \in K, s \in S, \quad (2.15)$$

Desta forma, todo o modelo foi transformado. Para conhecimento, pode-se observar como ficou a classe de restrições (2.6) após as modificações das variáveis.

$$\sum_{e \in E} l_e x_e^k \leq \bar{l}_k, \quad \forall k \in K, \quad (2.16)$$

$$\sum_{a \in A(G')} \sum_{s=1}^{\bar{s}} l_a f_a^{sk} \leq \bar{l}_k, \quad \forall k \in K, \quad (2.17)$$

$$\sum_{a \in A(G')} l_a f_a^{sk} \leq \bar{l}_k \sum_{a \in \delta^+(o_k)} f_a^{sk}, \quad \forall k \in K, s \in S, \quad (2.18)$$

Em seu estudo, [Colares, Kerivin e Wagler \(2021\)](#) comprovaram que tal formulação é realmente mais forte, pois, embora *RSA-BASE* e *RSA-EXT* tenham o mesmo número de desigualdades, *RSA-EXT* usa menos que o dobro do número de variáveis usadas em *RSA-BASE*. Portanto, a parte matemática deste trabalho foi baseada na formulação *RSA-EXT*.

## 2.2 Branch and Cut

Muitos problemas de otimização combinatória podem ser formulados como problemas de programação linear inteira mista. Eles podem então serem resolvidos pelo método *Branch-and-Cut*. De acordo com [Cunha, Simonetti e Lucena \(2014\)](#), esse algoritmo caracteriza-se como um algoritmo do tipo *Branch-and-Bound* no qual planos de cortes são gerados para os diferentes problemas definidos nos nós da árvore de enumeração. E isso é executado com o intuito de fortalecer os limitantes duais associados aos mesmos e eventualmente reduzir o número de nós da árvore.

De acordo com [Wolsey e Nemhauser \(1999\)](#) esse método funciona resolvendo uma sequência de relaxações de programação linear da programação inteira problema, usando o algoritmo *Simplex*. Quando uma solução ótima é encontrada e uma das variáveis inteiras desejadas tem um valor não inteiro, usa-se um algoritmo de plano secante para encontrar uma restrição linear satisfeita por todas as soluções inteiras, mas violada pela solução fracionária. Se tal restrição for encontrada, então é adicionada ao programa linear, fazendo com que seja fornecida uma solução com menos valores não inteiros. Este processo é repetido até que uma solução inteira seja encontrada ou até que nenhum plano secante possa ser encontrado.

São muitos os exemplos de aplicação deste método na literatura, para solucionar o modelo do problema *RSA* descrito na seção anterior, [Hadhbi, Kerivin e Wagler \(2019\)](#) e [Colares, Kerivin e Wagler \(2021\)](#), usaram o algoritmo *Branch and Cut*. A fim de garantir a obtenção de uma solução do problema do caixeiro viajante, [Loparic \(1996\)](#), combinou o algoritmo de planos cortantes com o *Branch and Bound*, trabalhando com um limitante inferior e um limitante superior do valor ótimo do problema. Já [González et al. \(2017\)](#), propõe um algoritmo *Branch-and-Cut* para resolver o problema de localização de contadores de tráfego, que consiste em determinar o número e a localização dos postos de pesquisa que melhor atendam uma rede de transporte com a finalidade de estimar matrizes de origem e destino. [Abreu et al. \(2018\)](#), ainda vai além, combinando o método *Branch and Cut* com heurísticas para explorar o problema de localização de sensores de contagem de veículos, instalados em segmentos viários, para o problema de estimativa de fluxo de veículos, também entre pares de origem e destino.

Tal como no método *Branch and Cut*, a proposta de melhoria do projeto baseia-se em ultrapassar os limites, para que a solução ótima seja encontrada mais rapidamente, através do acréscimo de uma nova classe de desigualdades.

## 2.3 Biblioteca *Tkinter*

De acordo com a documentação oficial do Python, disponível em <https://docs.python.org/3/library/tkinter.html>, *Tkinter* é a biblioteca padrão do *Python* para a criação de interfaces gráficas utilizando o conjunto de ferramentas *Tk*. Uma das suas principais vantagens é sua simplicidade, o que a torna ideal para iniciantes em programação de interfaces gráficas. No entanto, a biblioteca também é capaz de criar aplicações mais complexas e personalizadas, com a inclusão de elementos, como menus, botões, caixas de texto e muitos outros.

Esta biblioteca é comumente utilizada, em um estudo de caso realizado por [Afolabi, Salawu e Olugbara \(2020\)](#), foi desenvolvido um sistema de gerenciamento de biblioteca utilizando a biblioteca *TKinter*. Os resultados mostraram que a interface do usuário criada era clara e fácil de usar, proporcionando uma experiência positiva para o usuário.

Outro estudo realizado por [Kim e Han \(2020\)](#) explorou a utilização de técnicas de *design* de interação para melhorar a usabilidade da interface criada com a biblioteca *Tkinter*. O estudo mostrou que a aplicação de princípios de *design* resultaram em uma interface mais intuitiva e eficiente para o usuário.

Além disso, a biblioteca *Tkinter* também é frequentemente utilizada em projetos de pesquisa científica, como o estudo realizado por [Ma e Sun \(2021\)](#) que desenvolveram uma interface de usuário para analisar dados de imagens médicas usando a biblioteca *Tkinter*. O estudo concluiu que a biblioteca é capaz de criar uma interface de usuário eficiente e fácil de usar para aplicação em pesquisas científicas.

Em resumo, a biblioteca *Tkinter* é uma ótima opção para a criação de interfaces em *Python*, especialmente para aqueles que estão começando a programar interfaces gráficas. A simplicidade da biblioteca permite a criação de interfaces básicas, enquanto técnicas de *design* podem ser utilizadas para aprimorar a usabilidade e melhorar a experiência do usuário. Além disso, a biblioteca também é capaz de lidar com projetos mais complexos e personalizados. Desta forma, ela será usada neste trabalho.

## 3 Metodologia

Esta pesquisa pode ser classificada quanto a sua natureza, como, básica estratégica, partindo do pressuposto de que pretende-se deixar bases para resolução de problemas futuros. De acordo com [Gil \(2010\)](#) uma pesquisa básica combina estudos que tem como objetivo completar uma lacuna no conhecimento, e que em seu âmbito estratégico, busca novos conhecimentos direcionados a amplas áreas com vistas à solução de reconhecidos problemas práticos. Já [Schwartzman \(1979\)](#) afirma que a pesquisa básica é aquela que acumula conhecimentos e informações que podem eventualmente levar a resultados acadêmicos ou aplicados importantes, mas sem fazê-lo diretamente.

No que se refere aos objetivos, a pesquisa é, sobretudo, normativa, de acordo com [Bertrand e Fransoo \(2002\)](#) este tipo de pesquisa é interessada no desenvolvimento de políticas, estratégias e ações para aperfeiçoar os resultados disponíveis na literatura existente. Sua abordagem pode ser considerada combinada, qualitativa e quantitativa, composta de dois momentos distintos. Para [Martins \(2011\)](#) uma abordagem combinada é complexa e poderosa, sendo desafiante para os pesquisadores.

No que diz respeito ao método empregado pode ser classificado como “modelagem e simulação”, de acordo com [Turrioni e Mello \(2012\)](#), uma vez que se utiliza modelos matemáticos para representar um problema específico. Quanto aos procedimentos, a pesquisa é de caráter bibliográfico. Para [Fachin \(2001\)](#), a pesquisa bibliográfica é a base para as demais. [Ruiz \(1996\)](#) revela que esta consiste no exame e análise do que já se produziu sobre determinado tema. Já para [Lakatos e Andrade \(2010\)](#), a pesquisa bibliográfica refere-se àquela na qual se realiza a partir de material disponível, decorrente de pesquisas anteriores em documentos impressos, como livros periódicos, artigos e outros.

No que se refere aos passos da pesquisa, foi feita uma análise predominantemente qualitativa documental, constituída de revisão de literatura para entendimento e delineamento do escopo do problema trabalhado. De acordo com [Echer \(2001\)](#) a revisão da literatura tem como finalidade reconhecer a unidade e a diversidade interpretativa existente no eixo temático em que se insere o problema em estudo, visando ampliar e ramificar a análise interpretativa, bem como para compor as abstrações e síntese que forneçam um alicerce para as argumentações do pesquisador. Além disso, esse método de pesquisa evita que haja duplicação de esforços e fornece uma melhor visão do tema estudado e segurança em relação à sua pertinência.

Posteriormente foi realizada uma abordagem prática, em que surgiu a necessidade de instalação de diversos pacotes e bibliotecas para a compilação e execução do projeto, cuja autoria pertence a [Colares, Kerivin e Wagler \(2021\)](#), que serviu de base para todo este estudo. Esta etapa do trabalho demandou muito tempo e estudo, pois haviam erros no código base, elaborado em C++.

Concomitante a isso, iniciou-se a criação de uma interface de usuário, com o objetivo de fazer com que uma pessoa não especialista em programação ou em pesquisa operacional pudesse alterar os parâmetros da formulação em um momento futuro, além de auxiliar na análise das soluções geradas, a partir de relaxação linear. Após a execução e conhecimento das soluções, foram feitos estudos, baseados, em sua grande parte, no trabalho de [Brunetta, Conforti e Fischetti \(2000\)](#), com a finalidade de propor uma melhoria no modelo, a partir da metodologia usada no artigo.

## 4 Resultados

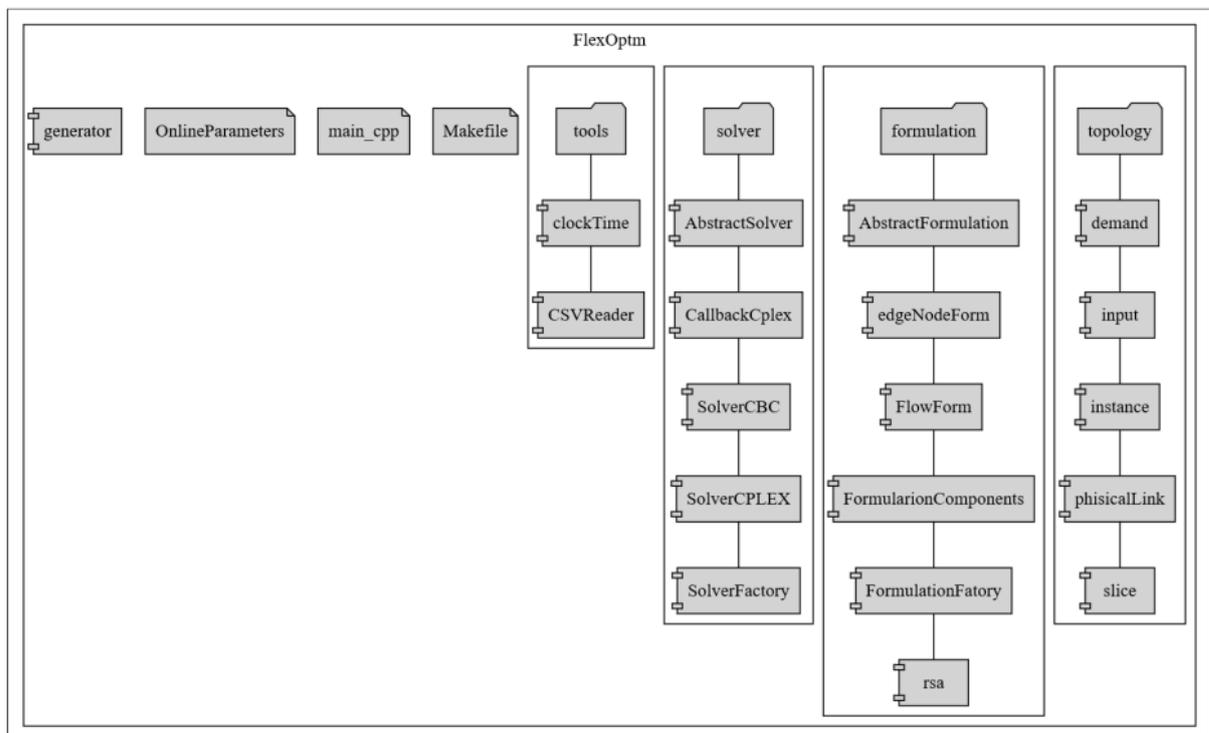
### 4.1 Compilação do código do *RSA-EXT*

Nesta etapa do trabalho, foram feitas as instalações dos pacotes e bibliotecas necessárias para a compilação e execução do modelo matemático do problema denominado por Colares, Kerivin e Wagler (2021) como *RSA-EXT*. Este projeto computacional foi desenvolvido no LIMOS, financiado pela ANR, recebeu o nome de *FlexOptim* e possui uma estrutura bem robusta, como será detalhada a seguir.

#### 4.1.1 Projeto *FlexOptim*

Em um estudo também baseado neste projeto, Sá e Kerivin (2021) desenvolveram um diagrama, apresentado pela Figura 4, que tem por objetivo mostrar todas as pastas e arquivos que compõem o projeto de implementação.

Figura 4 – Como os arquivos e pastas são organizados no *FlexOptim*



Elaborado por Sá e Kerivin (2021)

O código foi escrito na linguagem de programação C++ e está disponível no repositório *GitHub* do principal desenvolvedor, Rafael Colares, e pode ser acessado através do link <<https://github.com/rafael-colares/FlexOptim>>.

No que pode-se chamar de primeiro nível, o projeto possui três arquivos *Makefile*, *OnlineParameters* e *main code*, um objeto de classe denominado *generator* e quatro pastas, *formulation*, *solver*, *tools* e *topology*.

- O *OnlineParameters* é o arquivo usado para identificar todos os parâmetros do modelo.
- O *main.cpp* é o arquivo principal do programa em C++, em que está localizada a função principal, e onde o programa inicia e termina sua execução.
- O *Makefile* é responsável por fazer a ligação entre todos os códigos e todas as bibliotecas.
- A pasta *tools* contém apenas duas classes de exploração: *CSVReader*, que implementa um leitor de arquivo *.csv* e é usada para ler arquivos de entrada e a *clockTime*, que é usada para controlar o tempo gasto pelo programa.
- A pasta *solver* contém todas as implementações relacionadas à resolução do problema **RSA**. Dentro dela, pode-se observar a classe *AbstractSolver* que descreve uma classe abstrata que modela um solucionador de problemas de **Programação Inteira Mista (MIP)**. Ela deriva-se em duas outras classes, para dois solucionadores diferentes: A classe *SolverCPLEX* que implementa uma formulação do **RSA** usando *IBM ILOG CPLEX* e *SolverCBC*, que implementa uma formulação do **RSA** usando *CBC*. Essas duas classes estão relacionadas à classe *SolverFactory*, que é responsável por fornecer um solucionador concreto. Além disso, a pasta ainda possui uma classe derivada de uma classe nativa da biblioteca *ILOG CPLEX*, a *callbackCPLEX*, esta implementa uma interface de retorno de chamada genérica, com duas funções principais: *add User cuts* e *add Lazy Constraints*.
- A pasta *formulation* é responsável por implementar o modelo. Dentro dela observa-se, entre outros arquivos, o *FormulationComponents*, que contém várias estruturas de dados usadas para a formulação **MIP** do problema.
- Por fim, a pasta *topology* contém todas as estruturas de dados utilizadas pelo modelo. A classe *input* mantém todas as informações fornecidas pelo arquivo de entrada *OnlineParameters*, como localização da topologia e arquivos de requisição, número de requisições roteadas simultaneamente, onde escrever o arquivo de saída, o limite de tempo de execução do código, qual função objetivo é usada, o solucionador escolhido e alguns outros parâmetros específicos do solucionador. Cada um desses parâmetros representa um atributo da entrada da classe e seu código fonte contém todos os *getters* e *setters*, métodos de conversão de tipo.

#### 4.1.2 Instalação de pacotes

Para a compilação do projeto, foi usado um computador pessoal, cuja as características são: processador Intel(R) Core(TM) i7-7500U de 2,70 GHz e 64 bits com 16 GB de RAM.

Foram utilizadas diversas ferramentas a partir da linha de comando para instalar as dependências do projeto. O *Git* foi utilizado para clonar os diretórios e a execução dos *scripts* exigiu a utilização do *GNU make* e *Cmake*. Algumas dessas ferramentas são compatíveis com diferentes sistemas operacionais, enquanto outras são projetadas especificamente para *Linux* e não levam em consideração as particularidades do *Windows*. Sabendo disso, foi necessário utilizar uma máquina virtual com a distribuição Ubuntu, um sistema operacional de código aberto baseado em *Linux*, para realizar os procedimentos. Na próxima seção, será explicado como o *Windows Subsystem for Linux (WSL)* foi usado para permitir o uso do sistema operacional. Antes disso, no entanto, serão apresentados os passos que foram seguidos para instalar os pacotes e bibliotecas necessários para a construção *build* do projeto.

- O código do problema *RSA* foi desenvolvido para o sistema *Linux*. Mas pode ser usado no *Windows*, como mencionado anteriormente, instalando o *WSL*. Para isso foram seguidos os passos descritos em um arquivo, que pode ser acessado através do seguinte link, disponibilizado pelo professor Hervé Kerivin, orientador do projeto: <<https://docs.microsoft.com/en-us/windows/wsl/install-win10>>
- Feito isso, foi necessária a instalação da biblioteca *LEMON COIN-OR*, que foi utilizada para modelar os gráficos. Para fazer isso, as instruções presentes no seguinte link foram seguidas: <<https://lemon.cs.elte.hu/trac/lemon/wiki/InstallGuide>>
- Em seguida, instalou-se a biblioteca *Boost*, usada para simplificar a verificação de arquivos .csv. A biblioteca *Boost* pode ser encontrada aqui: <[https://www.boost.org/doc/libs/1\\_55\\_0/doc/html/bbv2/installation.html](https://www.boost.org/doc/libs/1_55_0/doc/html/bbv2/installation.html)>
- O código disponibilizado usa *IBM ILOG CPLEX* versão 12.10 (versões mais recentes também são suportadas), tal ferramenta foi instalada seguindo as instruções disponíveis no seguinte link: <<https://developer.ibm.com/>>
- Para a resolução de problemas de programação inteira mista, utilizou-se a biblioteca *COIN-OR CBC*. O *CBC* foi instalado seguindo as instruções descritas aqui: <<https://projects.coin-or.org/Cbc>>
- Finalmente, para calcular a relação sinal óptico para ruído óptico (*OSNR*) de uma rota que pode ser usada com precisão, o *GNPY* foi usado. Para a sua instalação foi necessário seguir as instruções aqui descritas: <<https://gnpy.readthedocs.io/en/master/install.html>>

### 4.1.3 *Windows Subsystem for Linux (WSL)*

Uma máquina virtual é uma máquina implementada por meio de *software* que executa programas como um computador real, com recursos de *hardware* virtuais, (O. . . , 2023). Isso permite o uso do sistema operacional *Linux* em um ambiente virtualizado, sem a necessidade de mudar o sistema operacional do computador usado para a compilação.

O [WSL](#) é um recurso opcional disponível no *Windows 10* que permite executar binários e *scripts Linux* diretamente no *Windows*, traduzindo as instruções enviadas ao sistema em uma instrução válida para o *Windows* ([WINDOWS... , 2023](#)). Com o *WSL*, é possível ter um ambiente idêntico a uma distribuição *Linux*. Ele permite instalar várias distribuições *Linux* na máquina utilizada, cada uma com uma estrutura de arquivos isolada, independente do sistema e de outras distribuições, semelhante a uma máquina virtual.

É importante destacar que a linguagem de programação utilizada para criar o pacote de arquivos é compatível com o sistema operacional *Windows*, portanto o comando "dos2unix" foi usado para converter o arquivo .txt de *Windows* para *Linux*. O "dos2unix" é uma ferramenta que converte arquivos de texto do formato DOS/*Windows* (com quebra de linha CRLF) para o formato *Unix/Linux* (com quebra de linha LF), permitindo que os arquivos sejam utilizados corretamente no ambiente *Linux* ([COMO... , 2023](#)).

#### 4.1.4 *Makefile*

A compilação do projeto é feita através de um *makefile*. Para fazer isso, assumindo que possui-se todos os requisitos descritos anteriormente, foi preciso atualizar as entradas *CHANGEME* no *makefile* do projeto para os diretórios correspondentes onde cada uma das bibliotecas foi instalada.

De acordo com [Stallman e McGrath \(1991\)](#), o utilitário *make* determina automaticamente quais partes de um grande programa precisam ser recompiladas e emite comandos para recompilá-las. A maioria são programas *C*, pois são os mais comuns, mas pode-se usar *make* para qualquer linguagem de programação cujo compilador possa ser executado com um comando *shell*. De fato, *make* não está limitado a determinados programas. É possível usá-lo para descrever qualquer tarefa em que alguns arquivos precisem ser atualizados automaticamente a partir de outros sempre que os outros forem alterados. Uma vez que exista um *makefile* adequado, sempre que for necessário modificar algum arquivo de origem, deve-se usar este simples comando *shell*: *make*.

Considerando o exposto, o *makefile* foi usado para executar todos os comandos de inclusão de biblioteca e criar configurações para o projeto *FlexOptim*.

#### 4.1.5 Parâmetros do problema

O projeto criado por [Colares, Kerivin e Wagler \(2021\)](#) possui um arquivo *onlineParameters.txt* que permite modificar os parâmetros cada vez que o projeto é compilado. A seguir eles são apresentados:

- *topologyFile*: estabelece o endereço do arquivo que contém as informações dos links da topologia física.

- *initialMappingDemandFile*: designa o endereço do arquivo que contém as informações das requisições presentes no mapeamento inicial.
- *initialMappingAssignmentFile*: denomina o endereço do arquivo contendo as informações sobre o roteamento e a atribuição espectral das requisições presentes no mapeamento inicial.
- *demandToBeRoutedFolder*: caracteriza o endereço da pasta que contém todos os arquivos de requisição a serem roteados.
- *GNPY\_activation*: parâmetro booleano que indica se o *GNPY* está ativo ou não.
- *GNPY\_topologyFile*: caracteriza o endereço do arquivo que contém as informações de topologia específicas, partindo do pressuposto de que para se usar *GNPY*, a topologia deve ser definida com precisão.
- *GNPY\_equipmentFile*: nomina o endereço do arquivo que contém as informações específicas do dispositivo, considerando que para se usar o *GNPY*, o equipamento usado na rede também deve ser definido com precisão.
- *nbDemandsAtOnce*: indica quantas requisições são processadas em uma etapa de otimização.
- *formulation*: representa a escolha da formulação a ser utilizada. Sendo valor 0 para a formulação de fluxo e 1 para *Edge-Node*.
- *userCuts*: parâmetro booleano que indica se devem ser aplicados cortes do usuário.
- *obj*: permite que seja escolhida qual função objetivo otimizar. Sendo número 1 para minimizar o último *slot* usado por solicitação, 1p para minimizar o último *slot* usado por aresta, 2 para minimizar o número de saltos (arestas), 2p para minimizar as fatias usadas, 4 minimiza o comprimento do caminho e 8 minimiza o último *slot* usado no conjunto.
- *allowBlocking*: parâmetro booleano que indica se a otimização pode continuar após o bloqueio de uma requisição.
- *hopPenalty*: requisição de penalidade aplicada a cada salto.
- *partitionPolicy*: indica se o espectro é particionado. Sendo 0 para nenhuma partição e 1 para empurrar as solicitações maiores para a direita e as menores para a esquerda.
- *partitionLoad*: indica a carga mais alta que deve estar na partição esquerda.
- *partitionSlice*: se a estratégia de partição = 2, indica onde o espectro está dividido.
- *Solver*: descreve qual solucionador usar. Sendo 0 para *CPLEX* ou 1 para *CBC*.

- *método*: indica possíveis métodos a aplicar a cada nó da árvore de enumeração, 0 para relaxamento linear.
- *preprocessingLevel*: possíveis níveis de pré-processamento a serem aplicados para eliminar variáveis antes de chamar a otimização. Em que pode-se escolher 0 para remover apenas os arcos que não correspondem à demanda de carga, 1 para encontrar arcos que induziriam uma violação de comprimento e arcos cujos vizinhos não podem encaminhar a solicitação ou 2 para aplicar o nível 1 recursivamente até que nenhum arco possa ser excluído.
- *linearRelaxation*: parâmetro booleano utilizado para transformar variáveis inteiras em variáveis lineares.
- *relaxMethod*: parâmetro booleano usado para habilitar ou desabilitar a relaxação para o método.
- *lagrangianRelaxation*: parâmetro booleano utilizado para habilitar ou desabilitar a relaxação lagrangeana.
- *outputPath*: parâmetro que refere-se ao endereço da pasta para onde os arquivos de saída serão enviados ao final do procedimento de otimização.
- *outputLevel*: possíveis estratégias de saída a serem usadas. Podendo selecionar as seguintes opções, 0 para não criar nenhum arquivo de saída. 1 para gerar os arquivos de saída correspondentes ao último mapeamento. 2 para gerar arquivos de saída após cada iteração de otimização.
- *nbSlicesInOutputFile*: indica o número de fatias que serão exibidas no arquivo de saída.
- *globalTimeLimit*: refere-se ao tempo (em segundos) que pode ser gasto durante toda a otimização.
- *timeLimit*: refere-se ao tempo (em segundos) que pode ser transcorrido durante uma iteração de otimização.

A compilação do modelo exigiu muito esforço e tempo. Foi necessário reinstalar todas as bibliotecas do zero, e depois de muitas tentativas identificou-se um erro no arquivo *makefile*, algumas referências não estavam devidamente definidas. Desta forma, foi feita a correção, compilação e execução do código, obtendo assim as soluções derivadas da relaxação linear.

Tendo agora conhecimento sobre todo o contexto do problema em questão, partiu-se para o objetivo principal deste trabalho, que é a criação de uma interface de usuário e, a partir da análise das soluções visualizadas através dela, elaborar uma proposta de melhoria para a formulação do *RSA - EXT*.

## 4.2 Interface de usuário para o *RSA Problem (FlexOptim)*

A fim de criar um ambiente onde um usuário que não possuísse conhecimento em pesquisa operacional e nem em programação pudesse editar os parâmetros da formulação proposta por Colares, Kerivin e Wagler (2021) e que as soluções derivadas da relaxação linear pudessem ser melhor analisadas, foi sugerida a criação de uma interface de usuário. Para isso, a primeira etapa solicitada pelo orientador foi a escolha da linguagem de programação, esta se deu após algumas pesquisas em sites da internet e também em artigos científicos, conforme descrito a seguir.

### 4.2.1 Escolha da linguagem *Python* e da biblioteca *Tkinter*

Para implementar os códigos, foi escolhida a linguagem *Python*, que é de primeira linha, orientada a objetos, tipada dinamicamente e forte, interpretada e interativa, segundo (BORGES, 2014). A sintaxe clara e concisa dessa linguagem favorece a legibilidade do código fonte, tornando-a mais produtiva. Além disso, *Python* conta com vários *frameworks* de alto nível e uma extensa coleção de módulos prontos para uso.

A escolha da linguagem *Python* se deve não só à sua simplicidade e clareza, mas também à sua capacidade de gerenciar sistemas e desenvolver grandes projetos. Uma vantagem significativa dessa linguagem é a legibilidade dos programas escritos nela. Ao contrário de outras linguagens de programação, que utilizam marcações como ponto ou ponto e vírgula ao final de cada linha, além de blocos marcadores de início e fim, *Python* dispensa esses elementos, o que facilita a leitura do código.

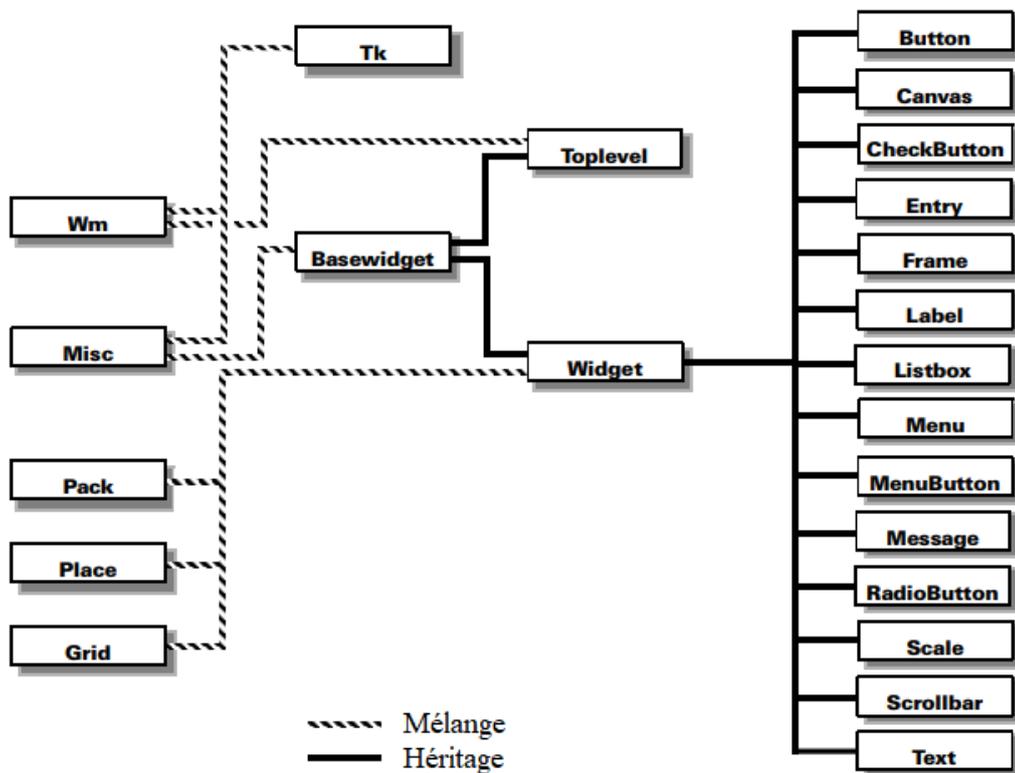
*Python* é uma linguagem completa, com bibliotecas para diversas finalidades, como acesso a bancos de dados, processamento de arquivos XML, criação de interfaces gráficas e jogos. Isso significa que é possível usar funções já existentes, escrevendo poucas linhas de código e aumentando a produtividade do programador. O uso de bibliotecas desenvolvidas e testadas por outras pessoas também reduz o número de erros e permite que o programador se concentre no problema a ser resolvido.

Para implementar a interface de usuário, foi utilizada a biblioteca *Tkinter* em *Python*. Grayson (2000) destaca que *Tkinter* fornece uma interface fácil de programar para os aplicativos *Python*, com uma coleção de ferramentas *Tk* que suportam a maioria das necessidades dos aplicativos. Embora *Tcl/Tk* seja mais utilizado em sistemas UNIX, desenvolvedores de várias áreas têm usado essa biblioteca em sistemas *Windows* e *MacOS*.

O funcionamento da biblioteca *Tkinter* é feito através do interpretador *Python*, utilizando o módulo *Tkinter* que, por sua vez, conta com a interface *Tkinter* que chama as bibliotecas *Tcl/Tk* e, em alguns casos, o interpretador *Tcl* para vincular propriedades aos *widgets*.

Ao contrário de muitos sistemas de interface, a hierarquia do *Tkinter* é bastante simples e não apresenta uma estrutura hierárquica. As classes *WM*, *Misc*, *Pack*, *Place* e *Grid* são misturadas para cada classe widget. A maioria dos programadores só precisa saber o nível mais baixo da árvore para executar as operações do dia a dia e muitas vezes é possível ignorar os níveis mais altos, conforme destaca [Grayson \(2000\)](#). A hierarquia nocional é ilustrada na Figura 5, criada pelo autor.

Figura 5 – Ferramenta *Tkinter* "hierarquia"

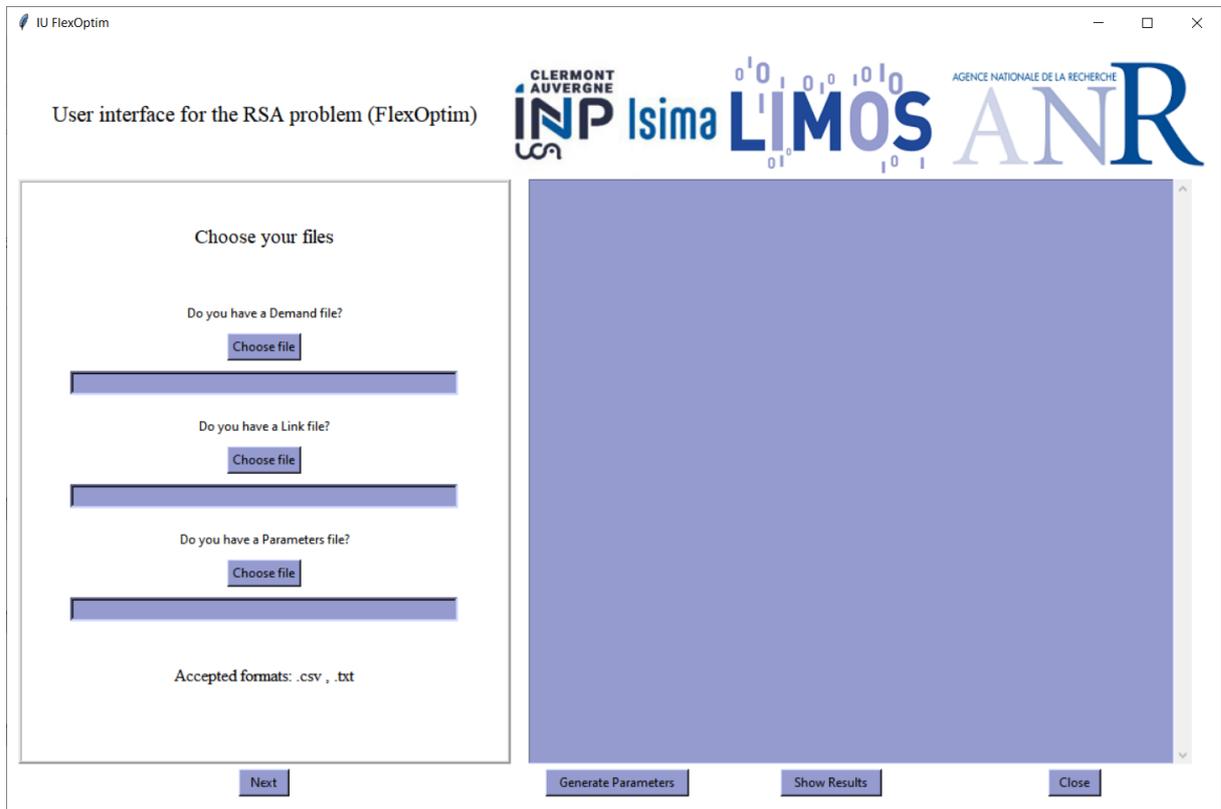


Elaborada por [Grayson \(2000\)](#)

#### 4.2.2 Funcionamento da Interface de Usuário

O objetivo do *design* da interface é tornar a interação digital o mais simples, fluida, intuitiva e eficiente possível. Assim, ela deve antecipar necessidades e garantir facilidade de acesso, entendimento e uso, maximizando a experiência do usuário.

As Figuras 6 e 7 ilustram a interface de usuário criada.

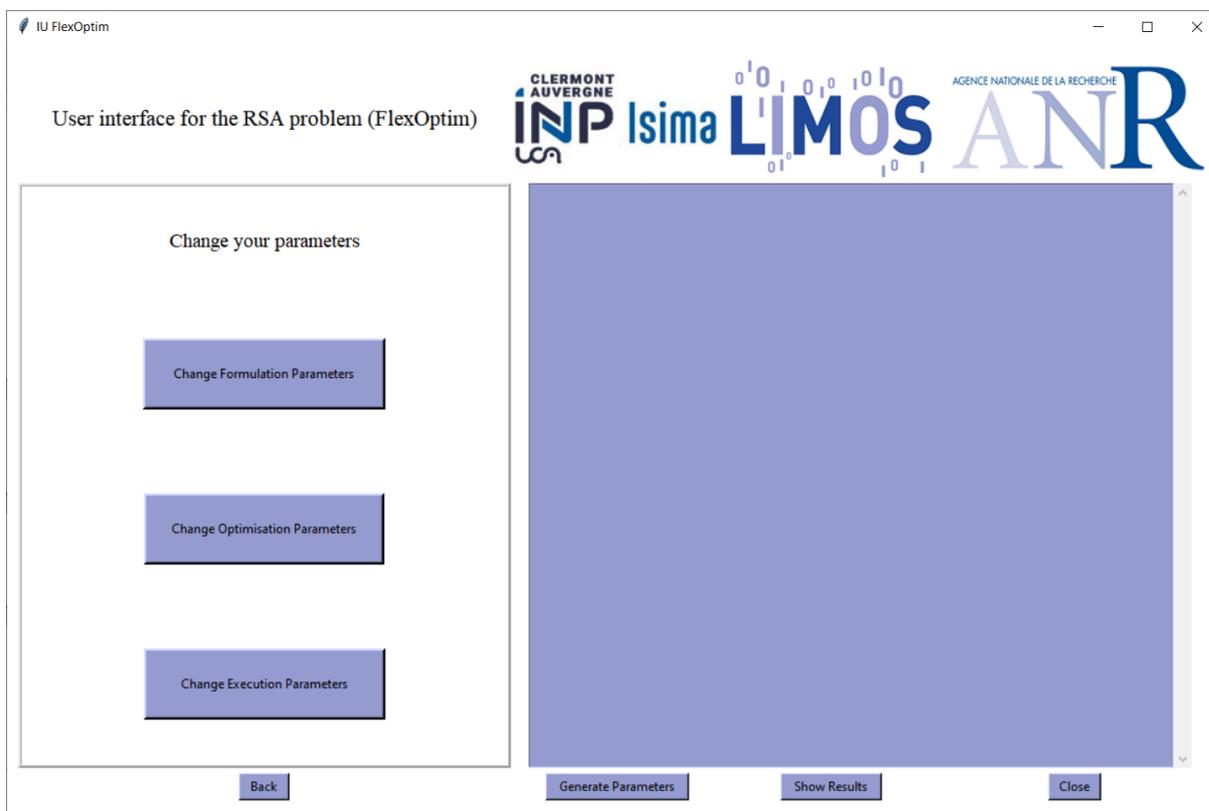
Figura 6 – Interface de usuário para o *RSA Problem (FlexOptim)*

Elaborada pela autora

A construção da interface gráfica promove a instanciação do problema *RSA* e funciona da seguinte forma:

- Quando clica-se em *Next*, a janela *Choose your files* incorporada torna-se *Change your parameters* e o botão *Next* torna-se *Back*. O inverso também acontece ao clicar em *Back*.
- Ao clicar em *Generate Parameters*, a interface criará o arquivo *onlineParameters.txt* de acordo com a combinação de parâmetros escolhida.
- Ao clicar em *Show results*, ele executa o comando no terminal: `./exec onlineParameters` e imprime os resultados.
- Na janela integrada *Choose your files*, ao selecionar *Choose file*, pode-se escolher o arquivo desejado. O caminho do arquivo será exibido na barra abaixo para cada tipo.
- Na janela *Choose your parameters*, ao clicar em *Change formulations Parameters*, *Change optimizations Parameters* ou *Change Execution Parameters*, pode-se alterar os parâmetros como deseja em uma nova janela que se abrirá.
- Ao clicar em *Close*, a interface é fechada.

Figura 7 – "Escolha de parâmetros": interface de usuário para o *RSA Problem (FlexOptim)*



Elaborada pela autora

Figura 8 – "Escolha de parâmetros": parâmetros de formulação

Formulation Parameters

n. Demands at Once :  
States how many demands are treated in one optimization step. Choose ▾

Formulation :  
Choice of the formulation to be used. 0 for Flow Formulation. 1 for Edge-Node. Choose ▾

User Cuts :  
Boolean parameter indicating whether user cuts should be applied. Choose ▾

Objective :  
Which objective to be optimized. 1 for minimize last slot used per demand. 1p for minimize last slot used per edge. 2 for minimize number of hops. 2p for minimize used slices. 4 minimize path length. 8 minimize last slot used overall. Choose ▾

Allow Blocking :  
Boolean parameter indicating whether the optimization can continue after a demand is blocked. Choose ▾

Hop Penalty :  
Demand's reach penalty applied on each hop. Choose ▾

Partition Policy :  
States whether the spectrum is partitioned. 0 for no partition at all. 1 for pushing bigger demands to the right and smaller to the left. Choose ▾

Partition Load :  
States the highest load that should be on the left partition. Choose ▾

Partition Slice :  
If partition policy = 2, states where the spectrum is divided. Choose ▾

Done

Elaborada pela autora

A alteração das configurações é feita através de menus de opções com texto explicando o funcionamento de cada parâmetro, conforme a Figura 8, como exemplo de alteração dos parâmetros da formulação. Ao clicar em *Done*, a janela fecha, as alterações feitas nos parâmetros serão aplicadas na próxima vez que você clicar em *Generate Parameters*.

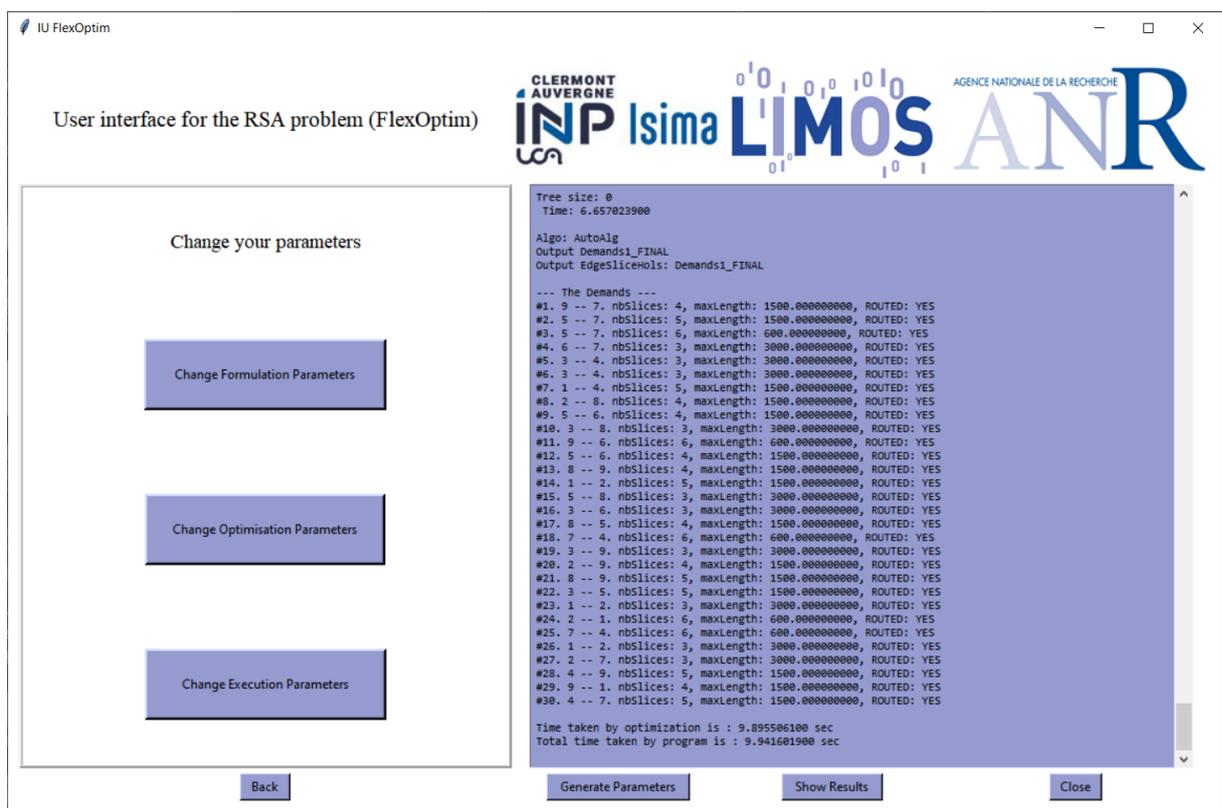
Estas funcionalidades são possíveis graças à biblioteca *os* do *Python* que permite executar comandos no terminal e possibilita a edição de arquivos de texto no *Python* (sem necessidade de importar uma biblioteca adicional). Desta forma, basta manipular as cadeias que serão utilizadas para criar o arquivo de parâmetros (*onlineParameters.txt*) de acordo com a combinação que o usuário fornecer antes de clicar no botão *Show results*.

### 4.2.3 Introdução de uma nova desigualdade e proposta de melhoria do modelo

Para justificar a necessidade de introduzir novas desigualdades na formulação atual do problema *RSA*, é importante considerar suas soluções, que podem ser visualizadas e analisadas por meio da interface de usuário previamente desenvolvida.

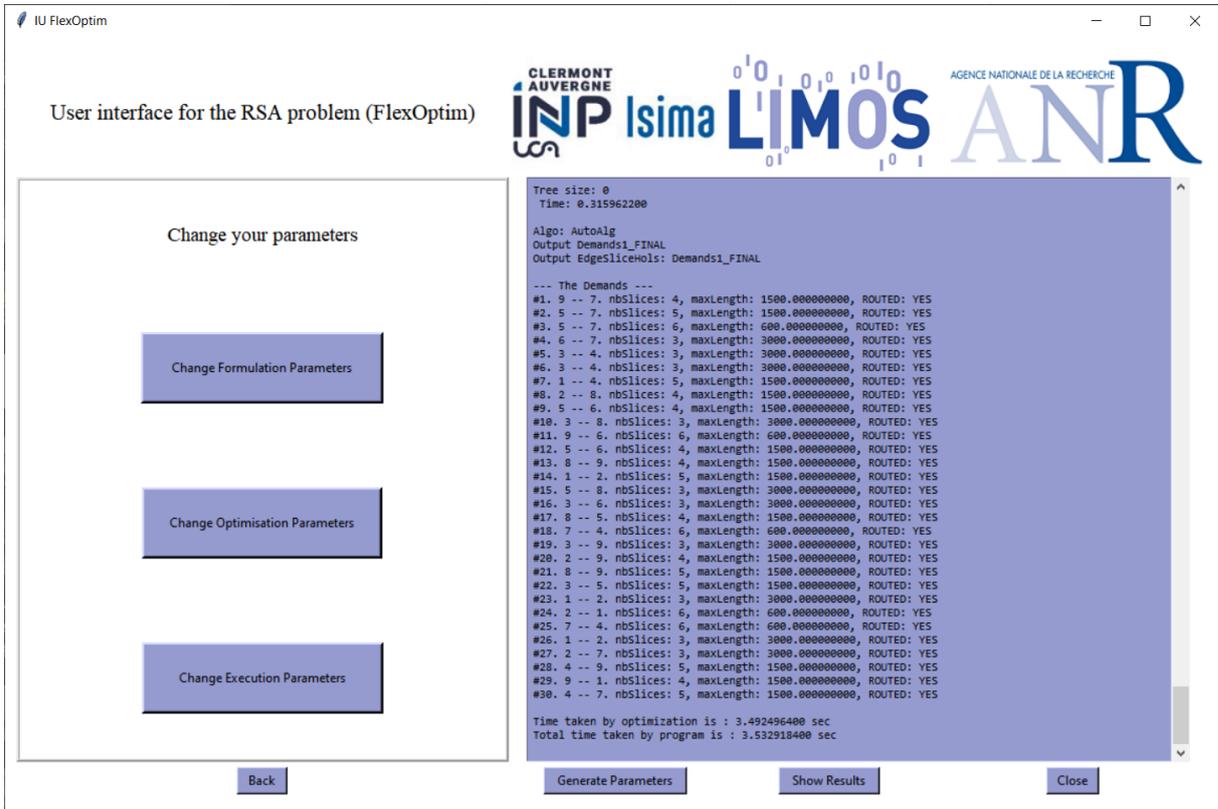
Com o auxílio dessa interface, é possível executar vários tipos de instâncias por meio de linhas de código que funcionam em segundo plano. A Figura 9 mostra o resultado da execução de uma instância com a função objetivo 1, que é minimizar o último *slot* utilizado por requisição. A Figura 10 apresenta o resultado da execução de uma instância com a função objetivo 2, que é minimizar o último *slot* utilizado por aresta. Além disso, a Figura 11 mostra o resultado da execução de uma instância com a função objetivo 2p, que é minimizar o último *slot* utilizado por requisição. Já a Figura 12 exibe o resultado da execução de uma instância com o objetivo 4, que é minimizar o tamanho do caminho. Por fim, a Figura 13 apresenta o resultado da execução de uma instância com a função objetivo 8, que é minimizar o último *slot* usado globalmente.

Figura 9 – Executando uma instância, selecionando a função objetivo 1 na interface



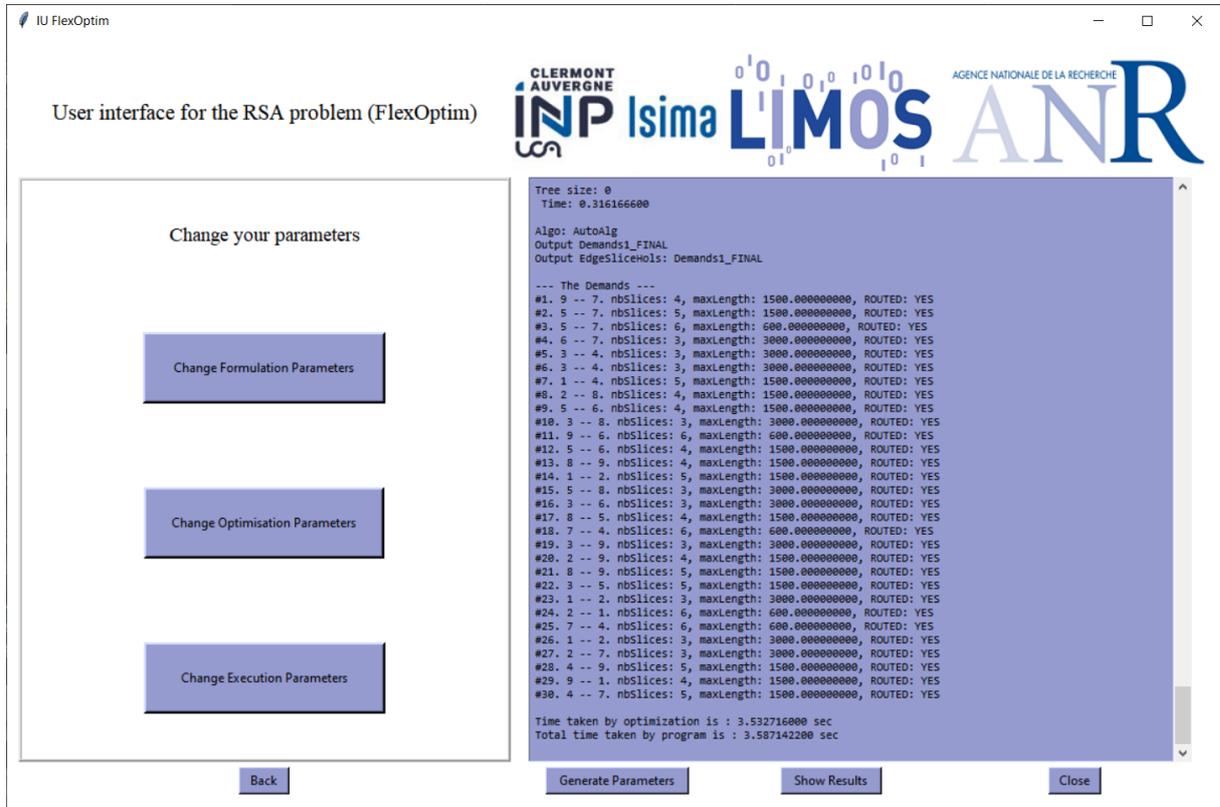
Elaborada pela autora

Figura 10 – Executando uma instância, selecionando a função objetivo 2 na interface



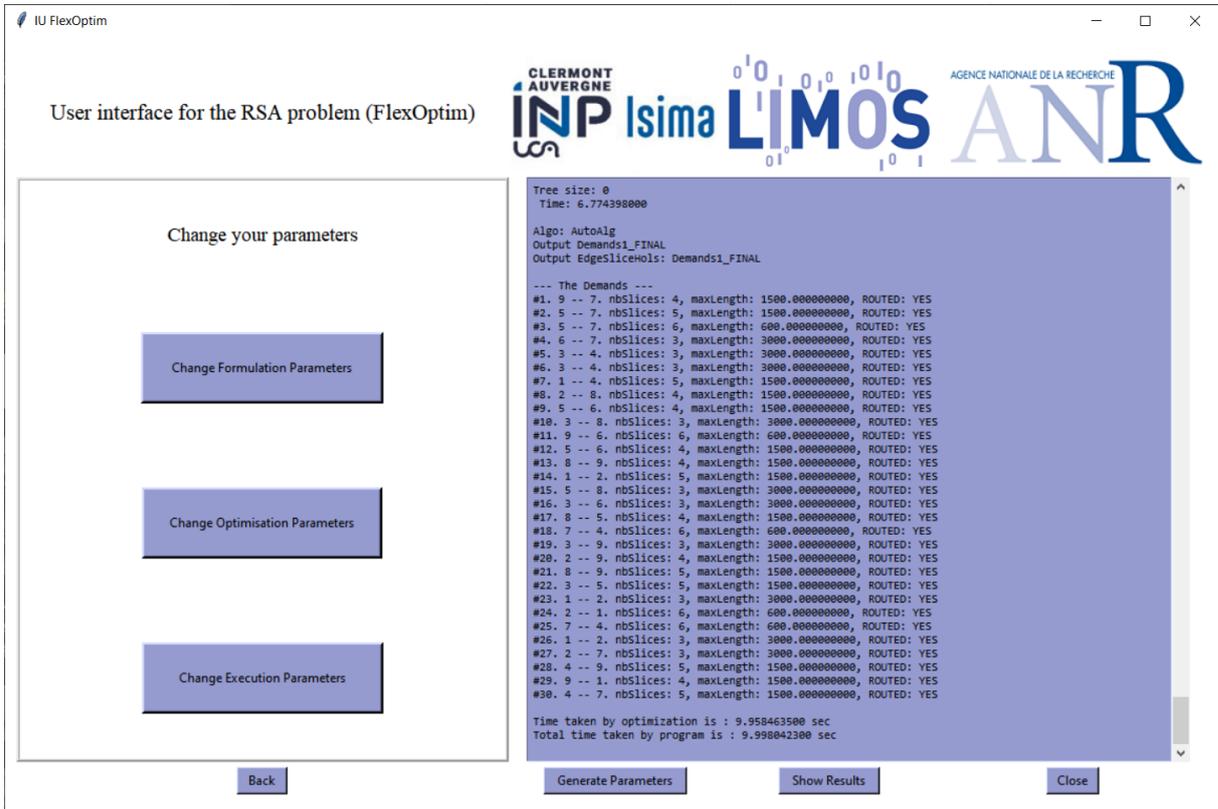
Elaborada pela autora

Figura 11 – Executando uma instância, selecionando a função objetivo 2p na interface



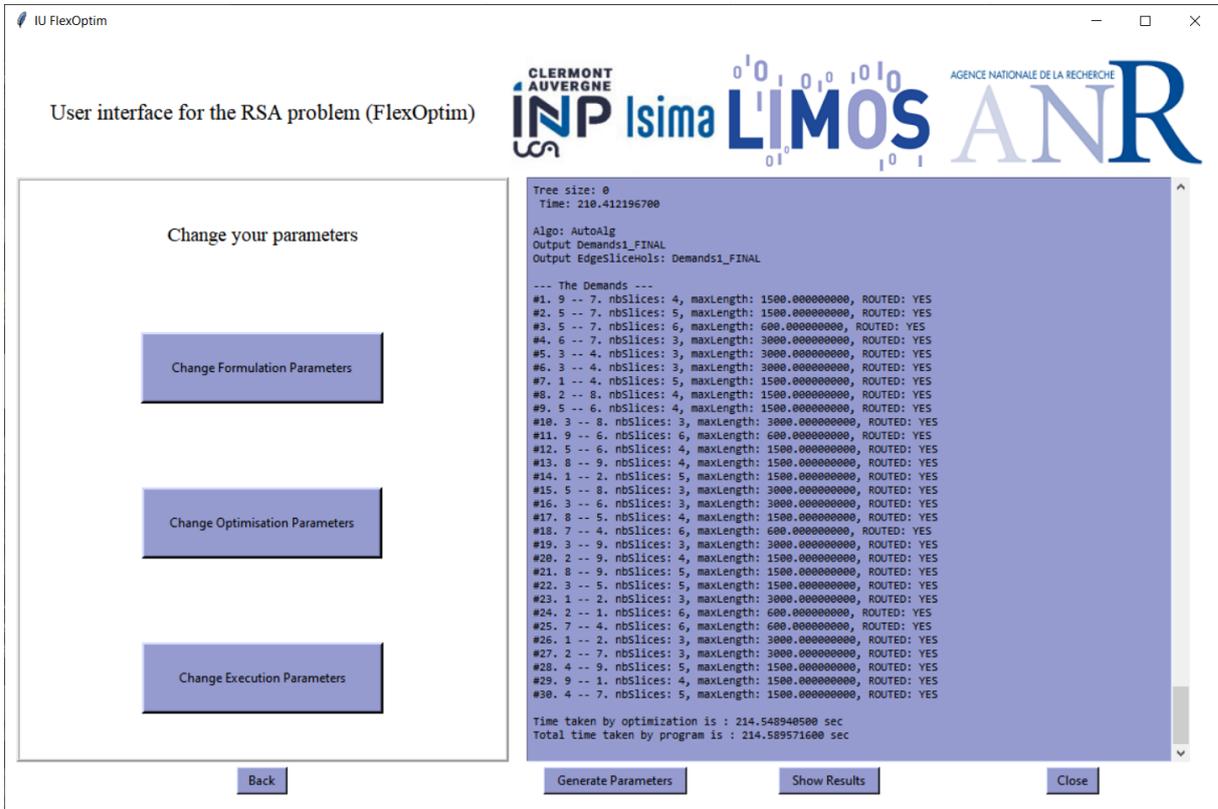
Elaborada pela autora

Figura 12 – Executando uma instância, selecionando a função objetivo 4 na interface



Elaborada pela autora

Figura 13 – Executando uma instância, selecionando a função objetivo 8 na interface



Elaborada pela autora

Ao analisar todas as execuções realizadas, observa-se que o tempo necessário para atingir a solução é consideravelmente longo, especialmente para a última execução. Diante desse cenário, é possível presumir que a formulação em questão pode ser aprimorada por meio da introdução de uma nova desigualdade, a qual pode contribuir significativamente para a redução do tempo gasto. Isso se deve ao fato de que, com a adição dessa nova desigualdade, o método *Branch and Cut* pode realizar menos cortes, o que resulta em uma solução mais eficiente e em um tempo menor de execução. Dessa forma, é possível concluir que a utilização de uma formulação mais otimizada pode contribuir para aprimorar a performance do método utilizado na resolução do problema em questão.

Para introduzir uma nova desigualdade e assim fortalecer o modelo, baseou-se principalmente em estudos de (BRUNETTA; CONFORTI; FISCHETTI, 2000). Neste artigo, o problema consiste em determinar a quantidade de cada bem que deve ser transportada entre os nós da rede, de modo que as restrições de capacidade da rede e a demanda de cada bem sejam atendidas, enquanto se minimiza o custo total de transporte. Para isso, um algoritmo de ramificação e corte é proposto para a solução exata do problema, que é considerado NP-difícil, assim como o RSA, e encontra importantes aplicações em transporte e telecomunicações.

No modelo em questão, a rede de fluxo é representada por um grafo direcionado ponderado, onde cada nó representa um local de origem ou destino de um bem e cada aresta representa um caminho que pode ser percorrido pelo bem. Cada bem é representado por uma *commodity*, e a demanda de cada uma delas é especificada como uma quantidade inteira.

Após análises, foi possível observar que, embora os modelos matemáticos de Brunetta, Conforti e Fischetti (2000) e Colares, Kerivin e Wagler (2021) abordem problemas diferentes em redes, ambos tem semelhanças em relação à abordagem matemática e aos métodos de resolução. Os dois modelos contam com uma formulação inicial, com um grafo conexo e não orientado ( $G = (V, E)$ ), que posteriormente foi reforçada, passando a ser orientada.

Além disso, em ambos os modelos, a abordagem matemática é baseada em programação linear inteira, onde as variáveis de decisão são inteiras e as restrições do problema são expressas como equações ou desigualdades lineares. Ademais, ambos os modelos buscam minimizar uma função objetivo, que representa o custo total de uma solução viável. Ambos os modelos também utilizam a técnica de relaxação linear, que consiste em relaxar as restrições do modelo para um problema de programação linear contínuo, que é mais fácil de resolver.

No problema RSA, a variável  $f_s^k$  indica se a requisição  $k$  é roteada pelo arco  $a$  e se  $s$  é o último *slot* no canal. Em Brunetta, Conforti e Fischetti (2000) a variável  $x^i$  é semelhante, refere-se a uma variável inteira que descreve se a demanda  $i$  passou pela aresta  $e$  ou não, onde  $i = 1, \dots, k$ .

Brunetta, Conforti e Fischetti (2000) então adiciona em sua formulação original uma nova grande classe de restrições válidas, as desigualdades denominadas *multi-handle comb*, esta classe de desigualdades é considerada pelos autores como uma classe de restrições nova, em que somente é conhecido na literatura o caso especial polinomialmente solúvel em que  $k = 1$ ,  $c = 1$  e  $e \in E$ . Elas possuem uma forma complexa, como mostrado a seguir:

$$\sum_{i=1}^k [x^i(\delta(W_i) \cap T) - x^i(\delta(W_i)/T)] \leq c(T) - 1 \quad (4.1)$$

onde  $k$  é o número de commodities distintas, nesse contexto,  $x^i$  é um vetor de fluxo inteiro associado à commodity  $i$ ,  $\delta(W_i)$  é o conjunto de arcos que saem do conjunto de nós  $W_i$ ,  $T$  é um conjunto de arcos da rede de transporte,  $c(T)$  é a capacidade total dos arcos em  $T$ .

A desigualdade estabelece que a soma dos fluxos de commodities que atravessam os arcos em  $T$  e que saem do conjunto de nós  $W_i$ , menos a soma dos fluxos que atravessam os arcos em  $T$  e que saem do conjunto de nós  $W_i$  e que não estão em  $T$ , deve ser menor ou igual à capacidade total dos arcos em  $T$ , menos 1. Isso implica que o fluxo total de commodities através de  $T$  não pode exceder a capacidade disponível, reduzida em uma unidade, de forma a evitar congestionamentos e manter o equilíbrio na rede de transporte.

De acordo com Brunetta, Conforti e Fischetti (2000) eles conseguiram melhorar o relaxação linear do modelo base considerado em seu estudo, a restrição *mult-handle comb* tornou o modelo mais eficiente, pois permitiu que ele descartasse soluções inválidas mais rapidamente, restringindo o espaço de busca do modelo, e conseqüentemente, reduziu o tempo necessário para encontrar a solução ótima.

Portanto, é possível que a classe de desigualdades *multi-handle comb* possa ser adaptada e aplicada ao modelo de Colares, Kerivin e Wagler (2021), para lidar com as restrições de capacidade da rede, desde que seja relevante para o problema abordado e ajude a melhorar a precisão e eficiência da solução do modelo. No entanto, essa adaptação exigiria uma análise cuidadosa das restrições do modelo de Colares, Kerivin e Wagler (2021) para garantir que ela possa ser aplicada corretamente.

## 5 Considerações Finais

Quando iniciou-se o trabalho de pesquisa constatou-se que existia a necessidade da criação de uma interface de usuário para o problema *RSA - EXT*, a fim de facilitar e aprimorar a interação do usuário, melhorando a sua experiência, para que os parâmetros fossem alterados e as soluções pudessem ser mostradas de uma forma melhor, exigindo um nível mínimo de conhecimento técnico possível. Ademais, notou-se a possibilidade de introdução de uma nova classe de desigualdades na formulação atual do modelo *RSA - EXT*, pressupondo que com a introdução de novas desigualdades válidas seria possível encontrar soluções de forma mais rápida e eficiente, considerando que a formulação atual leva muito tempo para encontrar boas soluções.

Sendo assim, é possível afirmar que o objetivo geral deste trabalho foi atingido, considerando que foi apresentado o *design* de uma interface de usuário e que esta possibilitou a análise das soluções derivadas da relaxação linear, fazendo com que se verificasse a oportunidade de introdução de uma nova classe de desigualdades na formulação de um modelo matemático já existente, baseando-se, sobretudo, nos estudos de [Colares, Kerivin e Wagler \(2021\)](#) e [Brunetta, Conforti e Fischetti \(2000\)](#).

Considerando os objetivos específicos desta pesquisa, em um primeiro momento, estudou-se o problema através de revisões bibliográficas e compilação do código existente, para isso foram utilizados vários pacotes e bibliotecas, conforme descrito no decorrer deste documento, o que foi muito trabalhoso e demorado, estendendo-se por alguns meses, devido a erros de referência no código fonte.

Realizou-se o *design* da interface de usuário, usando a linguagem de programação *Python* e a biblioteca *Tkinter* como ferramentas principais. Dessa forma, após sua elaboração foi possível visualizar as soluções resultantes da relaxação linear e através desta análise, mostrar a necessidade de se introduzir novas desigualdades na formulação atual, para fortalecer o modelo e o método *Branch and Cut*.

A proposta de melhoria ao modelo abordou o âmbito matemático da pesquisa operacional, visando propor a introdução de uma nova classe de desigualdades, denominada *multi-handle comb*, com base em estudos relevantes presentes na literatura. Levando em conta, a semelhança entre as variáveis, objetivos e métodos de solução dos modelos de [Colares, Kerivin e Wagler \(2021\)](#) e [Brunetta, Conforti e Fischetti \(2000\)](#), torna-se viável a tentativa de melhorar o tempo de execução da atual formulação do *RSA - EXT*.

Como perspectivas para a elaboração de trabalhos futuros, sugere-se a implementação da nova classe de desigualdades para que se possa verificar se elas são de fato capazes de produzir resultados satisfatórios e melhores do que a formulação atual e, assim, fortalecer a formulação existente.

# Referências

- ABREU, V.; GONZÁLEZ, P. H.; RIBEIRO, G. M.; ORRICO, R. D.; GOMES, H.; PERIM, L. R. Técnicas heurísticas e um branch-and-cut para o problema de localização de sensores de tráfego com limitações quanto ao número de sensores e de faixas observadas. **Anais do L Simpósio Brasileiro de Pesquisa Operacional, SBPO**, 2018.
- AFOLABI, A. O.; SALAWU, A. S.; OLUGBARA, O. O. Development of a library management system using python tkinter. **International Journal of Advanced Computer Science and Applications**, v. 11, n. 1, p. 491–496, 2020.
- AGARWAL, Y. K.; ANEJA, Y. P. Fixed charge multicommodity network design using p-partition facets. **European Journal of Operational Research**, Elsevier, v. 258, n. 1, p. 124–135, 2017.
- BERTRAND, J. W. M.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. **International journal of operations & production management**, MCB UP Ltd, v. 22, n. 2, p. 241–264, 2002.
- BIENSTOCK, D.; MURATORE, G. Strong inequalities for capacitated survivable network design problems. **Mathematical programming**, Springer, v. 89, n. 1, p. 127–147, 2000.
- BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. [S.l.]: Novatec Editora, 2014.
- BRUNETTA, L.; CONFORTI, M.; FISCHETTI, M. A polyhedral approach to an integer multicommodity flow problem. **Discrete Applied Mathematics**, Elsevier, v. 101, n. 1-3, p. 13–36, 2000.
- CHRISTODOULOPOULOS, K.; TOMKOS, I.; VARVARIGOS, E. A. Elastic bandwidth allocation in flexible ofdm-based optical networks. **Journal of Lightwave Technology**, IEEE, v. 29, n. 9, p. 1354–1366, 2011.
- COLARES, R.; KERIVIN, H.; WAGLER, A. An extended formulation for the Constraint Routing and Spectrum Assignment Problem in Elastic Optical Networks \*. *Travail en cours ou préimpression*. 2021. Disponível em: <<https://hal.uca.fr/hal-03156189>>.
- COMO converter arquivos de texto no Linux. 2023. <<https://linuxize.com/post/how-to-convert-text-files-between-dos-and-unix-format/>>. Acesso em: 21 fev. 2023.
- CUNHA, J. O.; SIMONETTI, L.; LUCENA, A. Um algoritmo branch-and-cut para o problema da mochila quadrática 0-1. 2014.
- ECHER, I. C. A revisão de literatura na construção do trabalho científico. **Revista Gaúcha de Enfermagem**, Porto Alegre, v. 22, n. 2, p. 5–20, jul. 2001. ISSN 0102-6933. Disponível em: <<https://seer.ufrgs.br/RevistaGauchadeEnfermagem/article/view/4365>>. Acesso em: 11 nov. 2019.
- FACHIN, O. **Fundamentos de metodologias**. [S.l.]: Saraiva Educação SA, 2001.
- GIL, A. C. Como elaborar projetos de pesquisa. são paulo: Atlas, 2006. gil, antônio carlos. **Como elaborar projetos de pesquisa**, v. 5, 2010.

GONZÁLEZ, P. H.; MARACANÃ, A.; MARACANÃ, R. d. J.-R.; CLIMACO, G.; SIMONETTI, L.; VIEIRA, B. S.; RIBEIRO, G. M.; JÚNIOR, N. F. R. C.; ABRAMIDES, C. A.; NUNES, A. de O. Um algoritmo branch-and-cut para o problema de localização ótima de contadores de tráfego em redes de transporte. **XLIX Simpósio Brasileiro de Pesquisa Operacional**. <https://doi.org/10.13140>, 2017.

GRAYSON, J. E. **Python and Tkinter programming**. [S.l.]: Manning Publications Co. Greenwich, 2000.

HADHBI, Y.; KERIVIN, H.; WAGLER, A. A novel integer linear programming model for routing and spectrum assignment in optical networks. In: IEEE. **2019 Federated Conference on Computer Science and Information Systems (FedCSIS)**. [S.l.], 2019. p. 127–134.

JINNO, M.; KOZICKI, B.; TAKARA, H.; WATANABE, A.; SONE, Y.; TANAKA, T.; HIRANO, A. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. **IEEE Communications Magazine**, IEEE, v. 48, n. 8, p. 138–145, 2010.

KIM, Y.; HAN, J. Designing usable user interface with python tkinter. **International Journal of Human-Computer Interaction**, Taylor & Francis, v. 36, n. 14, p. 1341–1348, 2020.

LAKATOS, E. M.; ANDRADE, M. de. Maria. metodologia científica. **Eva Maria Lakatos, Mariana de Andrade Marconi: -5. ed.—4. reimpresso—São Paulo: Atlas**, 2010.

LOPARIC, M. **Uma aplicação do método de branch-and-cut a um problema de roteamento de veículos**. Tese (Doutorado) — Universidade de São Paulo, 1996.

MA, X.; SUN, L. An efficient and user-friendly interface for medical image analysis using python tkinter. **International Journal of Medical Informatics**, Elsevier, v. 152, p. 104490, 2021.

MARTINS, R. A. Abordagem quantitativa e qualitativa. In: MIGUEL, P. A. C. (Ed.). **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. 2. ed. [S.l.]: Elsevier, 2011. cap. 3, p. 45–61. ISBN 853524891-9.

O que é uma máquina virtual. 2023. <<https://www.vmware.com/br/topics/glossary/content/virtual-machine>>. Acesso em: 21 fev. 2023.

RUIZ, J. Á. Metodologia científica. **Guia para eficiência nos estudos**, v. 13, p. 131, 1996.

SÁ, J. V. D. P.; KERIVIN, H. Routing and spectrum assignment problem : An in-deep analysis of formulation and implementation. Rapport de fin de stage. 2021.

SCHWARTZMAN, S. Pesquisa acadêmica, pesquisa básica e pesquisa aplicada em duas comunidades científicas. **Termos de referência de pesquisa, não publicado, janeiro de**, 1979.

STALLMAN, R. M.; MCGRATH, R. Gnu make-a program for directing recompilation. **The Free Software Foundation**, Citeseer, 1991.

TURRIONI, J. B.; MELLO, C. H. P. **Metodologia de Pesquisa em Engenharia de Produção: Estratégias, métodos e técnicas para condução de pesquisas quantitativas e qualitativas**. Itajubá, MG, 2012. Disponível em: <[http://www.marco.eng.br/adm-organizacao-I/Apostila\\_Metodologia\\_Completa\\_2012\\_%20UNIFEI.pdf](http://www.marco.eng.br/adm-organizacao-I/Apostila_Metodologia_Completa_2012_%20UNIFEI.pdf)>. Acesso em: 11 nov. 2019.

---

WINDOWS Subsystem for Linux. 2023. <<https://docs.microsoft.com/pt-br/windows/wsl/about>>. Acesso em: 21 fev. 2023.

WOLSEY, L. A.; NEMHAUSER, G. L. **Integer and combinatorial optimization**. [S.l.]: John Wiley & Sons, 1999. v. 55.