



Trabalho de Conclusão de Curso

# **Otimização na seleção de alvos em uma interceptação telefônica**

Larissa Artemis Luna Monteiro  
lalm@ic.ufal.br

Orientador:  
Rian Gabriel Santos Pinheiro

Maceió, 22 de fevereiro de 2022

Larissa Artemis Luna Monteiro

## **Otimização na seleção de alvos em uma interceptação telefônica**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Rian Gabriel Santos Pinheiro

Maceió, 22 de fevereiro de 2022

**Catálogo na fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**  
Bibliotecária: Livia Silva dos Santos – CRB-4 – 1670

M772o Monteiro, Larissa Artemis Luna.  
Otimização na seleção de alvos em uma interceptação telefônica / Larissa Artemis  
Luna Monteiro. – 2022.  
38 f.:il.

Orientador: Rian Gabriel Santos Pinheiro.  
Monografia (Trabalho de Conclusão de Curso em Engenharia da Computação) –  
Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.

Bibliografia: f. 37-38

1. Programa linear interna (PLI). 2. Interceptação telefônica. 3. Criminofísica.  
4. Modelos de redes. 5. Método simplex. 6. Cobertura de vértices. I. Título.

CDU: 004.343.9

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação do Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.



---

Rian Gabriel Santos Pinheiro - Orientador  
Instituto de Computação  
Universidade Federal de Alagoas



---

Bruno Costa e Silva Nogueira - Examinador  
Instituto de Computação  
Universidade Federal de Alagoas

---

André Luiz Lins de Aquino - Examinador  
Instituto de Computação  
Universidade Federal de Alagoas

# Agradecimentos

Agradeço imensamente aos meus pais, Walterlan e Christine, bem como aos amigos mais próximos, por todo o apoio e carinho.

Ao Ariques, por ler este trabalho tanta vezes quanto eu, me motivando e me ouvindo sempre.

Agradeço demais ao Raumário, pelo líder que é e por incentivar o meu interesse nesta área, levando ao desenvolvimento deste trabalho.

Agradeço também ao professor Rian, meu orientador, por todos os conhecimentos e por conduzir, de maneira excelente, o que era apenas uma ideia até a sua consolidação.

Aos demais professores do curso, pelos conhecimentos transmitidos dentro e fora da sala de aula.

*Se eu vi mais longe, foi por estar sobre ombros de gigantes.*

– Isaac Newton

# Resumo

Neste trabalho foi proposta uma forma de modelar os relacionamentos entre membros de uma organização criminosa (ORCRIM) que possuem autorização judicial para quebra de sigilo telefônico e dados cadastrais. A motivação para tal proposta veio durante a Operação Flash-back I, na qual foram percebidos diversas chamadas duplicadas provocadas pela forte interação entre os alvos interceptados, o que leva ao desperdício de recursos humanos (tempo) e computacionais (como o armazenamento das chamadas no servidor, a alocação do canal que desvia a chamada telefônica para agência, etc). A fim de minimizar tais efeitos, os relacionamentos entre os indivíduos monitorados foram representados usando grafos a partir dos extratos reversos fornecidos pelas operadoras de telefonia. Na modelagem, os vértices dos grafos representam os interlocutores e as arestas representam as chamadas efetuadas. Cada aresta contém um peso que representa o total de horas das chamadas dos indivíduos que se comunicaram. A partir do grafo de relacionamento, foi utilizada uma variação do problema de cobertura de vértices ao grafo para obter a sugestão de um conjunto de indivíduos a serem monitorados no ciclo de interceptação seguinte. Esta variante, conhecida como  $k$ -cobertura de vértice máxima, tem como entrada um grafo ponderado e um inteiro  $k$ , o objetivo do problema é encontrar a  $k$ -cobertura (subconjunto com  $k$  vértices) que maximize a soma dos pesos das arestas. Assim, considerando que este problema é NP-completo, não é conhecido um algoritmo que resolva este problema em tempo polinomial, porém, há estratégias que o fazem em tempo viável. Neste trabalho, foi projetada uma solução a partir da formulação matemática de programação linear inteira. Com a solução proposta, foi possível encontrar soluções ótimas em instâncias reais e sintéticas (até 50 vezes maior que as reais) com um baixo custo computacional. Portanto, de maneira sucinta, as contribuições deste trabalho incluem (i) um modelo matemático para a seleção de alvos a serem monitorados, (ii) uma forma de quantificar numericamente quanto da rede formada pela organização criminosa e seus interlocutores está sendo acompanhada e (iii) um algoritmo de que maximize o peso da  $k$ -cobertura de vértices do grafo formado pelos indivíduos monitorados e seus interlocutores, cujo objetivo é sugerir indivíduos a serem monitorados no ciclo de interceptação seguinte.

**Palavras-chave:** interceptação telefônica, criminofísica, organizações criminosas, grafos, cobertura de vértices, programação linear inteira.

# Abstract

In this study, a way to model the relations between members of a criminal organization who have judicial authorization to break telephone secrecy and registration data was proposed. The motivation for this proposal came during Operation Flashback I, in which several duplicate calls were noticed caused by the strong interaction between the intercepted targets, which leads to a waste of human (time) and computational resources (such as the storage of calls on the server, the allocation of the channel that diverts the telephone call to the branch, etc.). For the purpose of minimize such effects, the relation between monitored individuals were represented using graphs from reverse extracts provided by telephone operators. In the modeling, the vertices of the graphs represent the interlocutors and the edges represent the calls made. Each edge contains a weight that represents the total hours of calls from individuals who communicated. From the relation graph, a variation of the problem of coverage of vertices to the graph was used to obtain the suggestion of a set of individuals to be monitored in the following interception cycle. This variant, known as  $k$ -maximum vertex coverage, takes as input a weighted graph and an integer  $k$ , the objective of the problem is to find the  $k$ -coverage (subset with  $k$  vertices) that maximizes the sum of the edge weights. Therefore, considering that this problem is NP-complete, an algorithm that solves this problem in polynomial time is not known, however, there are strategies that do it in viable time. In this work, a solution was designed based on the mathematical formulation of integer linear programming. With the proposed solution, it was possible to find optimal solutions in real and synthetic instances (up to 50 times larger than the real ones) with a low computational cost. Accordingly, succinctly, the contributions of this work include (i) a mathematical model for the selection of targets to be monitored, (ii) a way to numerically quantify how much of the network formed by the criminal organization and its interlocutors is being monitored and (iii) ) an algorithm that maximizes the weight of the  $k$ -coverage of vertices of the graph formed by the monitored individuals and their interlocutors, whose objective is to suggest individuals to be monitored in the following interception cycle.

**Keywords:** telephone interception, criminophysics, criminal organizations, graphs, vertex cover, integer linear programming.

# Sumário

Lista de Figuras . . . . .	v
Lista de Tabelas . . . . .	vi
<b>1 Introdução</b>	<b>1</b>
1.1 Criminofísica . . . . .	1
1.2 Interceptação telefônica e intervenções policiais . . . . .	2
1.2.1 Interceptação telefônica . . . . .	2
1.2.2 Organizações criminosas . . . . .	4
1.2.3 Operação Flashback I . . . . .	5
1.3 Motivação e objetivos . . . . .	6
1.4 Trabalhos relacionados . . . . .	8
1.5 Estrutura do texto . . . . .	8
<b>2 Fundamentação Teórica</b>	<b>9</b>
2.1 Teoria dos grafos . . . . .	9
2.1.1 Aplicações de grafos . . . . .	11
2.1.2 Problema de cobertura de vértices . . . . .	12
2.1.3 Modelos de redes complexas . . . . .	13
2.2 Programação linear . . . . .	14
2.2.1 Método Simplex . . . . .	15
2.2.2 Programação linear inteira - PLI . . . . .	16
<b>3 Proposta</b>	<b>19</b>
3.1 Problemática . . . . .	19
3.2 Análise dos dados e modelagem do grafo . . . . .	20
3.3 Formulação Matemática . . . . .	25
<b>4 Resultados e Discussões</b>	<b>28</b>
4.1 Ferramentas . . . . .	28
4.2 Instância real . . . . .	28
4.3 Instâncias simuladas . . . . .	30
4.3.1 Instâncias com o mesmo tamanho (n = 2957) . . . . .	30
4.3.2 Instâncias 10x maior (n = 29570) . . . . .	32
4.3.3 Instâncias 50x maior (n = 147850) . . . . .	34
<b>5 Considerações Finais</b>	<b>36</b>
<b>Referências bibliográficas</b>	<b>37</b>

# Lista de Figuras

1.1	Fluxograma do ciclo de interceptação telefônica . . . . .	4
1.2	Símbolos do Primeiro Comando da Capital - PCC . . . . .	5
1.3	Grafo da Flashback . . . . .	6
1.4	Fluxograma de renovação de ciclo . . . . .	7
1.5	Novo fluxo de atividades para renovação de ciclo . . . . .	7
2.1	Pontes de Königsberg . . . . .	9
2.2	Exemplo de um grafo . . . . .	10
2.3	Grafo ponderado e direcionado . . . . .	11
2.4	Cobertura de vértices . . . . .	12
2.5	Modelos de redes . . . . .	14
2.6	Exemplo de PPL . . . . .	15
2.7	Método Simplex . . . . .	16
2.8	Comparação de domínios: PLI x PPL . . . . .	17
3.1	Alvos que geram áudios duplicados . . . . .	20
3.2	Exemplo de um ORCRIM modelada no <i>software</i> IBM i2 . . . . .	21
3.3	Operadoras mais usadas no Brasil . . . . .	22
3.4	Grafo criado a partir dos extratos reversos . . . . .	22
3.5	Gráfico de graus do grafo . . . . .	23
3.6	Distribuição de pesos nas arestas . . . . .	24
4.1	Grafo do extrato simplificado . . . . .	29
4.2	Resultados para n=2957 . . . . .	32
4.3	Resultados para n=29570 . . . . .	34
4.4	Gráfico de resultados para n=147850 . . . . .	35

# Lista de Tabelas

3.1	Tabela de erro: distribuição de pesos nas arestas do grafo . . . . .	24
4.1	Instância real: variando $k$ . . . . .	30
4.2	Tabela de resultados para $n=2957$ . . . . .	31
4.3	Tabela de resultados para $n=29570$ . . . . .	33
4.4	Tabela de resultados para $n=147850$ . . . . .	35

# 1

## Introdução

*"A sociedade que separa seus estudiosos de seus guerreiros terá seu pensamento feito por covardes e sua luta por tolos"*

– Tucídides

### 1.1 Criminofísica

A criminofísica (da Cunha, 2021) é a área de conhecimento que surge ao focar as aplicações da matemática, física e ciências de redes para modelar problemas nos contextos criminais. Isto se deve ao fato empírico que, apesar de o comportamento humano individual ser imprevisível, nosso comportamento coletivo apresenta uma enormidade de propriedades em rede que são bastante regulares e matematicamente tratáveis (Barabási, 2010).

Dentro dos fenômenos criminais, em especial, ao considerar as organizações criminosas (ORCRIMs) e suas interações, surge a necessidade de modelar tal sistema e as relações entre as partes que o compõe. O ramo da matemática que estuda as interações entre entes abstratos é chamado de teoria de grafos (Barabási, 2016). Neste ramo, um grafo (que representa o sistema ou rede abstrata) é representado por um conjunto não vazio de vértices (ou nós) ligados por arestas (ou conexões). Considerando as ORCRIMs, os vértices representam os indivíduos e as arestas representam suas interações.

Além da teoria de grafos, tal modelagem sugere também a aplicação dos conceitos das ciências das redes. De forma geral, nas palavras de Barabási (2003), "redes são estruturas complexas que permeiam todo tipo de fenômeno natural que envolva relacionamento entre partes constituintes". Portanto, engloba desde a propagação de uma pandemia como a da COVID-19, passando pelas redes sociais on-line e nossas redes de contato diárias até chegar ao crime organizado (da Cunha, 2021).

Segundo o professor [da Cunha \(2021\)](#), em seu livro *Criminofísica: a ciência das interações criminais*:

A ciência de redes vem provando ser muito útil no estudo de sistemas criminais. Em termos gerais, a ciência de rede é o estudo quantitativo das relações elementos conectados, que juntos formam entidades complexas conhecidas como redes. Matematicamente falando, uma rede é um conjunto de arestas e vértices conectados de acordo com alguma regra, arquitetura ou topologia, como os cientistas da rede gostam de dizer. Caso esses vértices representem criminosos, ter-se-á o que é conhecido como rede criminal. De fato, a regra segundo a qual esses indivíduos se conectam geralmente segue padrões matemáticos bem conhecidos que podem nos dar informações sobre a natureza do fenômeno criminal, revelando estratégias eficientes de enfrentamento e combate.

Um exemplo bem-sucedido de aplicação de redes ao contexto criminal ocorreu em Chicago, nos Estados Unidos, com a criação da estratégia VRS (Group Violence Reduction Strategy, ou Estratégia de Redução de Violência em Grupo), pela Polícia de Chicado<sup>1</sup>. Esta estratégia foi desenvolvida para entender e minimizar os tiroteios envolvendo gangues. Num primeiro momento, foi feita a identificação das gangues mais ativas, modelando-as como vértices, e criada uma aresta caso tenha existido um conflito entre elas. Num segundo momento, estudou-se cada grupo individualmente, para identificar seus alvos mais importantes. A seguir, as intervenções policiais buscaram atuar seguindo a priorização topológica, o que gerou uma redução de 23% nos tiroteios e de 32% na vitimização entre facções.

## 1.2 Intercepção telefônica e intervenções policiais

Como visto, do ponto da criminofísica, uma rede criminal é uma entidade matemática consistindo de criminosos (vértices), conectada por algum tipo de relacionamento (arestas) que pode ser intermediado de diversas maneiras ([da Cunha \(2021\)](#)). Na VRS, tal relacionamento existe caso tenha havido um conflito entre os grupos. Por outro lado, pode-se relacionar alvos através de chamadas telefônicas, dados de inteligência, fluxo de dinheiro, etc.

Considerando que um dos usos da intercepção das chamadas telefônicas está voltado ao acompanhamento sistemático de organizações criminosas no estado de Alagoas, se faz necessário entender todo um arcabouço de legislação específica.

### 1.2.1 Intercepção telefônica

A intercepção telefônica, autorizada na Lei Nº 9.296, de 24 de julho de 1996 [BRASIL \(1996\)](#), consiste na captação das ligações vinculadas a um determinado terminal telefônico, que

<sup>1</sup><https://home.chicagopolice.org/inside-cpd/violence-reduction-strategy-vrs/>

pode ser tanto um número de telefone quanto um IMEI (*International Mobile Equipment Identity* ou Identidade Internacional de Equipamento Móvel, um código global e único de cada celular ou *smartphone*). Tal captação pode ser feita apenas mediante autorização judicial e se houver indícios razoáveis da autoria ou participação em infração penal do usuário daquele terminal. Uma vez autorizado, é emitida uma ordem judicial para as operadoras telefônicas (como a TIM, Oi, Vivo, Claro e Nextel) desviarem as chamadas do terminal telefônico monitorado para um terminal específico, normalmente de uma agência de inteligência ou investigação.

As chamadas desviadas, por sua vez, são administradas por *softwares* como o Guardiã, da Dígitro, que armazena as chamadas em um servidor local da agência (para que possam ser escutadas em momento oportuno, não em tempo real) e fornece uma interface de acesso para os analistas, onde é possível ouvir os áudios como um *player* de música, com opções de pausar, aumentar ou reduzir a velocidade de reprodução, reduzir ruídos de fundo, etc.

O Guardiã, ou *software* similar, é a principal ferramenta do analista que acompanha uma interceptação telefônica, porém, atendidos os devidos requisitos, o analista pode também solicitar às operadoras tanto os dados cadastrais quanto o extrato reverso de ligações (que contém a lista das chamadas realizadas ou recebidas em um certo período de tempo) vinculados àquele terminal monitorado.

A autorização judicial da quebra do sigilo telefônico dura 15 dias e, ao fim deste período, um profissional (chamado de analista) deve sugerir a prorrogação do terminal, se comprovadas atividades ilícitas, ou exclusão, caso contrário. O analista também pode sugerir a inclusão de interlocutores (terminais que ligaram ou receberam chamadas do interceptado), desde que estejam vinculados às atividades ilegais.

Em equipes com vários analistas, é comum realizar-se uma reunião ao fim do período de interceptação para decidir, em conjunto, quais alvos serão prorrogados ou excluídos e se algum novo terminal será incluído no monitoramento.

Por fim, é produzido um relatório contendo a análise da interceptação telefônica, que será avaliado por um juiz. Em seguida, uma nova ordem judicial às operadoras é expedida e o ciclo se repete. Este processo está descrito na Figura 1.1 abaixo.

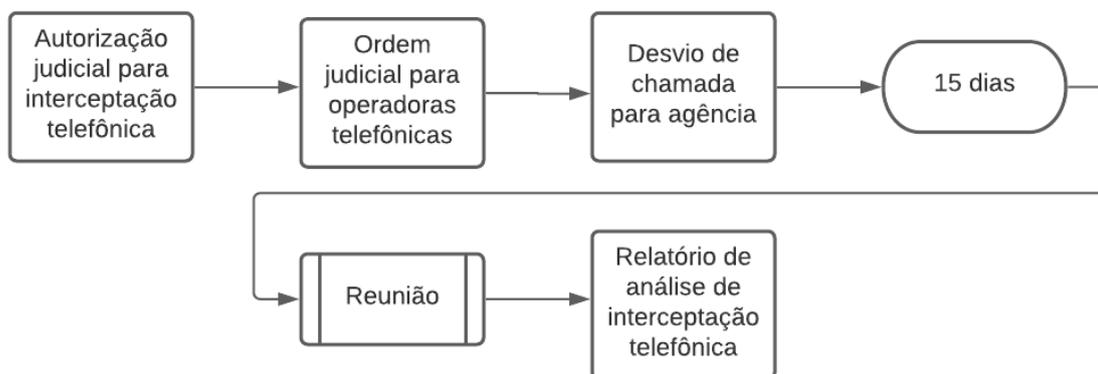


Figura 1.1: Fluxograma do ciclo de interceptação telefônica

## 1.2.2 Organizações criminosas

Conforme a Lei Nº 12.850, de 2 de agosto de 2013 [BRASIL \(2013\)](#), denomina-se organização criminosa ou ORCRIM:

(...) associação de 4 (quatro) ou mais pessoas estruturalmente ordenada e caracterizada pela divisão de tarefas, ainda que informalmente, com objetivo de obter, direta ou indiretamente, vantagem de qualquer natureza, mediante a prática de infrações penais cujas penas máximas sejam superiores a 4 (quatro) anos, ou que sejam de caráter transnacional.

Em Alagoas, as principais organizações criminosas são as facções Primeiro Comando da Capital (PCC) e Comando Vermelho (CV), que estão em constante disputa de poder. Tais facções constituem a chamada criminalidade complexa, que difere dos crimes comuns (executados isoladamente) por possuir características próprias e assemelhar sua estrutura organizacional com redes sociais complexas, nas quais os líderes possuem toda uma importância ideológica.

Dentre as características das facções, podemos citar a auto-declaração de ser uma organização criminosa, a uso de símbolos próprios (conforme Figura 1.2), a presença de estatuto, bandeira, forte questão carcerária, aceitação da hierarquia e existência além de fronteiras físicas.

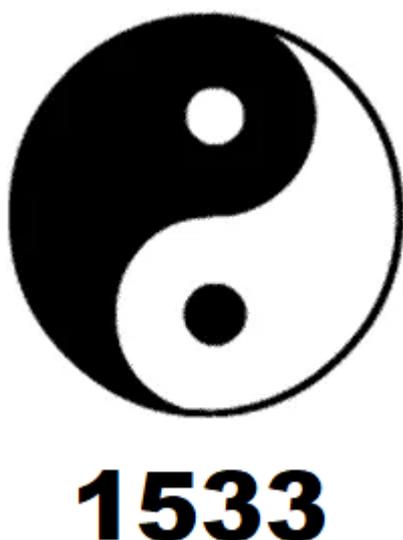


Figura 1.2: Símbolos do Primeiro Comando da Capital - PCC: os números 1533 correspondem a ordem das letras da facção no alfabeto.

### 1.2.3 Operação Flashback I

Nesse contexto de acompanhamento sistemático de facções criminosas no estado de Alagoas, em 2019 foi realizada a Operação Flashback I, que durante sete meses de investigação a respeito dos membros do PCC qualificou mais de 100 membros atuantes da facção em 7 estados, entre eles Alagoas, Mato Grosso do Sul, Minas Gerais, Paraná, Pernambuco, São Paulo, Tocantins e Sergipe. A operação esteve delimitada nos conhecimentos produzidos a partir das ligação telefônicas (ou linhas vermelhas, no vocabulário da facção) e contou com a atuação de diversos órgãos de segurança de Alagoas e demais estados, como a Secretaria de Estado da Segurança Pública (SSP/AL), Ministério da Justiça, Ministério Público do Estado de Alagoas, por meio do Grupo de Atuação Especial de Combate ao Crime Organizado (GAECO) e do Grupo Nacional de Combate às Organizações Criminosas (GNCO), a 17ª Vara Criminal da Capital, Batalhão de Operações Especiais (BOPE) e a Secretaria de Ressocialização e Inserção Social (SERIS).

Ao todo, a Flashback I acompanhou um total de 329 alvos através da interceptação telefônica por 13 analistas, conforme ilustrado na Figura 1.3. Tal esforço resultou em 200 mandados de prisão e busca e apreensão de lideranças de Alagoas e de outros estados que tinham atuação direta com o chamado tribunal do crime do PCC, que consiste num julgamento por parte dos faccionados segundo seu próprio estatuto aos infratores ou membros de facção oposta.

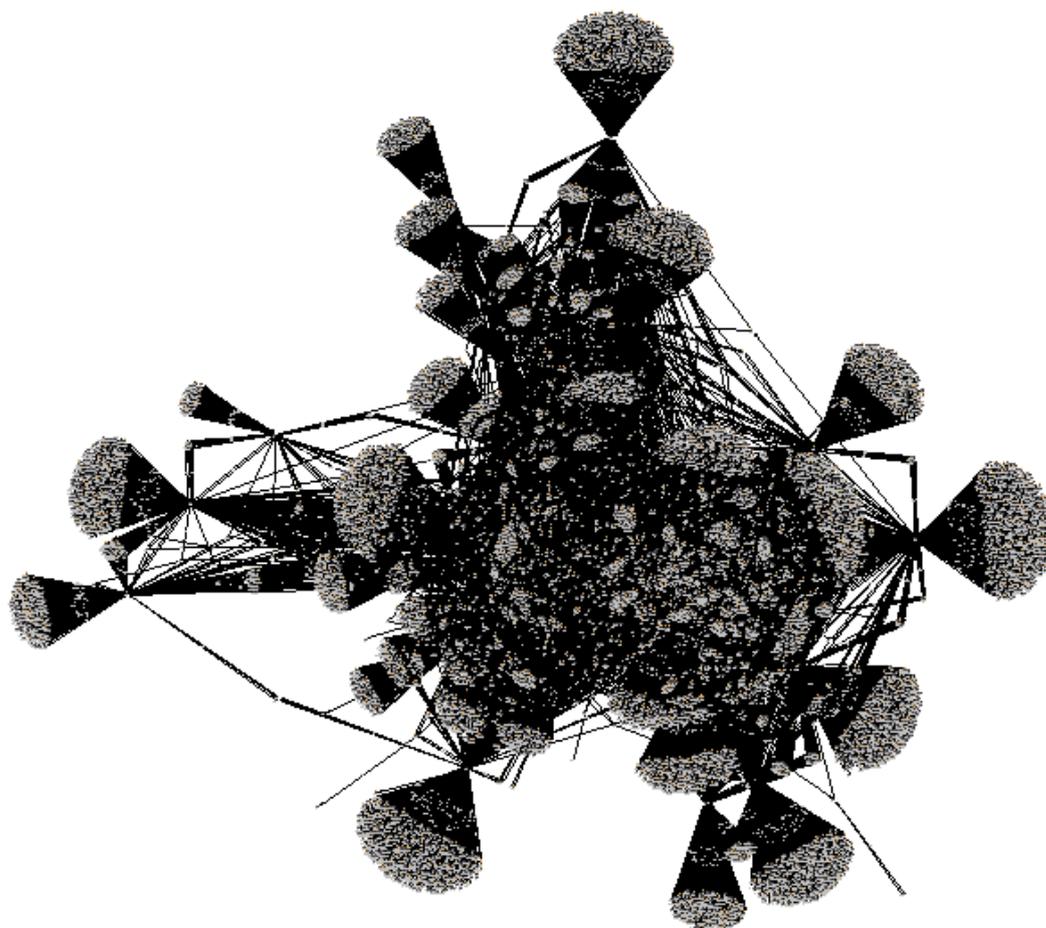


Figura 1.3: Grafo da ORCRIM acompanhada durante a Operação Flashback

### 1.3 Motivação e objetivos

Durante a operação Flashback I, a escolha dos alvos a serem interceptados foi feita da forma tradicional: ao fim de cada período de 15 dias de interceptação (ciclo) realizava-se uma reunião em que os analistas, a partir de sua percepção do que é relevante e de sua experiência, sugeriam as inclusões, prorrogações e exclusões de terminais, conforme a Figura 1.4.

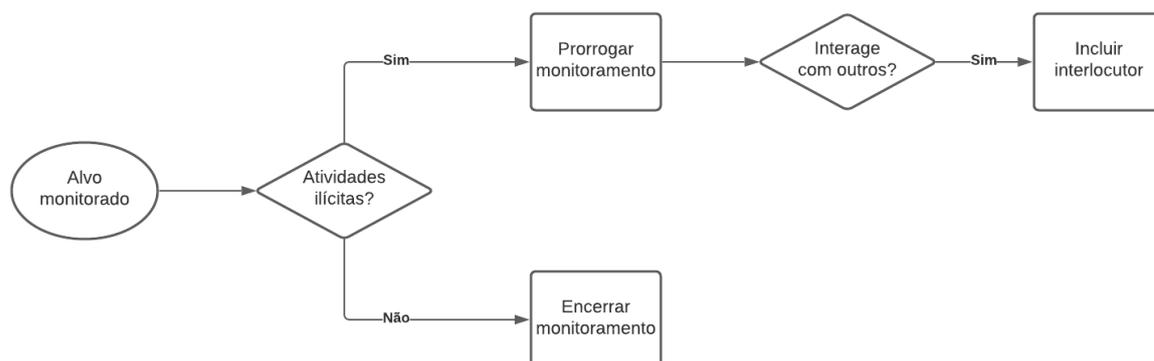


Figura 1.4: Fluxograma de renovação de ciclo

Porém, percebeu-se que devido a interação entre os alvos, diversas ligações foram interceptadas de forma duplicada, em mais de um alvo, o que gera um desperdício de recursos humanos e computacionais (como o armazenamento das chamadas no servidor, a alocação do canal que desvia a chamada telefônica para agência, etc). Em termos práticos, o ideal seria minimizar a ocorrência desse fato através da seleção adequada de alvos monitorados.

Assim, tendo em mente obter a maior quantidade de informações sobre a composição da ORCRIM e considerando a impossibilidade de acompanhar todos os membros de uma organização criminosa devido à limitações já mencionadas, a motivação deste trabalho consiste em projetar uma ferramenta que auxilie o(s) analista(s) na escolha de quais alvos interceptar baseado na sua interação e minimizando o desperdício de recursos.

A proposta é que durante a reunião entre os analistas que ocorre a cada ciclo, seja executada a ferramenta, assim, isto modificará um pouco o fluxo de atividades de renovação de ciclo, acrescentando uma etapa, conforme ilustrado na Figura 1.5.

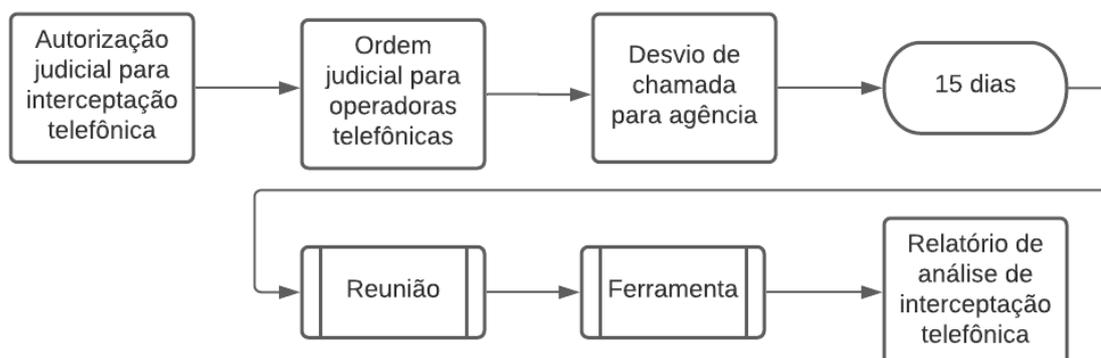


Figura 1.5: Novo fluxo de atividades para renovação de ciclo

Portanto, constituem objetivos específicos deste trabalho:

- Propor um modelo matemático para a seleção de alvos a ser monitorados;

- Criar um protocolo para fixação de alvos na interceptação;
- Propor um conjunto de alvos como futuros monitorados no ciclo de interceptação seguinte;
- Validar a solução proposta.

Conforme será abordado nos Capítulos 3 e 4, tais objetivos serão alcançados mediante as técnicas de programação linear inteira, a qual resolveu a instância de dados do problema real de maneira ótima, com um baixo custo computacional. Vale ressaltar que esta abordagem foi preferida devido ao seu desempenho: a princípio, pensou-se em usar métodos heurísticos, porém a escalabilidade se mostrou um empecilho, em especial, nas instâncias simuladas da Seção 4.3.

## 1.4 Trabalhos relacionados

O ponto chave deste trabalho consiste em uma ferramenta que auxilie as equipes a otimizar a seleção de alvos de uma interceptação telefônica baseado nos registros telefônicos provenientes dos extratos reversos fornecidos pelas operadoras telefônicas.

Para alcançar este objetivo, uma abordagem é modelar a rede criminal como um grafo e, a partir de então, aplicar os conceitos relacionados a teoria dos grafos (neste trabalho, o problema de cobertura de vértices) para propor novos alvos a serem interceptados no ciclo seguinte baseado nas interações desses indivíduos.

Não foram encontrados, na literatura, trabalhos com este objetivo. Há outros trabalhos que estudam as propriedades de uma rede criminal usando os conceitos de grafos (como [Gavião et al. \(2021\)](#) e [Cavallaro et al. \(2021\)](#)), da mesma forma que há trabalhos que sugerem a modelagem dos vínculos destes indivíduos (como [Englefield \(2022\)](#)). Porém, isto difere do proposto aqui.

## 1.5 Estrutura do texto

As demais seções deste trabalho são constituídas por quatro capítulos, organizados por seu tema, a saber:

- Capítulo 2: traz a fundamentação teórica necessária;
- Capítulo 3: aponta a abordagem proposta;
- Capítulo 4: apresenta as ferramentas e os resultados obtidos;
- Capítulo 5: discorre sobre a conclusão e os trabalhos futuros.

# 2

## Fundamentação Teórica

Esta seção aborda os principais conceitos teóricos necessários à resolução do problema proposto.

### 2.1 Teoria dos grafos

A teoria dos grafos foi introduzida por Euler, motivado pelo problema das pontes de Königsberg, ilustrado na Figura 2.1. Königsberg trata-se de uma cidade da Prússia do século 18, que possuía um total de sete pontes para uma ilha, todas cruzando o rio Pregel, como na figura abaixo. Os habitantes da cidade questionaram-se sobre a possibilidade de cruzar as sete pontes, sem passar duas vezes por nenhuma uma delas.

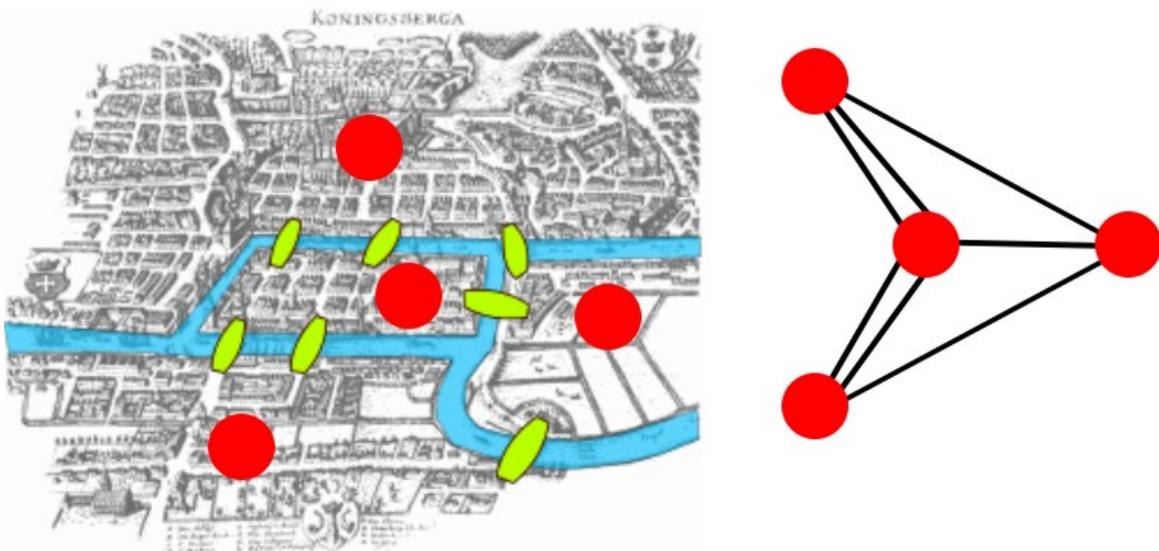


Figura 2.1: Pontes de Königsberg

Tal problema foi modelado a partir em estruturas hoje conhecidas por grafos, que possuem um conjunto não vazio de objetos chamados de nós (ou vértices) e outro conjunto de pares não ordenados de vértices, denominado de arestas. Normalmente, denota-se um grafo como  $G(V, E)$ , onde  $V$  é o conjunto dos vértices e  $E$ , o conjunto de arestas (do inglês, *edges*). Na Figura 2.2, a seguir, está representado um grafo  $G(V, E)$  onde  $V$  consiste em um conjunto de 6 vértices (numerados de 1 a 6) e  $E$  é formado por 5 arestas.

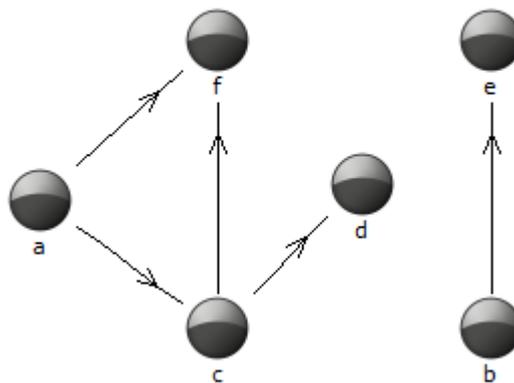


Figura 2.2: Exemplo de um grafo com 6 vértices

As arestas podem ter tanto peso quanto direção. Quando as arestas possuem uma direção (ou sentido) associado, dizemos que trata-se de um grafo orientado ou dirigido ou direcionado. Já quando um grafo é dito ponderado, ou valorado, suas arestas possuem um peso (valor numérico) associado. Não há empecilhos formais para que um mesmo grafo seja, simultaneamente, dirigido e ponderado, como na Figura 2.3.

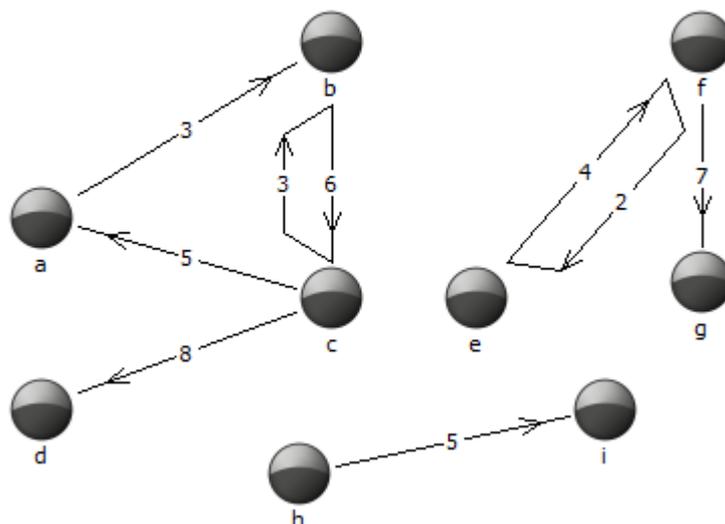


Figura 2.3: Grafo ponderado e direcionado

Além dos conceitos anteriores, há também o de grau, que refere-se à quantidade de arestas que incide sobre um vértice. Para o grafo da Figura 2.3, observe que o vértice  $b$  possui grau 3, enquanto o vértice  $h$  possui grau 1.

### 2.1.1 Aplicações de grafos

Grafos podem ser entendidos como uma abstração que permite codificar relacionamentos entre pares de objetos (Protti, 2021). Pensando assim, os objetos são mapeados como vértices e os relacionamentos como arestas.

Alguns exemplos de aplicações seriam:

**Transporte Aéreo:** neste contexto, pode-se entender as cidades como vértices e os vôos comerciais entre duas cidades como uma aresta;

**Web:** aqui as páginas web são os vértices, enquanto as ligações entre uma página e outra são as arestas;

**Redes sociais:** considere as pessoas como vértices e as interações entre elas como arestas;

**Interceptações telefônicas:** trate os números de telefones interceptados como vértices e as chamadas entre os números como arestas.

A aplicação da teoria dos grafos é multidisciplinar, inclusive, segundo Feofiloff et al. (2011), vários problemas de grafos são um desafio intelectual que possuem importantes aplicações práticas.

### 2.1.2 Problema de cobertura de vértices

Seja  $G = (V, E)$  um grafo não-dirigido, onde  $V$  é seu conjunto de vértices e  $E$  representa suas arestas. Dizemos que o conjunto cobertura será o conjunto de nós  $S$  tal que  $S \subseteq V$  cada aresta  $e \in E$  tem pelo menos uma extremidade em  $S$ .

Assim, o conjunto cobertura de um grafo (ou simplesmente sua cobertura) é o conjunto de vértices que cobre todas as arestas. Além disso, diz que uma cobertura é mínima se, para o grafo  $G$ , não existe outra menor (ou seja, não existe outro conjunto cobertura que possua menos vértices). Nesse caso, denominamos o conjunto de cobertura de vértices ótima.

Na Figura 2.4 a seguir, temos três momentos de um grafo: inicialmente, o grafo original. Em seguida, um vértice é adicionado ao conjunto cobertura e as arestas que incidem sobre ele (bem como os vértices na extremidade dessas arestas) são marcados em amarelo. Por fim, mais um vértice é adicionado à cobertura e repete-se o procedimento de marcação. Como resultado, os 6 vértices do grafo são cobertos por 2.

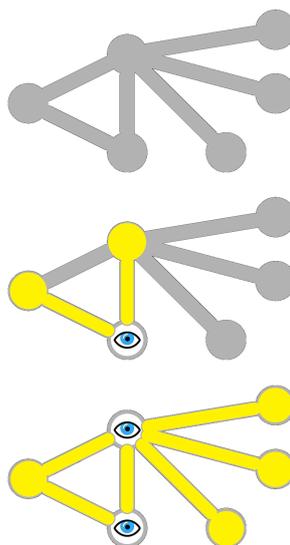


Figura 2.4: Cobertura de vértices

Sabe-se que o problema de cobertura de vértices é  $\mathcal{NP}$ -completo (Karp, 1972). Devido a sua natureza combinatória, um algoritmo por força bruta examinaria todas as  $2^n$  combinações de vértices possíveis, o que é inviável a depender do tamanho de  $n$ . Portanto, há algoritmos que fornecem aproximadas.

Embora não seja conhecido um algoritmo polinomial para resolver o problema de cobertura, há estratégias que encontram soluções viáveis. Uma delas, usada neste trabalho, consiste em usar programação inteira.

### 2.1.3 Modelos de redes complexas

Há diversos modelos que permitem determinar as propriedades estruturais do grafo, como distribuição de grau e coeficiente de aglomeração. Dentre eles, destacam-se o  $\mathbb{G}(n, m)$ ,  $\mathbb{G}(n, p)$  e *scale-free*.

$\mathbb{G}(n, m)$ : São redes nas quais tem-se  $n$  nós e  $m$  arestas e, portanto, também são denominadas redes regulares. Este conceito é bastante útil para estudar, por exemplo, os modelos atômicos.

$\mathbb{G}(n, p)$ : Neste modelo, proposto por Erdős & Rényi (1959), as redes possuem  $n$  vértices e cada vértice possui a probabilidade  $p$  de estar ligado a outro vértice. A formação de um grafo desse tipo ocorre com os nós ligando-se de forma aleatória, por isso, os nós tem (em média) a mesma quantidade de ligações.

**Scale-free:** No modelo de ligação preferencial (ou *scale-free*, em inglês), que foi definido por Barabási & Albert (1999), a criação do grafo difere dos outros modelos, por não ser construído de uma vez. Na verdade, é um processo que se dá ao longo do tempo a partir de duas características: a de crescimento e a de preferência de conexão. A primeira refere-se a adição de novos vértices a medida que o grafo cresce, enquanto a segunda relaciona-se a ligar os novos vértices adicionados aos nós de maior grau.

Deste modo, o algoritmo que resulta neste tipo de rede consiste em:

1. Para garantir que a rede não terá nós desconexos, a rede é iniciada com um número pequeno de vértices  $m_0$ , cujas arestas são arbitrariamente criadas;
2. A cada passo, é adicionado um novo nó à rede, com  $m \leq m_0$  ligações, conectando o novo vértice a outros  $m$  vértices. A probabilidade de haver um vértice receber uma nova ligação é proporcional a sua conectividade. Assim, a probabilidade  $p_i$  do vértice  $v_i$  receber a conexão de um novo vértice é definido por:

$$p_i = \frac{d(v_i)}{\sum_{v \in V} d(v)}$$

em que  $d(v_i)$  é o grau do vértice  $v_i$ .

3. Repete-se o Passo 2 até o tamanho desejado da rede.

Com isso, este tipo de rede possui elementos que agrupam ligações (*hubs*), como pode ser visto na Figura 2.5, que traz a comparação de uma rede aleatória (à esquerda) e uma de ligação preferencial (à direita).

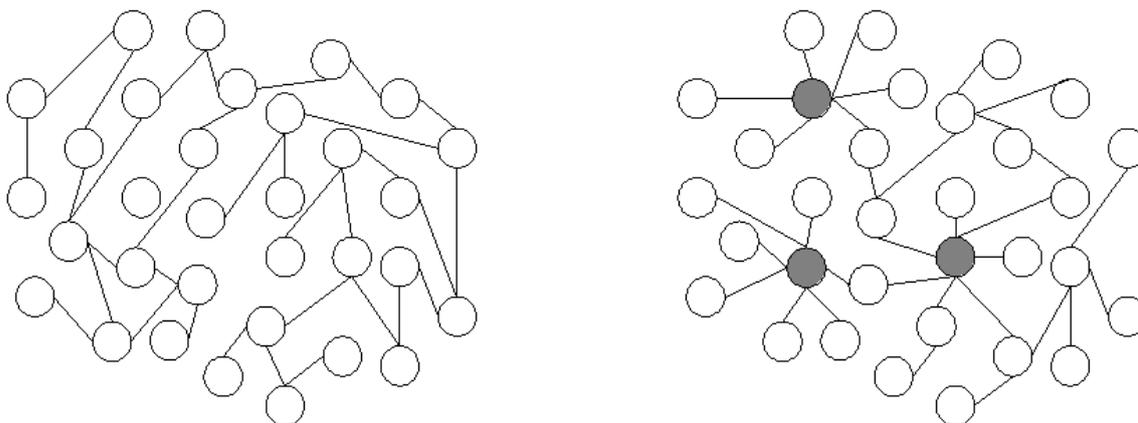


Figura 2.5: Modelos de redes: à esquerda, uma rede aleatória; à direita, uma rede de ligação preferencial com os nós que concentram ligações destacados em cinza

## 2.2 Programação linear

A programação linear que teve seu boom na 2<sup>a</sup> Guerra Mundial devido as aplicações militares, consiste essencialmente em uma função objetivo linear (que se deseja maximizar ou minimizar) e algumas restrições lineares, conforme exemplo na Figura 2.6.

A formulação de um Problema de Programação Linear (PPL) consiste em:

**Variável de decisão:** o que se deseja otimizar;

**Restrições:** o que separa uma solução viável de uma inviável;

**Função objetivo:** como decidir a melhor solução.

A depender do problema modelado, pode ser interessante também definir restrições de domínio, que simbolizam os valores que a variável de decisão pode assumir. Caso o domínio das variáveis seja inteiro, tem-se um problema de programação linear inteira.

$$\begin{aligned}
 \text{(PPL) max} \quad & 11x_1 + 12x_2 \\
 \text{s. a} \quad & x_1 + 4x_2 \leq 10 \\
 & 5x_1 + 2x_2 \leq 20 \\
 & x_1, x_2 \geq 0
 \end{aligned}$$

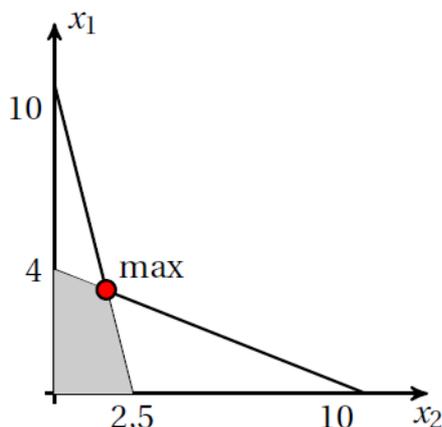


Figura 2.6: Exemplo de problema de programação linear. No topo, a função objetivo: maximizar  $11x_1 + 12x_2$ . Em seguida, as restrições lineares:  $x_1 + 4x_2 \leq 10$ ,  $5x_1 + 2x_2 \leq 20$ . Por fim, as restrições de domínio:  $x_1, x_2 \geq 0$ .

### 2.2.1 Método Simplex

Dentre os algoritmos para resolver um Problema de Programação Linear - PPL, um dos mais conhecidos é o método Simplex. Proposto pelo matemático George Dantzig, o método encontra boa aceitação onde as restrições influenciam em valores que precisam ser maximizados.

A respeito da complexidade, formalmente, o Simplex é tido como exponencial (Dagupta et al., 2006) devido ao seu caráter combinatório, conforme a equação abaixo, na qual  $n$  relaciona-se com a quantidade de variáveis e  $m$  com as restrições. Apesar disso, tanto Dagupta et al. (2006) quanto Kleinberg & Tardos (2006) relatam que, são poucos os problemas cujas instâncias, na prática, tomam esta complexidade e, por isto, o algoritmo Simplex é altamente utilizado.

Para usar o Simplex, é necessário transformar o PPL no formato padrão:

- A função objetivo deve ser uma maximização;
- As restrições precisam ser igualdades.

Por isso, em alguns casos, para a modelagem, pode ser necessário usar alguns artifícios: para a função objetivo, transformar considerando que  $\min(f(x)) = -\max(-f(x))$ ; para igualdades, usar estratégias como as variáveis de folga. Em geral, especialmente para o segundo caso, será necessário fazer algum tipo de ajuste, já que os problemas reais costumam trazer inequações.

Uma vez modelado corretamente, as restrições formam um conjunto de soluções viáveis (região sombreada na Figura 2.6), isto é, um conjunto de valores que as variáveis de decisão podem assumir. Assim, duas variáveis de decisão, resultam num polígono em um plano, três, resultam num poliedro, etc. Com isso, o método Simplex, percorrerá os vértices da região viável até encontrar uma solução ótima, conforme a Figura 2.7.

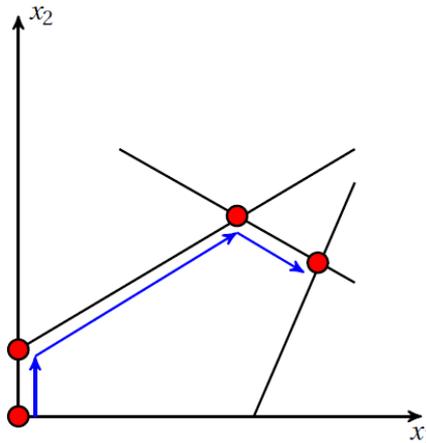


Figura 2.7: Método Simplex para duas variáveis de decisão: em vermelho, os vértices que serão percorridos em busca da solução ótima.

### 2.2.2 Programação linear inteira - PLI

A programação linear inteira consiste em um caso especial da programação linear, em que suas variáveis devem ser inteiras. Nesse caso, o domínio agora é um grid, tornando o espaço de soluções a ser testado muito menor, conforme a Figura 2.2.2.

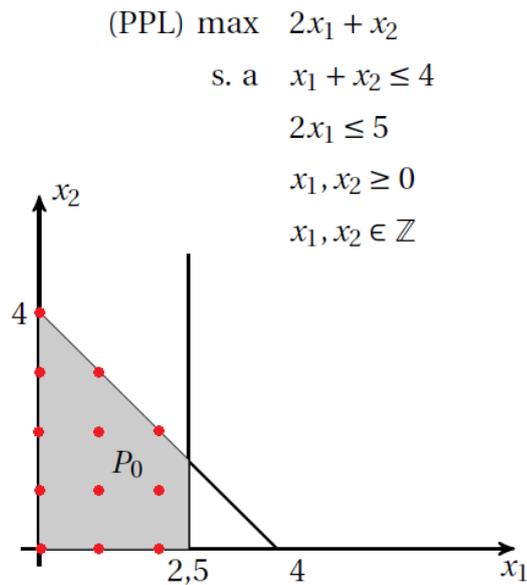


Figura 2.8: O domínio de um PLI (pontos em vermelho) é mais restrito que um PPL (plano em cinza), pois suas variáveis devem ser inteiras.

Frequentemente, esse tipo de restrição aplica-se quando a variável de decisão é binária: como quando  $x \in \{0, 1\}$ , significando que  $x$  pertence ou não ao conjunto decisão.

### Formulação do problema de cobertura

Dado um grafo  $G(V, E)$ , o problema de cobertura pode ser modelado com uma formulação de programação linear inteira da seguinte forma:

Desta forma, dada uma variável de decisão binária  $y_v$  para todo vértice  $v \in V$ , em que  $y_v = 1$  se e somente se o vértice  $v$  pertencer a cobertura ótima. A formulação de programação linear inteira para o problema de cobertura de vértice é definida como:

$$(\text{COBERTURA}) : \min \sum_{v \in V} y_v \quad (2.1)$$

$$\text{s. a } y_v + \sum_{u \in N(v)} y_u \geq 1 \quad \forall v \in V \quad (2.2)$$

$$y_v \in \{0, 1\} \quad \forall v \in V \quad (2.3)$$

Neste contexto, a Equação (2.1) minimiza a quantidade de vértices na cobertura, enquanto a Restrição (2.2) determina que, para cada vértice  $v$ , ele próprio ou pelo menos um de seus vizinhos deve pertencer ao conjunto cobertura. Por fim, a Restrição (2.3) determina o domínio das variáveis de decisão.

Além disso, cabe destacar que a natureza combinatória do problema se mantém, assim

---

como num PPL. Na literatura, este tipo de problema é  $\mathcal{NP}$ -difícil (Garey & Johnson, 1979), portanto, o tempo para obter uma solução ótima cresce exponencialmente de acordo com os dados de entrada.

# 3

## Proposta

Neste capítulo serão abordados os pontos relevantes para o desenvolvimento do algoritmo de cobertura de vértices.

### 3.1 Problemática

Como já explicado, durante a operação Flashback I, a escolha dos alvos a serem interceptados a cada ciclo foi decidida em reunião nas quais os analistas, baseados em seu conhecimento, sugeriam as inclusões, prorrogações e exclusões de terminais telefônicos. Porém, percebeu-se que alvos importantes e interceptados, por diversas vezes, conversavam entre si. Com isso, houveram diversas chamadas que foram interceptadas em mais de um alvo, gerando um desperdício de recursos que deseja-se minimizar.

Além das chamadas duplicadas, muitas vezes o mesmo fato era escutado em alvos distintos, pois estavam dentro do mesmo círculo de relacionamentos e falavam uns com os outros sobre os mesmos assuntos, o que também é desejável minimizar. Exceto para questões pontuais, como qualificar o alvo, pode ser interessante não interceptar simultaneamente dois (ou mais) alvos que interagem diretamente entre si, conforme a Figura 3.1.

Houve também um aspecto que dificultou a identificação de duplicidade de informações num primeiro momento: o número de alvos *versus* o número de analistas. Como cada analista era responsável por acompanhar alguns alvos, nem sempre o mesmo analista escutava a chamada repetida num segundo alvo. Portanto, houveram vezes em que tal redundância de trabalho não foi detectada.

Assim, a proposta é aplicar o problema de cobertura de vértices à ORCRIM, que será mapeada como um grafo ponderado a partir do extrato reverso de chamadas telefônicas fornecido pela operadora, conforme a Figura 3.2, que mostra um grafo modelado no *software* IBM i2,

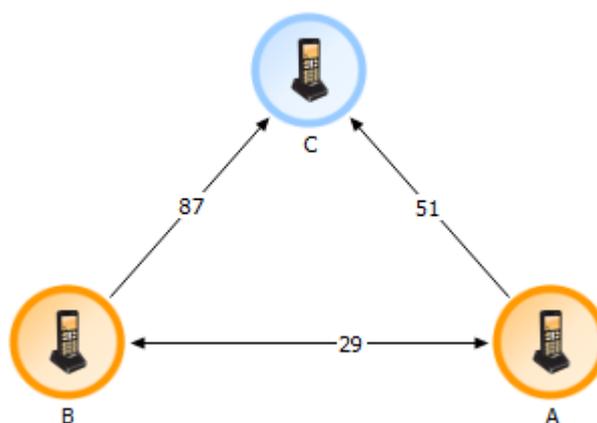


Figura 3.1: Alvos que geram áudios duplicados: ao interceptar A e B (em laranja), todas as chamadas entre os dois serão salvas no alvo A e no alvo B. Além disso, se C (em azul) também fosse interceptado, o conteúdo seria replicado ainda mais vezes.

usado em análises de vínculo. Neste grafo, os vértices representam os alvos e seus interlocutores, as arestas simbolizam as chamadas em si e os pesos nas arestas simbolizam o tempo de duração total das chamadas entre um alvo e um interlocutor.

A proposta é que durante a reunião entre os analistas que ocorre a cada ciclo, seja executada a ferramenta e esta sugira quais alvos interceptar. Para isso, será necessário fornecer como entrada os extratos reversos dos alvos monitorados, bem como o número limite de alvos que deseja-se interceptar. Como saída, será fornecida uma lista dos vértices que maximizam a soma total de horas (representadas nas arestas do grafo) das chamadas telefônicas dos interceptados. Esta lista será estudada pelos analistas para que decida-se incluir ou não os indivíduos sugeridos. De forma opcional, pode-se fixar (antes da execução da ferramenta) indivíduos cuja expertise dos analistas entende que devem ser monitorados e, portanto, devem pertencer à solução (ou ainda o contrário: indivíduos que não se deseja monitorar).

## 3.2 Análise dos dados e modelagem do grafo

cap3:analise

Num primeiro momento, foram estudados os dados oriundos dos extratos reversos de 34 alvos. Tais extratos foram solicitados durante o andamento da operação e disponibilizados agora já que o processo judicial relacionado já se tornou público.

Um extrato reverso é uma planilha que contém informações sobre as chamadas efetuadas e recebidas por um número de telefone ou IMEI durante certo período de tempo. Esta planilha fornecida pelas operadoras telefônicas, cada uma com sua formatação, que contém alguns campos, cada um em uma coluna. Dentre eles, pode-se citar:

- Número A, IMEI A: refere-se ao número de telefone e IMEI que realizou a chamada;

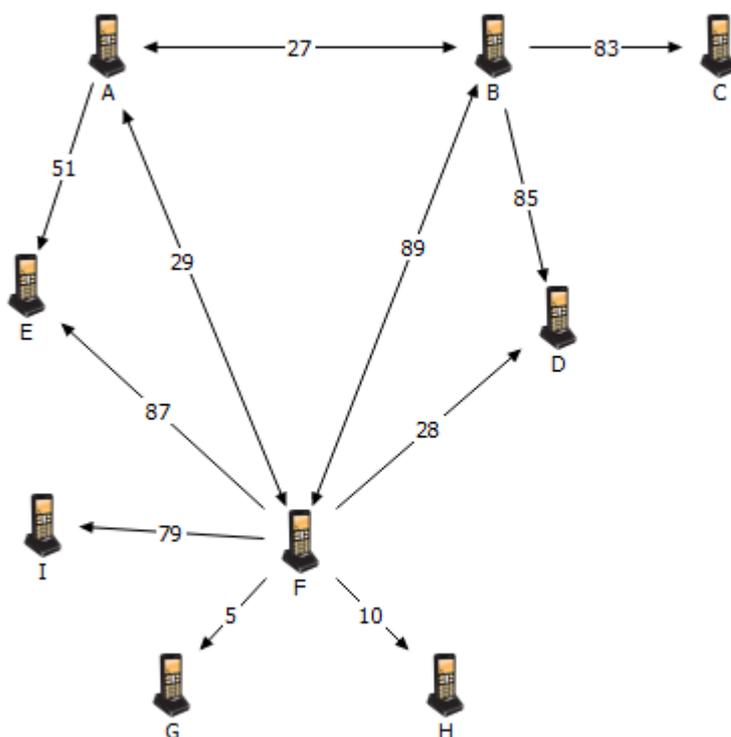


Figura 3.2: Exemplo de um ORCRIM modelada no *software* IBM i2

- Número B, IMEI B: refere-se ao número de telefone e IMEI que recebeu a chamada;
- Data Início, Hora Início, Duração: refere-se a data e hora de início da chamada, bem como sua duração (em segundos);
- Endereço ERB RE, Município ERB RE, Latitude RE, Longitude RE, Raio RE, Azimute RE, Endereço ERB RS, Município ERB RS, Latitude RS, Longitude RS, Raio RS, Azimute RS: refere-se aos dados da ERB (estação rádio base) de quem realizou e de quem recebeu a chamada.

Alguns desses dados podem sofrer variação no nome do campo, a depender da operadora. Porém, os campos mais relevantes para este trabalho são Número A, Número B e Duração, que comuns às quatro operadoras mais usadas no Brasil segundo *ranking* da Anatel: Tim, Claro, Oi e Vivo (ver Figura 3.3).

Como os dados disponíveis não foram obtidos para a finalidade a que se refere este trabalho, os extratos reversos da operação referem-se a períodos de tempos distintos, das diferentes operadoras. Assim, num primeiro momento, foi selecionado um espaço de tempo comum a todos os extrados, que pudesse ser usada de forma semelhante ao que acontecerá na prática. Finalmente, o espaço de tempo considerado nesse trabalho é de quinze dias.

A partir destes extratos filtrados, foi modelado um grafo (Figura 3.4), de forma que a existência do relacionamento entre as entidades representa uma ligação. Assim, considerou-se as colunas referentes a Número A e Número B como nós com uma aresta entre elas.

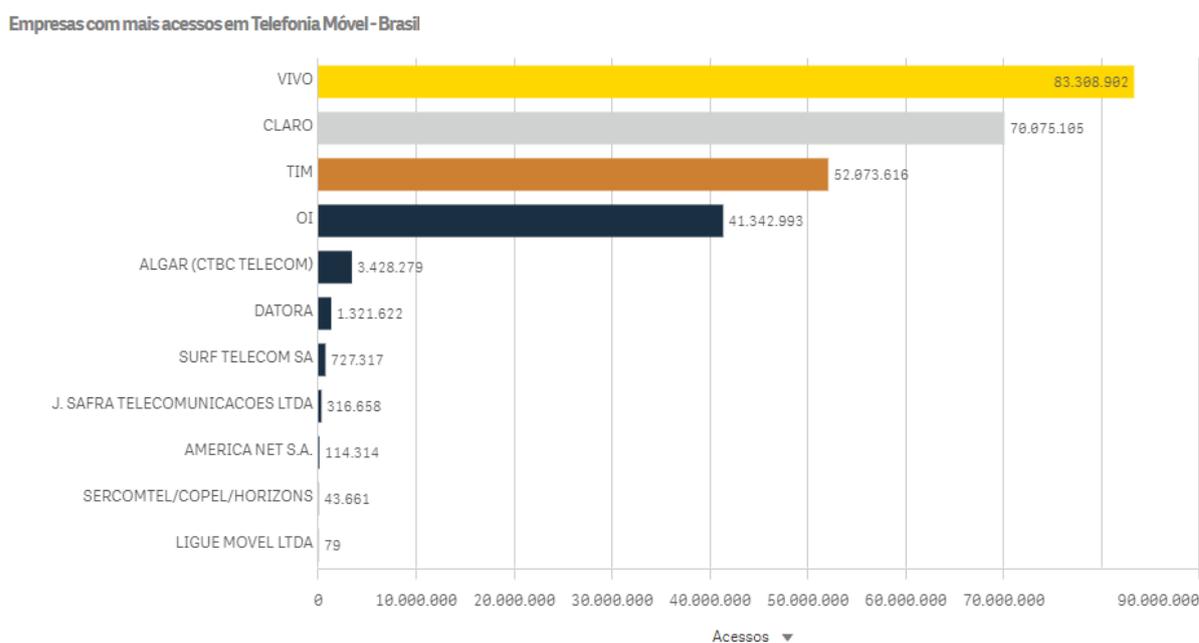


Figura 3.3: Operadoras mais usadas no Brasil em janeiro de 2022 segundo ranking da Anatel disponível em <https://informacoes.anatel.gov.br/paineis/acessos/ranking>

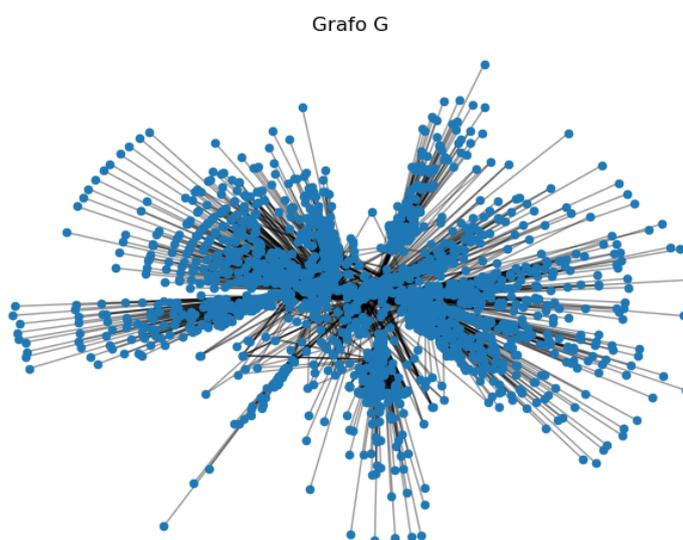


Figura 3.4: Grafo criado a partir dos extratos reversos

Entendeu-se também que é importante mensurar a relação entre os dois nós, ou seja, atribuir um peso às arestas. Isto poderia ser feito de duas maneiras: a partir da quantidade de chamadas ou pela duração total das chamadas entre ambos os alvos.

Optou-se pela segunda forma visto que uma chamada longa, de várias horas de duração, por exemplo, tende a ter mais conteúdo relevante para a operação do que diversas chamadas curtas, que duram apenas alguns minutos. Além disso, as direções dos arcos não foram levadas

em consideração, isto é, o grafo resultante não é dirigido. Após a criação do grafos ponderados, tornou-se necessário entender como as interações se comportam, para isso, foram analisados algumas características do grafo.

A instância real modelada possui um total de 2957 vértices e 3487 arestas. Observou-se também que o vértice mais conectado possui 532 arestas (ver Figura 3.5) e que os demais vértices possuem muito menos conexões. Além disso, não considerou-se relevante modelar o grafo orientado, visto que a simples existência da aresta é suficiente para mostrar a relação entre dois interlocutores, independente de quem ligou e quem recebeu a chamada.

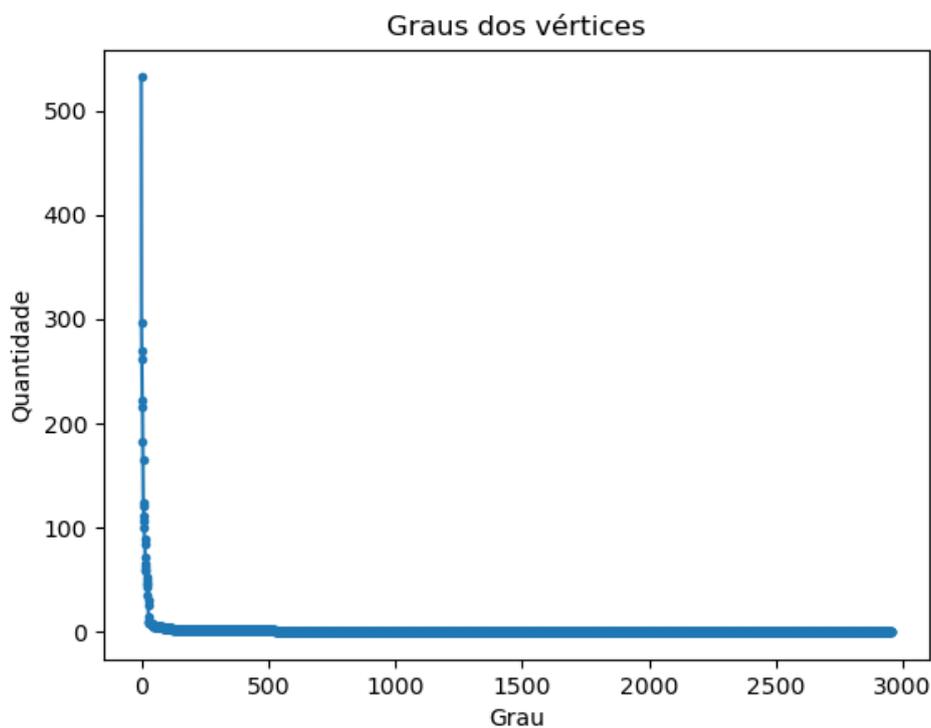


Figura 3.5: Gráfico de graus do grafo

Por fim, dados dois vértices quaisquer, a probabilidade de existir uma aresta ligando-os, ou seja, a densidade do grafo, é de 0,000797. A densidade de um grafo  $G(V, E)$  é definida pela quantidade de arestas em  $G$  dividido pela quantidade de arestas de um grafo completo com mesmo número de vértice, isto é:

$$den(G) = \frac{2|E|}{|V|(|V| - 1)}.$$

Somando-se isso às características mencionadas, percebeu-se que, conforme esperado, alguns nós concentram muitas chamadas, enquanto outros nós pouco aparecem. Tais nós também possuem grau mais elevado que os demais. Estas observações caracterizam a rede como no modelo de ligação preferencial.

Foi feito também uma análise sobre a distribuição dos pesos das arestas: procurou-se determinar a distribuição estatística mais se adequada aos dados. Para alcançar este objetivo, foram realizados alguns testes usando a biblioteca `Fitter`, do Python, cujos resultados estão detalhados na Tabela 3.1, que apresenta a soma do erros ao quadrado (SEQ) de cada distribuição.

Distribuição	SEQ
halflogistic	2,33e-11
laplace asymmetric	4,66e-10
expon	4,66e-10
chi2	3,42e-09
rayleigh	4,10e-09
exponpow	4,33e-09
norm	4,68e-09
powerlaw	4,84e-09
lognorm	4,99e-09
cauchy	6,12e-09

Tabela 3.1: Tabela de erro: distribuição de pesos nas arestas do grafo

A biblioteca mencionada permite que o teste estatístico seja gerado para distribuições específicas ou para um conjunto de mais de 100 distribuições suportadas pela biblioteca `SciPy`. O dados mostrados na tabela acima e na Figura 3.6 trazem apenas os melhores resultados para fins de visualização.

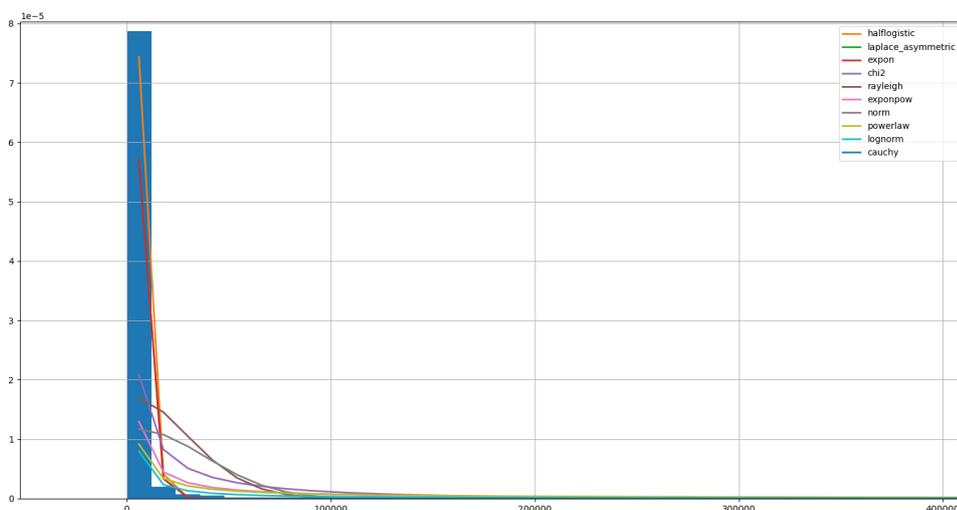


Figura 3.6: Gráfico de erro: distribuição de pesos nas arestas do grafo

Assim, concluiu-se que a distribuição de pesos nas arestas segue a distribuição estatística *half-logistic*, cuja função de densidade de probabilidade é:

$$f(x) = \frac{2e^{-x}}{(1+e^{-x})^2} = \frac{1}{2} \operatorname{sech}(x/2)^2, \quad \text{para } x \geq 0. \quad (3.1)$$

Em relação aos parâmetros que caracterizam esta curva, há que se considerar dois: localização e escala (*loc* e *esc*, respectivamente). O primeiro está relacionado a largura, enquanto o segundo influencia a função em termos de sua altura. Em conjunto, definem se a função é alta e achatada ou mais baixa e larga. A densidade de probabilidade acima é definida na forma “padronizada”. A função de densidade da *half-logistic* com parâmetros *loc* e *esc* é equivalente  $f(y)/esc$  com  $y = (x - loc)/esc$ . Para os dados em análise, verificou-se que os parâmetros são:  $loc = 0,99$  e  $esc = 3707,95$ . Mediante tais informações torna-se possível, em momento oportuno, criar outras instâncias semelhantes para aplicação do algoritmo de cobertura de vértices proposto.

### 3.3 Formulação Matemática

Esta seção apresenta a modelagem do problema de seleção de alvos usando programação linear inteira.

Foram consideradas, a princípio, duas abordagens para o problema de seleção de alvos: a primeira (denominada *k*-alvos) consiste em fixar a quantidade de vértices *k* da cobertura e tentar maximizar o peso total das arestas cobertas pelos *k* vértices; enquanto a segunda (denominada *T*-horas) consiste no contrário, limitar a soma dos pesos das arestas e pertencentes à cobertura e maximizar o valor de *k* (vértices na cobertura).

No primeiro caso, tem-se uma variante do problema de cobertura de vértices conhecida na literatura como *k*-cobertura de vértice máxima (*maximum k-vertex cover*) ou máxima cobertura parcial de vértices (*maximum partial vertex cover*) (Manurangsi, 2018). Neste problema, a entrada é um grafo ponderado nas arestas *G* e um inteiro *k*, e o objetivo é encontrar um subconjunto *V'* de *k* vértices que maximize o peso total das arestas cobertas por *V'* (chamada *k*-cobertura ou cobertura parcial). Uma aresta é dita coberta por *V'* se pelo menos uma de suas extremidades estiver em *V'*. Este problema também é  $\mathcal{NP}$ -difícil e sua formulação é definida

como:

$$(k\text{-alvos}) : \max \sum_{e \in E} w_e x_e \quad (3.2)$$

$$\text{s. a } y_v + y_u \leq 1 \quad \forall uv \in E \quad (3.3)$$

$$\sum_{v \in V} y_v = k \quad (3.4)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (3.5)$$

$$y_v \in \{0, 1\}. \quad \forall v \in V \quad (3.6)$$

Aqui, as variáveis binárias são  $x$  e  $y$ , associadas respectivamente às arestas e aos vértices do grafo. A função objetivo consiste em maximizar a soma dos pesos das arestas ( $w_e$ ), conforme a Equação (3.2). Em seguida, as Restrições (3.3) obrigam que apenas um entre o vértice e seu vizinho pertençam a  $k$ -cobertura. Já a Restrição (3.4) limita o valor máximo de vértices pertencentes a  $k$ -cobertura. Por fim, as Restrições (3.5) e (3.6) definem o domínio das variáveis. Na prática, perceba que esta modelagem está relacionada a equipe escolher interceptar  $k$  alvos. A escolha de  $k$  é subjetiva.

No segundo caso, a formulação é:

$$(T\text{-horas}) : \max \sum_{v \in V} y_v \quad (3.7)$$

$$\text{s. a } x_e \geq y_v \quad \forall e = uv \in E \quad (3.8)$$

$$\sum_{e \in E} w_e x_e \leq T \quad (3.9)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (3.10)$$

$$y_v \in \{0, 1\} \quad \forall v \in V \quad (3.11)$$

Neste contexto, as variáveis binárias também são  $x$  e  $y$ , associadas respectivamente às arestas e aos vértices do grafo. A função objetivo consiste em maximizar a quantidade de vértices pertencentes à solução, conforme a Equação (3.7). As Restrições (3.8) garantem que a aresta só será contabilizada se o vértice estiver incluso na solução. Já a Restrição (3.9) limita a soma dos pesos das arestas pertencentes à cobertura parcial a um valor  $T$ . Por fim, as Restrições (3.10) e (3.11) definem o domínio das variáveis.

Na prática, nessa segunda abordagem, a entrada seria o grafo  $G(V, E)$  com pesos nas arestas  $w_e$  e um tempo limite  $T$ , que representaria, em horas, o tempo que os analistas teriam disponível para escutar as interceptações. Porém, esta abordagem não trouxe boas soluções, uma vez que os nós pertencentes à solução eram (em sua maioria) nós folhas, que tinham pouco tempo em chamada, mas interagiam com os nós que concentram ligações.

De toda forma, como não há garantia que a duração das ligações de um ciclo será a mesma do outro, optou-se por restringir o subconjunto  $V' \in V$  pela quantidade  $k$  de vértices, em vez

de pelo peso total  $T$  nas arestas associadas a estes vértices. Com isso, o objetivo passa a ser encontrar o subconjunto de  $k$  vértice que maximiza a soma dos pesos das arestas incluídas na cobertura parcial.

Observe ambos os problemas são ligeiramente diferentes do problema clássico de cobertura, no qual cada aresta do grafo é incidente a, pelo menos, um vértice do conjunto solução. Em outras palavras, no problema clássico, todas as arestas do grafo deve ser cobertas. Nestas variantes, deseja-se encontrar uma cobertura parcial levando em consideração os pesos das arestas.

Além disso, vale ressaltar que é importante que o conhecimento do analista a respeito da operação tenha impacto direto sobre as restrições de alvos: pode ser interessante definir que um alvo sensível deve, obrigatoriamente, pertencer à cobertura parcial de vértices ou, por outro lado, pode ser preferível excluir alguém da solução (o que pode ocorrer devido a interlocutores que interajam com membros da ORCRIM mas que, na realidade, não participam de suas atividades ilícitas, como familiares).

Assim, opcionalmente, pode-se adicionar como restrições a um ou outro vértice que deseja-se fazer parte da solução (ou mesmo o contrário, que deseja-se que não pertença à solução). Na modelagem da programação linear inteira, isto é feito simplesmente definindo o valor da variável associada ao vértice como 0 ou 1.

Por fim, o algoritmo dará como saída um subconjunto  $V' \in V$  contendo  $k$  vértices que são a sugestão de números telefônicos a serem monitorados no ciclo de interceptação seguinte. Mais uma vez, não descarta-se nem pretende-se substituir o analista aqui: para garantir que os resultados são coerentes e estão dentro da legalidade, a equipe deve analisar a saída e detectar se tais interlocutores realmente praticam atividades ilícitas para que possam ser incluídos na interceptação.



## Resultados e Discussões

Este capítulo contém os resultados computacionais da ferramenta proposta, bem como algumas discussões a respeito deles.

### 4.1 Ferramentas

O algoritmo para solucionar o problema de seleção de alvos foi desenvolvido usando o solver CPLEX 12.9.0.0. Os dados foram recebidos das operadoras em planilhas, cada um com seu próprio formato. Portanto, foi necessário limpar e transformar os dados num formato padrão, em arquivo de texto, para que este seja lido no CPLEX. A limpeza dos dados foi feita usando Python 3.8.8. Por fim, o algoritmo foi executado em um computador com processador Intel(R) Core(TM) i3-9100F, 8 GB de memória RAM e Windows 10 Pro versão 21H2.

### 4.2 Instância real

Os dados analisados a partir dos extratos reversos de um período de 15 dias geraram um grafo contendo 2957 vértices e 3487 arestas (ver Figura 4.1). Foi escolhido como parâmetro de entrada o valor  $k = 20$ , por ser uma quantidade de alvos possível de monitorar com os recursos disponíveis à época. A execução do algoritmo proposto levou aproximadamente 266ms e conjunto de vértices resultante gerou uma cobertura parcial de 98,9% do total de horas das ligações.

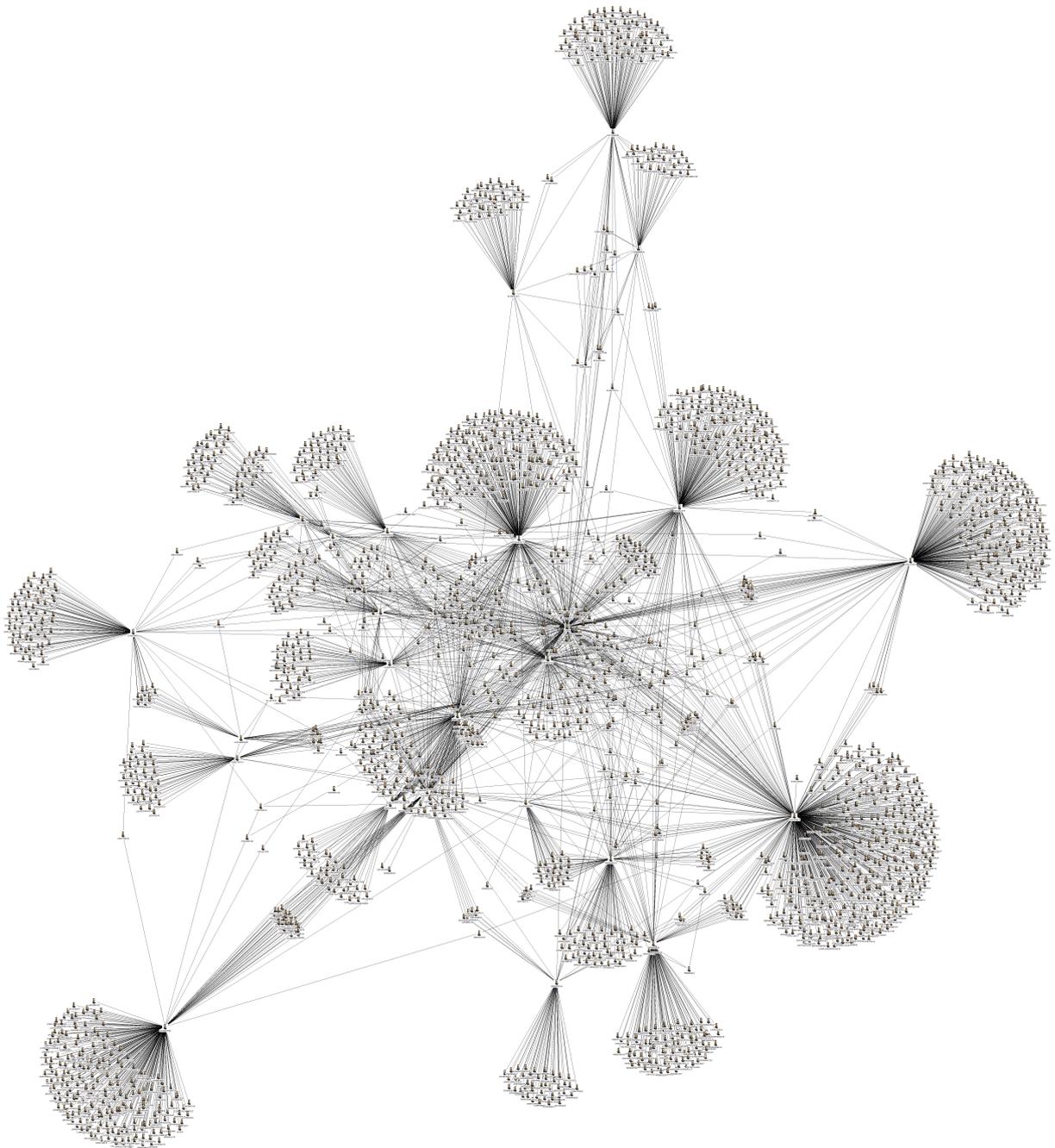


Figura 4.1: Grafo do extrato simplificado

O algoritmo foi também executado para outros valores de  $k$ , conforme a Tabela 4.1, que associa os valores de  $k$  com a porcentagem de peso da  $k$ -cobertura resultante, bem como o tempo de execução (em milissegundos). Perceba (analisando a terceira coluna) que, em certo ponto, o peso da  $k$ -cobertura passa a aumentar muito pouco em relação ao valor anterior de  $k$ , sugerindo que a adição de mais vértices a interceptação resulta em pouco conhecimento.

A título de comparação, foi feita uma escolha aleatória de 20 indivíduos na instância real e calculada a  $k$ -cobertura. Este procedimento foi repetido 500 vezes, resultando  $k$ -coberturas com peso médio de 1,28% (a pior  $k$ -cobertura foi de 0,0012% e a melhor de 18,91%). Perceba

k	Avaliação	Comparação	Tempo (ms)
5	0,53	-	406,41
10	0,828	+ 0,298	189,93
15	0,94	+ 0,112	237,97
20	0,989	+ 0,049	266,3
25	0,999	+ 0,01	140,32
30	0,999	0	338,29
35	1	+ 0,001	180,06
40	1	0	127,99

Tabela 4.1: Instância real: variando k

que o valor médio é muito inferior aos resultados encontrados usando o algoritmo exato, além disso, não há consistência: o melhor caso é simplesmente aquele que, por sorte, selecionou mais vértices bem conectados. Foi executado também um algoritmo guloso, que obteve uma  $k$ -cobertura de 96,09% do total de horas, considerando o valor de  $k = 20$ . Apesar do resultado ter sido muito bom, o algoritmo exato teve desempenho melhor, obtendo mais 2,81% do total de horas. Portanto, o *gap* entre os algoritmos, calculado através da Equação (4.1) é de 0,0284.

$$gap = \frac{exato - guloso}{exato} \quad (4.1)$$

### 4.3 Instâncias simuladas

Posteriormente, objetivando entender como a execução se dá em outras instâncias, foram gerados grafos com características semelhantes. Foi usado o modelo de ligação preferencial (*scale-free*) para criar o grafo e a distribuição *half-logistic* para determinar o peso nas arestas.

Num primeiro momento, gerou-se um grafo com o mesmo número de vértices do problema real. Em seguida, para simular instâncias maiores, a quantidade de vértices foi aumentada em 10 vezes e em 50 vezes. O algoritmo foi executado diversas vezes para estas quantidades, para verificar se o quão boa eram as soluções variaria muito.

#### 4.3.1 Instâncias com o mesmo tamanho (n = 2957)

Foram construídas 100 instâncias com as mesmas características do grafo obtido através dos dados, portanto, possui 2957 vértices. O parâmetro  $k$  foi mantido. A execução do algoritmo PLI, para todas as instâncias, demandou aproximadamente 16 segundos. Observe que, para uma instância apenas, o tempo de resolução seria baixíssimo.

A seguir, a Tabela 4.2 traz o número da solução e a porcentagem do peso da  $k$ -cobertura de cada solução. Todas as soluções foram ótimas, segundo o próprio CPLEX.

Nº	Avaliação	Nº	Avaliação	Nº	Avaliação
0	0,671	34	0,695	67	0,682
1	0,704	35	0,699	68	0,725
2	0,699	36	0,694	69	0,683
3	0,683	37	0,67	70	0,705
4	0,706	38	0,713	71	0,707
5	0,645	39	0,701	72	0,668
6	0,73	40	0,723	73	0,685
7	0,687	41	0,71	74	0,689
8	0,698	42	0,692	75	0,66
9	0,65	43	0,699	76	0,698
10	0,653	44	0,649	77	0,707
11	0,672	45	0,658	78	0,688
12	0,728	46	0,638	79	0,665
13	0,695	47	0,616	80	0,713
14	0,669	48	0,709	81	0,709
15	0,694	49	0,688	82	0,716
16	0,717	50	0,636	83	0,71
17	0,693	51	0,637	84	0,671
18	0,727	52	0,718	85	0,721
19	0,667	53	0,712	86	0,679
20	0,692	54	0,72	87	0,658
21	0,705	55	0,708	88	0,718
22	0,672	56	0,686	89	0,705
23	0,698	57	0,631	90	0,668
24	0,694	58	0,669	91	0,662
25	0,687	59	0,712	92	0,658
26	0,712	60	0,725	93	0,708
27	0,71	61	0,67	94	0,655
28	0,662	62	0,676	95	0,681
29	0,715	63	0,651	96	0,656
30	0,683	64	0,711	97	0,715
31	0,689	65	0,662	98	0,7
32	0,659	66	0,661	99	0,695
33	0,714	-	-	-	-

Tabela 4.2: Tabela de resultados para n=2957

Para melhor visualização, foi gerado um gráfico contendo os dados da tabela anterior (ver Figura 4.2). Sumarizando, tem-se que a melhor solução foi de 73% e a pior de 61,6%. Além disso, a porcentagem de peso da  $k$ -cobertura média foi de 68,77%.

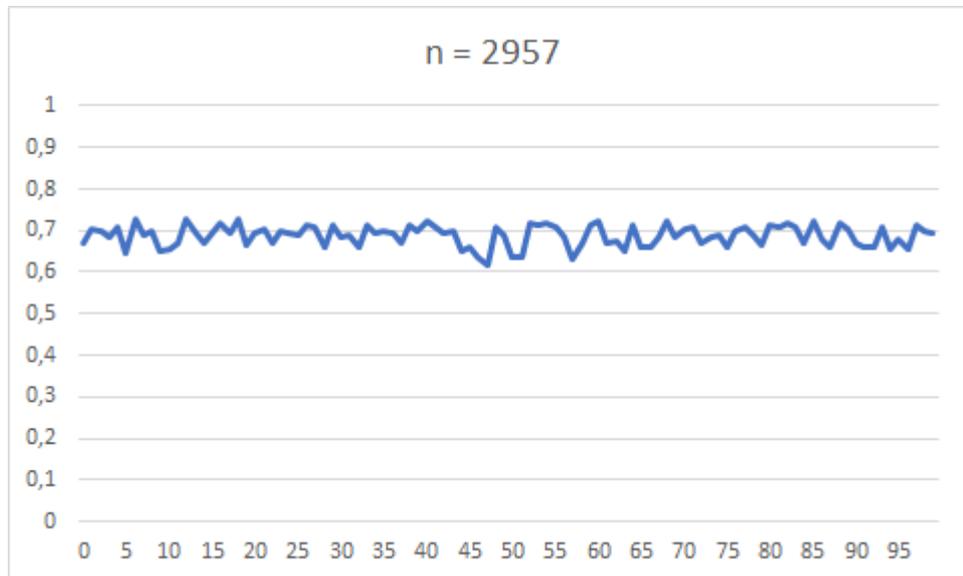


Figura 4.2: Resultados para  $n=2957$

### 4.3.2 Instâncias 10x maior ( $n = 29570$ )

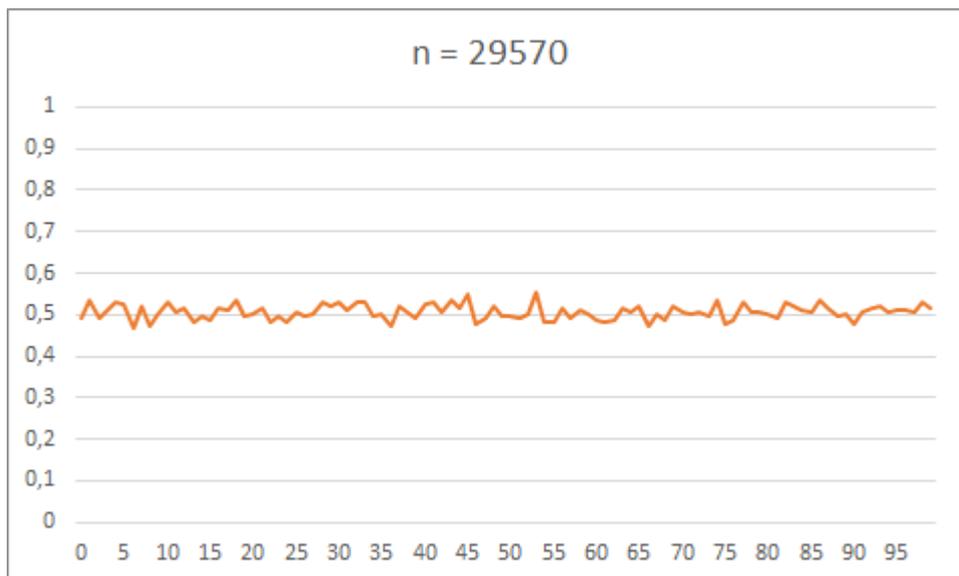
A seguir, uma instância 10 vezes maior que a original foi gerada, portanto, possui 29570 vértices. Note que o parâmetro  $k = 20$  foi mantido, porém, na prática, esta quantidade de interações iria requerer mais analistas trabalhando e, conseqüentemente, o valor de  $k$  deveria ser aumentado.

De maneira semelhante à instância  $n = 2957$ , o algoritmo foi executado 100 vezes, totalizando 676172ms (cerca de 11 minutos). Em média, pode-se dizer que cada execução levou 6761ms (menos de 7 segundos) para ser resolvida. Note que este tempo de espera é bastante viável em uma situação prática. Os resultados de todas as execuções estão condensados na Tabela 4.3 abaixo.

Para fins de melhorar a visualização, da mesma maneira que na seção anterior, os dados foram reestruturados no gráfico da Figura 4.3. Perceba que a melhor solução foi de 55,63% e a pior de 46,87%. Além disso, a porcentagem dos pesos da  $k$ -cobertura média foi de 50,67%.

Nº	Avaliação	Nº	Avaliação	Nº	Avaliação
0	0,490	34	0,498	67	0,503
1	0,535	35	0,504	68	0,488
2	0,494	36	0,474	69	0,520
3	0,509	37	0,521	70	0,504
4	0,532	38	0,507	71	0,502
5	0,523	39	0,490	72	0,506
6	0,469	40	0,527	73	0,494
7	0,521	41	0,531	74	0,535
8	0,475	42	0,508	75	0,479
9	0,502	43	0,536	76	0,488
10	0,532	44	0,514	77	0,532
11	0,507	45	0,551	78	0,508
12	0,514	46	0,475	79	0,505
13	0,483	47	0,492	80	0,500
14	0,498	48	0,519	81	0,491
15	0,487	49	0,498	82	0,530
16	0,518	50	0,494	83	0,519
17	0,510	51	0,493	84	0,509
18	0,533	52	0,504	85	0,507
19	0,497	53	0,556	86	0,534
20	0,504	54	0,484	87	0,516
21	0,517	55	0,483	88	0,497
22	0,482	56	0,515	89	0,501
23	0,497	57	0,489	90	0,478
24	0,483	58	0,512	91	0,508
25	0,505	59	0,500	92	0,517
26	0,495	60	0,488	93	0,523
27	0,500	61	0,482	94	0,505
28	0,528	62	0,489	95	0,510
29	0,521	63	0,514	96	0,509
30	0,528	64	0,508	97	0,507
31	0,510	65	0,521	98	0,530
32	0,528	66	0,471	99	0,518
33	0,528	-	-	-	-

Tabela 4.3: Tabela de resultados para n=29570

Figura 4.3: Resultados para  $n=29570$ 

### 4.3.3 Instâncias 50x maior ( $n = 147850$ )

Por fim, esta instância é 50 vezes maior que aquela gerada pelos dados originais, portanto, o número de vértices é  $n = 174.850$ . O valor de  $k$  foi mantido inalterado.

Vale ressaltar que esta instância foi executada menos vezes devido a demora em construir o grafo. Assim, foram simuladas 45 (em vez de 100) instâncias. O tempo de execução para resolver os PPLs foi de  $4.22836e+06$ ms, aproximadamente 1 hora e 10 minutos. Isto significa que, em média, cada instância levou cerca de 1 minuto e meio para ser resolvida.

Observe que, é extremamente improvável encontrar uma situação real com este tamanho, mesmo assim, o tempo de execução ainda é aceitável.

A seguir, na Tabela 4.4, estão os resultados. Os mesmos dados foram condensados na Figura 4.4 e nos mostram que a melhor solução foi de 43,73% e a pior de 37,12%. Além disso, a porcentagem dos pesos da  $k$ -cobertura média foi de 40,87%.

Nº	Avaliação	Nº	Avaliação	Nº	Avaliação
0	0,402	34	0,400	67	0,412
1	0,413	35	0,395	68	0,400
2	0,416	36	0,394	69	0,396
3	0,411	37	0,437	70	0,426
4	0,414	38	0,394	71	0,397
5	0,397	39	0,395	72	0,423
6	0,412	40	0,432	73	0,421
7	0,385	41	0,406	74	0,411
8	0,433	42	0,406	75	0,423
9	0,424	43	0,412	76	0,417
10	0,400	44	0,412	77	0,406
11	0,431	45	0,421	78	0,408
12	0,421	46	0,387	79	0,394
13	0,414	47	0,413	80	0,371
14	0,412	48	0,415	81	0,383

Tabela 4.4: Tabela de resultados para n=147850

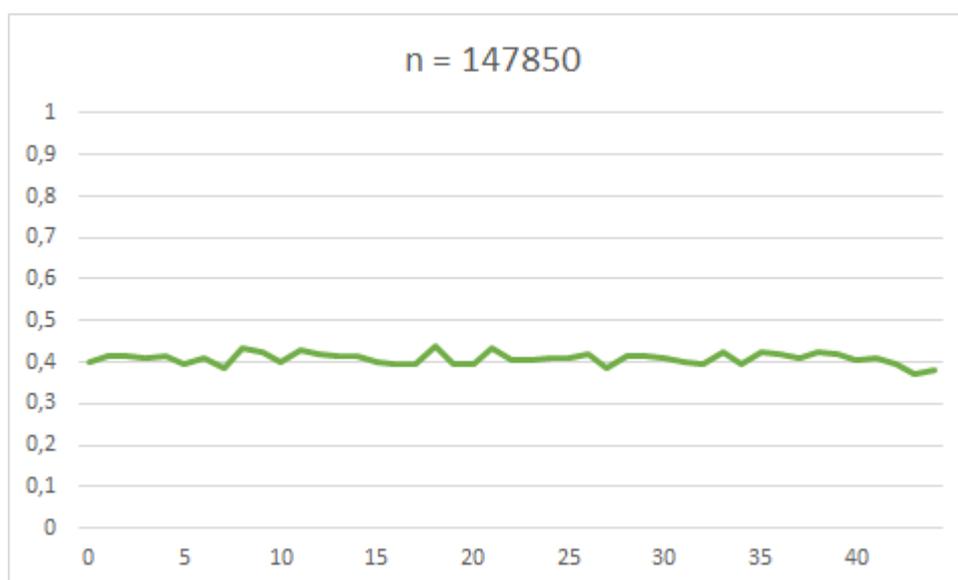


Figura 4.4: Gráfico de resultados para n=147850

# 5

## Considerações Finais

*"Em um mundo de redes criminais descentralizadas e flexíveis, o tempo disponível entre a análise (descobrir o que está acontecendo) e as operações (combater o problema) é cada vez menor"*

– Moisés Naim

Nesse trabalho foi implementado um algoritmo de  $k$ -cobertura de vértices para o grafo gerado a partir dos extratos telefônicos fornecidos pelas operadoras. Foram geradas soluções aleatórias a fim de comparar os resultados e percebeu-se que a escolha guiada de indivíduos pode ser de grande auxílio nas interceptações.

Dentre as contribuições deste trabalho, podemos citar:

- Um modelo matemático para a seleção de alvos a serem monitorados;
- Uma forma de quantificar numericamente quanto da rede formada pela organização criminosa e seus interlocutores está sendo acompanhada;
- Um algoritmo de que maximize o peso da  $k$ -cobertura de vértices do grafo formado pelos indivíduos monitorados e seus interlocutores, cujo objetivo é sugerir indivíduos a serem monitorados no ciclo de interceptação seguinte.

Para trabalhos futuros, pode-se procurar alternativas para a função de otimização, considerando outras variáveis como o tempo (em horas) disponível para a análise.

Além disso, pode-se avaliar outras formas de inserir os dados para criar o grafo, além daqueles oriundos da interceptação, como por exemplo Boletins de Ocorrência, interações de indivíduos em redes sociais ou ainda extratos telemáticos — agenda de contatos (Google, Whatsapp, Instagram, etc.) do indivíduo .

Por fim, cabe destacar que, de maneira nenhuma, o papel humano será substituído pelo algoritmo, que busca apenas auxiliar a tomada de decisão.

## Referências bibliográficas

- Barabási, A. & Albert, A.** (1999), *Emergence of scaling in random networks*, Journal Science.
- Barabási, A.-L.** (2003), *Linked: The new science of networks*, Perseus Books Group.
- Barabási, A.-L.** (2010), *Bursts: the hidden patterns behind everything we do, from email to bloody crusades*, Penguin.
- Barabási, A.-L.** (2016), *Network science*, Cambridge University Press.
- BRASIL** (1996), 'Lei nº 9.296, de 24 de julho de 1996'. [Acesso em: 03/10/2021].
- BRASIL** (2013), 'Lei nº 12.850, de 02 de agosto de 2013'. [Acesso em: 03/10/2021].
- Cavallaro, L., Bagdasar, O., De Meo, P., Fiumara, G. & Liotta, A.** (2021), 'Graph and network theory for the analysis of criminal networks'.
- da Cunha, B. R.** (2021), *Criminofísica: a ciência das interações criminais*, Editora Buqui.
- Daguspta, S., Papadimitriou, C. H. & e Vazirani, U. V.** (2006), *Algorithms*, AMGH.
- Englefield, A.** (2022), 'How to use graph technology to identify criminal activities in call records?', <https://linkurious.com/blog/how-to-use-phone-calls-and-network-analysis-to-identify-criminals/>. Online, acessado em fevereiro de 2022.
- Erdős, P. & Rényi, A.** (1959), 'On random graphs i.', *Publicationes Mathematicae* (6), 90–297.
- Feofiloff, P., Kohayakawa, Y. & Wakabayashi, Y.** (2011), 'Uma introdução sucinta à teoria dos grafos', <https://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf>. Online, acessado em novembro de 2021.
- Garey, M. & Johnson, D.** (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman.

- Gavião, L., Sant’Anna, A., Garcia, P., Silva, L., Kostin, S. & Lima, G.** (2021), ‘Multi-criteria decision support to criminology by graph theory and composition of probabilistic preferences’, *Brazilian Operations Research Society* (41), 1–44.
- Karp, R.** (1972), Reducibility among combinatorial problems, in R. Miller & J. Thatcher, eds, ‘Complexity of Computer Computations’, Plenum Press, pp. 85–103.
- Kleinberg, J. & Tardos, E.** (2006), *Algorithm Design*.
- Manurangsi, P.** (2018), A Note on Max k-Vertex Cover: Faster FPT-AS, Smaller Approximate Kernel and Improved Approximation, in J. T. Fineman & M. Mitzenmacher, eds, ‘2nd Symposium on Simplicity in Algorithms (SOSA 2019)’, Vol. 69 of *OpenAccess Series in Informatics (OASICS)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 15:1–15:21.
- Protti, F.** (2021), ‘Uma breve introdução sobre aplicações de grafos’, <http://www2.ic.uff.br/~fabio/Palestra-Aplicacoes-Grafos.pdf>. Online, acessado em outubro de 2021.