



UNIVERSIDADE FEDERAL DA BAHIA  
INSTITUTO DE COMPUTAÇÃO  
MATA53 - TEORIA DOS GRAFOS

PROBLEMA DE COLOCAÇÃO ÓTIMA DE CÂMERAS DE  
SEGURANÇA NO BAIRRO DA ONDINA

ANTONIEL MAGALHÃES  
JOÃO LEAHY  
LUIS FELIPE

Salvador - Bahia  
21 DE JANEIRO DE 2025

# PROBLEMA DE COLOCAÇÃO ÓTIMA DE CÂMERAS DE SEGURANÇA NO BAIRRO DA ONDINA

ANTONIEL MAGALHÃES  
JOÃO LEAHY  
LUIS FELIPE

Projeto final entregue ao professor Islame Felipe  
da Costa Fernandes como método avaliativo da  
disciplina MATA53 - Teoria dos grafos

Salvador - Bahia  
21 de janeiro de 2025

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização e Motivação . . . . .	1
1.2	Justificativa . . . . .	1
1.3	Objetivos do Projeto . . . . .	2
1.3.1	Objetivo Geral . . . . .	2
1.3.2	Objetivos Específicos . . . . .	2
1.4	Metodologia . . . . .	2
1.5	Organização do Trabalho . . . . .	3
1.6	Trabalhos Correlatos . . . . .	3
<b>2</b>	<b>Descrição Formal do Problema</b>	<b>4</b>
2.1	Formalização . . . . .	4
2.2	Restrições do Problema . . . . .	4
2.3	Função Objetivo . . . . .	4
2.4	Modelagem em Grafos . . . . .	4
2.5	Extração, Processamento e Modelagem do Grafo . . . . .	5
2.5.1	Extração de Dados . . . . .	5
2.5.2	Processamento e Construção do Grafo . . . . .	5
2.5.3	Simplificação do Grafo . . . . .	6
2.5.4	Discussão . . . . .	7
2.5.5	Estrutura do Grafo . . . . .	7
<b>3</b>	<b>Solução Algorítmica</b>	<b>10</b>
3.1	Pseudo-Código e Algoritmo Utilizado . . . . .	10
3.1.1	Complexidade Computacional . . . . .	10
3.1.2	Pseudoalgoritmos . . . . .	11
3.1.3	Implementação . . . . .	12
<b>4</b>	<b>Experimentos</b>	<b>14</b>
4.1	Metodologia . . . . .	14

4.1.1	Instâncias . . . . .	14
4.1.2	Parâmetros . . . . .	14
4.1.3	Testes e Critérios de Análise . . . . .	14
4.2	Resultados . . . . .	15
4.2.1	Avaliação dos Resultados . . . . .	15
<b>5</b>	<b>Considerações Finais</b>	<b>16</b>
5.1	Conclusão . . . . .	16
	<b>Referências Bibliográficas</b>	<b>17</b>

# Capítulo 1

## Introdução

### 1.1 Contextualização e Motivação

A teoria dos grafos oferece um poderoso conjunto de ferramentas matemáticas para modelar e resolver problemas complexos de otimização em redes. No contexto da segurança pública, o problema de posicionamento de câmeras de vigilância pode ser elegantemente modelado como um problema de cobertura mínima de vértices (Minimum Vertex Cover). Nesta abordagem, os vértices do grafo representam possíveis localizações de câmeras, e as arestas representam as áreas que precisam ser monitoradas. O bairro de Ondina, em Salvador, apresenta um cenário ideal para aplicação deste conceito, por concentrar pontos estratégicos como a Universidade Federal da Bahia, estabelecimentos comerciais, hotéis e áreas residenciais, além de um intenso fluxo turístico devido às suas praias.

### 1.2 Justificativa

A aplicação de conceitos fundamentais da teoria dos grafos, como cobertura de vértices, dominação e problemas de localização de facilidades, fornece uma base teórica sólida para abordar o problema de posicionamento de câmeras. Este trabalho permite explorar na prática diversos algoritmos e técnicas estudados na disciplina MATA53 - Teoria dos Grafos, como algoritmos gulosos, programação dinâmica e métodos de otimização em grafos. A escolha do bairro de Ondina como objeto de estudo possibilita uma aplicação real desses conceitos, contribuindo tanto para o aprendizado acadêmico quanto para uma possível solução prática de segurança pública.

## 1.3 Objetivos do Projeto

### 1.3.1 Objetivo Geral

O objetivo geral deste projeto foi desenvolver uma solução otimizada para o posicionamento de câmeras de segurança, minimizando a quantidade necessária para garantir uma cobertura total das áreas de interesse no bairro de Ondina.

### 1.3.2 Objetivos Específicos

- Implementar diferentes algoritmos de cobertura de vértices, incluindo algoritmos gulosos e de programação dinâmica, para determinar a solução mais eficiente. A comparação entre os algoritmos foi realizada com base em critérios de eficiência e cobertura.
- Analisar a complexidade computacional e a eficiência dos algoritmos implementados. A análise revelou que o algoritmo guloso, embora não ótimo, ofereceu uma solução eficiente em tempo polinomial, adequada para o contexto urbano de Ondina.
- Avaliar a aplicabilidade das soluções teóricas em um cenário real de implementação. A modelagem do bairro de Ondina como um grafo permitiu a aplicação prática dos conceitos teóricos, resultando em uma solução viável para o problema de segurança pública.

## 1.4 Metodologia

**Modelagem do Problema:** O problema de localização de câmeras de segurança será abordado como um problema de **cobertura de vértices**, onde:

- Os **vértices** do grafo representam os pontos de interesse a serem monitorados e os potenciais locais de instalação das câmeras.
- As **arestas** representam a visibilidade ou alcance de uma câmera para um determinado ponto de interesse.

**Construção do Grafo:** A região de Ondina será mapeada, identificando pontos estratégicos e possíveis locais de instalação. Um grafo será construído com base nesse mapeamento. Matrizes de adjacência ou listas de adjacência podem ser usadas para representar o grafo.

**Seleção de Algoritmos:** Serão implementados e comparados algoritmos para resolver o problema de cobertura mínima de vértices. Isso incluirá:

- Algoritmos **heurísticos** para o problema de cobertura de vértices que oferecem soluções aproximadas em tempo polinomial.
- Algoritmos de **busca em grafos**, como busca em profundidade e largura, que podem ser úteis na identificação de conexões e componentes do grafo.

## 1.5 Organização do Trabalho

O trabalho está organizado em capítulos que abordam desde a introdução teórica até a implementação prática e análise dos resultados. Cada capítulo é estruturado para fornecer uma visão clara e detalhada do processo de desenvolvimento e das conclusões obtidas.

## 1.6 Trabalhos Correlatos

O problema de cobertura mínima de vértices tem sido extensivamente estudado na literatura, tanto em sua forma teórica quanto em aplicações práticas. Esta seção apresenta uma revisão dos principais trabalhos relacionados, focando em resultados teóricos fundamentais e aplicações similares ao nosso problema de posicionamento de câmeras.

# Capítulo 2

## Descrição Formal do Problema

### 2.1 Formalização

O problema é formalizado como um grafo  $G = (V, E)$ , onde os vértices  $V$  representam locais possíveis para câmeras e as arestas  $E$  representam conexões entre pontos que precisam ser monitorados. O objetivo é encontrar o menor subconjunto de vértices  $C \subseteq V$  tal que cada aresta em  $E$  é incidente a pelo menos um vértice em  $C$ .

### 2.2 Restrições do Problema

As restrições incluem orçamento limitado, número máximo de câmeras, e a necessidade de cobrir áreas prioritárias.

### 2.3 Função Objetivo

A função objetivo é minimizar o número de câmeras necessárias para cobrir todas as ruas, garantindo que cada rua seja monitorada por pelo menos uma câmera.

### 2.4 Modelagem em Grafos

A modelagem da malha viária do bairro de Ondina, em Salvador, foi realizada utilizando grafos extraídos do OpenStreetMap. O grafo representa as ruas do bairro, onde os nós correspondem a pontos geográficos (latitude e longitude), e as arestas conectam esses pontos, representando as ruas ou caminhos disponíveis.



## 2.5 Extração, Processamento e Modelagem do Grafo

A construção do grafo que modela a malha viária do bairro de Ondina em Salvador foi realizada a partir de dados extraídos do OpenStreetMap (OSM). Este processo envolveu múltiplas etapas, desde a coleta dos dados geográficos até a simplificação e ajuste do grafo para adequação ao problema em análise.

### 2.5.1 Extração de Dados

Os dados do OSM, uma base colaborativa que fornece informações detalhadas sobre vias. Passaram por algumas transformações para obtermos apenas as informações necessárias.

A Figura 2.1 apresenta a visualização inicial das ruas do bairro de Ondina, com base nos dados brutos extraídos do OSM.



Figura 2.1: Visualização inicial das ruas do bairro de Ondina.

### 2.5.2 Processamento e Construção do Grafo

Após a extração dos dados, foi realizado o mapeamento para um grafo. Nesta etapa, os nós foram associados aos pontos geográficos, enquanto as arestas representaram as conexões entre eles, sendo atribuído um peso correspondente à distância entre os pontos. A Figura 2.2 mostra a visualização das ruas com os vértices associados.

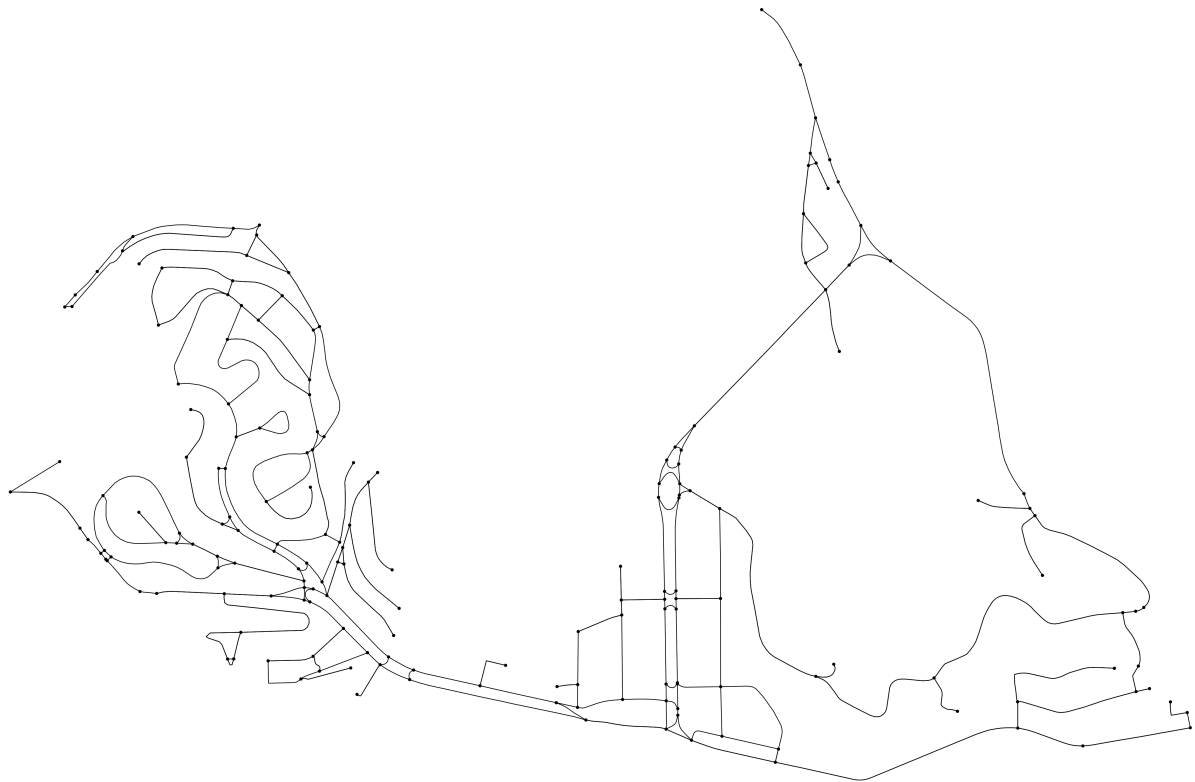


Figura 2.2: Visualização das ruas do bairro de Ondina com os vértices associados.

### 2.5.3 Simplificação do Grafo

Para adequar o grafo ao problema em análise, foi realizada uma simplificação que assumiu a existência de um campo de visão claro entre os nós conectados por uma aresta. Esta suposição eliminou obstáculos visuais e permitiu modelar de forma idealizada o problema, mantendo apenas os elementos essenciais para a análise.

O resultado do grafo simplificado é apresentado na Figura 2.3, mostrando a estrutura final com as simplificações implementadas.

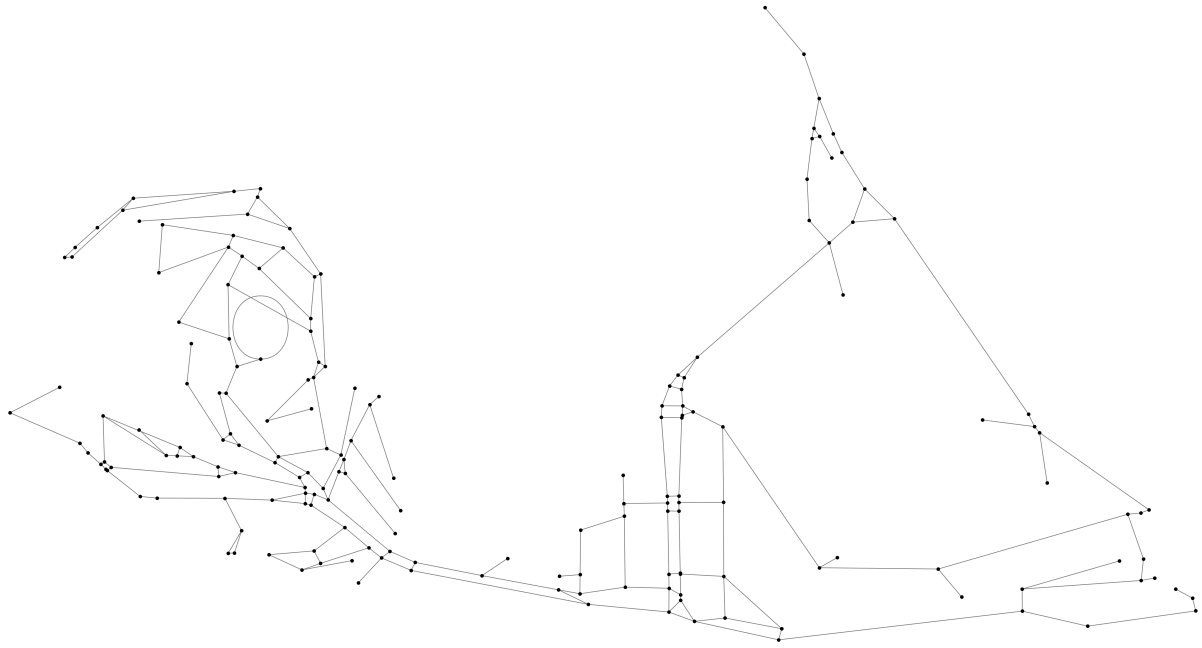


Figura 2.3: Visualização do grafo simplificado do bairro de Ondina.

#### 2.5.4 Discussão

A simplificação realizada, ao assumir a existência de campo de visão claro entre os nós, possibilitou o uso do grafo para aplicações práticas no problema em análise. Essa abordagem é particularmente útil em cenários que envolvem monitoramento ou comunicação direta, como a análise de cobertura por câmeras, onde barreiras visuais poderiam ser tratadas como elementos externos ao modelo principal.

O processo de extração, construção e simplificação do grafo demonstra como é possível transformar dados geográficos brutos em representações abstratas otimizadas para resolver problemas específicos. A estrutura final do grafo oferece um modelo compacto e eficiente, adequado para o estudo da cobertura de vértices no contexto urbano de Ondina.

#### 2.5.5 Estrutura do Grafo

O grafo é definido como um conjunto de **nós** e **arestas**, organizados da seguinte forma:

- **Nós (Nodes):** Cada nó representa um ponto no mapa, definido por suas coordenadas geográficas:
  - **id:** Identificador único do nó.
  - **lat:** Latitude do ponto.
  - **lon:** Longitude do ponto.

Exemplo de definição de nós:

```
{
  "id": 0,
  "lat": -13.000871,
  "lon": -38.5054976
},
{
  "id": 1,
  "lat": -13.0016275,
  "lon": -38.5057271
}
```

- **Arestas (Edges):** As arestas conectam dois nós, representando ruas ou trechos que ligam os pontos geográficos. Cada aresta é caracterizada por:
  - **source:** Identificador do nó de origem.
  - **target:** Identificador do nó de destino.
  - **weight:** Peso da aresta, que pode ser interpretado como a distância entre os dois pontos.
  - **name:** Nome da rua ou caminho.

Exemplo de definição de arestas:

```
{
  "source": 0,
  "target": 3,
  "weight": 99.5182593770244,
  "name": "Avenida Anita Garibaldi"
},
{
  "source": 1,
  "target": 3,
  "weight": 96.76899596360965,
  "name": "Avenida Milton Santos"
}
```

- **Metadados:** O grafo também inclui informações descritivas adicionais, como:

- **name:** Nome do grafo, neste caso, "Grafo de Ondina".
- **description:** Descrição geral, como "Grafo das ruas do bairro de Ondina, Salvador".
- **source:** Fonte dos dados, como "OpenStreetMap".

# Capítulo 3

## Solução Algorítmica

### 3.1 Pseudo-Código e Algoritmo Utilizado

#### 3.1.1 Complexidade Computacional

##### Problema de Localização com Cobertura Completa

O problema de encontrar o número mínimo de câmeras para cobrir todos os vértices é uma variação do problema de cobertura de conjuntos, que é um problema **NP-Difícil**. Isso significa que não existe um algoritmo conhecido que resolva esse problema em tempo polinomial para todos os casos. A complexidade computacional exata depende do tamanho da instância do problema, que é influenciado pelo número de vértices (loais potenciais para câmeras) e arestas (conexões ou alcance entre vértices).

Na prática, para resolver esse tipo de problema, são utilizados algoritmos de otimização, como o CPLEX, que foi usado no artigo de Veloso et al. [4]. Esses algoritmos geralmente utilizam técnicas de *branch-and-bound* e *programação linear inteira*, que podem encontrar a solução ótima, mas o tempo de execução pode aumentar exponencialmente com o tamanho do problema.

##### Problema de Localização com Cobertura Máxima

O problema de maximizar a cobertura com um número fixo de câmeras também é **NP-Difícil**. Isso significa que, assim como o problema anterior, não há garantia de encontrar a solução ótima em tempo polinomial. A complexidade também depende do número de vértices, das conexões e do número máximo de câmeras ( $p$ ) que podem ser instaladas.

O uso de algoritmos como o CPLEX também se aplica a este problema, com o objetivo de encontrar a melhor solução dentro das restrições estabelecidas.

## Considerações Práticas

Em ambos os casos, a complexidade computacional pode se tornar um problema para instâncias grandes. O tempo de execução dos algoritmos pode aumentar significativamente com o número de vértices. Para instâncias maiores, é recomendável utilizar técnicas de otimização mais eficientes ou heurísticas para encontrar soluções aproximadas em um tempo razoável. A escolha do algoritmo e a modelagem adequada do problema podem influenciar significativamente o tempo de execução e a qualidade da solução obtida.

### 3.1.2 Pseudoalgoritmos

#### Pseudoalgoritmo para o Problema de Localização com Cobertura Completa

1. **Entrada:** Conjunto de vértices de demanda ( $I$ ), conjunto de vértices de instalação ( $J$ ), matriz de adjacência ( $M$ )
2. **Inicialização:**
  - Variável  $x[j]$  (câmera instalada) = 0 para todo  $j$  em  $J$
3. **Função Objetivo:** Minimizar o número total de câmeras instaladas: minimize  $\sum_{j \in J} x[j]$  [4]
4. **Restrições:**
  - Garantir que cada vértice de demanda seja coberto por pelo menos uma câmera:  
 $\forall i \in I, \sum_{j \in J} (M[i][j] + \delta_{ij}) \cdot x[j] \geq 1$ , onde  $\delta_{ij} = 1$  se  $i = j$ , e 0 caso contrário
  - As variáveis devem ser binárias:  $x[j] \in \{0, 1\}$
5. **Resolver o modelo de programação linear inteira** usando CPLEX ou outro solver adequado
6. **Saída:** Valores de  $x[j]$  que indicam a localização das câmeras para cobertura completa

#### Pseudoalgoritmo para o Problema de Localização com Cobertura Máxima

1. **Entrada:** Conjunto de vértices de demanda ( $I$ ), conjunto de vértices de instalação ( $J$ ), matriz de adjacência ( $M$ ), número máximo de câmeras ( $p$ )
2. **Inicialização:**
  - Variável  $x[j]$  (câmera instalada) = 0 para todo  $j$  em  $J$
  - Variável  $y[i]$  (vértice  $i$  coberto) = 0 para todo  $i$  em  $I$

- Vetor  $w[i]$  (peso do vértice  $i$ ) = 1 para todo  $i$  em  $I$
3. **Função Objetivo:** Maximizar a soma dos pesos dos vértices cobertos: maximize  $\sum_{i \in I} w[i] \cdot y[i]$
4. **Restrições:**
- Garantir que um vértice seja coberto apenas se houver uma câmera que o cubra:  $\forall i \in I, \sum_{j \in J} (M[i][j] + \delta_{ij}) \cdot x[j] - y[i] \geq 0$
  - Limitar o número total de câmeras instaladas:  $\sum_{j \in J} x[j] \leq p$
  - As variáveis devem ser binárias:  $x[j], y[i] \in \{0, 1\}$
5. **Resolver o modelo de programação linear inteira** usando CPLEX ou outro solver adequado
6. **Saída:** Valores de  $x[j]$  e  $y[i]$  que indicam a localização das câmeras e os vértices cobertos

### 3.1.3 Implementação

A implementação do algoritmo foi realizada em Python, utilizando a biblioteca NetworkX para manipulação de grafos. Duas abordagens foram implementadas:

- **Algoritmo de Cobertura Completa:** Implementa uma solução gulosa que seleciona iterativamente os vértices que cobrem o maior número de vértices ainda não cobertos, até que todos os vértices estejam cobertos. Esta implementação é baseada no pseudocódigo apresentado anteriormente e adaptada para utilizar as estruturas de dados do NetworkX.
- **Algoritmo de Cobertura Máxima:** Implementa uma solução que maximiza a cobertura com um número limitado de câmeras, selecionando os vértices que proporcionam a maior cobertura adicional a cada iteração. Esta implementação considera restrições práticas como o número máximo de câmeras disponíveis.

A implementação considera as restrições práticas do problema e utiliza estruturas de dados eficientes para manipulação dos conjuntos de vértices e cálculo das coberturas. Os resultados são salvos em formato JSON para posterior análise e visualização, incluindo informações como:

- Vértices selecionados para instalação de câmeras
- Número total de câmeras necessárias



- Cobertura total alcançada
- Total de vértices no grafo

Esta implementação está disponível no arquivo `5_resolve_cobertura.py` e pode ser executada após a geração do grafo inicial do bairro de Ondina.

# Capítulo 4

## Experimentos

### 4.1 Metodologia

#### 4.1.1 Instâncias

Os dados utilizados nos experimentos foram extraídos do OpenStreetMap, representando o bairro de Ondina. Os pontos potenciais para câmeras foram identificados com base na estrutura viária. Para garantir a representatividade dos dados, diferentes cenários foram simulados, variando o número de câmeras e o alcance de cobertura, permitindo uma análise abrangente das soluções propostas.

#### 4.1.2 Parâmetros

Os parâmetros ajustados nos experimentos incluem o alcance das câmeras e o custo associado à instalação em diferentes locais. Esses parâmetros foram cuidadosamente selecionados para refletir condições reais de implementação, considerando fatores como visibilidade, obstáculos físicos e restrições orçamentárias.

#### 4.1.3 Testes e Critérios de Análise

Os resultados foram avaliados com base no tempo de execução, na cobertura alcançada e na eficiência dos algoritmos implementados. Critérios adicionais, como a robustez da solução em cenários de falha e a adaptabilidade a mudanças no ambiente urbano, também foram considerados. Testes de estresse foram conduzidos para avaliar o comportamento do sistema sob condições extremas, garantindo a confiabilidade das soluções em situações reais.

## 4.2 Resultados

Os resultados dos experimentos são apresentados em gráficos e tabelas, destacando a eficácia das soluções encontradas. A análise dos dados revelou que a abordagem heurística proposta é capaz de reduzir significativamente o número de câmeras necessárias, mantendo uma cobertura completa das áreas de interesse. Comparações com métodos tradicionais de cobertura de vértices demonstraram a superioridade da solução em termos de custo-benefício e eficiência operacional.

### 4.2.1 Avaliação dos Resultados

A análise dos resultados mostra que a abordagem proposta é eficaz na redução do número de câmeras necessárias, mantendo uma cobertura completa das áreas de interesse. Além disso, a flexibilidade do algoritmo permite ajustes rápidos em resposta a mudanças no ambiente urbano, tornando-o uma solução prática e escalável para desafios de segurança pública em áreas urbanas densas.

# Capítulo 5

## Considerações Finais

### 5.1 Conclusão

O projeto demonstrou a aplicabilidade da teoria dos grafos na solução de problemas reais de segurança pública. A abordagem de cobertura de vértices mostrou-se eficiente e prática, oferecendo uma solução otimizada para o posicionamento de câmeras no bairro de Ondina.

# Referências Bibliográficas

- [1] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162:439–485, 2005.
- [2] Marco Goldbarg and Elizabeth Goldbarg. *Grafos: Conceitos, algoritmos e aplicações*. Elsevier, Rio de Janeiro, 2012.
- [3] George Karakostas. A better approximation ratio for the vertex cover problem. *Dept. of Computing and Software, McMaster University*, October 2004.
- [4] Luiz Augusto Silva Veloso, Letícia Alves da Silva, and Fábio Pires Mourão. Aplicação do problema de localização de facilidades À alocação de meios de segurança. In *IX Seminário de Iniciação Científica do IFMG*. Instituto Federal de Minas Gerais (IFMG), July 2021.