



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE XXXXXXXXX
DEPARTAMENTO DE XXXXXXXXXX
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

BILLY GRAHAN ALVES RODRIGUES

ALGORITMOS DE BUSCA PARA DETERMINAR ROTAS ENTRE ATIVOS NA
ZELADORIA PÚBLICA NO CONTEXTO DE CIDADES INTELIGENTES

RUSSAS

2025

BILLY GRAHAN ALVES RODRIGUES

ALGORITMOS DE BUSCA PARA DETERMINAR ROTAS ENTRE ATIVOS NA
ZELADORIA PÚBLICA NO CONTEXTO DE CIDADES INTELIGENTES

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Centro de XXXXXXXX da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de bacharel em Ciência da
Computação.

Orientador: Prof. Dr. Mayrton Dias de
Queiroz

RUSSAS

2025

BILLY GRAHAN ALVES RODRIGUES

ALGORITMOS DE BUSCA PARA DETERMINAR ROTAS ENTRE ATIVOS NA
ZELADORIA PÚBLICA NO CONTEXTO DE CIDADES INTELIGENTES

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Centro de XXXXXXXX da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de bacharel em Ciência da
Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Mayrton Dias de Queiroz (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. XXXXXXXX XXXXXXXX XXXXXXXX
Universidade do Membro da Banca Dois (SIGLA)

Prof. Dr. XXXXXXXX XXXXXXXX XXXXXXXX
Universidade do Membro da Banca Três (SIGLA)

Prof. Dr. XXXXXXXX XXXXXXXX XXXXXXXX
Universidade do Membro da Banca Quatro (SIGLA)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

“O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado.”

(Ariano Suassuna)

RESUMO

A gestão eficiente de ativos públicos municipais constitui um desafio logístico para as administrações públicas, especialmente quando demanda visitas periódicas para inspeção e manutenção de equipamentos urbanos distribuídos por diferentes regiões da cidade. Este trabalho propõe o desenvolvimento de uma solução algorítmica baseada em técnicas heurísticas para otimização de rotas de visitação a ativos públicos, aplicando o Problema do Caixeiro Viajante ao contexto da gestão pública municipal. Foi realizada uma revisão sistemática da literatura para identificar as principais abordagens metodológicas e algoritmos utilizados em problemas de otimização de rotas em ambientes urbanos e cidades inteligentes. A metodologia envolveu a caracterização do problema, a identificação de técnicas heurísticas adequadas e a análise comparativa de algoritmos utilizando dados de ativos municipais. Os resultados demonstram a viabilidade da aplicação de técnicas de otimização para redução de custos operacionais, economia de tempo de deslocamento e melhoria na eficiência dos serviços prestados à população. A solução proposta insere-se no contexto das cidades inteligentes, contribuindo para a modernização da gestão pública em municípios de pequeno e médio porte, onde as limitações orçamentárias tornam essencial o uso eficiente dos recursos disponíveis. Este trabalho evidencia que a otimização de rotas constitui ferramenta estratégica para maximizar a eficiência operacional da administração pública, promovendo maior agilidade no atendimento às demandas de zeladoria urbana e fortalecendo a qualidade dos serviços municipais.

Palavras-chave: Otimização de rotas. Problema do Caixeiro Viajante. Gestão de ativos públicos. Heurísticas. Cidades inteligentes.

ABSTRACT

Efficient management of municipal public assets represents a logistical challenge for public administrations, especially when it requires periodic visits for the inspection and maintenance of urban equipment distributed across different areas of the city. This work proposes the development of an algorithmic solution based on heuristic techniques for optimizing visitation routes to public assets, applying the Traveling Salesman Problem to the context of municipal public management. A systematic literature review was conducted to identify the main methodological approaches and algorithms used in route optimization problems in urban environments and smart cities. The methodology involved characterizing the problem, identifying suitable heuristic techniques, and performing a comparative analysis of algorithms using municipal asset data. The results demonstrate the feasibility of applying optimization techniques to reduce operational costs, save travel time, and improve the efficiency of services provided to the population. The proposed solution aligns with the smart city framework, contributing to the modernization of public management in small and medium-sized municipalities, where budget constraints make the efficient use of available resources essential. This work highlights that route optimization is a strategic tool for maximizing the operational efficiency of public administration, promoting greater agility in addressing urban maintenance demands and strengthening the quality of municipal services.

Keywords: Route optimization. Traveling Salesman Problem. Public asset management. Heuristics. Smart cities.

LISTA DE FIGURAS

| | | |
|----------|--|----|
| Figura 1 | – Duas representações de um grafo não dirigido. (a) Um grafo não dirigido G com cinco vértices e sete arestas. (b) Uma representação de G por lista de adjacências. (c) A representação de G por matriz de adjacências | 23 |
| Figura 2 | – Duas representações de um grafo dirigido. (a) Um grafo dirigido G com seis vértices e oito arestas. (b) Uma representação de G por lista de adjacências. (c) A representação de G por matriz de adjacências | 23 |
| Figura 3 | – Exemplo de execução da Busca em Largura (BFS) em um grafo | 25 |
| Figura 4 | – Exemplo de execução da Busca em Profundidade (DFS) em um grafo | 26 |
| Figura 5 | – Exemplo de aplicação do algoritmo A^* no problema de roteamento entre cidades da Romênia | 28 |
| Figura 6 | – Resultados da Questão de Pesquisa 1 (QP1). | 32 |
| Figura 7 | – Resultados da Questão de Pesquisa 2 (QP2). | 33 |
| Figura 8 | – Gráfico de tensão considerando a impedância humana | 39 |
| Figura 9 | – Produção anual das dissertações de mestrado e teses de doutorado entre os anos de 1990 e 2008 | 40 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Quantidade de artigos em cada base de dados. | 31 |
| Tabela 2 – Notas dos participantes nas avaliações A, B e C | 40 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------------|--|
| <i>API</i> | <i>Application Programming Interface</i> / Interface de Programação de Aplicações |
| <i>BFS</i> | <i>Breadth-First Search</i> / Busca em Largura |
| <i>CIM</i> | <i>City Information Modeling</i> / Modelagem de Informações Urbanas |
| <i>DAG</i> | <i>Directed Acyclic Graph</i> / Grafo Acíclico Direcionado |
| <i>DFS</i> | <i>Depth-First Search</i> / Busca em Profundidade |
| <i>FIFO</i> | <i>First In, First Out</i> / Primeiro a Entrar, Primeiro a Sair |
| <i>GRASP</i> | <i>Greedy Randomized Adaptive Search Procedure</i> / Procedimento de Busca Adaptativa Gulosa Aleatória |
| <i>ITS</i> | <i>Intelligent Transportation Systems</i> / Sistemas Inteligentes de Transporte |
| <i>IoT</i> | <i>Internet of Things</i> / Internet das Coisas |
| <i>LIFO</i> | <i>Last In, First Out</i> / Último a Entrar, Primeiro a Sair |
| <i>TSP</i> | <i>Traveling Salesman Problem</i> / Problema do Caixeiro Viajante |
| <i>UCS</i> | <i>Uniform-Cost Search</i> / Busca de Custo Uniforme |
| AG | Algoritmos Genéticos |
| OX | Cruzamento de Ordem |
| QP | Questões de Pesquisa |

LISTA DE SÍMBOLOS

| | |
|---------------|---|
| b | Fator de ramificação |
| C^* | Custo da solução ótima |
| E | Conjunto de arestas de um grafo |
| e | Aresta de um grafo |
| $f(n)$ | Função de avaliação no algoritmo A* ($f(n) = g(n) + h(n)$) |
| G | Grafo, definido como $G = (V, E)$ |
| $g(n)$ | Custo do caminho da origem até o vértice n |
| $h(n)$ | Função heurística que estima o custo de n até o objetivo |
| n | Vértice ou nó em um grafo |
| O | Notação Big-O para complexidade de tempo ou espaço |
| u | Vértice em um grafo |
| v | Vértice em um grafo |
| V | Conjunto de vértices de um grafo |
| w | Função peso que atribui valores às arestas ($w : E \rightarrow \mathbb{R}$) |
| ε | Menor custo de aresta |

SUMÁRIO

| | | |
|----------------|--|-----------|
| 1 | INTRODUÇÃO | 13 |
| 1.1 | Objetivos | 13 |
| 1.1.1 | <i>Objetivos específicos</i> | 13 |
| 1.2 | Justificativa | 14 |
| 1.3 | Organização do Trabalho | 15 |
| 2 | CONTEXTUALIZAÇÃO | 16 |
| 2.1 | Cidades inteligentes | 16 |
| 2.1.1 | <i>Eixos das Cidades Inteligentes</i> | 16 |
| 2.1.1.1 | <i>Governança Inteligente</i> | 16 |
| 2.1.1.2 | <i>Mobilidade Urbana Inteligente</i> | 17 |
| 2.1.1.3 | <i>Urbanismo e Planejamento Urbano</i> | 17 |
| 2.2 | Gestão de ativos públicos | 17 |
| 2.2.1 | <i>Zeladoria Urbana</i> | 18 |
| 2.3 | Métodos de Otimização | 19 |
| 2.3.1 | <i>Métodos Exatos</i> | 19 |
| 2.3.2 | <i>Heurísticas</i> | 20 |
| 2.3.3 | <i>Meta-heurísticas</i> | 21 |
| 2.3.3.1 | <i>Algoritmos Genéticos</i> | 21 |
| 2.4 | Teoria dos Grafos | 22 |
| 2.4.1 | <i>Grafos Direcionados e Não-Direcionados</i> | 22 |
| 2.4.2 | <i>Grafos Ponderados</i> | 23 |
| 2.4.3 | <i>Conceitos Adicionais</i> | 24 |
| 2.4.4 | <i>Representação Computacional</i> | 24 |
| 2.5 | Algoritmos de Busca | 24 |
| 2.5.1 | <i>Busca em Largura (BFS)</i> | 25 |
| 2.5.2 | <i>Busca em Profundidade (DFS)</i> | 25 |
| 2.5.3 | <i>Busca de Custo Uniforme</i> | 26 |
| 2.5.4 | <i>Algoritmo A* (A Estrela)</i> | 27 |
| 3 | REVISÃO DA LITERATURA | 29 |
| 3.1 | Contextualização | 29 |

| | | |
|--------------|---|----|
| 3.2 | Planejamento | 29 |
| 3.2.1 | <i>Definição de Questões de Pesquisa</i> | 29 |
| 3.2.2 | <i>Definição da string de busca</i> | 30 |
| 3.2.3 | <i>Definições de bases de dados</i> | 30 |
| 3.2.4 | <i>Definição dos critérios de inclusão e exclusão</i> | 30 |
| 3.3 | Condução | 31 |
| 3.4 | Resultados | 31 |
| 3.5 | Considerações Finais | 33 |
| 4 | METODOLOGIA | 35 |
| 4.1 | Exemplo de alíneas | 35 |
| 4.2 | Usando Fórmulas Matemáticas | 36 |
| 4.3 | Usando Código-fonte | 38 |
| 4.4 | Usando Teoremas, Proposições, etc | 38 |
| 4.5 | Usando Questões | 38 |
| 5 | RESULTADOS | 39 |
| 5.1 | Resultados do Experimento A | 39 |
| 5.2 | Resultados do Experimento B | 40 |
| 6 | CONCLUSÕES E TRABALHOS FUTUROS | 42 |
| | REFERÊNCIAS | 43 |
| | APÊNDICES | 45 |
| | APÊNDICE A – Exemplo de apêndice | 45 |
| | APÊNDICE B – Questionário utilizado para... | 46 |
| | APÊNDICE C – Códigos-fontes utilizados para... | 47 |
| | APÊNDICE D – <i>IEEE CEFC 2016</i> | 49 |
| | ANEXOS | 51 |
| | ANEXO A – Exemplo de um anexo | 51 |
| | ÍNDICE | 52 |

1 INTRODUÇÃO

A gestão eficiente dos ativos públicos constitui um desafio constante para as administrações municipais, especialmente quando demanda visitas periódicas para inspeção, manutenção ou levantamento de informações. Esses ativos — como praças, escolas, unidades de saúde, postes de iluminação, equipamentos urbanos, entre outros — encontram-se com algum Problema por diferentes regiões da cidade, exigindo um planejamento logístico criterioso que minimize o tempo de deslocamento e os custos operacionais das equipes responsáveis.

No contexto das cidades inteligentes, a otimização de rotas torna-se fundamental para a eficiência operacional dos serviços públicos. A ausência de planejamento adequado resulta em desperdício de recursos, aumento do tempo de resposta às demandas da população e ineficiência na prestação de serviços essenciais. Este cenário é particularmente crítico em municípios de pequeno e médio porte, onde as limitações orçamentárias e de infraestrutura são mais acentuadas.

O desafio consiste em determinar a rota mais eficiente para que profissionais da prefeitura percorram todos os pontos de visitação necessários e retornem ao ponto de origem, minimizando a distância total percorrida e, conseqüentemente, o tempo e os custos operacionais.

1.1 Objetivos

Diante do contexto apresentado, o objetivo geral deste trabalho consiste em desenvolver e avaliar um algoritmo capaz de gerar rotas otimizadas para a visitação de ativos públicos por profissionais da administração municipal, aplicando técnicas heurísticas para resolver o Problema do Caixeiro Viajante (*Traveling Salesman Problem – Traveling Salesman Problem / Problema do Caixeiro Viajante (TSP)*) adaptado à realidade da gestão pública.

1.1.1 Objetivos específicos

Para alcançar o objetivo geral, estabelecem-se os seguintes objetivos específicos:

- Realizar uma revisão sistemática da literatura para identificar os principais trabalhos relacionados à otimização de rotas em contextos urbanos e cidades inteligentes, bem como as técnicas e algoritmos empregados;
- Identificar o Problema do Caixeiro Viajante e suas variantes, contextualizando suas aplicações na gestão de ativos públicos municipais;

- Obter um algoritmo heurístico para geração de rotas otimizadas, considerando as especificidades da gestão pública;
- Obter uma avaliação do desempenho do algoritmo desenvolvido por meio de análise comparativa utilizando dados de ativos municipais;

1.2 Justificativa

A relevância deste trabalho fundamenta-se em múltiplas dimensões que abrangem aspectos técnicos, econômicos e sociais. Do ponto de vista técnico, o Problema do Caixeiro Viajante representa um desafio clássico da ciência da computação, classificado como NP-difícil, cuja solução exata torna-se impraticável para instâncias de grande porte. A aplicação de técnicas heurísticas permite obter soluções de boa qualidade em tempo computacional razoável, viabilizando sua implementação em contextos práticos.

No âmbito econômico, a otimização de rotas contribui diretamente para a redução de custos operacionais no setor público. A economia gerada pela minimização das distâncias percorridas reflete-se na diminuição do consumo de combustível, na redução do desgaste dos veículos e no aproveitamento mais eficiente do tempo dos profissionais. Essa otimização representa uma ferramenta estratégica para maximizar a eficiência dos recursos disponíveis.

Sob a perspectiva social, a melhoria na eficiência operacional da administração pública impacta diretamente a qualidade dos serviços prestados à população. Rotas otimizadas permitem maior agilidade no atendimento às demandas de manutenção urbana, reduzindo o tempo de resposta e aumentando a capacidade de atendimento das equipes municipais. Esse ganho de eficiência fortalece a confiança da população nas instituições públicas e contribui para a melhoria da qualidade de vida nos centros urbanos.

Além disso, este trabalho insere-se no contexto contemporâneo das cidades inteligentes, onde a aplicação de tecnologias da informação e técnicas de otimização tornam-se instrumentos essenciais para a modernização da gestão pública. A solução proposta pode ser adaptada e replicada em diversos municípios, especialmente aqueles de pequeno e médio porte, democratizando o acesso a ferramentas tecnológicas que promovam eficiência administrativa e transparência na gestão dos recursos públicos.

1.3 Organização do Trabalho

Este trabalho está estruturado em cinco capítulos, organizados de forma a proporcionar uma compreensão progressiva e sistemática do tema abordado.

O Capítulo 1 apresenta a contextualização do problema, os objetivos da pesquisa, a justificativa e a organização geral do trabalho, fornecendo uma visão panorâmica dos elementos centrais desta investigação.

O Capítulo 2 expõe a fundamentação teórica necessária para a compreensão do trabalho, abordando os conceitos de cidades inteligentes, gestão de ativos públicos, zeladoria urbana, métodos de otimização e teoria dos grafos. São apresentados os principais algoritmos de busca e técnicas heurísticas relevantes para a resolução do problema proposto.

O Capítulo 3 descreve a revisão sistemática da literatura realizada, detalhando as etapas de planejamento, condução e análise dos resultados. São apresentados os principais trabalhos relacionados à otimização de rotas em contextos urbanos e cidades inteligentes, identificando estratégias metodológicas, algoritmos empregados e lacunas de pesquisa.

O Capítulo 4 apresenta a metodologia empregada no desenvolvimento da solução, descrevendo o algoritmo proposto, os dados utilizados, os procedimentos de implementação e os critérios de avaliação de desempenho.

Por fim, o Capítulo 5 consolida as conclusões do trabalho, sintetizando os principais resultados obtidos, discutindo as limitações da pesquisa e apontando direções para trabalhos futuros.

2 CONTEXTUALIZAÇÃO

Nesta seção serão abordados os assuntos e conceitos utilizados atualmente para soluções de melhoria de bem-estar da população, como: cidades inteligentes, gestão de ativos públicos, zeladoria urbana, e os algoritmos de busca.

2.1 Cidades inteligentes

O conceito de cidades inteligentes (do inglês, *smart cities*) é definido como um ambiente urbano que utiliza ferramentas de tecnologia da informação e comunicação (TIC) para melhorar a infraestrutura urbana, a utilização inteligente dos recursos, a mobilidade, a governança e os serviços públicos, tendo como principal beneficiado o cidadão (WEISS *et al.*, 2015). Embora o conceito de cidades inteligentes seja amplamente implementado em grandes centros urbanos e metrópoles, é possível realizar sua adaptação a pequenos e médios municípios brasileiros. O uso de tecnologias em tempo real, automação de processos e a participação ativa da população possibilitam a adaptação e a melhoria da qualidade da gestão pública local (SANTIAGO, 2023).

As cidades inteligentes estruturam-se em múltiplos eixos de atuação que, integrados, promovem o desenvolvimento urbano sustentável e centrado no cidadão. Esses eixos abrangem diferentes dimensões da vida urbana e demandam soluções tecnológicas específicas para enfrentar os desafios contemporâneos das cidades (CARVALHO, 2024).

2.1.1 Eixos das Cidades Inteligentes

2.1.1.1 Governança Inteligente

A governança inteligente refere-se à aplicação de tecnologias digitais para promover a participação cidadã, transparência administrativa e eficiência na gestão pública. Neste eixo, os sistemas de informação permitem que gestores públicos tomem decisões baseadas em dados, realizem o monitoramento de serviços em tempo real e promovam a interação direta com a população (WEISS *et al.*, 2015). Ferramentas como *dashboards* interativos e plataformas de dados abertos são exemplos de aplicações que facilitam o acesso à informação e fortalecem a democracia participativa (DIAS *et al.*, 2019). A gestão orientada por dados contribui para a otimização de recursos públicos e para o planejamento estratégico de longo prazo, especialmente

em contextos de restrição orçamentária.

2.1.1.2 Mobilidade Urbana Inteligente

A mobilidade urbana inteligente integra sistemas de transporte, tecnologias de geolocalização e análise de dados para otimizar o deslocamento de pessoas e bens nas cidades. Sistemas Inteligentes de Transporte (*Intelligent Transportation Systems* / Sistemas Inteligentes de Transporte (*ITS*), do inglês *Intelligent Transportation Systems*) utilizam sensores, câmeras e algoritmos de análise para monitorar o fluxo de tráfego, identificar congestionamentos e propor rotas alternativas (COSTA, 2023). Além disso, soluções de estacionamento inteligente baseadas em sistemas ciber-físicos e agentes inteligentes contribuem para a redução do tempo de busca por vagas e diminuição da poluição urbana (BOTELHO *et al.*, 2019). A aplicação de técnicas de aprendizado de máquina permite a análise de padrões espaço-temporais da mobilidade, auxiliando no planejamento de intervenções estruturantes no sistema viário (MELONIO, 2021).

2.1.1.3 Urbanismo e Planejamento Urbano

O eixo de urbanismo inteligente envolve o uso de geotecnologias, modelagem de informações urbanas e análise espacial para o planejamento e gestão do território urbano. A adoção de metodologias como *City Information Modeling* (*City Information Modeling* / Modelagem de Informações Urbanas (*CIM*)) permite a integração de dados geoespaciais, infraestrutura urbana e informações socioeconômicas em modelos tridimensionais que auxiliam gestores no planejamento territorial e na tomada de decisões (SANTIAGO, 2023). Aplicações mobile para relato de problemas urbanos por parte da população, integradas a sistemas de geolocalização, possibilitam o mapeamento colaborativo de demandas e a priorização de intervenções no espaço urbano (SANTOS *et al.*, 2019). A simulação urbana integrada a sistemas multi-agentes e Internet das Coisas (*Internet of Things* / Internet das Coisas (*IoT*)) permite testar cenários de desenvolvimento urbano antes de sua implementação física (CASTRO *et al.*, 2020).

2.2 Gestão de ativos públicos

Os ativos públicos são recursos, direitos e serviços que são acessíveis e disponíveis para todos os membros de uma sociedade, com diversas finalidades, tais como: infraestrutura, serviços essenciais, bens comuns, patrimônio cultural e recursos naturais. A gestão de ativos

públicos consiste no conjunto de práticas e processos que visam controlar, inventariar, manter e planejar a utilização desses recursos de forma eficiente e sustentável ao longo de seu ciclo de vida.

No contexto das cidades inteligentes, a gestão de ativos públicos torna-se estratégica para garantir a qualidade dos serviços urbanos e a otimização dos recursos públicos. O gerenciamento eficaz desses ativos demanda sistemas de informação capazes de permitir o acompanhamento em tempo real, com dados precisos de localização, estado de conservação e histórico de ações realizadas. A integração de tecnologias de geolocalização, sensores *IoT* e plataformas de gerenciamento permite que gestores públicos tenham visibilidade completa sobre o patrimônio municipal (SANTIAGO, 2023).

Soluções informatizadas de gestão de ativos reduzem erros operacionais, previnem retrabalhos e promovem melhor uso dos recursos públicos (STORCK *et al.*, 2017). Isso é particularmente importante em contextos onde há limitação orçamentária, como em municípios de pequeno e médio porte. A implementação de sistemas integrados de gestão possibilita a criação de inventários digitais georreferenciados, o planejamento preventivo de manutenções e a geração de indicadores de desempenho para a tomada de decisão baseada em evidências.

Além disso, a gestão de ativos públicos no contexto de cidades inteligentes permite a integração com sistemas de participação cidadã, onde a população pode reportar problemas relacionados à infraestrutura urbana através de aplicativos móveis (SANTOS *et al.*, 2019). Essa integração entre gestão de ativos e participação popular fortalece a governança inteligente e contribui para a melhoria contínua dos serviços públicos, promovendo maior transparência e efetividade nas ações do poder público (OLIVEIRA, 2020).

2.2.1 Zeladoria Urbana

A zeladoria urbana aborda um conjunto de atividades que visam manter em boas condições de uso os ambientes públicos para a população, realizando serviços essenciais de manutenção e conservação, tais como: manutenção de ruas, poda de árvores, limpeza de vias, conservação de praças e calçadas. Em cidades de pequeno porte, especialmente no interior dos estados do Nordeste brasileiro, é possível encontrar desafios como: limitações operacionais, de infraestrutura e recursos humanos especializados (SANTOS *et al.*, 2019).

Os processos nesses contextos costumam não ser automatizados e sistematizados, o que compromete seriamente o planejamento, a priorização das demandas, a transparência das

ações dos gestores e pode propiciar o mau uso de verba pública. A implementação de sistemas digitais de gestão de demandas de zeladoria urbana, integrados a tecnologias de geolocalização e priorização inteligente, pode contribuir significativamente para a eficiência operacional e para a melhoria da qualidade dos serviços prestados à população (OLIVEIRA, 2020).

Aplicativos móveis para reporte de problemas urbanos, como buracos em vias, iluminação pública defeituosa e lixo acumulado, têm se mostrado eficazes na mediação entre cidadãos e gestores públicos (SANTOS *et al.*, 2019). Quando integrados a sistemas de gestão de ativos e algoritmos de otimização de rotas, esses aplicativos possibilitam não apenas o registro das demandas, mas também o planejamento eficiente das equipes de manutenção, a priorização baseada em critérios objetivos e o acompanhamento transparente do andamento das solicitações.

2.3 Métodos de Otimização

Problemas de otimização são encontrados em diversas aplicações práticas, como o planejamento de rotas, alocação de recursos e *scheduling*. Esses problemas buscam encontrar uma solução factível dentre um conjunto de soluções possíveis, de acordo com determinado critério de otimização (CORMEN *et al.*, 2012). Os métodos para resolver problemas de otimização podem ser classificados em diferentes categorias, cada uma com características, vantagens e limitações específicas.

2.3.1 Métodos Exatos

Métodos exatos são aqueles que garantem encontrar a solução ótima para um problema de otimização. Esses métodos exploram sistematicamente o espaço de soluções, avaliando todas as possibilidades ou utilizando técnicas de poda para eliminar soluções que comprovadamente não podem ser ótimas (CORMEN *et al.*, 2012).

Entre as principais técnicas exatas, destacam-se:

- **Enumeração Completa:** Consiste em avaliar todas as soluções possíveis e selecionar a melhor. Embora garanta a solução ótima, torna-se impraticável para problemas com grande espaço de busca, pois geralmente possuem um crescimento exponencial do número de soluções.
- **Programação Dinâmica:** Técnica que resolve problemas complexos dividindo-os em subproblemas menores e armazenando os resultados desses subproblemas para evitar recál-

culos (CORMEN *et al.*, 2012). É aplicável quando o problema possui subestrutura ótima (a solução ótima contém soluções ótimas dos subproblemas) e subproblemas sobrepostos.

- **Branch-and-Bound:** Método que particiona o espaço de soluções em subconjuntos (ramificação) e utiliza limitantes para podar ramos que não podem conter a solução ótima. É amplamente utilizado em problemas de otimização combinatória, como o problema do caixeiro viajante.
- **Programação Linear e Inteira:** Técnicas matemáticas que formulam o problema de otimização como um conjunto de equações e inequações lineares. Enquanto a programação linear lida com variáveis contínuas, a programação inteira trata de variáveis discretas, sendo esta última geralmente mais complexa computacionalmente.

A principal limitação dos métodos exatos é a complexidade computacional. Parte dos problemas de otimização pertencem à classe NP-difícil, onde não se conhece algoritmo que encontre a solução ótima em tempo polinomial. Para instâncias de grande porte, o tempo necessário para obter a solução ótima pode ser impraticável, motivando o uso de métodos aproximados (CORMEN *et al.*, 2012).

2.3.2 Heurísticas

Heurísticas são métodos que buscam encontrar boas soluções em tempo computacional razoável, sem necessariamente garantir a otimalidade. Uma heurística é uma regra prática, estratégia ou técnica que simplifica a resolução de problemas complexos (RUSSELL; NORVIG, 2013).

As principais características das heurísticas incluem:

- **Eficiência computacional:** Geralmente possuem complexidade de tempo polinomial, permitindo resolver instâncias grandes em tempo razoável.
- **Qualidade de solução:** Embora não garantam a solução ótima, frequentemente produzem soluções de boa qualidade, próximas ao ótimo.
- **Especificidade:** Parte das heurísticas são desenvolvidas explorando características específicas do problema, tornando-as eficazes para determinados tipos de instâncias.

Exemplos de heurísticas construtivas incluem o algoritmo do vizinho mais próximo para o problema do caixeiro viajante, onde, partindo de um vértice inicial, sempre se escolhe o vértice não visitado mais próximo até completar o circuito. Heurísticas de melhoria, como a busca local, partem de uma solução inicial e iterativamente aplicam modificações locais que

melhoram a qualidade da solução (CORMEN *et al.*, 2012).

2.3.3 Meta-heurísticas

Meta-heurísticas são estratégias de alto nível que guiam heurísticas subordinadas na busca por soluções de qualidade. Diferentemente das heurísticas clássicas, as meta-heurísticas são mais genéricas e podem ser aplicadas a diversos problemas de otimização com pequenas adaptações (RUSSELL; NORVIG, 2013).

A escolha entre métodos exatos, heurísticas ou meta-heurísticas depende do contexto da aplicação, considerando fatores como o tamanho da instância, requisitos de qualidade da solução, tempo disponível para computação e recursos computacionais (CORMEN *et al.*, 2012).

2.3.3.1 Algoritmos Genéticos

Os Algoritmos Genéticos (Algoritmos Genéticos (AG)) constituem uma meta-heurística inspirada nos princípios da evolução biológica, simulando o processo de seleção natural para evoluir uma população de soluções candidatas ao longo de gerações sucessivas (RUSSELL; NORVIG, 2013). São particularmente eficazes para problemas de otimização combinatória NP-difíceis, como o problema do caixeiro viajante.

A estrutura básica de um AG envolve os seguintes componentes principais: (i) **Cromossomo** - representação de uma solução candidata, codificada como sequência de genes (no TSP, uma permutação das cidades); (ii) **População** - conjunto de soluções que evoluem simultaneamente; (iii) **Função de Aptidão (*Fitness*)** - métrica que avalia a qualidade de cada solução, determinando sua probabilidade de reprodução.

O algoritmo opera através de três operadores genéticos fundamentais:

- **Seleção:** Escolhe indivíduos para reprodução favorecendo aqueles com maior aptidão. Métodos comuns incluem seleção por torneio, roleta e ranking.
- **Cruzamento (*Crossover*):** Combina características de dois cromossomos pais para gerar descendentes. Para problemas de permutação como o *TSP*, utiliza-se técnicas como o Cruzamento de Ordem (Cruzamento de Ordem (OX)), que preserva a ordem relativa dos elementos. Outras variantes incluem cruzamento de um ponto, dois pontos e uniforme.
- **Mutação:** Introduce pequenas alterações aleatórias nos cromossomos (como troca de posições ou inversão de segmentos), mantendo a diversidade genética e evitando convergência prematura para ótimos locais.

O ciclo evolutivo segue a sequência: (1) Inicialização de população aleatória; (2) Avaliação da aptidão de cada indivíduo; (3) Seleção dos melhores para reprodução; (4) Aplicação de cruzamento e mutação para gerar nova geração; (5) Substituição da população antiga, frequentemente mantendo os melhores indivíduos (elitismo). O processo se repete até atingir um critério de parada, como número máximo de gerações ou qualidade de solução satisfatória.

No contexto de otimização de rotas, os AGs permitem encontrar soluções de boa qualidade em tempo computacional razoável, mesmo para instâncias com grande número de pontos, onde métodos exatos seriam inviáveis. Suas principais vantagens incluem: exploração simultânea do espaço de soluções, robustez a ótimos locais, flexibilidade para diferentes representações e capacidade de paralelização (CORMEN *et al.*, 2012).

2.4 Teoria dos Grafos

Um grafo é uma estrutura matemática utilizada para modelar relações entre objetos, sendo amplamente empregado na ciência da computação para representar redes, mapas, circuitos e diversos outros sistemas (CORMEN *et al.*, 2012). Formalmente, um grafo G é definido pelo par ordenado $G = (V, E)$, onde:

- **Vértices (ou Nós):** V é um conjunto finito não-vazio de elementos chamados vértices. Cada vértice representa uma entidade do sistema modelado. Por exemplo, em um grafo representando uma rede de cidades, cada vértice corresponde a uma cidade.
- **Arestas:** E é um conjunto de pares de vértices, denominados arestas, que representam as conexões ou relações entre os vértices. Uma aresta $e \in E$ conecta dois vértices, podendo ser denotada como (u, v) , onde $u, v \in V$.

Os grafos podem ser classificados segundo diferentes critérios (CORMEN *et al.*, 2012):

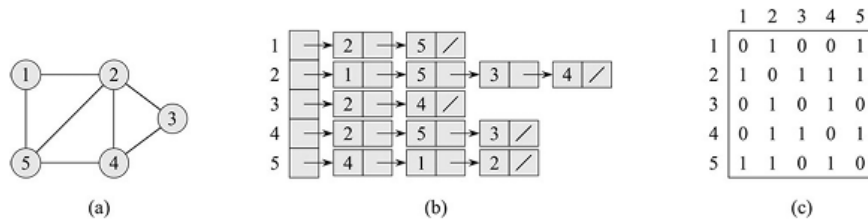
2.4.1 Grafos Direcionados e Não-Direcionados

- **Grafo Não-Direcionado:** As arestas não possuem direção, ou seja, a aresta (u, v) é equivalente a (v, u) . Representa relações simétricas, como estradas de mão dupla entre cidades.
- **Grafo Direcionado (Dígrafo):** As arestas possuem direção, representadas por setas. A aresta (u, v) indica uma conexão de u para v , mas não necessariamente de v para u . Útil

para modelar ruas de mão única, dependências entre tarefas, etc.

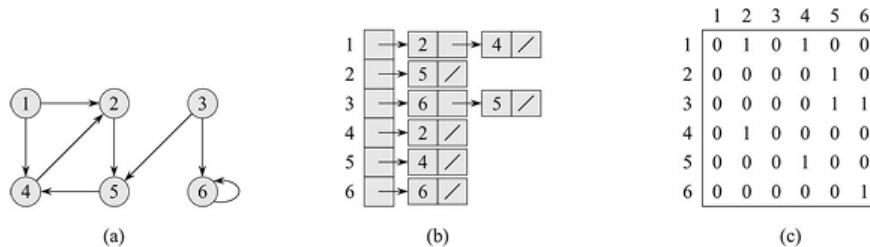
A Figura 1 ilustra duas representações de um grafo não direcionado: o grafo propriamente dito, sua representação por lista de adjacências e por matriz de adjacências. Já a Figura 2 apresenta as mesmas representações para um grafo direcionado, evidenciando o sentido das conexões através de setas.

Figura 1 – Duas representações de um grafo não dirigido. (a) Um grafo não dirigido G com cinco vértices e sete arestas. (b) Uma representação de G por lista de adjacências. (c) A representação de G por matriz de adjacências



Fonte: (CORMEN *et al.*, 2012).

Figura 2 – Duas representações de um grafo dirigido. (a) Um grafo dirigido G com seis vértices e oito arestas. (b) Uma representação de G por lista de adjacências. (c) A representação de G por matriz de adjacências



Fonte: (CORMEN *et al.*, 2012).

2.4.2 Grafos Ponderados

Em muitas aplicações práticas, as arestas possuem pesos ou custos associados. Um grafo ponderado é aquele em que existe uma função $w : E \rightarrow \mathbb{R}$ que atribui um valor numérico a cada aresta (CORMEN *et al.*, 2012). Esses pesos podem representar:

- Distâncias entre localizações geográficas;
- Tempo de deslocamento entre pontos;
- Custo de transporte;
- Capacidade de transmissão de dados em redes.

2.4.3 Conceitos Adicionais

- **Adjacência:** Dois vértices são adjacentes se existe uma aresta conectando-os.
- **Grau de um Vértice:** Em grafos não-direcionados, o grau de um vértice v é o número de arestas incidentes a ele. Em grafos direcionados, distingue-se entre grau de entrada (arestas que chegam) e grau de saída (arestas que saem).
- **Caminho:** Sequência de vértices onde cada par consecutivo está conectado por uma aresta. O comprimento de um caminho é o número de arestas que o compõem.
- **Ciclo:** Caminho que começa e termina no mesmo vértice, sem repetir arestas.
- **Grafo Conexo:** Grafo não-direcionado onde existe um caminho entre qualquer par de vértices.

2.4.4 Representação Computacional

Grafos podem ser representados computacionalmente de duas formas principais (CORMEN *et al.*, 2012):

- **Matriz de Adjacência:** Matriz A de dimensão $|V| \times |V|$, onde $A[i][j] = 1$ se existe aresta entre os vértices i e j , e $A[i][j] = 0$ caso contrário. Para grafos ponderados, $A[i][j]$ armazena o peso da aresta. Ocupa espaço $O(V^2)$.
- **Lista de Adjacência:** Para cada vértice, mantém-se uma lista dos vértices adjacentes. Mais eficiente em termos de espaço para grafos esparsos, ocupando $O(V + E)$.

Os algoritmos de busca em grafos são fundamentais para resolver problemas de caminho mínimo, conectividade, detecção de ciclos e otimização de rotas, constituindo ferramentas essenciais para aplicações em logística, redes de computadores e planejamento urbano (CORMEN *et al.*, 2012).

2.5 Algoritmos de Busca

Nessa seção, serão apresentados alguns dos principais algoritmos de busca em grafos, sendo eles a Busca em Largura (BFS), a Busca em Profundidade (DFS), a Busca de Custo Uniforme (UCS) e o Algoritmo A* (A Estrela).

2.5.1 Busca em Largura (BFS)

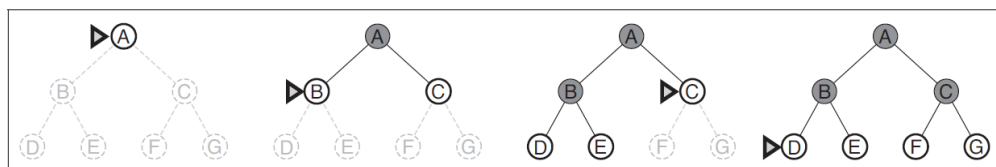
A Busca em Largura (*Breadth-First Search* - *Breadth-First Search* / Busca em Largura (*BFS*)) explora o grafo sistematicamente visitando todos os vértices a uma mesma distância da origem antes de avançar para níveis mais profundos (CORMEN *et al.*, 2012). O algoritmo utiliza uma estrutura de fila (*First In, First Out* / Primeiro a Entrar, Primeiro a Sair (*FIFO*) - *First In, First Out*) para controlar a ordem de visitação dos vértices.

A complexidade de tempo da BFS é $O(V + E)$, onde V é o número de vértices e E é o número de arestas. O algoritmo garante encontrar o caminho mais curto em termos de número de arestas quando todas têm o mesmo peso. A BFS é particularmente útil para:

- Encontrar o menor caminho em grafos não ponderados;
- Testar a conectividade de um grafo;
- Encontrar todos os vértices alcançáveis a partir de um vértice de origem;
- Detectar ciclos em grafos não direcionados.

A Figura 3 ilustra o funcionamento da busca em largura em um grafo, mostrando a ordem de visitação dos vértices nível por nível.

Figura 3 – Exemplo de execução da Busca em Largura (BFS) em um grafo



Fonte: (CORMEN *et al.*, 2012).

2.5.2 Busca em Profundidade (DFS)

A Busca em Profundidade (*Depth-First Search* - *Depth-First Search* / Busca em Profundidade (*DFS*)) percorre o grafo explorando o máximo possível cada caminho antes de retroceder (CORMEN *et al.*, 2012). Pode ser implementada recursivamente ou utilizando uma pilha (*Last In, First Out* / Último a Entrar, Primeiro a Sair (*LIFO*) - *Last In, First Out*). A DFS também possui complexidade $O(V + E)$.

A DFS é especialmente adequada para:

- Detecção de ciclos em grafos direcionados e não direcionados;
- Ordenação topológica em grafos acíclicos direcionados (*Directed Acyclic Graph* / Grafo

Acíclico Direcionado (*DAG*));

- Decomposição de grafos em componentes fortemente conexos;
- Resolução de quebra-cabeças e problemas que exigem exploração de todas as possibilidades, como labirintos.

expansão, os custos dos nós vizinhos são atualizados se um caminho mais barato for encontrado. A UCS é completa e ótima, garantindo encontrar a solução de menor custo quando todos os custos das arestas são não-negativos (RUSSELL; NORVIG, 2013).

A complexidade de tempo e espaço da busca de custo uniforme é $O(b^{1+\lceil C^*/\epsilon \rceil})$, onde b é o fator de ramificação, C^* é o custo da solução ótima, e ϵ é o menor custo de aresta. Na prática, com implementação eficiente usando *heap* binário, a complexidade é $O((V + E) \log V)$.

A UCS é particularmente adequada para:

- Encontrar caminhos de menor custo em grafos ponderados com pesos não-negativos;
- Problemas de roteamento onde diferentes caminhos têm custos variados;
- Planejamento de rotas considerando distâncias, tempos ou custos operacionais;
- Situações onde não há informação heurística disponível sobre a proximidade do objetivo.

A principal diferença entre a UCS e algoritmos informados como o A^* é que a UCS não utiliza conhecimento adicional (heurística) sobre a distância até o objetivo, expandindo nós apenas com base no custo acumulado desde a origem (RUSSELL; NORVIG, 2013).

2.5.4 Algoritmo A^* (A Estrela)

O algoritmo A^* é uma extensão da busca de custo uniforme que incorpora uma função heurística para guiar a busca em direção ao objetivo (RUSSELL; NORVIG, 2013). A função de avaliação é definida como $f(n) = g(n) + h(n)$, onde $g(n)$ é o custo do caminho da origem até o vértice n , e $h(n)$ é a heurística que estima o custo de n até o objetivo.

Para que o A^* garanta encontrar o caminho ótimo, a heurística deve ser admissível, ou seja, nunca superestimar o custo real até o objetivo. Heurísticas comuns incluem a distância euclidiana e a distância de Manhattan em espaços bidimensionais (RUSSELL; NORVIG, 2013).

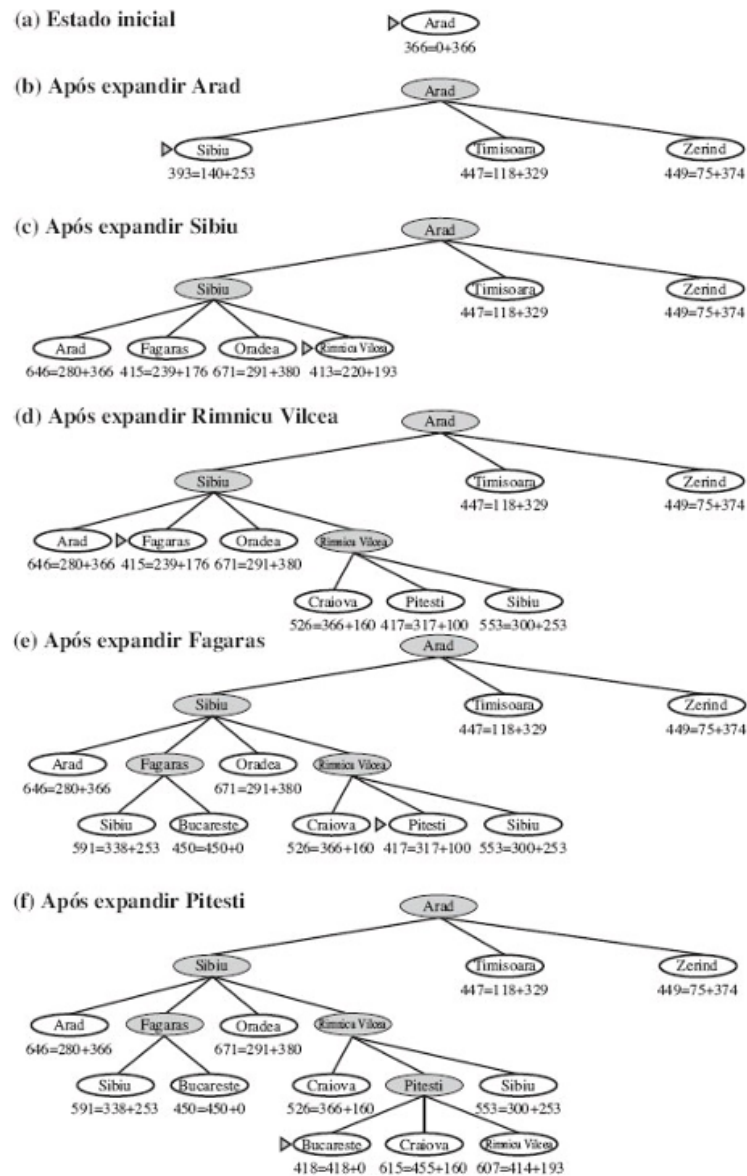
O A^* é particularmente eficiente quando existe uma boa heurística disponível, pois reduz significativamente o número de vértices explorados em comparação com Dijkstra. É amplamente utilizado em:

- Jogos e simulações para movimentação de personagens e entidades;
- Robótica para planejamento de trajetórias;
- Sistemas de navegação com informação geográfica;
- Otimização de rotas considerando múltiplos critérios.

A Figura 5 apresenta um exemplo clássico de aplicação do algoritmo A^* para encontrar o caminho mais curto entre cidades romenas, demonstrando como a heurística guia a

busca de forma eficiente.

Figura 5 – Exemplo de aplicação do algoritmo A* no problema de roteamento entre cidades da Romênia



Fonte: (RUSSELL; NORVIG, 2013).

3 REVISÃO DA LITERATURA

Neste capítulo serão descritas as etapas realizadas durante a Revisão da Literatura do trabalho.

3.1 Contextualização

Diante do objetivo do trabalho, foi necessário a realização de uma revisão sistemática com o intuito de identificar os principais trabalhos relacionados com essa pesquisa. Esta fase da pesquisa foi organizada em três etapas: Planejamento, Condução e Resultados que serão descritas nas próximas seções.

3.2 Planejamento

Segundo (KITCHENHAM *et al.*, 2009), o Planejamento é a etapa de definição de cada passo do protocolo, que consiste em: definição das Questões de Pesquisa; definição da *string* de busca; definição das bases de dados e definição dos critérios de inclusão e exclusão. Sendo assim, nas próximas seções, serão descritos os passos deste protocolo.

3.2.1 Definição de Questões de Pesquisa

Questões de Pesquisa (QP)1 - Qual a estratégia adotada no trabalho?

- Criação de um framework
- Simulação
- Análise conceitual de algum tema relacionado às cidades inteligentes
- Análise de dados
- Abordagem com internet das coisas
- Desenvolvimento/uso de algoritmos de otimização

QP2 - Qual(is) algoritmo(s) adotado(s) no trabalho?

- Algoritmo genético
- Busca local
- Colônia de formigas
- Grasp
- Programação linear

- Algoritmo Híbrido

3.2.2 Definição da string de busca

Após a definição das questões de pesquisas, foi definida a seguinte string de busca: *"cidades inteligentes"AND "framework"AND "problema do caixeiro viajante"*

3.2.3 Definições de bases de dados

De posse da string de busca, foram selecionadas as bases de dados, conforme mostra o Quadro 1:

Quadro 1 – Bases de dados selecionadas.

| Base de dados | Link |
|---------------------|---|
| Google acadêmico | https://www.periodicos.capes.gov.br |
| Periódicos da capes | https://scholar.google.com.br/ |
| SBC-OpenLib | https://sol.sbc.org.br |

Fonte: elaborado pelo autor (2025).

3.2.4 Definição dos critérios de inclusão e exclusão

Para concluir essa etapa do planejamento, foram definidos os critérios de inclusão e exclusão que todos os artigos deverão satisfazer. Os critérios definidos são: **Critérios de inclusão:**

- Estudos completos publicados em revistas ou conferências alinhados ao problema de pesquisa;
- Estudos teóricos ou experimentais com o objetivo de apresentar conceitos para o entendimento da área;
- Acessível eletronicamente e ter sido publicado no período de 2015 a 2025.

Critérios de exclusão:

- Estudos que não estejam relacionados ao problema de pesquisa;
- Estudos que não respondem a nenhuma das questões de pesquisa;
- Artigos duplicados, ou seja, aqueles encontrados em mais de uma base de dados;
- Artigos convidados, tutoriais, relatórios técnicos que não passam pelo critério de avaliação das conferências ou revistas;
- Estudos não disponíveis para download.

Tabela 1 – Quantidade de artigos em cada base de dados.

| Base de dados | Resultados da busca | Download | Inclusão/Exclusão |
|---------------------|---------------------|----------|-------------------|
| Google acadêmico | 26 | 13 | 3 |
| Periódicos da capes | 10 | 6 | 4 |
| SBC-OpenLib | 3 | 3 | 1 |

Fonte: Elaborado pelo autor (2025).

3.3 Condução

Na fase de condução, foi colocado em prática o planejamento definido na seção de Planejamento, dessa forma, a *string* de busca foi inserida em cada base de dados, respeitando as especificações de cada base. De posse dos resultados da busca, foi realizado o *download* dos estudos disponíveis. Em seguida, os trabalhos passaram pelos critérios de inclusão e exclusão. Na Tabela 1, é possível identificar a quantidade de trabalhos em cada base durante a busca, a realização do *download* e a aplicação dos critérios de inclusão e exclusão.

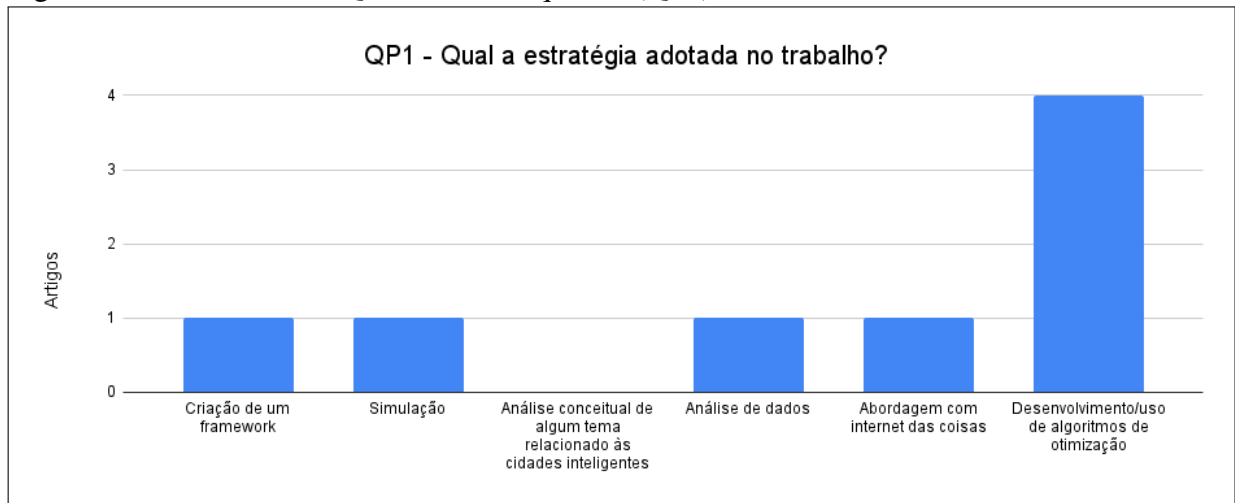
3.4 Resultados

Nesta seção são apresentados e discutidos os resultados obtidos a partir da análise dos trabalhos selecionados, organizados de acordo com as duas questões de pesquisa estabelecidas.

A Figura 6 apresenta um gráfico de barras com os resultados da QP1, referente às estratégias metodológicas adotadas nos trabalhos selecionados. Observa-se que apenas um estudo utilizou a estratégia de criação de *framework*: o trabalho de (SILVA, 2024), no qual os autores desenvolveram uma solução baseada em *Application Programming Interface* / Interface de Programação de Aplicações (*API*)s externas para calcular rotas entre pontos estratégicos visando a resolução de Problemas de Roteamento de Veículos.

A estratégia mais prevalente identificada na QP1 foi o desenvolvimento e/ou uso de algoritmos de otimização, totalizando quatro trabalhos. (PAZ, 2023) desenvolveu um algoritmo para otimizar o transporte urbano considerando o menor trajeto no menor tempo possível, evitando gargalos como congestionamentos. (SILVA *et al.*, 2022) utilizou a ferramenta externas de código aberto para resolver o Problema do Caixeiro Viajante em centros urbanos, aplicado ao contexto de entregas dos correios. (SILVA; OCHI, 2017) propôs um algoritmo para resolver uma variação do Problema do Caixeiro Viajante, denominada Caixeiro Alugador, visando otimizar viagens com aluguel de veículos entre cidades. Por fim, (CHAVES *et al.*, 2007) apresentou um método heurístico híbrido para resolver aproximadamente o Problema do Caixeiro Viajante com

Figura 6 – Resultados da Questão de Pesquisa 1 (QP1).



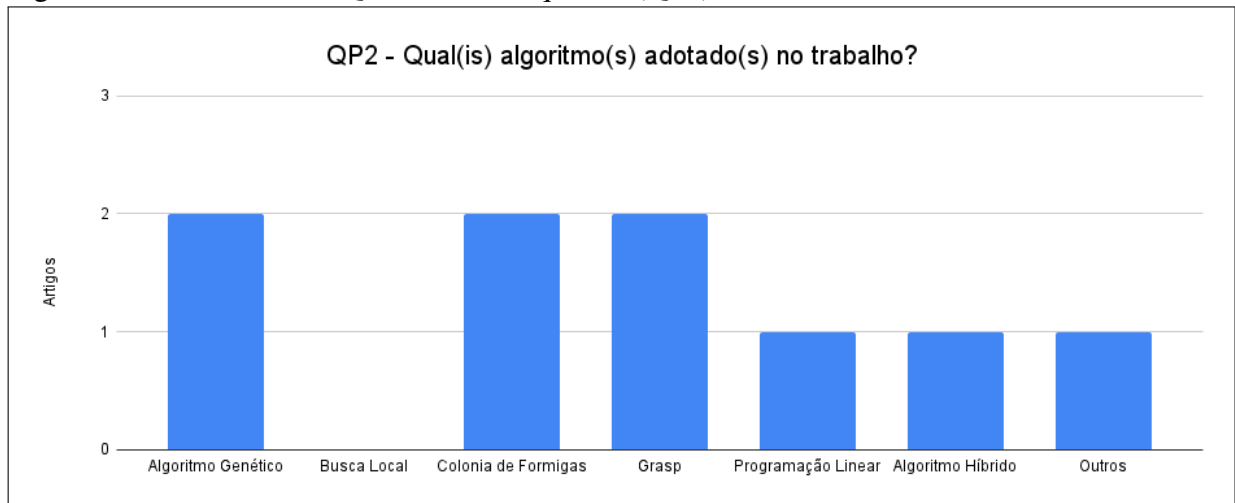
Fonte: elaborada pelo autor (2025).

Coleta de Prêmios.

Ainda em relação à QP1, na alternativa de Abordagem com Internet das Coisas, (BARTH, 2016) desenvolveu uma metodologia para elaboração de um sistema computacional baseado em um conjunto de regras definidas por modelagem matemática, visando apoiar o projeto de redes híbridas (ópticas e sem fio) para atendimento das demandas de cidades inteligentes. Na alternativa de simulação, (SATI *et al.*, 2020) utilizou e comparou três diferentes modelos matemáticos, analisando os benefícios de cada um para a resolução do problema de roteamento de veículos em centros urbanos voltado ao transporte de funcionários. Quanto à alternativa de análise de dados, (GEORGES, 2014) apresentou um estudo das rotas de coleta de materiais recicláveis de uma cooperativa, aplicando um modelo conceitual do Problema do Caixeiro Viajante para reordenação dos pontos de coleta.

A Figura 7 apresenta um gráfico de barras com os resultados da QP2, referente aos algoritmos adotados nos trabalhos selecionados. (PAZ, 2023) utilizou o Algoritmo Genético combinado com conceitos de probabilidade para otimização de rotas e, para fins comparativos, também empregou os algoritmos Colônia de Formigas, *Greedy Randomized Adaptive Search Procedure* / Procedimento de Busca Adaptativa Gulosa Aleatória (GRASP), A* (A estrela) e Dijkstra. (BARTH, 2016) adotou o algoritmo Colônia de Formigas em seu projeto de redes híbridas para cidades inteligentes. (SILVA; OCHI, 2017) e (CHAVES *et al.*, 2007) utilizaram, respectivamente, o Algoritmo Genético e um método híbrido de GRASP com outras técnicas de otimização para refinar a solução construída, ambos aplicados à resolução de variantes do Problema do Caixeiro Viajante. Por sua vez, (SATI *et al.*, 2020) adotou Programação Linear para resolver o problema de roteamento de veículos em centros urbanos.

Figura 7 – Resultados da Questão de Pesquisa 2 (QP2).



Fonte: elaborada pelos autores (2025).

Os demais trabalhos, (SILVA, 2024), (SILVA *et al.*, 2022) e (GEORGES, 2014), não adotaram nenhum dos algoritmos especificados na QP2. Estes estudos utilizaram ferramentas externas para resolução dos problemas propostos, tais como OR-Tools (*software* de código aberto do Google), OpenStreetMap, Open Source Routing Machine e/ou modelos conceituais, porém sem implementação direta de algoritmos ou realização de análises comparativas entre diferentes abordagens.

3.5 Considerações Finais

A realização desta revisão sistemática da literatura foi fundamental para mapear o estado da arte em otimização de rotas e gestão de ativos públicos no contexto de cidades inteligentes. Por meio das questões de pesquisa estabelecidas, foi possível identificar as principais estratégias metodológicas adotadas pelos pesquisadores da área, bem como os algoritmos e técnicas de otimização empregados para resolver problemas relacionados ao Problema do Caixeiro Viajante e suas variantes em ambientes urbanos.

Os resultados evidenciaram também a crescente utilização de ferramentas e APIs externas, como OR-Tools, OpenStreetMap e Open Source Routing Machine, que facilitam a implementação de soluções práticas sem a necessidade de desenvolvimento de algoritmos do zero.

Esta revisão forneceu subsídios importantes para o desenvolvimento do presente trabalho, orientando tanto a escolha da abordagem metodológica quanto a seleção de técnicas de otimização adequadas ao problema proposto. A análise comparativa dos algoritmos identificados

na literatura servirá como referencial para a avaliação da solução desenvolvida, permitindo posicionar a contribuição desta pesquisa no cenário científico atual da área de cidades inteligentes e otimização de rotas para gestão pública.

[illegible]

$$k_{n+1} = n^2 + k_n^2 - k_{n-1}. \quad (4.3)$$

[illegible]

$$\cos(2\theta) = \cos^2 \theta - \sin^2 \theta. \quad (4.4)$$

[illegible]

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}. \quad (4.5)$$

[illegible]

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}. \quad (4.6)$$

[illegible]

6 CONCLUSÕES E TRABALHOS FUTUROS

Parte final do texto na qual se apresentam as conclusões apoiadas no desenvolvimento do assunto. É a recapitulação sintética dos resultados obtidos. Pode apresentar recomendações e sugestões para pesquisas futuras.

REFERÊNCIAS

- BARTH, M. J. **Otimização multi-nível para projeto de redes híbridas (ópticas e sem fio) para implementação de cidades inteligentes**. Dissertação (Mestrado em Computação Aplicada) — Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo - RS, Brasil, 2016. Orientador: José Vicente Canto dos Santos.
- BOTELHO, P.; BORGES, A.; ALVES, G. Proposta de implantação de um sistema ciberfísico para um smart parking baseado em agentes inteligentes. In: **Anais do XIII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações**. Porto Alegre, RS, Brasil: SBC, 2019. p. 259–264. ISSN 2326-5434. Disponível em: <<https://sol.sbc.org.br/index.php/wesaac/article/view/33361>>.
- CARVALHO, M. D. S. d. **Smart cities: uso de sensores e dados secundários para cidades inteligentes centradas no cidadão**. Dissertação (Mestrado) — Universidade de Caxias do Sul, Caxias do Sul, Brasil, 2024.
- CASTRO, L.; MANOEL, F.; JESUS, V.; PANTOJA, C.; BORGES, A.; ALVES, G. Integrando sistemas multi-agentes embarcados, simulação urbana e aplicações de iot. In: **Anais do XIV Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações**. Porto Alegre, RS, Brasil: SBC, 2020. p. 165–176. ISSN 2326-5434. Disponível em: <<https://sol.sbc.org.br/index.php/wesaac/article/view/33389>>.
- CHAVES, A. A.; BIAJOLI, F. L.; MINE, O. M.; SOUZA, M. J. F. Metaheurísticas híbridas para resolução do problema do caixeiro viajante com coleta de prêmios. **Produção**, ABEPRO, Brasil, v. 17, n. 2, p. 263–272, 2007.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos: teoria e prática**. 3. ed. Rio de Janeiro: Elsevier, 2012.
- COSTA, I. S. Bacharelado em Engenharia Urbana, **Proposição de elementos para uso de sistemas inteligentes de transportes para potencialização de intervenções estruturantes no sistema viário de cidades de médio porte: uma proposta para o novo anel viário de Bom Despacho-MG**. Ouro Preto - MG, Brasil: [s.n.], 2023.
- DIAS, C.; LOPES, F.; LEITE, J. Smartnode dashboard: um framework front-end baseado em node-red para criação de city dashboards. In: **Anais do II Workshop Brasileiro de Cidades Inteligentes**. Porto Alegre, RS, Brasil: SBC, 2019. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/wbci/article/view/6744>>.
- GEORGES, M. R. R. Rotas solidárias: um estudo das rotas de coleta de materiais recicláveis numa cooperativa popular de coleta e seleção de recicláveis. **Revista Gestão Industrial**, UTFPR, Ponta Grossa - PR, Brasil, v. 10, n. 1, p. 1–21, 2014.
- KITCHENHAM, B.; BRERETON, O. P.; BUDGEN, D.; TURNER, M.; BAILEY, J.; LINKMAN, S. Systematic literature reviews in software engineering – a systematic literature review. **Information and Software Technology**, v. 51, n. 1, p. 7–15, 2009. ISSN 0950-5849. Special Section - Most Cited Articles in 2002 and Regular Research Papers. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950584908001390>>.
- MELONIO, A. C. C. **Smart São Paulo: um estudo da mobilidade urbana sob a ótica de Machine Learning e aspectos espaço-temporais**. Dissertação (Mestrado) — Universidade Presbiteriana Mackenzie, 2021.

OLIVEIRA, V. A. T. **Gestão de operações de serviços de emergência no contexto de cidades inteligentes e sustentáveis: o caso da Polícia Militar do Paraná.** Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2020.

PAZ, F. A. **Planejamento de Rotas Veiculares e Otimização de Mobilidade Urbana Utilizando Algoritmo Bioinspirado e Paralelo.** Dissertação (Proposta de Dissertação de Mestrado em Ciência da Computação) — Universidade Federal de Sergipe, São Cristóvão – SE, Brasil, 2023. Orientador: Rubens de Souza Matos Júnior; Coorientador: Ricardo José Paiva de Britto Salgueiro.

RUSSELL, S. J.; NORVIG, P. **Inteligência artificial.** 3. ed. Rio de Janeiro: Elsevier, 2013. Tradução de: Artificial intelligence, 3rd ed. ISBN 978-85-352-3701-6.

SANTIAGO, T. E. T. **Cidades inteligentes, gestão urbana e geotecnologias: proposta de city information modeling (CIM) aplicado ao Município de Madre de Deus-BA.** Dissertação (Mestrado) — Universidade Católica do Salvador (UCSal), 2023.

SANTOS, L.; NASCIMENTO, L.; NUNES, S. Reclamando app: um aplicativo para auxiliar na reivindicação de problemáticas urbanas. In: **Anais da XIX Escola Regional de Computação Bahia, Alagoas e Sergipe.** Porto Alegre, RS, Brasil: SBC, 2019. p. 184–189. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/erbase/article/view/8976>>.

SATI, T. N.; SCARPIN, C. T.; JUNIOR, J. E. P.; LOPES, R. M. Z. Diferentes formulações de programação linear inteira para a roteirização no transporte de funcionários. **Produção**, UFPR, Curitiba - PR, Brasil, v. 17, n. 2, p. 263–272, 2020.

SILVA, A. R. Villela da; OCHI, L. S. Um algoritmo evolutivo para o problema do caixeiro alugador. In: **Proceeding Series of the Brazilian Society of Applied and Computational Mathematics.** Gramado - RS, Brasil: SBMAC, 2017. v. 5, n. 1, p. 460–467.

SILVA, E. C. d.; LIMA, D. R.; SANTOS, L. C. d.; SILVA, F. D. V. d. Uso da ferramenta or-tools na geração de rotas otimizadas em agência dos correios utilizando algoritmos para o problema do caixeiro viajante. In: **Anais de evento acadêmico (IFCE - Campus Aracati).** Aracati - CE, Brasil: [s.n.], 2022.

SILVA, M. d. O. Bacharelado em Engenharia de Produção, **Desenvolvimento de uma aplicação gráfica para resolução de problemas de roteamento de veículos.** João Pessoa - PB, Brasil: [s.n.], 2024. Orientador: Luciano Carlos Azevedo da Costa.

STORCK, C.; SALES, E.; ZÁRATE, L.; FIGUEIREDO, F. Proposta de um framework baseado em mineração de dados para redes 5g. **Revista Eletrônica de Sistemas de Informação**, v. 16, 2017. ISSN 1677-3071. Disponível em: <<https://www.periodicosibepes.org.br/index.php/reinfo/article/view/2488>>.

WEISS, M. C.; BERNARDES, R. C.; CONSONI, F. L. Cidades inteligentes como nova prática para o gerenciamento dos serviços e infraestruturas urbanos: a experiência da cidade de porto alegre. **Urbe. Revista Brasileira de Gestão Urbana**, SciELO Brasil, v. 7, p. 310–324, 2015.

APÊNDICE A – EXEMPLO DE APÊNDICE

Um apêndice é um documento elaborado pelo autor, diferentemente do anexo. Geralmente, se coloca como apêndice, questionários, códigos de programação, tabelas que tomariam muito espaço no meio do trabalho. Artigos, resumos ou qualquer publicação relacionada ao trabalho podem ser utilizados como apêndice.

APÊNDICE B – QUESTIONÁRIO UTILIZADO PARA...

Questão 1. Esta é a primeira questão com alguns itens:

- (a) Este é o primeiro item
- (b) Segundo item

Questão 2. Esta é a segunda questão:

- (a) Este é o primeiro item
- (b) Segundo item

Questão 3. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc dictum sed tortor nec viverra. consectetur adipiscing elit. Nunc dictum sed tortor nec viverra.

- (a) consectetur
- (b) adipiscing
- (c) Nunc
- (d) dictum

APÊNDICE C – CÓDIGOS-FONTES UTILIZADOS PARA...

Código-fonte 1 – Hello World em C++

```

1 #include <iostream>
2 using namespace std;
3 int main() {
4     cout<<"Hello World!"<<endl;
5     system("pause");
6 }

```

Código-fonte 2 – Hello World em Java

```

1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello World!");
4     }
5 }

```

Algoritmo 1: Roleta de Seleção

Input: Lista de pares (*objeto*, *distancia*)

Output: Objeto selecionado

foreach (*objeto*, *distancia*) *em roleta do*

if *distancia* = 0 **then**

return *objeto*

end

 Adicione *objeto* à lista de objetos;

 Adicione *distancia* à lista de distancias;

end

foreach *distancia em distancias do*

 Calcule *peso* = 1/*distancia*;

 Adicione *peso* à lista de pesos;

end

Calcule *probabilidades* = *pesos* / \sum *pesos*;

Escolha aleatoriamente um objeto com base nas probabilidades;

return objeto escolhido;

Algoritmo 2: Gerar Solução

Output: Melhor rota gerada
 Inicialize $melhor_rota_copia \leftarrow [(inicio, 0)]$;
 $ativos_copia \leftarrow copia(ativos)$;
 $origem \leftarrow inicio$;
while $ativos_copia \neq \emptyset$ **do**
 Inicialize lista *roleta*;
 foreach *ativo em* $ativos_copia$ **do**
 if $ativo \notin distancia_entre_ativos[origem]$ **then**
 $rota, distancia, nos, ramificacao \leftarrow algoritmo(origem, ativo)$;
 Adicione *rota* a *caminho_entre_ativos*;
 Atualize $distancia_entre_ativos[origem][ativo] \leftarrow distancia$;
 end
 Adicione $(ativo, distancia)$ à *roleta*;
 end
 $elemento \leftarrow Roleta(roleta)$;
 Adicione $(elemento, 0)$ à $melhor_rota_copia$;
 Remova *elemento* de $ativos_copia$;
 $origem \leftarrow elemento$;
end
if $inicio \notin distancia_entre_ativos[origem]$ **then**
 $rota, distancia, nos, ramificacao \leftarrow algoritmo(origem, inicio)$;
 Adicione *rota* a *caminho_entre_ativos*;
 Atualize $distancia_entre_ativos[origem][inicio] \leftarrow distancia$;
end
 Adicione $(inicio, 0)$ à $melhor_rota_copia$;
 Chame $recalcula_distancias(melhor_rota_copia)$;
return $melhor_rota_copia$;

Algoritmo 3: Melhor Rota

Output: Atualiza a melhor rota encontrada
for $i \leftarrow 0$ **to** $iteracoes - 1$ **do**
 $melhor_rota_copia \leftarrow Gerar_Solucao()$;
 if $melhor_rota = \emptyset$ **ou** $melhor_rota_copia[-1][1] < melhor_rota[-1][1]$ **then**
 $melhor_rota \leftarrow melhor_rota_copia$;
 end
end

APÊNDICE D – IEEE CEFC 2016

Digest submetido ao The 17th Biennial Conference on Eletromagnetic Field Computation, Miami FL - NOV 13-16, 2016, USA.

Lightning Incidence Model Based on the Electric Field Gradient: 3D Electrostatic Analyses

Ednardo M. Rodrigues, Ricardo S. T. Pontes and Tobias R. Fernandes Neto

Federal University of Ceará, Department of Electrical Engineering, Fortaleza CE, BRAZIL
ednardorodrigues@dee.ufc.br

Abstract— The paper deals with the 3D electrostatic analysis of a lightning strike in a hangar and a power transmission line. The lightning incidence model is based on the electric field gradient. Finally, the simulation results are described and discussed.

Index Terms—Lightning, Electrostatic, Finite element.

I. INTRODUCTION

In [1], a 2D electrostatic analysis of a new lightning incidence model based on the electric field gradient (EFG) was presented. Moreover, the simulations results were carried out for a building and a power transmission line and they were compared with the classical electrogeometrical model (EGM), the rolling-sphere technique (RST) and the leader progression model (LPM) [2]. The present paper estimates the trajectory of lightning strikes from the thundercloud to a grounded metal roof of a hangar. Furthermore, the same procedure will be carried out for 500kV power transmission lines.

II. ELECTROSTATIC ANALYSES

A lightning occurs when the electric field is higher than the breakeven field (400kV/m —3MV/m) [2]. This model is based on the electric field gradient described by

$$\vec{E}_L(\vec{r}) \approx \vec{E}_b(\vec{r}) + \lambda_t \nabla E_b(\vec{r}), \quad (1)$$

$\vec{E}_b(\vec{r})$ is the background electric field, which is function of the position \vec{r} and it is generated by the electric potential difference (EPD) between the cloud and the ground. λ_t is the lightning step length (~50m) [3], and $\vec{E}_L(\vec{r})$ is the lightning electric field. More details about Eq. (1) can be found in [1].

A 3D finite element method (FEM) model of a hangar and a power transmission line (TL) were designed by using the electrostatic module. All simulations were carried out within a cubic domain of 250m x 250m x 250m. The upper level of each domain is defined with -12.5MV, while the lower level is the ground. This is equivalent to a real thundercloud with a potential of (-100MV) at 2km of altitude [4].

The dimensions of the hangar are: 8.60m height, 77.37m width and 229.00m length. The aluminum metal roof has 0.7mm thickness and it is grounded. The second simulation is for a TL composed by three phase conductors, equally spaced by 11.5m and positioned at 40.5m above the reference plane. The TLs are protected by two earth wires spaced by 19m over 54.47m of the reference plane.

III. RESULTS

In order to evaluate the 3D model, the software COMSOL Multiphysics® was used in a computer with quad-core processor of 2.6GHz. For the hangar, the simulation time was around 4s. The necessary physical memory for the simulation

was 1.34GB and 5.6GB of virtual memory. The electric field is very intense at the roof (about 80MV/m) and the lightning (cyan lines) strikes the building roof, as shown in Fig. 1a. In summary, it is not necessary to add air terminals, as long the roof is grounded. The simulation time for the TL was around 6 min, using 15GB of physical memory and 32GB of virtual memory. As shown in Fig 1b, the cyan lines strike the earth wires.

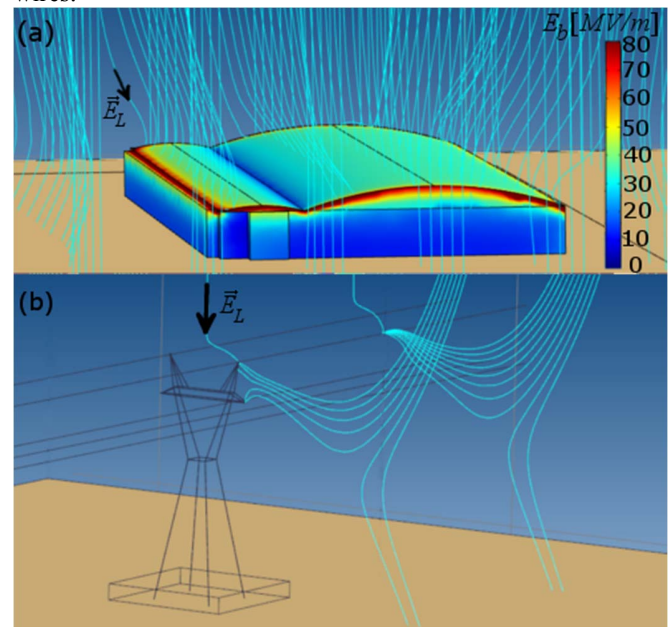


Fig. 1. Case of studies: (a) hangar and (b) power transmission line.

IV. CONCLUSIONS

The EFG simulations predicted that the aluminum metal roof is able to protect the hangar against lightning strikes. In the TL simulation, the earth wires have fulfilled the protection for the phase conductors. Finally, the protection zone and the design of lightning protection system can be evaluated by 3D electrostatic analyses, which are closer to the reality than the 2D analyses. However, 3D models are often more complex and require more simulation time.

REFERENCES

- [1] E. M. Rodrigues, *Novel Lightning Incidence Model Based on the Electric Field Gradient: 2D Electrostatic Analyses*. GROUND'2016 & 7th LPE, 2016.
- [2] V. Cooray, *Lightning protection*, The Institution of Engineering and Technology, 2009.
- [3] V.A. Rakov and M. A. Uman, *Lightning: physics and effects*. Cambridge University Press, 2007.
- [4] S. Visacro, *Descargas atmosféricas: Uma Abordagem de Engenharia (Lightning strike: An Engineering approach)*, Artliber, 2005.

ANEXO A – EXEMPLO DE UM ANEXO

Um anexo é um documento que não foi elaborado pelo autor, ou seja, o autor apenas anexa. Anexos podem ser tabelas, mapas, diagramas, *datasheets*, manuais e etc.

ÍNDICE

Adobe

Illustrator, 36

Photoshop, 36

CorelDraw, 36

Gimp, 36

InkScape, 36